Applied Mathematics in Engineering

Edited by: Olga Moreira

ARCLER PRESS

Applied Mathematics in Engineering

Applied Mathematics in Engineering

Edited by:

Olga Moreira

ARCLER PRESS

www.arclerpress.com

Applied Mathematics in Engineering

Olga Moreira

Arcler Press
224 Shoreacres Road
Burlington, ON L7L 2H2
Canada
www.arclerpress.com
Email: orders@arclereducation.com

e-book Edition 2023

ISBN: 978-1-77469-559-3 (e-book)

ISBN: 978-1-77469-474-9 (Hardcover)

Arcler Press publishes wide variety of books and eBooks. For more information about Arcler Press and its products, visit our website at www.arclerpress.com

DECLARATION

Some content or chapters in this book are open access copyright free published research work, which is published under Creative Commons License and are indicated with the citation. We are thankful to the publishers and authors of the content and chapters as without them this book wouldn't have been possible.

ABOUT THE EDITOR

Olga Moreira is a Ph.D. and M.Sc. in Astrophysics and B.Sc. in Physics/ Applied Mathematics (Astronomy). She is an experienced technical writer and data analyst. As a graduate student, she held two research grants to carry out her work in Astrophysics at two of the most renowned European institutions in the fields of Astrophysics and Space Science (the European Space Agency, and the European Southern Observatory). She is currently an independent scientist, peer-reviewer and editor. Her research interest is solar physics, machine learning and artificial neural networks.

TABLE OF CONTENTS

LIST OF CONTRIBUTORS

Xinming Huo
Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, Tianjin 300350, China

Shuofei Yang
Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon 999077, Hong Kong, China

Binbin Lian
Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, Tianjin 300350, China

Tao Sun
Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, Tianjin 300350, China

Yimin Song
Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, Tianjin 300350, China

Abd-Elsalam R. Abd-Elhay
National Authority for Remote Sensing and Space Sciences (NARSS), 23 Jozeph Tito St., Cairo, Egypt

Wael A. Murtada
National Authority for Remote Sensing and Space Sciences (NARSS), 23 Jozeph Tito St., Cairo, Egypt

Mohamed I. Yosof
Department of Electrical Engineering, Faculty of Engineering, Al-Azher University, Cairo, Egypt

Sebastian Blauth
Fraunhofer Institute for Industrial Mathematics ITWM, Kaiserslautern, Germany
TU Kaiserslautern, Kaiserslautern, Germany

Frank Hübner
Institute for Diagnostic and Interventional Radiology of the J.W. Goethe University Hospital, Frankfurt/Main, Germany

Christian Leithäuser
Fraunhofer Institute for Industrial Mathematics ITWM, Kaiserslautern, Germany

Norbert Siedow
Fraunhofer Institute for Industrial Mathematics ITWM, Kaiserslautern, Germany

Thomas J. Vogl
Institute for Diagnostic and Interventional Radiology of the J.W. Goethe University Hospital, Frankfurt/Main, Germany

Wasim Ullah Khan
School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China

Ali Imran
Department of Mathematics, COMSATS University Islamabad, Attock Campus, Kamra Road, Attock, Pakistan

MuhammadAsif Zahoor Raja
Future Technology Research center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan, ROC

Muhammad Shoaib
Department of Mathematics, COMSATS University Islamabad, Attock Campus, Kamra Road, Attock, Pakistan

Saeed EhsanAwan
Department of Electrical and Computer Engineering, COMSATS University Islamabad, Attock Campus, Kamra road, Attock, Pakistan

Khadija Kausar
Department of Mathematics, COMSATS University Islamabad, Attock Campus, Kamra Road, Attock, Pakistan

Yigang He
School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China

JunyiGao
School of Architecture and Civil Engineering, Yan'an University, Yan'an 716000, China
Shandong Provincial Lunan Geology and Exploration Institute, Jining 272100, China

Qipeng Shi
Shandong Provincial Lunan Geology and Exploration Institute, Jining 272100, China
Shandong Geothermal Clean Energy Exploration and Development Engineering Research Center, Jining 272100, China

Zakir Ullah
Department of Mathematics, University of Malakand, Chakdara, Dir(L), Khyber Pakhtunkhwa 18800, Pakistan

Ikram Ullah
Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Peshawar, KP 25000, Pakistan

Gul Zaman
Department of Mathematics, University of Malakand, Chakdara, Dir(L), Khyber Pakhtunkhwa 18800, Pakistan

Hamda Khan
Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Islamabad, Pakistan

Taseer Muhammad
Department of Mathematics, College of Sciences, King Khalid University, Abha 61413, Saudi Arabia

Saeed Tavakkoli Moghaddam
Young Researchers and Elites Club, Science and Research Branch, Islamic Azad University, Tehran, Iran

Mehrdad Javadi
Department of Mechanical Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

Seyyed Mohammad Hadji Molana
Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

F. Forouzanfar
Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

R. Tavakkoli-Moghaddam
School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
LCFC, Arts et Me´tier Paris Tech, Metz, France

M. Bashiri
Department of Industrial Engineering, Faculty of Engineering, Shahed University, Tehran, Iran

A. Baboli
DISP Laboratory, INSA-Lyon, University of Lyon, Villeurbanne, France

S. M. Hadji Molana
Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

Hamid Tohidi
College of Industrial Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

Alireza Namdari
College of Engineering, Industrial Engineering Department, Western New England University, Springfield, MA, USA

Thomas K. Keyser
College of Engineering, Industrial Engineering Department, Western New England University, Springfield, MA, USA

Julie Drzymalski
Drexel University, Philadelphia, PA 19104, USA

Martin Eigel
Weierstrass Institute for Applied Analysis and Stochastics, 10117 Berlin, Germany

Marvin Haase
Department of Mathematics, Technical University Berlin, 10623 Berlin, Germany

Johannes Neumann
Rafinex Ltd., Great Haseley OX44 7JQ, UK

Qingxin Liu
School of Computer Science and Technology, Hainan University, Haikou 570228, China

Ni Li
School of Mathematics and Statistics, Hainan Normal University, Haikou 571158, China
Key Laboratory of Data Science and Intelligence Education of Ministry of Education, Hainan Normal University, Haikou 571158, China

Heming Jia
School of Information Engineering, Sanming University, Sanming 365004, China

Qi Qi
School of Computer Science and Technology, Hainan University, Haikou 570228, China

Laith Abualigah
Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan
School of Computer Science, Universiti Sains Malaysia, Gelugor 11800, Malaysia

Yuxiang Liu
College of Physics and Information Engineering, Fuzhou University, Fuzhou 350108, China

Shafiq Ahmad
Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia

Firoz Ahmad
Department of Management Studies, Indian Institute of Science, Bangalore 560012, India
Department of Statistics and Operations Research, Aligarh Muslim University, Aligarh 202002, India

Intekhab Alam
Department of Statistics and Operations Research, Aligarh Muslim University, Aligarh 202002, India

Abdelaty Edrees Sayed
Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia

Mali Abdollahian
School of Science, College of Sciences, Technology, Engineering, Mathematics, RMIT University, Melbourne, VIC 3001, Australia

Jie Gao
Department of Engineering Mechanics, School of Aerospace Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
Hubei Key Laboratory for Engineering Structural Analysis and Safety Assessment, Huazhong University of Science and Technology, Wuhan 430074, China

Mi Xiao
The State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Yan Zhang
School of Machinery and Automation, Wuhan University of Science and Technology, Wuhan 430081, China

Liang Gao
The State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Andrea Chierici
Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409, USA

Valentina Giovacchini
Laboratory of Montecuccolino, Department of Industrial Engineering, University of Bologna, Via dei Colli 16, 40136 Bologna, Italy

Sandro Manservisi
Laboratory of Montecuccolino, Department of Industrial Engineering, University of Bologna, Via dei Colli 16, 40136 Bologna, Italy

Akhlaq Husain
Department of Applied Sciences, BML Munjal University, Gurgaon 122413, India

Manikyala Navaneeth Nanda
School of Engineering & Technology, BML Munjal University, Gurgaon 122413, India

Movva Sitaram Chowdary
School of Engineering & Technology, BML Munjal University, Gurgaon 122413, India

Mohammad Sajid
Department of Mechanical Engineering, College of Engineering, Qassim University, Buraydah 51452, Saudi Arabia

LIST OF ABBREVIATIONS

CNN	convolutional neural network
CVaR	conditional value at risk
DNN	deep neural network
FEM	finite element method
GNN	graph neural network
LSTM	long short-term memory
MC	Monte Carlo
NN	neural network
PDE	partial differential equation
TCNN	topology CNN
TLSTM	topology LSTM
TO	topology optimisation

PREFACE

Mathematical engineering is an interdisciplinary field devoted to the application of mathematical methods and techniques in engineering and industry. This book covers methods and techniques that are currently being developed for solving mathematical engineering problems. The primary focus of this book is on the real-world applicability of mathematical modelling and analysis, as well as optimization problems in engineering and industry.

The first part of the book (chapters 1 to 9) includes examples of applications of mathematical modelling for solving real-world problems in mechanical engineering (chaps. 1 and 2), biomedical industry (chaps. 3 and 4), computational fluid dynamics (chaps. 5 and 6), and other fields of industrial engineering (chaps. 7,8, and 9).

Mechanical Engineering:

Chapter 1 reviews three mathematical tools (Lie group and Lie algebra; dual quaternion and pure dual quaternion; finite screw and instantaneous screw) and their application in the design of mechanisms and robots. The aim of this review is to help readers select the appropriate method when implementing the analysis and design of robotic mechanisms.

Chapter 2 presents a spacecraft reaction wheel mathematical model that utilizes a Radial Basis Function Neural Network (RBFNN) and an improved variant of the Quantum Behaved Particle Swarm Optimization (QPSO).

Biomedical Industry:

Chapter 3 presents a mathematical model of vaporization of water inside organic materials for treating liver cancer with laser-induced thermotherapy.

Chapter 4 presents a mathematical model for the motion of cilia using non-linear rheological fluid in a symmetric channel, which is based on an analytical perturbation technique.

Computational Fluid Dynamics:

Chapter 5 presents a mathematical model for calculating thermal exploration efficiency under various geothermal well layout conditions.

Chapter 6 presents a mathematical model for the stagnation-point flow of magnetohydrodynamic Prandtl-Eyring fluid over a stretchable cylinder. This has significant applications in natural and industrial phenomena, including a flow of fluid over the tips of various objects (e.g., ships, submarines, aircrafts, and rockets) and a blood-flow in the blood vessel at the branch/sub-branch that separates into two or more directions.

Other Fields of Industrial Engineering:

Chapter 7 presents a mathematical model for the reverse supply chain of perishable goods, taking into account the sustainable production system.

Chapter 8 proposes a mathematical modelling scheme based on the Non-dominated Sorting genetic Algorithm (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO) algorithm for solving a location-routing-inventory problem in a multi-period closed-loop supply chain in the car industry.

Chapter 9 presents a mathematical model for evaluating the performance of a team associated with I.T. and the optimized size of subteams.

The second part of the book (chapters 10 to 14) includes examples of mathematical optimization methods that are relevant to engineering problems:

Chapter 10 presents a neural network-based method for solving topology optimization problems that are relevant to different engineering problems where the distribution of materials in a confined domain is distributed in some optimal manner, and it is subject to a predefined cost function representing the desired properties and constraints.

Chapter 11 presents the Hybrid Arithmetic Optimization and Golden Sine Algorithm (HAGSA) for solving industrial engineering design problems.

Chapter 12 utilizes a bounded geometric programming approach for modelling and optimizing nonlinear optimization problems in various engineering fields, such as gravel-box design, bar–truss region texture, and system reliability optimization.

Chapter 13 offers a comprehensive review of the Isogeometric Topology Optimization (ITO) methods and their applications in mechanical metamaterials, splines, and computational cost.

Chapter 14 uses mathematical tools based on the optimal control theory to show the possibility of systematically controlling natural and mixed convection flow, which is important for the field of engineering and industry.

The third part of the book, chapter 15 is devoted to an overview of fractal mathematics and its engineering-driven, industry-oriented, commercial and emerging applications (e.g. fractal landscape generation, fractal antennas, fractal image compression, and more).

Chapter

1

A SURVEY OF MATHEMATICAL TOOLS IN TOPOLOGY AND PERFORMANCE INTEGRATED MODELING AND DESIGN OF ROBOTIC MECHANISM

Xinming Huo[1] , Shuofei Yang[2] , Binbin Lian[1] , Tao Sun[1] and Yimin Song[1]

[1] Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, Tianjin 300350, China
[2] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon 999077, Hong Kong, China

ABSTRACT

Topology and performance are the two main topics dealt in the development of robotic mechanisms. However, it is still a challenge to connect them by integrating the modeling and design process of both parts in a unified frame. As the properties associated with topology and performance, finite motion and instantaneous motion of the robot play key roles in the procedure. On the

Citation: (APA): Huo, X., Yang, S., Lian, B., Sun, T., & Song, Y. (2020). A survey of mathematical tools in topology and performance integrated modeling and design of robotic mechanism. Chinese Journal of Mechanical Engineering, 33(1), 1-15.(15 pages).

purpose of providing a fundamental preparation for integrated modeling and design, this paper carries out a review on the existing unified mathematic frameworks for motion description and computation, involving matrix Lie group and Lie algebra, dual quaternion and pure dual quaternion, finite screw and instantaneous screw. Besides the application in robotics, the review of the work from these mathematicians concentrates on the description, composition and intersection operations of the finite and instantaneous motions, especially on the exponential-differential maps which connect the two sides. Furthermore, an in-depth discussion is worked out by investigating the algebraical relationship among these methods and their further progress in integrated robotic development. The presented review offers insightful investigation to the motion description and computation, and therefore would help designers to choose appropriate mathematical tool in the integrated design and modeling and design of mechanisms and robots.

INTRODUCTION

Mechanism, serving as the execution unit, is one of the essential subsystems of robot. The development of robot meeting the requirements from application scenarios depends largely on the analysis and design of robotic mechanism, which focus on topology and performance [1, 2]. Topology denotes the mechanical structure of the robotic mechanism. Topology analysis and design, also named as type synthesis, arrange the limbs and joints according to the demands on mechanism mobility, including number, sequence, type and axis (or direction) [3, 4]. Performance describes the output motion or/and force of the robotic mechanism. Performance analysis studies the kinematic, stiffness or dynamic mappings between joint space and operated space [5, 6], and performance design searches for the optimal parameters to guide the prototyping based on the task requirements [7, 8]. Conventionally, type synthesis, performance analysis and design of robotic mechanism are carried out in sequence [9]. This development procedure is to firstly invent the topological structures, select one type, build the performance models, and finally implement the optimal design. In this process, however, the type synthesis and performance design were separately implemented. The disconnection between topology and performance models leads to: (1) the difficulty in choosing particular topological structure as the performance features are usually regarded as the selecting criteria, and (2) the failure in concerning mechanism types in the optimal design since different topological structures behave differently. Therefore, it has long been a desire to unify the topology and performance analysis and design of robotic mechanisms.

Motion is the property considered in every stage of the development procedure, which is divided into two categories: finite and instantaneous motions [10, 11]. When a robotic mechanism moves along a continuous path, finite motion describes the total movement of the mechanism with respect to the initial pose [10], and instantaneous motion evaluates the velocity (acceleration, jerk, etc.) of the mechanism at current pose [11]. Literature review shows that type synthesis starts from predefined mobility described either by motion pattern based on finite motion computations [12, 13] or by constraint analysis based on instantaneous motion properties. Kinematic, stiffness and dynamic performance of the robotic mechanism relates directly to the displacement, velocity and acceleration mappings, which are analyzed either at finite motion or instantaneous motion level. It indicates that the finite and instantaneous motion description and computation are the fundamental preparation for the development of robotic mechanism. Hence, a unified mathematical framework for the finite and instantaneous motions is essential for the integrated topology and performance analysis and design.

The unified mathematical framework involves the analytical description, algebraic computation and mapping relations of the finite and instantaneous motions. The computations include mainly composition and intersection of motions. Composition is the operation for the accumulation of motions that can be the successive motions of a rigid body or the resulted motion by several rigid bodies connected by joints. For instance, the finite/instantaneous motion of a serial mechanism is calculated by the composition of finite/instantaneous motions of joints [14]. Intersection is to obtain the common part of different motions. Such operation is applied in the occasion like the finite motion of parallel mechanism whose calculation is performed by the intersection of finite motions of limbs [15]. Specially, the mapping relation of the finite and instantaneous motions is of vital importance because it is the main reason for the disconnection between topology synthesis and performance analysis of robotic mechanisms. So far, there are three mathematical tools that have been applied to the descriptions, computations and mappings of finite and instantaneous motions, i.e. matrix Lie group and Lie algebra [16], dual quaternion and pure dual quaternion [17], finite screw and instantaneous screw [18].

In the matrix Lie group, a special Euclidean group consisting of a rotation matrix and a translation vector is denoted by SE(3), whose element is rewritten into a homogenous matrix. The linear transformation can be implemented in a homogenous form, resulting in describing any finite motion by an element of the matrix representation of SE(3) [16]. By exploring the

computation rules, matrix Lie group was introduced to the mobility analysis [19,20,21] and type synthesis of mechanism [22,23,24,25,26,27]. The matrix form of Lie algebra se(3) was employed to describe instantaneous motion of mechanisms. There exists an exponential map between matrix representations of SE(3) and se(3) [28].

Dual quaternion is the extension of quaternion from real number to dual number. The composition and intersection operations are investigated, allowing the dual quaternion being used in displacement modeling of mechanism [29]. Pure dual quaternion, the dual vector, describes instantaneous motion, which was adopted to the kinematics [30] and dynamics [31]. There is an exponential map between the dual quaternion and pure dual quaternion [32].

Finite screw is proposed to describe the finite motion of rigid body in the framework of screw theory [33]. A screw triangle product [34] was defined to accomplish the composition, and the algebraic method [35] to perform the intersection of finite motions was investigated, which are employed in the type synthesis of mechanism [36]. Instantaneous screw was described as the twist of rigid body in the beginning [34]. Twist and wrench, known as the infinitesimal displacement and external force, are widely applied to the kinematic [37, 38], stiffness [39, 40], dynamic analysis and design [41,42,43] of mechanisms. It has been rigorously proved that a differential map exists in the finite and instantaneous screws.

Although these three mathematical tools have been applied at different stages of mechanism development, their capabilities in unifying the topology and performance analysis and design have not been realized. Aiming at helping designers find out effective methods in implementing integrated analysis and design so as to meet different requirements, this paper provides a comprehensive review on the mathematical tools for this topic. The paper is organized as follows. In Section 2, motions in integrated topology and performance modeling and design is discussed. Section 3 to 5 introduce the matrix Lie group and Lie algebra, dual quaternion and pure dual quaternion, finite screw and instantaneous screw, respectively, including history of development, description, computations and mapping relations of finite and instantaneous motions. A comparison of the three mathematic frameworks is illustrated from the view of algebraic structures in Section 6 following with the applications of unified mathematic tools in integrated topology and performance modeling and design. The conclusions are drawn in Section 7.

RELATIONSHIP BETWEEN DEVELOPMENT OF ROBOTIC MECHANISM AND MOTIONS

It is a long-term challenge to unify the topology and performance modeling and design in the development of robotic mechanism. To address this problem, an integrated mathematical framework should be prepared, for which the relationship between topology/performance and motions is firstly analyzed.

Topology, considered as the skeleton of a robot, includes the numbers and types of kinematic limbs as well as the adjacency and incidence among kinematic joints [44]. One particular topology corresponds to a motion pattern of the robot. Hence, type synthesis is to obtain all the possible topologies according to the expected motion pattern. The description of expected motion patterns can be classified into two formats [45, 46]. One takes the finite motion form, which expresses the displacement of the robot from the initial pose to another. Referred to the summary of the generalized procedure of type synthesis by Gao [47], the available limbs are generated by the composition and intersection operations of finite motions. The other methods begin with the instantaneous motion description. Instantaneous motion is the infinitesimal motion of the robot at the given moment. Composition operation of instantaneous motions is the basis to get the available limbs and assembly conditions in the type synthesis of robotic mechanisms. Therefore, the topology model is related with the description and calculation of finite or instantaneous motion.

Performance determines the behavior of robots in practical application. Denoted by finite and instantaneous motions, the performances of a robot can be categorized by displacement, velocity and acceleration. The displacement model of the robotic mechanism is sometimes interpreted as forward or inverse kinematics, which focuses on the mapping between the displacements of actuations and the pose of the end-effector [48]. The displacement model is constructed and calculated by the finite motions. For example, the displacement model of a serial mechanism is built by the composition of finite motions of each kinematic joint. In the case of parallel mechanisms, both composition and intersection operations of the finite motions are involved. With the displacement model, the reachable workspace of the robot can also be analyzed. The next level of performance, i.e., the velocity of robotic mechanism, is described and calculated by instantaneous motion, because both instantaneous motion and velocity denote the infinitesimal motion at given pose. The velocity and force

mapping between joint space and operated space lay the foundation of the kinematic performance analysis of serial and parallel mechanisms, which are carried out by the composition and intersection of instantaneous motions of joints and limbs. Regarding the static deformation as the perturbation of displacement, stiffness can be classified as the performance at velocity level. The stiffness modeling and analysis also rely on the composition and intersection of instantaneous motions. Finally, the performance at acceleration level refers to the dynamics, in which the velocity, acceleration and forces of the robotic mechanisms are involved. Since acceleration model is obtained by the first-order of velocity model, the performance at acceleration level are analyzed by instantaneous motions. In summary, the performance model is formulated by the description and calculation of finite or instantaneous motion.

From the above analysis, it is concluded that the topology and performance of robotic mechanisms are completely reflected by finite and instantaneous motions. Therefore, the kernel of the integrated modeling lies in the algebraic derivation between finite and instantaneous motions. Because of intrinsic connections between displacement and velocity, finite and instantaneous motions could be connected by differential and integral mappings. In this manner, if the finite motion of a continuous path is known, the instantaneous motion at the given pose could be derived, and vice versa. The composition and intersection operations of the resultant finite and instantaneous motions can also be connected, which is beneficial for implementing the integrated topology and performance modeling and design of robotic mechanism. However, these mappings cannot be performed when topology and performance models are established by different mathematical tools. Consequently, a unified mathematic framework for finite and instantaneous motions is essential for the integrated modeling.

As illustrated in Figure 1, the description, computation and mapping of finite and instantaneous motions involving in integrated modeling should be covered in a unified mathematic framework. Till now, there are three mathematical tools that have been applied, including matrix Lie group and Lie algebra, dual quaternion and pure dual quaternion, finite screw and instantaneous screw. To provide an algebraic foundation of integrated modeling and design, these unified mathematic tools are reviewed in terms of the topics in the following sections, respectively.

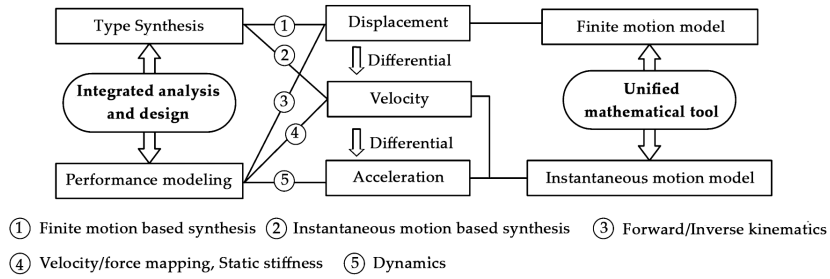


Figure 1: Relationship between motions and development of robotic mechanism.

MATRIX LIE GROUP AND LIE ALGEBRA BASED METHOD

Among the three methods applied in topology and performance integrated modeling and design of robotic mechanisms, i.e., matrix Lie group and Lie algebra based method, dual quaternion and pure dual quaternion based method, finite screw and instantaneous screw based method, the matrix based method is introduced in this section. At first, the developments on the applications of matrix Lie group and Lie algebra in robotic mechanisms are reviewed in detail, which is followed by the introductions on their expressions and computations. Based upon these, the exponential and differential mappings between them are illustrated.

Matrix Lie Group and Lie Algebra

When rotation and translation are respectively described by linear transformation and translation vector, each 6-dimensional finite motion in 3-dimensional space is thus represented as a pair of 3-dimensional orthogonal matrix and vector. In this way, the entire set of finite motions forms a Lie group under motion composition, which is called the special Euclidean group (SE(3)). Correspondingly, when 3-dimensional skew-symmetric matrix and vector are used to respectively describe angular and linear velocities, the entire set of 6-dimensional instantaneous motions constituted by the pairs of velocities form the Lie algebra se(3) of SE(3).

The matrix Lie group and Lie algebra are originated from the Erlangen program proposed by Klein [16] in the late 19th century, from then, the pairs in SE(3) and se(3) are rewritten into homogenous matrices. Both finite

motion and instantaneous motion can be expressed in homogenous forms, resulting in that any finite motion is described by an element in the matrix representation of SE(3), and that any instantaneous motion is described by an element in the matrix representation of se(3).

It was Hervé [19] who introduced the matrix Lie group into mobility analysis of mechanisms. In the 1980s and 1990s, he had been investigating description and calculation of mechanism displacement by the sub-groups of SE(3) [20, 49, 50]. The application of matrix Lie group in geometry and kinematics of mechanisms was discussed. On this basis, Hervé and Sparacino [51] employed matrix Lie group to the type synthesis (structure synthesis) of parallel mechanisms. This work was later developed by Li and Hervé [7, 8, 10, 26, 52], Lee and Hervé [53,54,55,56]. Owing to their efforts, a systematic type synthesis method by matrix Lie group was proposed. Specially, Li introduced the sub-manifolds of SE(3) as the extension of sub-groups to describe the displacements of parallel mechanisms and their limbs. Many novel parallel mechanisms were invented, including five degree-of-freedom (DoF) parallel mechanisms that could not be synthesized due to the lack of 5-dimensional sub-groups of SE(3). Besides applying matrix Lie group to type synthesis, Fanghella and Galletti [57, 58] discussed the approximate computation algorithms of matrix Lie group. Composition of two sub-groups was computed by their minimum envelope group, while the intersection of two sub-groups was performed by searching for the maximum common group. All possible cases of sub-group composition and intersection were listed. This computation method is different from the analytical algorithms in Baker-Campbell-Hausdorff formula [59, 60] and is easier to be directly applied. Meng [61] also engaged in giving the clear intersection algorithms of sub-groups. They obtained the intersection of Lie sub-groups by solving the intersection of the corresponding Lie sub-algebras. In their work, the matrix form of Lie algebra se(3) was employed to describe instantaneous motion of mechanisms. The similar method was employed by Wu [62,63,64] in type synthesis of quotient mechanisms, and by Liu [65] in type synthesis of mechanisms with adjoint-invariant sub-manifolds of SE (3). All these contributions lead to the topology modeling by using matrix Lie group and its sub-sets. In 1983, Brockett [28, 66] established the framework of matrix Lie group and Lie algebra for mechanism modeling and analysis. By investigating the exponential mapping between matrix representations of SE(3) and se(3), he set up the connection between finite and instantaneous motions of mechanisms. His work was further extended by Li [67,68,69], Park [70,71,72], Chen [73,74,75], Chen [76,77,78] and their colleagues,

leading to an integrated framework for kinematics, dynamics, calibration, and control of mechanisms.

Matrix Lie Group and Its Computations

As introduced in Section 2.1, the matrix representation of SE(3) is the entire set of homogeneous matrices that describe all the linear transformations in the Euclidean space. This matrix Lie group can be used to describe all the finite motions of a rigid body or a mechanism. Hence, the finite motion description based upon matrix Lie group can be expressed as,

$$\mathrm{SE}(3) = \left\{ \boldsymbol{g} \middle| \boldsymbol{g} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}, \boldsymbol{R} \in \mathrm{SO}(3), \boldsymbol{t} \in \mathbb{R}^3 \right\}, \tag{1}$$

where SO(3) denotes the special orthogonal group consisting of the orthogonal matrices that describe rotations, $\mathbb{R}^3$ denotes the 3-dimensional vector space, R is an arbitrary element in SO(3) which represents the rotation matrix about the Chasles' axis, t is the translation vector along that axis. R and t involve the Chasles' axis $\left(\boldsymbol{s}_f^{\mathrm{T}} \ (\boldsymbol{r}_f \times \boldsymbol{s}_f)^{\mathrm{T}} \right)^{\mathrm{T}}$ together with the corresponding rotational angle θ and translational distance t. The expressions of R and t can be referred to Ref. [28], as

$$\boldsymbol{R} = \boldsymbol{E}_3 + \sin\theta \tilde{\boldsymbol{s}}_f + (1 - \cos\theta)\left(\tilde{\boldsymbol{s}}_f\right)^2, \tag{2}$$

$$\boldsymbol{t} = (\boldsymbol{E}_3 - \boldsymbol{R})\left(\boldsymbol{r}_f - \boldsymbol{s}_f^{\mathrm{T}} \boldsymbol{r}_f \boldsymbol{s}_f\right) + t\boldsymbol{s}_f, \tag{3}$$

where E_3 is a three-order unit matrix, $\tilde{\boldsymbol{s}}_f$ is the skew-symmetric matrix that denotes the cross product of $\boldsymbol{s}_f . \boldsymbol{r}_f$ expresses the position vector of the Chasles' axis.

When the matrix Lie group theory is applied in topology modeling and analysis of robotic mechanisms, the finite motion generated by each 1-DoF joint can be described by a 1-dimensional sub-group of SE(3). Following this manner, the motion of each limb is the composition of all its joints' motions, and the mechanism motion is the intersection of the limbs' motions.

The composition of matrix Lie sub-groups is performed by matrix multiplication. This is because any Lie sub-group can be regarded as the composition of several 1-dimensional sub-groups. Hence, the composition of finite motions can be expressed by the multiplication of a sequence of 1-DoF finite motions as,

$$\boldsymbol{M} = \boldsymbol{M}_n \cdots \boldsymbol{M}_2 \boldsymbol{M}_1, \tag{4}$$

where $\boldsymbol{M}_k \ (k = 1, 2, \ldots, n)$ denotes the 1-dimensional sub-group that describes the *k*th finite motion in the sequence.

Consider that the elements in each 1-dimensional sub-group can be expressed by exponential expressions, M_k can be obtained as

$$\boldsymbol{M}_k = \left\{ e^{\theta_k \tilde{\boldsymbol{\xi}}_{f,k}} \middle| \theta_k \in \mathbb{R} \right\}, \tag{5}$$

where ξξ~f,kξξ~f,k is the homogenous matrix that represents the Chasles' axis that corresponds to MMkMMk and a pitch, as

$$\tilde{\boldsymbol{\xi}}_{f,k} = \begin{bmatrix} \tilde{\boldsymbol{s}}_{f,k} & \boldsymbol{r}_{f,k} \times \boldsymbol{s}_{f,k} + \frac{t_k}{\theta_k} \boldsymbol{s}_{f,k} \\ \boldsymbol{0} & 0 \end{bmatrix}. \tag{6}$$

The denotations of the symbols in Eq. (6) can be referred to those in Eqs. (2) and (3).

Taking the exponential form, Eq. (4) can be rewritten as,

$$\boldsymbol{M} = \left\{ e^{\theta_n \tilde{\boldsymbol{\xi}}_{t,n}} \cdots e^{\theta_2 \tilde{\boldsymbol{\xi}}_{t,2}} e^{\theta_1 \tilde{\boldsymbol{\xi}}_{t,1}} \middle| \theta_1, \theta_2, \ldots, \theta_n \in \mathbb{R} \right\}. \tag{7}$$

In order to obtain the expansion form of Eq. (7), the Baker-Campbell-Hausdorff formula is employed. The composition of two 1-DoF finite motions could be performed as,

$$e^{\theta_{k+1} \tilde{\boldsymbol{\xi}}_{f,k+1}} e^{\theta_k \tilde{\boldsymbol{\xi}}_{f,k}} = e^{f\left(\theta_{k+1} \tilde{\boldsymbol{\xi}}_{f,k+1}, \theta_k \tilde{\boldsymbol{\xi}}_{f,k}\right)}, \tag{8}$$

where

$$\begin{aligned} f\left(\theta_{k+1} \tilde{\boldsymbol{\xi}}_{f,k+1}, \theta_k \tilde{\boldsymbol{\xi}}_{f,k}\right) = {} & \theta_k \tilde{\boldsymbol{\xi}}_{f,k} + \theta_{k+1} \tilde{\boldsymbol{\xi}}_{f,k+1} + \frac{1}{2}\left[\theta_{k+1} \tilde{\boldsymbol{\xi}}_{f,k+1}, \theta_k \tilde{\boldsymbol{\xi}}_{f,k}\right] \\ & + \frac{1}{12}\left(\left[\theta_{k+1} \tilde{\boldsymbol{\xi}}_{f,k+1}, \left[\theta_{k+1} \tilde{\boldsymbol{\xi}}_{f,k+1}, \theta_k \tilde{\boldsymbol{\xi}}_{f,k}\right]\right]\right. \\ & \left. + \left[\theta_k \tilde{\boldsymbol{\xi}}_{f,k}, \left[\theta_k \tilde{\boldsymbol{\xi}}_{f,k}, \theta_{k+1} \tilde{\boldsymbol{\xi}}_{f,k+1}\right]\right]\right) + \cdots \end{aligned}$$

Herein, $\left[\theta_{k+1} \tilde{\boldsymbol{\xi}}_{f,k+1}, \theta_k \tilde{\boldsymbol{\xi}}_{f,k}\right] = \theta_k \theta_{k+1}\left(\tilde{\boldsymbol{\xi}}_{f,k+1} \tilde{\boldsymbol{\xi}}_{f,k} - \tilde{\boldsymbol{\xi}}_{f,k} \tilde{\boldsymbol{\xi}}_{f,k+1}\right)$ is defined as the Lie bracket. It is found that algebraic computation becomes more complicated and difficult because of higher order items, especially for the cases of more than two motions.

Intersection of finite motions is the maximum common sub-group or sub-manifold contained in all motions. By using the property of the exponential

expression in Eq. (5), Meng [20] partly solved this problem by mapping the intersection of the Lie sub-groups to Lie algebra level. Till now, intersection of finite motions by matrix Lie sub-groups and the composited manifolds (the product of several Lie sub-groups) is mainly based upon specific principles, such as the cases given by Fanghella and Galletti [16, 17]. However, these operations are difficult to implement in an analytical manner and be applied for all the motion patterns. There is no generic intersection algorithm for matrix Lie sub-groups and the composited manifolds yet.

Matrix Lie Algebra and Its Computations

As the counterpart of matrix Lie group SE(3), its matrix Lie algebra se(3) is employed to describe the instantaneous motions of robotic mechanisms, as

$$\mathrm{se}(3) = \left\{ \omega\tilde{\boldsymbol{\xi}}_t | \omega\tilde{\boldsymbol{\xi}}_t = \begin{pmatrix} \tilde{\boldsymbol{\omega}} & \boldsymbol{v} \\ \mathbf{0} & 0 \end{pmatrix}, \quad \boldsymbol{\omega}, \boldsymbol{v} \in \mathbb{R}^3 \right\}, \tag{9}$$

where $\boldsymbol{\omega}$ and $\boldsymbol{v}$ are angular and linear velocities in 3-dimensional vector forms.

Any element in se(3) can be rewritten into vector form as

$$(\boldsymbol{\omega} \quad \boldsymbol{v})^{\mathrm{T}} = \omega\boldsymbol{\xi}_t, \tag{10}$$

$$\boldsymbol{\xi}_t = (\boldsymbol{s}_t \quad \boldsymbol{r}_t \times \boldsymbol{s}_t + p_t\boldsymbol{s}_t)^{\mathrm{T}}, \tag{11}$$

where ξ_t is the normalized unit velocity, ω is its amplitude, and p_t denotes the pitch. r_t expresses the position of the Mozzi's axis.

When the matrix Lie algebra theory is applied in performance modeling and analysis of robotic mechanisms, 1-dimensional sub-space of se(3) is employed to describe the instantaneous motion generated by 1-DoF joint. In this way, the composition of the motions of all joints in a limb leads to the limb motion, and the intersection of all the limbs' motions results in the mechanism motion.

As is well known, se(3) is a 6-dimensional vector space. The composition of matrix Lie sub-spaces is performed by linear addition as,

$$\begin{aligned} \boldsymbol{T} &= \mathrm{span}\{\boldsymbol{T}_1 \cup \boldsymbol{T}_2 \cup \cdots \cup \boldsymbol{T}_n\} \\ &= \boldsymbol{T}_1 \oplus \boldsymbol{T}_2 \oplus \cdots \oplus \boldsymbol{T}_n, \end{aligned} \tag{12}$$

where '$\oplus$' denotes the combination operation of linear vector spaces. The intersection of several sub-spaces can be obtained through linear computations, as

$$\begin{aligned} \boldsymbol{T} &= \boldsymbol{T}_1 \cap \boldsymbol{T}_2 \cdots \cap \boldsymbol{T}_n \\ &= \left(\boldsymbol{T}_1^{\perp} \oplus \boldsymbol{T}_2^{\perp} \cdots \oplus \boldsymbol{T}_n^{\perp} \right)^{\perp}, \end{aligned} \tag{13}$$

where $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, …, $\boldsymbol{T}_n$ denote n sub-spaces of se(3). The computations shown in Eqs. (12) and (13) are easy to be conducted because they both fall in the area of linear algebra.

Mappings between Matrix Lie Group and Lie Algebra

According to the physical principle, finite motion (displacement) is the integral of instantaneous motion (velocity), and velocity is the differential of displacement. When the displacement and velocity are described by matrix Lie group and Lie algebra, a differential-exponential mapping can be formulated between them as follows,

$$\begin{aligned} d\boldsymbol{g} &= de^{\theta\tilde{\boldsymbol{\xi}}_f} \\ &= \dot{\theta}\tilde{\boldsymbol{\xi}}_f e^{\theta\tilde{\boldsymbol{\xi}}_f} \\ &= \omega\tilde{\boldsymbol{\xi}}_f e^{\theta\tilde{\boldsymbol{\xi}}_f}, \end{aligned} \tag{14}$$

$$e^{\theta\tilde{\boldsymbol{\xi}}_t} = \boldsymbol{g}, \tag{15}$$

The interpretations on the above two equations are given as follows:

1. The differential of g at $\theta = 0$ is $\omega\tilde{\boldsymbol{\xi}}_f$. As the Chasles' axis is coincident with the axis of the velocity when $\theta = 0$, the differential of g at $\theta = 0$ is an element of se(3). This is because se(3) is the tangent space of SE(3) at the identity element (the unit matrix).
2. The exponential of $\omega\tilde{\boldsymbol{\xi}}_t$ with respect to the time results in g, which means that the exponential of any elements in se(3) with respect to the time leads to the elements in SE(3).

The differential-exponential mapping between matrix Lie group SE(3) and Lie algebra se(3) leads to the following 1-DoF case, as

$$\begin{aligned} d\boldsymbol{M}_k|\theta_k = 0 &= \left\{ de^{\theta_k \tilde{\boldsymbol{\xi}}_{f,k}} |\theta_k = 0 \right\} \\ &= \left\{ \dot{\theta}_k \tilde{\boldsymbol{\xi}}_{f,k} e^{\theta_k \tilde{\boldsymbol{\xi}}_{f,k}} |\theta_k = 0 \right\} \\ &= \left\{ \omega_k \tilde{\boldsymbol{\xi}}_{t,k} | \omega_k \in \mathbb{R} \right\} \\ &= \boldsymbol{T}_k, \end{aligned} \tag{16}$$

$$\left\{ e^{\theta_n \tilde{\boldsymbol{\xi}}_{f,n}} |\theta_k \in \mathbb{R} \right\} = \boldsymbol{M}_k, \tag{17}$$

and multi-DoF cases, as

$$\begin{aligned} & d(\boldsymbol{M}_n \cdots \boldsymbol{M}_2 \boldsymbol{M}_1)\big|_{\theta_k=0,\, k=1,2,\ldots,n} \\ &= \left\{ d\left(e^{\theta_n \tilde{\boldsymbol{\xi}}_{f,n}} \cdots e^{\theta_2 \tilde{\boldsymbol{\xi}}_{f,2}} e^{\theta_1 \tilde{\boldsymbol{\xi}}_{f,1}} \right) \Big|_{\theta_k=0,\, k=1,2,\ldots,n} \right\} \\ &= \left\{ \omega_1 \tilde{\boldsymbol{\xi}}_{t,1} + \omega_2 \tilde{\boldsymbol{\xi}}_{t,2} + \cdots + \omega_n \tilde{\boldsymbol{\xi}}_{t,n} \Big| \omega_k \in \mathbb{R},\, k = 1, 2, \ldots, n \right\} \\ &= \boldsymbol{T}_1 \oplus \boldsymbol{T}_2 \oplus \cdots \oplus \boldsymbol{T}_n, \end{aligned} \tag{18}$$

$$\left\{ e^{\theta_n \tilde{\boldsymbol{\xi}}_{t,n}} \cdots e^{\theta_2 \tilde{\boldsymbol{\xi}}_{t,2}} e^{\theta_1 \tilde{\boldsymbol{\xi}}_{t,1}} \Big| \theta_1, \theta_2, \ldots, \theta_n \in \mathbb{R} \right\} = \boldsymbol{M}_n \cdots \boldsymbol{M}_2 \boldsymbol{M}_1. \tag{19}$$

DUAL QUATERNION AND PURE DUAL QUATERNION BASED METHOD

The review of dual quaternion and pure dual quaternion based method is provided in this section. Firstly, the application of this method in topology and performance modeling and design of robotic mechanisms is traced. Secondly, the basic formats together with their composition and intersection operations are discussed. Finally, the exponential/Cayley- differential maps between finite and instantaneous motions are constructed in the form of quaternionic algebras.

Dual Quaternion and Pure Dual Quaternion

As the representations of SE(3) and se(3), respectively, dual quaternion and pure dual quaternion are applied to describe the transformation from one pose to another and the velocity at any instant. Dual quaternion utilizes eight parameters by presenting a scalar with the cosine of half the dual angle [17] and further six numbers by integrating the direction and position of the motion axis with the sine of half the dual angel. Herein, dual angle integrated

the rotational angle and linear displacement by dual operator. Pure dual quaternion is also called dual vector, which includes six elements and is defined by means of the unit axis and amplitude of instantaneous motion.

The dual quaternion and pure dual quaternion based method can be traced back to Euler-Rodrigues' parameters and Euler-Rodrigues' formula [79] in the 18th century. Hamilton [80] and Rodrigues [81] did some pioneering work in this field. Based on that, Clifford [82] transformed rotation about an axis into translation parallel to the axis and proposed the concept of "biquaternion" in the investigation of geometry and algebra. Biquaternion was then applied to motion description and termed as dual quaternion [17].

It was pointed out the dual quaternion is the extension of quaternion from real number to dual number. According to the "transference principle" [83, 84], the algorithms for quaternions can be applied to the algorithms for dual quaternions. In this way, the composition of two dual quaternions could be computed by quaternion multiplication [85], i.e., Euler-Rodrigues' formula with dual angles. As for the intersection algorithms, Sun [86] employed analytical derivations to deal with the intersection of the sets of dual quaternions. Mechanism analysis by dual quaternion was implemented by McAulay [87] for the first time who utilized dual quaternion to describe rigid body displacement. Later on, dual quaternion was used in the kinematics of mechanisms from a geometrical prospective by Refs. [88, 89] and Blaschke [90]. Kong studied the method for motion mode analysis of single-loop and closed-loop spatial mechanisms by formulating a set of kinematic loop equations based on dual quaternions [91, 92]. It was proved in ref. [93] that dual quaternions facilitate to avoid singularities in the analysis of finite motion. Besides robotic kinematics, joint stiffness identification and deformation compensation algorithms for serial robots were constructed [94]. Apart from the applications of dual quaternion in finite motion description, pure dual quaternion (dual vector) was adopted to describe instantaneous motion. For instance, Yang and Freudenstein [29, 95] combined both dual quaternion and pure dual quaternion to analyze the displacement and velocity of a spatial four-link mechanism. Similar researches on the mechanism kinematic analysis by dual quaternions can be found in [30, 96, 97]. For the mechanism design, McCarthy et al. [98, 99] formulated forward and inverse kinematic equations of spatial serial chains and proposed a semi-analytical design method. These kinematic equations are obtained by the exponential map between pure dual quaternion and dual quaternion. Selig [32] built the dynamic model of mechanisms

using quaternions [31]. In his research, the Cayley map in dual quaternion theory was constructed concerning that the entire set of dual quaternions is a double cover of SE(3). The intrinsic connections between quaternion exponential map and Euler-Rodrigues' formula were deeply investigated by Dai [100], relating dual quaternions with other representations of SE(3). Taking advantages of these mappings, the integrated method was also used in calibration algorithms [101, 102], path planning and control strategies [103, 104]. Motivated by the arithmetic operations of dual quaternions, Cohen developed the concept of hyper dual quaternion currently, which was applied for the displacement and velocity modeling of serial mechanisms [105].

Dual Quaternion and its Computations

The dual quaternion is the extension of quaternion from real number to dual number. Rotation axis and rotational angle in quaternion can be replaced with dual axis and dual angle. Thus, the 1-DoF finite motion is described by dual quaternion as

$$\boldsymbol{D} = \cos\frac{\hat{\theta}}{2} + \sin\frac{\hat{\theta}}{2}\boldsymbol{L}_f^{\wedge}, \tag{20}$$

where $\hat{\theta} = \theta + \varepsilon t$ denotes the dual angle. It has the cosine and sine functions as

$$\cos\frac{\hat{\theta}}{2} = \cos\frac{\theta}{2} - \frac{t}{2}\sin\frac{\theta}{2}\varepsilon, \quad \sin\frac{\hat{\theta}}{2} = \sin\frac{\theta}{2} + \frac{t}{2}\cos\frac{\theta}{2}\varepsilon,$$

where ε is the dual unit and $\varepsilon^2=0$. ()^ in this paper denotes a vector in pure dual quaternion form. $\boldsymbol{L}_f^{\wedge}$ is the pure dual quaternion form of the Plücker coordinates of the Chasles' axis, which can be denoted as $\boldsymbol{L}_f^{\wedge} = \boldsymbol{s}_f^{\wedge} + \varepsilon \boldsymbol{r}_f^{\wedge} \times \boldsymbol{s}_f^{\wedge}$. Herein, $\boldsymbol{s}_f^{\wedge}$ and $\boldsymbol{r}_f^{\wedge}$ are the unit direction dual vector and position dual vector of the Chasles' axis.

$$\boldsymbol{s}_f^{\wedge} = s_{f,1}\boldsymbol{i} + s_{f,2}\boldsymbol{j} + s_{f,3}\boldsymbol{k},$$

$$\boldsymbol{r}_f^{\wedge} = r_{f,1}\boldsymbol{i} + r_{f,2}\boldsymbol{j} + r_{f,3}\boldsymbol{k},$$

where $s_{f,u}$ and $r_{f,u}$ $(u = 1, 2, 3)$ are scalar coefficients of Plücker coordinates. i, j, k are plural units with the properties,

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = -1, \quad \boldsymbol{ij} = \boldsymbol{k}, \quad \boldsymbol{ijk} = -1. \tag{21}$$

For a serial mechanism or limbs in parallel mechanism, the finite motion generated by all 1-DoF joints can be solved by the composition operation, which can be rewritten utilizing quaternion multiplication [80, 81],

$$\boldsymbol{D}_{12\ldots n} = \boldsymbol{D}_n \ldots \boldsymbol{D}_2 \boldsymbol{D}_1, \tag{22}$$

$$\boldsymbol{D}_{12\ldots n} = \boldsymbol{D}_n \ldots \left(\begin{array}{l} \cos\dfrac{\hat{\theta}_1}{2}\cos\dfrac{\hat{\theta}_2}{2} + \cos\dfrac{\hat{\theta}_1}{2}\sin\dfrac{\hat{\theta}_2}{2}\boldsymbol{L}_{f,2}^{\wedge} \\ + \sin\dfrac{\hat{\theta}_1}{2}\cos\dfrac{\hat{\theta}_2}{2}\boldsymbol{L}_{f,1}^{\wedge} \\ + \sin\dfrac{\hat{\theta}_1}{2}\sin\dfrac{\hat{\theta}_2}{2}\boldsymbol{L}_{f,2}^{\wedge}\boldsymbol{L}_{f,1}^{\wedge} \end{array} \right), \tag{23}$$

It is noted that the motion of moving platform in a parallel mechanism and that generated by each limb is in equilibrium. Therefore, having the analytical resultant motion of limbs at hand, the finite motion of the moving platform can be obtained by the intersection operation as,

$$\boldsymbol{D} = \boldsymbol{D}_1 = \boldsymbol{D}_2 = \cdots = \boldsymbol{D}_i = \cdots = \boldsymbol{D}_m, \quad i = 1, 2, \ldots, m. \tag{24}$$

Thanks to the expression and quaternion multiplication defined in Eqs. (20), (21), the finite motion of each limb could be determined by formulating equations as Eq. (24).

Pure Dual Quaternion and Its Computations

The velocity of the rigid body at any instant is specified by a dual vector, which connects two 3-D vectors by dual operator. In this way, the format of pure dual quaternion is introduced here

$$\boldsymbol{d} = \omega \boldsymbol{L}_t^{\wedge} + \varepsilon \omega p \boldsymbol{s}_t^{\wedge}, \tag{25}$$

where $\boldsymbol{L}_t^{\wedge} = \boldsymbol{s}_t^{\wedge} + \varepsilon \boldsymbol{r}_t^{\wedge} \times \boldsymbol{s}_t^{\wedge}$ is the pure dual quaternion form of the Plücker coordinates of the Mozzi's axis. Herein, $\boldsymbol{s}_t^{\wedge}, \boldsymbol{r}_t^{\wedge}$ are the unit direction dual vector and position dual vector of the Mozzi's axis.

$$\boldsymbol{s}_t^{\wedge} = s_{t,1}\boldsymbol{i} + s_{t,2}\boldsymbol{j} + s_{t,3}\boldsymbol{k},$$

$$\boldsymbol{r}_t^{\wedge} = r_{t,1}\boldsymbol{i} + r_{t,2}\boldsymbol{j} + r_{t,3}\boldsymbol{k},$$

where $s_{t,u}$ and $r_{t,u}$ $(u = 1, 2, 3)$ are scalar coefficients of Plücker coordinates.

The pure dual quaternions are Lie algebra elements with both well-defined addition and multiplication. Thus, when pure dual quaternion is applied in the performance modeling and design of robotic mechanisms, the composition and intersection operations can be performed as linear algebra, referring to Eq. (12) and Eq. (13), respectively.

Mappings between Dual Quaternion and Pure Dual Quaternion

Similar to matrix Lie group and Lie algebra, the exponential map and Cayley map exist from pure dual quaternion to dual quaternion, which are given by,

$$\begin{aligned} e^{\hat{\theta}\boldsymbol{L}_f^{\wedge}} &= \left(1 - \frac{\hat{\theta}^2}{2} + \frac{\hat{\theta}^4}{2} + \cdots\right) + \left(1 - \frac{\hat{\theta}^3}{3} + \frac{\hat{\theta}^5}{5} + \cdots\right)\boldsymbol{L}_f^{\wedge} \\ &= \cos\hat{\theta} + \sin\hat{\theta}\boldsymbol{L}_f^{\wedge} = \boldsymbol{D}, \end{aligned} \tag{26}$$

$$\begin{aligned} \mathrm{Cay}_{\boldsymbol{D}}\left(\boldsymbol{L}_f^{\wedge}\right) &= \frac{1 + 2\left|\boldsymbol{s}_t^{\wedge}\right|^2 - \left|\boldsymbol{s}_t^{\wedge}\right|^4}{\left(1 + \left|\boldsymbol{s}_t^{\wedge}\right|^2\right)^2} + \frac{2 + 4\left|\boldsymbol{s}_t^{\wedge}\right|^4}{\left(1 + \left|\boldsymbol{s}_t^{\wedge}\right|^2\right)^2}\boldsymbol{L}_f^{\wedge} \\ &\quad + \frac{2}{\left(1 + \left|\boldsymbol{s}_t^{\wedge}\right|^2\right)^2}\boldsymbol{L}_f^{\wedge 2} + \frac{2}{\left(1 + \left|\boldsymbol{s}_t^{\wedge}\right|^2\right)^2}\boldsymbol{L}_f^{\wedge 3} = \boldsymbol{D}. \end{aligned} \tag{27}$$

In the modeling process of robots, exponential map facilitates to connect the velocity and the possible displacements allowed by the joint. It would be convenient to formulate the topology or kinematic models of serial mechanisms or open-loop limbs by taking the axes and motion variables of joints in an analytical manner. Cayley map is always used in numerical methods since it does not need so many trigonometric function calls and will avoid cost consuming. For multi-DoF, the maps could be expanded as

$$e^{\hat{\theta}_n \boldsymbol{L}_{f,n}^{\wedge}} \cdots e^{\hat{\theta}_2 \boldsymbol{L}_{f,2}^{\wedge}} e^{\hat{\theta}_1 \boldsymbol{L}_{f,1}^{\wedge}} \Big|_{\theta_1,\theta_2,\cdots,\theta_n \in \mathbb{R}} = \boldsymbol{D}_n \cdots \boldsymbol{D}_2 \boldsymbol{D}_1, \tag{28}$$

$$\mathrm{Cay}_{\boldsymbol{D}}\left(\boldsymbol{L}_{f,n}^{\wedge}\right) \cdots \mathrm{Cay}_{\boldsymbol{D}}\left(\boldsymbol{L}_{f,1}^{\wedge}\right) = \boldsymbol{D}_n \cdots \boldsymbol{D}_1. \tag{29}$$

When the topology/displacement models are obtained at first, differential mapping between dual quaternion and pure dual quaternion would help to get the velocities. It could be executed by taking differentiations of dual quaternion D with respect to time.

$$\dot{\boldsymbol{D}}\big|_{\hat{\theta}=0} = \dot{\hat{\theta}} \boldsymbol{L}_f^{\wedge} e^{\hat{\theta} \boldsymbol{L}_f^{\wedge}} \Big|_{\hat{\theta}=0} = \boldsymbol{L}_f^{\wedge} = \boldsymbol{L}_t^{\wedge}. \tag{30}$$

It indicates that the time derivative of D at the initial pose is exactly the corresponding pure dual quaternion $\boldsymbol{L}_t^{\wedge}$ at the instant $\hat{\theta} = 0$. This rule is also proved in the multi-DoF cases,

$$\begin{aligned} & d(\boldsymbol{D}_n \cdots \boldsymbol{D}_2 \boldsymbol{D}_1)\big|_{\hat{\theta}_k=0,\, k=1,2,\ldots,n} \\ & = d\left(e^{\hat{\theta}_n \boldsymbol{L}_{f,n}^{\wedge}} \cdots e^{\hat{\theta}_2 \boldsymbol{L}_{f,2}^{\wedge}} e^{\hat{\theta}_1 \boldsymbol{L}_{f,1}^{\wedge}}\right)\Big|_{\substack{\hat{\theta}_k=0,\\ k=1,2,\cdots,n}} \\ & = \dot{\hat{\theta}}_1 \boldsymbol{L}_{f,1}^{\wedge} + \dot{\hat{\theta}}_2 \boldsymbol{L}_{f,2}^{\wedge} + \cdots + \dot{\hat{\theta}}_n \boldsymbol{L}_{f,n}^{\wedge} \Big| \dot{\hat{\theta}}_n \in \mathbb{R},\ k = 1, 2, \ldots, n. \end{aligned} \tag{31}$$

FINITE SCREW AND INSTANTANEOUS SCREW BASED METHOD

In this section, integrated screw theory based method is presented beginning with the progress achieved in topology and performance modeling and design of robotic mechanisms. Then the description and computation of motions by finite and instantaneous screws are introduced. After that, the differential mapping between them is formulated.

Finite Srew and Instantaneous Screw

According to Chasles's theorem [10], a general rigid-body displacement could be described as a rotation about a line followed by a translation in the same direction as the rotation axis. Such a line is specified by the finite motion axis, a rotation angle, and a pitch. Motivated by this point, finite screw is invented to describe the finite motion in a 6-D quasi-vector format. Meanwhile, instantaneous motion could be expressed by the line in linear

subspace, representing instantaneous motion axis with angular and linear velocities. Instantaneous screw was proposed based on spatial vectors with the definition of pitch. By this means, finite and instantaneous motions are depicted in the view of geometry by finite and instantaneous screws.

The finite screw and instantaneous screw based method origins from screw theory proposed in the 19th century. In the beginning, Chasles [10] proposed the concept of twist motion of a rigid body. It was further developed by Poinsot and Plücker [11], in which screw coordinates of infinitesimal displacement and external force were involved. They were named as twist and wrench, respectively. The reciprocal property of twist and wrench was later explored by Ball [106] and Klein [107, 108].

In the book "A treatise on the theory of screws" [109], Ball discussed kinematics and dynamics of an arbitrary rigid body by screw theory. It laid a solid foundation for the mechanism analysis by Hunt [110] who proposed the screw based kinematic and dynamic modeling method for serial, parallel and closed-loop mechanisms. Following Hunt's work, substantial researches were carried out for the mechanism analysis and design based on instantaneous screw, such as type synthesis [3, 45, 111], statics and kinetics [112, 113], performance evaluation and optimization [114, 115]. Besides the applications of instantaneous screw, finite screw, termed by Dimentberg [116], was proposed to describe the finite motion of rigid body. On this track, the format of finite screw, including the pitch and amplitude, was intensively studied by Parkin [117, 118], Hunt [119], Dai [33] and Huang [120,121,122]. Other than description of finite screw, the computation was another difficult problem. To this end, Roth [123] defined screw triangle product to accomplish finite screw composition with the aid of Euler-Rodrigues' formula. This definition had been widely accepted. From then on, many scholars focused on finding out concise algorithm for the screw triangle product [124,125,126,127,128,129]. Through the linear combination of two original screws, their translational parts and the screw along their common perpendicular, Huang [130] simplified the screw triangle product. However, the nonlinear intersection of finite screws was analyzed in linear subspaces [131], which leads to inappropriate results. In terms of the finite screw intersection, Sun [18, 35, 36] presented an algebraic method. For the first time, Dai [34] formulated the mapping between finite and instantaneous screws, and defined correlations among screw theory, matrix Lie group and quaternions [132]. Based on the contribution of Dai, Sun [18, 133] expanded the differential mapping to the analysis of spatial mechanisms. For the applications of finite screw to mechanism analysis, Huang [120,121,122]

built the forward kinematic equations of some serial mechanisms. Sun and his colleagues [133,134,135,136] proposed a generic method to formulate motion equations for different types of mechanisms. Finite motion based type synthesis and instantaneous motion based kinematic analysis of parallel mechanisms are integrated by a consistent algebraic manner in their method.

Finite Screw and Its Computations

Finite motion description by screw directly reflect the Chasles' axis together with the angular and linear displacements. The 1-DoF finite motion could be parameterized as finite screw in 6-dimensional quasi-vector form as

$$\boldsymbol{S}_f = 2\tan\frac{\theta}{2}\begin{pmatrix}\boldsymbol{s}_f\\ \boldsymbol{r}_f\times\boldsymbol{s}_f\end{pmatrix} + t\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{s}_f\end{pmatrix}, \tag{32}$$

where $\boldsymbol{s}_f, \boldsymbol{r}_f, \theta, t$ have the same meanings as given in Eqs. (2), (3).

Composition operation of finite screws could be performed by screw triangle product signed as "$\triangle$". The composition of two 1-dimensional finite screws results in a linear combination of the two original screws, their translational parts and the screw along their common perpendicular. In this way, the analytical expression of the composited motion can be easily obtained in an approximately linear manner, which simplifies the nonlinear composition of finite motions

$$\boldsymbol{S}_{f,1\cdots n} = \boldsymbol{S}_{f,1} \triangle \boldsymbol{S}_{f,2} \cdots \triangle \boldsymbol{S}_{f,n}, \tag{33}$$

where

$$\boldsymbol{S}_{f,1}\triangle\boldsymbol{S}_{f,2} = \frac{1}{1-\tan\frac{\theta_1}{2}\tan\frac{\theta_2}{2}\boldsymbol{s}_{f,2}^{\mathrm{T}}\boldsymbol{s}_{f,1}}\begin{pmatrix}\boldsymbol{S}_{f,1}+\boldsymbol{S}_{f,2}-\frac{1}{2}\boldsymbol{S}_{fc,12}\\ -\boldsymbol{S}_{fp,1}-\boldsymbol{S}_{fp,2}\end{pmatrix},$$

$$\boldsymbol{S}_{f,i} = 2\tan\frac{\theta_i}{2}\begin{pmatrix}\boldsymbol{s}_{f,i}\\ \boldsymbol{r}_{f,i}\times\boldsymbol{s}_{f,i}\end{pmatrix} + t_i\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{s}_{f,i}\end{pmatrix}, \quad i=1,2,$$

$$\boldsymbol{S}_{fc,12} = \begin{pmatrix}2\tan\frac{\theta_1}{2}\boldsymbol{s}_{f,1}\times 2\tan\frac{\theta_2}{2}\boldsymbol{s}_{f,2}\\ \left(2\tan\frac{\theta_1}{2}\boldsymbol{r}_{f,1}\times\boldsymbol{s}_{f,1}+t_1\boldsymbol{s}_{f,1}\right)\times 2\tan\frac{\theta_2}{2}\boldsymbol{s}_{f,2}\\ +2\tan\frac{\theta_1}{2}\boldsymbol{s}_{f,1}\times\left(2\tan\frac{\theta_2}{2}\boldsymbol{r}_{f,2}\times\boldsymbol{s}_{f,2}+t_2\boldsymbol{s}_{f,2}\right)\end{pmatrix},$$

$$\boldsymbol{S}_{fp,1} = \tan\frac{\theta_1}{2}\tan\frac{\theta_2}{2}t_2\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{s}_{f,1}\end{pmatrix}, \quad \boldsymbol{S}_{fp,2} = \tan\frac{\theta_1}{2}\tan\frac{\theta_2}{2}t_1\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{s}_{f,2}\end{pmatrix}.$$

Similar to the intersection algorithm of dual quaternions, the intersection of finite screws is achieved through formulating the simultaneous equations and solving the common range of the finite screw expressions

$$\boldsymbol{S}_{f,1} = \boldsymbol{S}_{f,2} = \cdots = \boldsymbol{S}_{f,i} = \cdots = \boldsymbol{S}_{f,m}, \quad i = 1, 2, \ldots, m. \tag{34}$$

Instantaneous Screw and Its Computations

Instantaneous motion description by screw directly reflect the Mozzi's axis together with the amplitude of velocity. The instantaneous motion of rigid body could be parameterized as instantaneous screw in 6-D vector form as

$$\boldsymbol{S}_t = \omega \begin{pmatrix} \boldsymbol{s}_t \\ \boldsymbol{r}_t \times \boldsymbol{s}_t + p\boldsymbol{s}_t \end{pmatrix}, \tag{35}$$

where $\boldsymbol{s}_t$, $\boldsymbol{r}_t$, ω and p have the same meanings as given in Eq. (11).

For robotic mechanism, the velocity of moving platform relative to the fixed platform forms a screw system, which is composed by a set of 1-DoF screws. In the process of performance modeling and design of robots, screw system plays an important role in mobility analysis and Jacobian formulation. For serial mechanisms, screw system could be measured as the combination of the instantaneous screws producing by each kinematic joint. When mechanisms with parallel structures, intersection operation of the screw systems generated by a series of connected chains is carried out. Due to the work of Rico and Duffy [137,138,139], screw systems were classified and proved to be subspaces, sometimes even sub-algebras of the Lie algebra se(3) of the Euclidean group SE (3). Therefore, the combination and intersection operation could be written as the form in Eq. (12) and Eq. (13), respectively.

Mappings between Finite and Instantaneous Screws

As far as we know, the exponential map does not exist between instantaneous screw and finite screw. That is because finite screw describes the displacement in a Gibson form, which break the linear transformation format of finite motion description of matrix Li Group and dual quaternion.

In spite of the lack of exponential map, differential map between displacement and velocity can be performed directly by taking differentiations of finite screw S_f with respect to time. For 1-DoF or multi-DoF finite screw S_f, the corresponding instantaneous screw system would be formulated as

$$\dot{\boldsymbol{S}}_f\Big|_{\substack{\theta=0\\t=0}} = \dot{\theta}\begin{pmatrix} \boldsymbol{s}_f \\ \boldsymbol{r}_f \times \boldsymbol{s}_f \end{pmatrix} + \dot{t}\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{s}_f \end{pmatrix} = \boldsymbol{S}_t, \tag{36}$$

$$\begin{aligned} \dot{\boldsymbol{S}}_{f,12\cdots n}\big|_{\substack{\theta_k=0\\ t_k=0,k=1,2,\cdots,n}} &= \dot{\boldsymbol{S}}_{f,1}\big|_{\substack{\theta_1=0\\ t_1=0}} + \dot{\boldsymbol{S}}_{f,2}\big|_{\substack{\theta_2=0\\ t_2=0}} + \cdots + \dot{\boldsymbol{S}}_{f,1}\big|_{\substack{\theta_n=0\\ t_n=0}} \\ &= \boldsymbol{S}_{t,1} + \boldsymbol{S}_{t,2} + \cdots + \boldsymbol{S}_{t,n}. \end{aligned} \tag{37}$$

DISCUSSIONS

After respectively reviewing the three mathematical tools applied in topology and performance modeling and analysis of robotic mechanisms, further discussions on comparisons among them and their applications will be given in this section.

Comparisons among the Three Methods

Based upon Sections 3–5, it can be seen that the instantaneous screws, matrix Lie algebra, and pure dual quaternions for instantaneous motion description are all linear vector spaces, and their algebraic structures are isomorphic to each other. Thus, only the mathematical tools for finite motion description will be compared here. The differences among matrix Lie group, dual quaternions, and finite screws rise from their different algebraic structures. In order to discuss the differences of these three mathematical tools in describing rigid body finite motion, we firstly look into their algebraic structures and the relationships among them and SE(3).

Any transformation matrix in the matrix Lie group can be represented by a 4×4 real matrix, a 6×6 real matrix, or a 3×3 dual matrix etc. Because these three representations are isomorphic with each other, we take 4×4 real matrix representation as an example in Section 3. The entire set of each kind of these matrices has the same inner closure and associative properties with SE(3). Hence, the matrix Lie group forms a homomorphism of SE(3). Furthermore, it is an isomorphism of SE(3), since there exists a bijective mapping between them. The matrix Lie group is also a representation of SE(3). This is because the matrix operations play as linear transformations acting on the 6-dimensional vector space. Dual quaternions have similar features. Half part of the entire set of dual quaternions with positive rotational angles is also an isomorphism and a representation of SE(3). Thus, the entire set of dual quaternion is a double cover of SE(3). The transformation matrices in matrix Lie group can be composited by multiplication with

linear transformation formats. The same operation can be performed by dual quaternions.

Different from transformation matrix and dual quaternion, finite screw is invented to break the linear transformation format of finite motion description, which can be regarded as a general form of Gibbs vector. Finite screw does not act on any vector space, and cannot transform any coordinate of geometric point or line. It is a mathematical tool purely for finite motion description, and it can express the basic elements of Chasles' motion in a straightforward manner. The composition algorithm of finite screws, i.e., screw triangle product, maintains the screw format, which directly leads to the expressions of basic elements of the resultant Chasles' motion. Although the entire set of finite screws under screw triangle product has the same inner closure and associative properties with SE(3), it is not a representation of SE(3). In other words, it only forms a isomorphism of SE(3).

Any element of SE(3) is a combination of rotation matrix and translation vector. It is a homogeneous transformation of the coordinates of points. In this way, all representations of SE(3) cannot break the inherent linear transformation formats. Hence, only finite screw with screw triangle product can express and composite finite motions in a non-redundant and direct manner.

All the three methods reviewed in Sections 3–5 could be used to describe and compute all situations of finite motions. To further investigate the relationships among them, we rewrite the element in dual quaternion in the following way,

$$\boldsymbol{D} = \left(\cos\frac{\theta}{2} - \varepsilon\frac{t}{2}\sin\frac{\theta}{2}\right) + \left(\sin\frac{\theta}{2}\boldsymbol{s}_f + \varepsilon\left(\sin\frac{\theta}{2}(\boldsymbol{r}_f \times \boldsymbol{s}_f) + \frac{t}{2}\cos\frac{\theta}{2}\boldsymbol{s}_f\right)\right). \tag{38}$$

Compare the above equation and Eq. (1)–(3) with finite screw in Eq. (32). It is noted that the information of a finite motion, i.e., the Chasles' axis and the corresponding rotational angle and translational distance, is involved in the 3×3 rotation matrix and 3×1 translation vector in, and is not easy to be extracted. Hence, for elements in matrix Lie group, at least 12 items are needed to describe the 6-dimensional finite motion. For dual quaternion, 8 items are needed, and the dual vector in D covers all the finite motion characteristics while the dual scalar is redundant. Finite screw contains the whole finite motion characteristics in the 6-dimensional quasi-vector form. Thus, it is non-redundant. Motion descriptions by finite screw are

more concise. On this basis, the composition of two finite motions could be obtained by three cross product computations and their linear combination. The redundancy of matrix Lie group and dual quaternions results in more operations in the process of computing the composition and intersection of finite motions. In the whole, finite and instantaneous screw based method has the most concise formats to describe mechanism motions, and provides the most explicit algorithms for the computation.

Future Works on Applications of the Three Methods

From the discussion in previous sections, three unified mathematical tools are proved to have the abilities of description, computation and mapping of finite and instantaneous motions. With the aid of the unified mathematical frameworks, the integrated topology and performance modeling and design can be studied, which is meaningful but still rarely investigated in the current researches. Therefore, the next problem is how to apply the above mathematical tools to the integrated modeling and design. Since both the topology and performances are considered, the integrated modeling and design process can be interpreted as (1) finding out all possible topologies having the same desired mobility, (2) formulating the performances of every topological structure, and (3) searching for the optimal topology and performances. Having the above unified mathematical tools, type synthesis and performance modeling can be carried out in the same mathematical framework, as shown in Figure 2. For instance, type synthesis can be implemented by the finite motion based methods. By the mapping between finite and instantaneous motions, the performance model would be done by instantaneous motion based methods. Hence, both the topological and performance parameters can be defined in the optimal design.

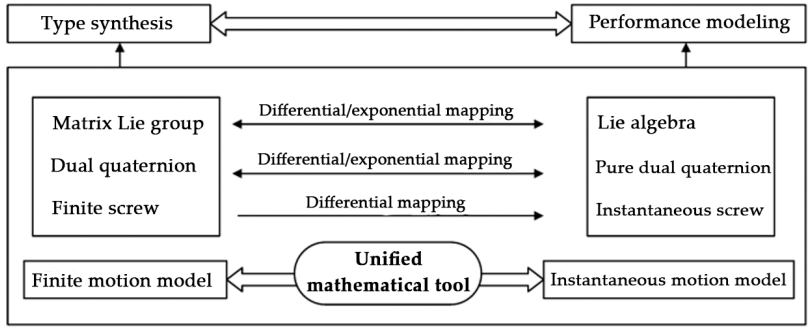


Figure 2: Application of unified mathematic tools in integrated analysis and design.

The detail integrated modeling and design process might be conducted as follows. First of all, the expected motion is described in a finite motion format based on matrix Lie group, dual quaternion or finite screw. By taking the advantages of intersection and composition operations, the available limbs and mechanisms would be generated. More details are referred to [35, 51, 86]. Because the type synthesis is implemented in an algebraic manner, the parameterized topology models are obtained. Then, the finite motion based topology model is directly applied as the displacement model relative to the initial pose. In order to construct the performance models with topology parameters, the differential mapping between matrix Lie group and Lie algebra, dual quaternion and pure dual quaternion, finite screw and instantaneous screw are utilized. In this way, the velocity model of 1-DoF kinematic joint, multi-DoF limbs and end-effector could be obtained in the forms of Lie algebra, pure dual quaternion and instantaneous screw, respectively. With the velocity model available at hand, the velocity/force features, stiffness performance can be further analyzed. By the first-order derivation of velocity, accelerations would be further formulated, with which the dynamic model is obtained. Up to this point, the integrated modeling for topology, kinematic, stiffness and dynamic is captured. Finally, both topological and dimensional parameters can be taken as design variables in optimal design, resulting in optimized topological structure with its dimensions.

Besides the methodology of integrated topology and performance modeling and design, another possible application of the reviewed mathematical tools is the automatic software development. It could be seen that every step of the integrated modeling and design is performed by algebraic expressions and computations, which facilitates this procedure to be realized in automatic manner using computer programming languages. By applying computation software like Matlab and Maple, composition, intersection and mapping algorithms of finite and instantaneous motions based on the three unified mathematic tools could be compiled as modularized programs. In this way, for given motion pattern, type synthesis can be automatically implemented to obtain all the feasible robotic mechanisms. The topology models are regarded as the displacement models. Then performance models in terms of velocity and acceleration can be directly constructed and analyzed by taking the first- and second-order derivation of its displacement model. The automatic software in the future work will improve the efficiency of integrated robot design and make the methods to be easily applied by the mechanical engineers without studying the mathematical knowledge.

CONCLUSIONS

Topology and performance of mechanism are the main focuses in the development of robotic mechanism. It has long been a desire to carry out the integrated analysis and design as topology and performance are mutually affected each other. A unified mathematical framework is the fundamental preparation. Three mathematical tools, i.e., Lie group and Lie algebra, dual quaternion and pure dual quaternion, finite screw and instantaneous screw, are comprehensively reviewed. The history, finite motion, instantaneous motion and the mapping relation of each mathematical tool are introduced, in which the description, computation and intersection of two types of motions are given. A discussion on the three mathematical tools is also presented. This paper aims at providing a reference on the mathematical tools in topology and performance integrated analysis and design, and helps reader select the appropriate method when implementing the analysis and design of robotic mechanisms.

REFERENCES

1. J Angeles. *Fundamentals of robotic mechanical systems: Theory, methods, and algorithms*. 4th ed. New York: Springer, 2014.
2. J P Merlet. *Parallel robots*. Netherlands: Springer, 2006.
3. X W Kong, C M Gosselin. *Type synthesis of parallel mechanisms*. Berlin: Springer, 2007.
4. T L Yang, A X Liu, Q Jin, et al. Position and orientation characteristic equation for topological design of robot mechanisms. *ASME Journal of Mechanical Design*, 2019, 131(2): 021001-1-021001-17.
5. M Sokolova, G Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 2009, 45(4): 427-437.
6. H N Huynh, A Hamed, R Edouard, et al. Modelling the dynamics of industrial robots for milling operations. *Robotics and Computer-Integrated Manufacturing*, 2020, 61, https://doi.org/10.1016/j.rcim.2019.101852.
7. V Muralidharan, A Bose, K Chatra, S Bandyopadhyay. Methods for dimensional design of parallel manipulators for optimal dynamic performance over a given safe working zone. *Mechanism and Machine Theory*, 2019, 147: 103721.
8. Z Gao, D Zhang Performance analysis, mapping and multiobjective optimization of a hybrid robotic machine tool. *IEEE Transact on Industrial Electronics*, 2015, 62(1): 423-433.
9. X J Liu, J S Wang. *Parallel mechanism: type, kinematics, and optimal design*. Berlin: Springer, 2014.
10. M Chasles. Note on the general properties of the system of two similar body between them and in any manner places in space; and on the finished moving or infinitely petis of free solid body. *Bull Math Ferussac*, 1830, 14: 321-326.
11. J Plücker. On a new geometry of space. *Philosophical Transactions*, 1865, 155: 725-791.
12. T L Yang, A X Liu, Q Jin, et al. Position and orientation characteristic equation for topological design of robot mechanisms. *ASME Journal of Mechanical Design*, 2009, 131(2): 021001-1-021001-17.
13. F Gao, J L Yang, Q J Ge. Type synthesis of parallel mechanisms having the second class GF sets and two dimensional rotations. *ASME Journal*

of Mechanism and Robotics, 2011, 3(1): 011003 (8 pages).

14. Z Fu, W Yang, Z Yang. Solution of inverse kinematics for 6R robot manipulators with offset wrist based on geometric algebra. *ASME Journal of Mechanism and Robotics*, 2015, 5(3): 310081-310087.
15. Q Jin, T L Yang. Theory for topology synthesis of parallel manipulators and its application to three-dimension-translation parallel manipulators. *ASME Journal of Mechanical Design*, 2004, 126(4): 625-639.
16. F A Klein. Comparative review of recent researches in geometry. *Mathematische Annalen*, 1893, 43: 63-100.
17. I M Yaglom. *Complex numbers in geometry*. New York: Academic, 1968.
18. T Sun, S F Yang, T Huang, et al. A way of relating instantaneous and finite screws based on the screw triangle product. *Mechanism and Machine Theory*, 2017, 108: 75-82.
19. J M Hervé. Analyse structurelle des mécanismes par groupe des déplacements. *Mechanism and Machine Theory*, 1978, 13: 437-450.
20. J M Hervé. Intrinsic formulation of problems of geometry and kinematics of mechanisms. *Mechanism and Machine Theory*, 1982, 17: 179-184.
21. Y J Wang, B Belzile, J Angeles, et al. Kinematic analysis and optimum design of a novel 2PUR-2RPU parallel robot. *Mechanism and Machine Theory*, 2019, 139: 407-423.
22. X D Jin, Y F Fang, D Zhang. Design of a class of generalized parallel mechanisms with large rotational angles and integrated end-effectors. *Mechanism and Machine Theory*, 2019, 134: 117-134.
23. L Q Li, Y F Fang, L Wang. Design of a family of multi-DOF drive systems for fewer limb parallel mechanisms. *Mechanism and Machine Theory*, 2020, 148: 103802.
24. Q C Li, Z Huang, J M Herve. Type synthesis of 3R2T 5-DOF parallel mechanisms using the Lie group of displacements. *IEEE Transactions on Robotics and Automation*, 2004, 20: 173-180.
25. Q C Li, J M Herve. Structural shakiness of nonoverconstrained translational parallel mechanisms with identical limbs. *IEEE Transactions on Robotics*, 2009, 25: 158-164.
26. Q C Li, J M Herve. Parallel mechanisms with bifurcation of Schoenflies motion. *IEEE Transactions on Robotics*, 2009, 25: 25-36.

27. Q C Li, J M Herve. 1T2R parallel mechanisms without parasitic motion. *IEEE Transactions on Robotics*, 2010, 26: 401-410.
28. R W Brockett. Robotic manipulators and the product of exponential formula. In: *Mathematical theory of networks and systems*. Berlin: Springer, 1984: 120-129.
29. A T Yang, F Freudenstein. Application of dual number quaternion algebra to the analysis of spatial mechanisms. *Journal of Applied Mechanics*, 1964, 86: 300-308.
30. J Angeles. The application on dual algebra to kinematic analysis. In: *Computational methods in mechanical systems, (NATO ASI Series)*, J Angeles, E Zakhariev, eds. Berlin: Springer, 1998.
31. J M Selig, E Bayro. Rigid body dynamics using Clifford algebra. *Advances in Applied Clifford Algebras*, 2010, 20: 141-154.
32. J M Selig. Exponential and Cayley maps for dual quaternions. *Advances in Applied Clifford Algebras*, 2010, 20: 923-936.
33. J S Dai, N Holland, D R Kerr. Finite twist mapping and its application to planar serial manipulators with revolute joints. *Proceedings of the IMechE Part C: Journal of Mechanical Engineering Science*, 1995, 209, 263-271.
34. J S Dai. Geometrical foundations and screw algebra for mechanisms and robotics. Beijing: Higher Education Press, 2014. (Translated from J S Dai. Screw Algebra and Kinematic Approaches for Mechanisms and Robotics. London: Springer, 2016.)
35. S F Yang, T Sun, T Huang, et al. A finite screw approach to type synthesis of three-DOF translational parallel mechanisms. *Mechanism and Machine Theory*, 2016, 104: 405-419.
36. S F Yang, T Sun, T Huang. Type synthesis of parallel mechanisms having 3T1R motion with variable rotational axis. *Mechanism and Machine Theory*, 2017, 109: 220-230.
37. L T Schreiber, C Gosselin. Schonflies motion PARAllel robot (SPARA), a Kinematically Redundant Parallel Robot with Unlimited Rotation Capabilities. *IEEE/ASME Transactions on Mechatronics*, 2019, 24(5): 2273-2281.
38. M Schappler, S Tappe, T Ortmaier. Modeling parallel robot kinematics for 3T2R and 3T3R tasks using reciprocal sets of Euler angles. *Robotics*, 2019, 8(3): 68. https://doi.org/10.3390/robotics8030068.

39. W A Cao, H Ding. A method for stiffness modeling of 3R2T overconstrained parallel robotic mechanisms based on screw theory and strain energy. *Precision Engineering*, 2018, 51: 10-29.

40. Y Shneor, V T Portman. Stiffness of 5-axis machines with serial, parallel, and hybrid kinematics: evaluation and comparison. *CIRP Annals Manufacturing Technology*, 2010, 59: 409-412.

41. J Wu, X L Chen, L P Wang. Design and dynamics of a novel solar tracker with parallel mechanism. *IEEE/ASME Transactions on Mechatronics*, 2016, 21(1): 88-97.

42. P J Shao, Z Wang, S F Yang. Dynamic modeling of a two-DoF rotational parallel robot with changeable rotational axes. *Mechanism and Machine Theory*, 2019, 131(1): 318-335.

43. J Gallardo, J M Rico, A Frisoli, et al. Dynamic of parallel manipulators by means of screw theory. *Mechanism and Machine Theory*, 2003, 38(11): 1113-1131.

44. C-H Kuo, J S Dai, H-S Yan. Reconfiguration principles and strategies for reconfigurable mechanisms. *ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*, 2009, 22-24, June London, United Kingdom.

45. Z Huang, Q C Li. Type synthesis of symmetrical lower-mobility parallel mechanisms using the constraint-synthesis method. *International Journal of Robotics Research*, 2003, 22: 59-79.

46. T L Yang, A X Liu, H P Shen, et al. Composition principle based on single-open-chain unit for general spatial mechanisms and its application-in conjunction with a review of development of mechanism composition principles. *ASME Journal of Mechanism and Robotics*, 2018, 10(5): 051005.

47. X D Meng, F Gao, S F Wu, et al. Type synthesis of parallel robotic mechanisms: Framework and brief review. *Mechanism and Machine Theory*, 2014, 78: 177-186.

48. H Ye, D Wang, J Wu, Y Yue, Y Zhou. Forward and inverse kinematics of a 5-DOF hybrid robot for composite material machining. *Robotics and Computer-Integrated Manufacturing*, 2020, 65: 101961.

49. J M Hervé. The mathematical group structure of the set of displacements. Mechanism and Machine Theory, 1994, 29: 73-81.

50. J M Hervé. The Lie group of rigid body displacements, a fundamental tool for mechanism design. *Mechanism and Machine Theory*, 1999, 34: 719-730.
51. J M Hervé, F Sparacino. Structural synthesis of parallel robots generating spatial translation. *Fifth International Conference on Advanced Robotics*, Pisa, Italy, 19-22 June 1991, 1991: 808-813.
52. Q C Li, J M Hervé, Type synthesis of 3-DOF RPR-equivalent parallel mechanisms. *IEEE Transaction on Robotics*, 2014, 30: 1333-1343.
53. C C Lee, J M Hervé. Type synthesis of primitive Schoenflies-motion generators. *Mechanism and Machine Theory*, 2009, 44: 1980-1997.
54. C C Lee, J M Hervé. Uncoupled actuation of overconstrained 3t-1r hybrid parallel manipulators. *Robotica*, 2009, 27: 103-117.
55. C C Lee, J M Hervé. Generators of the product of two Schoenflies motion groups. *European Journal of Mechanics A-Solid*, 2010, 29: 97-108.
56. C C Lee, J M Hervé. Isoconstrained parallel generators of Schoenflies motion. *ASME Journal of Mechanism and Robotics*, 2011, 3: 021006-1-021106-10.
57. P Fanghella, C Galletti. Mobility analysis of single-loop kinematic chains: an algorithmic approach based on displacement groups. *Mechanism and Machine Theory*, 1994, 29: 1187-1204.
58. P Fanghella, C Galletti. Metric relations and displacement groups in mechanism and robot kinematics. *ASME Journal of Mechanical Design*, 1995, 117: 470-478.
59. F Schur. Neue Begruendung der Theorie der endlichen Transformations gruppen. *Mathematische Annalen*, 1890, 35: 161-197.
60. J E Campbell. On a law of combination of operators. *Proceedings of the London Mathematical Society*, 1898, 29: 14-32.
61. J Meng, G F Liu, Z X Li. A geometric theory for analysis and synthesis of sub-6 dof parallel manipulators. *IEEE Transaction on Robotics*, 2007, 23: 625-649.
62. Y Q Wu, H Wang, Z X Li. Quotient kinematics machines: concept, analysis, and synthesis. *ASME Journal of Mechanism and Robotics*, 2011, 3: 041004-1-041004-11.
63. Y Q Wu, M Carricato. Symmetric subspace motion generators. *IEEE Transaction on Robotics*, 2018, 34: 716-735.

64. Y Q Wu, H Löwe, M Carricato, Z X Li. Inversion symmetry of the Euclidean group: theory and application to robot kinematics. *IEEE Transaction on Robotics*, 2016, 32: 312-326.

65. G F Liu, G Y Zhang, Y S Guan et al. Geometry of adjoint-invariant submanifolds of SE(3). *IEEE Transaction on Robotics*, 2019.

66. R W Brockett. Linear feedback systems and the groups of Galois and Lie. *Linear Algebra and its Applications*, 1983, 50: 45-60.

67. Z X Li, S S Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Transactions on Robotics and Automation*, 1988, 4: 32-44.

68. R Murray, Z X Li, S S Sastry. *A mathematical introduction to robotic manipulation*. Boca Raton: CRC Press, 1994.

69. Z X Li, J B Gou, Y X Chu. Geometric algorithms for workpiece localization. *IEEE Transactions on Robotics and Automation*, 1998, 14: 864-878.

70. F C Park. Computational aspects of the product-of-exponentials formula for robot kinematics. *IEEE Transactions on Robotics and Automation*, 1994, 39: 643-647.

71. K Okamura, F C Park. Kinematic calibration using the product of exponentials formula. *Robotics*, 1996, 14: 415-421.

72. C Han, J Kim, J Kim, F C Park. Kinematic sensitivity analysis of the 3-UPU parallel mechanism. *Mechanism and Machine Theory*, 2002, 37: 787-798.

73. G L Yang, I M Chen. Kinematic calibration of modular reconfigurable robots using product-of-exponentials formula. *Journal of Robotic System*, 1997, 14: 807-821.

74. G L Yang, I M Chen, W Chen, et al. Kinematic design of a six-DOF parallel-kinematics machine with decoupled-motion architecture. *IEEE Transaction on Robotics*, 2004, 20: 876-887.

75. Y Jin, I M Chen, G L Yang. Kinematic design of a 6-DOF parallel manipulator with decoupled translation and rotation. *IEEE Transaction on Robotics*, 2006, 22: 545-551.

76. G L Chen, H Wang, Z Q Lin. Determination of the identifiable parameters in robot calibration based on the POE formula. *IEEE Transaction on Robotics*, 2014, 30: 1066-1077.

77. G L Chen, H Wang, Z Q Lin, X M Lai. The principal axes decomposition of spatial stiffness matrices. *IEEE Transaction on Robotics*, 2015, 31:

191-207.

78. G L Chen, L Y Kong, Q C Li, et al. Complete, minimal and continuous error models for the kinematic calibration of parallel manipulators based on POE formula. *Mechanism and Machine Theory*, 2018, 121: 844-856.
79. J S Dai. An historical review of the theoretical development of rigid body displacements from Rodrigues parameters to the finite twist. *Mechanism and Machine Theory*, 2006, 41: 41-52.
80. W R Hamilton. *Elements of quaternions*. Cambridge: Cambridge University Press, 1899.
81. O Rodrigues, Des lois geometriques qui reagissent les deplacements d'un systeme solide dans l'espace. *Jde Mathematique Pures et Appliquees de Liouville*, 1840, 5: 380-440.
82. W K Clifford. Preliminary sketch of bi-quaternions. Proc. London Math Society, 1873, 4(64/65): 381-395.
83. J M R Martínez, J Duffy. The principle of transference: History, statement and proof. *Mechanism and Machine Theory*, 1993, 28: 165-177.
84. D P Chevallier. On the transference principle in kinematics: its various forms and limitations. *Mechanism and Machine Theory*, 1996, 31: 57-76.
85. O P Agrawal. Hamilton operators and dual-number-quaternions in spatial kinematics. *Mechanism and Machine Theory*, 1987, 22: 569-575.
86. Y Qi, T Sun, Y M Song. Type synthesis of parallel tracking mechanism with varied axes by modeling its finite motions algebraically. *ASME Journal of Mechanism and Robotics*, 2017, 9: 054504-1-054504-6.
87. A McAulay. Octonion: a development of Clifford's Bi-quaternions. Cambridge: Cambridge University Press,1898.
88. E Study. Von den bewegungen und umlegungen. *Mathematische Annalen*, 1891, 39: 441-565.
89. E Study. *Die geometrie der dynamin*. Leipzig, 1903: 437.
90. W Blaschke. *Kinematic and quaternionen*. Berlin: VEB Verlag, 1960.
91. X W Kong. Reconfiguration analysis of multimode single-loop spatial mechanisms using dual quaternions. *ASME Journal of Mechanism and Robotics*, 2017, 9(5): 051002.

92. K Liu, X W Kong, J J Yu. Operation mode analysis of lower-mobility parallel mechanisms based on dual quaternions. *Mechanism and Machine Theory*, 2019, 142: 103577.

93. A S Oliveira, E R Pieri, U F Moreno, et al. A new approach to singularity-free inverse kinematics using dual-quaternionic error chains in the Davies method. *Robotica*, 2016, 34(4): 942-956.

94. G Z Li, F H Zhang, Y L Fu, et al. Joint stiffness identification and deformation compensation of serial robots based on dual quaternion algebra. *Applied Sciences*, 2019, 9: 65.

95. A T Yang. Calculus of screws. In: W R Spiller ed. *Basic questions of design theory*. New York: American Elsevier Publishing Company, 1974: 265-281.

96. G R Veldkamp. On the use of dual numbers, vectors and matrices in instantaneous spatial kinematics. *Mechanism and Machine Theory*, 1976, 11: 141-156.

97. A P Kotelnikov. Screw calculus and some applications to geometry and mechanics. *Annals of the Imperial University of Kazan*, 1895.

98. A Perez-Gracia, J M McCarthy. Dual quaternion synthesis of constrained robotic systems. *ASME Journal of Mechanical Design*, 2004, 126: 425-435.

99. A Perez-Gracia, J M McCarthy. Kinematic synthesis of spatial serial chains using Clifford algebra exponentials. *Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science*, 2006, 220: 953-968.

100. J S Dai. Euler-Rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory*, 2015, 92: 144-152.

101. G Li, F Zhang, Y Fu, et al. Kinematic calibration of serial robot using dual quaternions. *Industrial Robot*, 2019, 46(2): 247-258,

102. K Daniilidis. Hand-eye calibration using dual quaternions. *International Journal of Robotics Research*, 1990, 18: 286-298.

103. C E Cea-Montufar, E A Merchán-Cruz, J Ramírez-Gordillo, et al. Multi-objective GA for collision avoidance on robot manipulators based on artificial potential field. In: Martínez-Villaseñor L, et al, Eds. *Lecture Notes in Computer Science*. Springer Nature, 2019: 687-700.

104. X K Wang, D P Han, C B Yu, et al. The geometric structure of unit dual quaternion with application in kinematic control. *Journal of*

Mathematical Analysis and Applications, 2012, 389(215): 1352-1364.

105. A Cohen, M Shoham. Hyper Dual Quaternions representation of rigid bodies kinematics. *Mechanism and Machine Theory*, 2020, 150: 103861.
106. R S Ball. The theory of screws: a geometrical study of the kinematics, equilibrium, and small oscillations of a rigid body. *The Transactions of the Royal Irish Academy*, 1871, 25: 137-217.
107. F Klein. The general linear transformation of linear coordinates. *Mathematische Annalen*, 1869, 2: 366-371.
108. F Klein. On the theory of linear complex of first and second degree. *Mathematische Annalen*, 1869, 2: 198-226.
109. R S Ball. *A treatise on the theory of screws*. Cambridge: Cambridge University Press, 1900.
110. K H Hunt. *Kinematic geometry of mechanisms*. Oxford: Oxford University Press, 1978.
111. J J Yu, S Z Li, H J Su. Screw theory based methodology for the deterministic type synthesis of flexure mechanisms. *Journal of Mechanism and Robotics*, 2011, 3(3): 031008.
112. T Sun, B B Lian, Y M Song. Stiffness analysis of a 2-DoF over-constrained RPM with an articulated traveling platform. *Mechanism and Machine Theory*, 2016, 96: 165-178.
113. H T Liu, T Huang, D G Chetwynd. A method to formulate a dimensionally homogeneous Jacobian of parallel manipulators. *IEEE Transaction on Robotics*, 2011, 27(1): 150-156.
114. X J Liu, X Chen, M Nahon. Motion/force constrainability analysis of lower-mobility parallel manipulators. *ASME Journal of Mechanism and Robotics*, 2014, 6: 031006-1-031006-9.
115. Y Z Zhao, J L Wang, Y C Cao, et al. Constant motion/force transmission analysis and synthesis of a class of translational parallel mechanisms. *Mechanism and Machine Theory*, 2017, 108: 57-74.
116. F M Dimentberg. *The screw calculus and its applications in mechanics*. Moskau: Nauka, 1965.
117. I A Parkin. Co-ordinate transformations of screws with applications to screw systems and finite twists. *Mechanism and Machine Theory*, 1990, 25: 689-699.
118. I A Parkin. A third conformation with the screw systems: finite twist

displacements of a directed line and point. *Mechanism and Machine Theory*, 1992, 27: 177-188.

119. K H Hunt, I A Parkin. Finite displacements of points, planes, and lines via screw theory. *Mechanism and Machine Theory*, 1995, 30: 177-192.
120. C T Huang. The finite screw systems associated with a prismatic-revolute dyad and the screw displacement of a point. *Mechanism and Machine Theory*, 1994, 29: 1131-1142.
121. C T Huang, B Roth. Analytic expressions for the finite screw systems. *Mechanism and Machine Theory*, 1994, 29: 207-222.
122. C T Huang. Notes on screw product operations in the formulations of successive finite displacements. *ASME Journal of Mechanical Design*, 1997, 119: 434-439.
123. B Roth. On the screw axes and other special lines associated with spatial displacements of a rigid body. *Journal of Engineering for Industry*, 1967, 89: 102-110.
124. J M McCarthy. *Introduction to theoretical kinematics*. Cambridge: MIT Press, 1990.
125. A T Yang. *Application of quaternion algebra and dual numbers to the analysis of spatial mechanisms*. New York, USA: Dept. of Mechanical Engineering, Columbia University, 1963.
126. L W Tsai, B Roth. Design of dyads with helical, cylindrical, spherical, revolute and prismatic joints. *Mechanism and Machine Theory*, 1972, 7: 85-102.
127. J Angeles. *Spatial kinematic chains: Analysis, Synthesis, optimization*. New York: Springer-Verlag, 1982.
128. I A Parkin. Unifying the geometry of finite displacement screws and orthogonal matrix transformations. *Mechanism and Machine Theory*, 1997, 3: 975-991.
129. I A Parkin. Dual systems of finite displacement screws in the screw triangle. *Mechanism and Machine Theory*, 1997, 32: 993-1003.
130. C T Huang, C M Chen. The linear representation of the screw triangle-a unification of finite and infinitesimal kinematics. *ASME Journal of Mechanical Design*, 1995, 117: 554-560.
131. C T Huang. The cylindroid associated with finite motions of the Bennett mechanism. *ASME Journal of Mechanical Design*, 1997, 119: 521-524.

132. J S Dai. Historical relation between mechanisms and screw theory and the development of finite displacement screws. *Journal of Mechanical Engineering*, 2015, 51: 13-26. (in Chinese)

133. T Sun, S F Yang, T Huang, J S Dai. A finite and instantaneous screw based approach for topology design and kinematic analysis of 5-axis parallel kinematic machines. *Chinese Journal of Mechanical Engineering*, 2018, 31(2): 66-75.

134. T Sun, C Y Liu, B B Lian, et al. Calibration for precision kinematic control of an articulated serial robot. *IEEE Transactions on Industrial Electronics*, 2020, https://doi.org/10.1109/tie.2020.2994890.

135. T Sun, B B Lian, S F Yang, et al. Kinematic calibration of serial and parallel robots based on finite and instantaneous screw theory. *IEEE Transactions on Robotics*, 2020, 36(3): 816-834.

136. T Sun, S F Yang, B B Lian, *Finite and instantaneous screw theory in robotic mechanism*. Springer: Singapore, 2020.

137. J M Rico, J Duffy. Classification of screw systems-II. Three-systems. *Mechanism and Machine Theory*, 1992, 27(4): 471-490.

138. J M Rico, J Duffy. Classification of screw systems-I. One- and two-systems. *Mechanism and Machine Theory*, 1992, 27(4): 459-470.

139. J M Rico, J Duffy. Orthogonal spaces and screw systems. *Mechanism and Machine Theory*, 1992, 27(4): 451-458.

Chapter

2

A HIGH ACCURACY MODELING SCHEME FOR DYNAMIC SYSTEMS: SPACECRAFT REACTION WHEEL MODEL

Abd-Elsalam R. Abd-Elhay[1], Wael A. Murtada[1] and Mohamed I. Yosof[2]

[1] National Authority for Remote Sensing and Space Sciences (NARSS), 23 Jozeph Tito St., Cairo, Egypt
[2] Department of Electrical Engineering, Faculty of Engineering, Al-Azher University, Cairo, Egypt

ABSTRACT

Reaction wheels are crucial actuators in spacecraft attitude control subsystem (ACS). The precise modeling of reaction wheels is of fundamental need in spacecraft ACS for design, analysis, simulation, and fault diagnosis applications. The complex nature of the reaction wheel leads to modeling difficulties utilizing the conventional modeling schemes. Additionally, the

Citation: (APA): Abd-Elhay, A. E. R., Murtada, W. A., & Yosof, M. I. (2022). A high accuracy modeling scheme for dynamic systems: spacecraft reaction wheel model. Journal of Engineering and Applied Science, 69(1), 1-22. (22 pages).

absence of reaction wheel providers' parameters is crucial for triggering a new modeling scheme. The Radial Basis Function Neural Network (RBFNN) has an efficient architecture, alluring generalization properties, invulnerability against noise, and amazing training capabilities. This research proposes a promising modeling scheme for the spacecraft reaction wheel utilizing RBFNN and an improved variant of the Quantum Behaved Particle Swarm Optimization (QPSO). The problem of enhancing the network parameters of the RBFNN at the training phase is formed as a nonlinear constrained optimization problem. Thus, it is proposed to efficiently resolve utilizing an enhanced version of QPSO with mutation strategy (EQPSO-2M). The proposed technique is compared with the conventional QPSO algorithm and different variants of PSO algorithms. Evaluation criteria rely upon convergence speed, mean best fitness value, stability, and the number of successful runs that has been utilized to assess the proposed approach. A non-parametric test is utilized to decide the critical contrast between the results of the proposed algorithm compared with different algorithms. The simulation results demonstrated that the training of the proposed RBFNN-based reaction wheel model with enhanced parameters by EQPSO-2M algorithm furnishes a superior prediction accuracy went with effective network architecture.

INTRODUCTION

In spacecraft missions that need a high pointing accuracy, Attitude Control Subsystem (ACS) with specific actuators shall be used. The reaction wheel (RW) is a vital actuator for the spacecraft ACS [1]. The accurate modeling of the spacecraft reaction wheel is recommended for the design, simulation, analysis, and fault identification applications. Meanwhile, increasing the accuracy of the reaction wheel modeling will improve the overall accuracy of the ACS modeling process. There are three common approaches for modeling dynamic systems. The noteworthy models are white box, black box, and gray box models. The white-box modeling is characterized by a good understanding of model parameters compared to black-box modeling that needs some measurements for the model inputs and outputs. Furthermore, the high accuracy modeling process can be achieved by black-box rather than white-box. Despite the black-box modeling accuracy, the generalization characteristics are proven to be superior in the case of white-box modeling rather than black-box modeling. Due to the complexity of the reaction wheel modeling, it is recommended to be in a white-box modeling manner. This is to satisfy the appropriate accuracy and generalization characteristics [2].

Unfortunately, many manufacturers provide insufficient information in their datasheets, which is needed to accurately model the dynamics of the reaction wheel. Therefore, building the white-box mathematical model is very difficult. Thus, the researchers have proposed many artificial intelligence (AI) schemes for modeling the reaction wheels [3,4,5,6,7,8]. For instance, Al-Zyoud and Khorasani [3] proposed a dynamic multilayer perceptron scheme for modeling the spacecraft reaction wheel. The dynamic properties were introduced into the multilayer perceptron network by adding delays between layers. Furthermore, the optimal results were obtained using six neurons at the hidden layers. Thus, a tiny training error was noticed in order of 0.04. However, the simulation results have shown that the dynamic multilayer perceptron had an improved performance compared to the linear reaction wheel model. There are some limitations for dynamic multilayer perceptron like the model complexity and noticeably low modeling accuracy. In [4], the three-layer Elman neural network is introduced to model the dynamics of the spacecraft reaction wheel. Therefore, the proposed Elman neural network had two inputs, 25 hidden neurons, and 1 output. Moreover, the network was trained through 5000 epochs to get a small mean square error of about 10^{-3}. Furthermore, simulation results have demonstrated the superiority of the Elman neural network-based observer compared to the linear observer for fault detection and identification. It was noticed that the former model has a computational complexity due to a large number of hidden neurons. Thus, this imposes a long computation time. Later on, Mousavi and Khorasani [5] proposed a reaction wheel model that represents four spacecraft formation flight missions. Thus, reaction wheel dynamics have been introduced by using an infinite impulse response filter with dynamic hidden layer neurons. Therefore, hopeful results were achieved from the four spacecraft constellations. The first one has a training error of 0.05 using a neural network architecture with ten hidden neurons. Furthermore, the second, third, and fourth spacecraft have a training error near of 0.018, 0.015, and 0.03 with eight, eight, and six neurons at their hidden layers, respectively. The drawback of the aforementioned proposed model is the use of the infinite impulse response filter that consumes tremendous computational resources.

Radial Basis Function Neural Networks (RBFNNs) are considered to be promising for modeling nonlinear dynamic systems like spacecraft reaction wheels. Moreover, RBFNN facilitates the modeling process due to its simple architecture, good generalization performance, low sensitivity against noise, and training capability [9]. Therefore, to address the drawbacks in related researches, this research proposes an efficient high accuracy modeling

scheme for spacecraft reaction wheel using RBFNN. Many researchers have proposed RBFNN as a modeling paradigm in different research areas [9,10,11,12,13,14,15]. For instance, in [9], RBFNN had been used for online modeling and adaptive control of nonlinear systems. Furthermore, it is proved that RBFNN has a noticeable performance with the effect of noise and parameters' variations. Nevertheless, the results also proved that RBFNN has a better performance than the feedforward neural network. Ali N. et al. [11] investigated the superiority of RBFNN over multilayer perceptron for predicting the welding features. The results proved the effectiveness of the high accuracy modeling capability for RBFNN over multilayer perceptron in modeling dynamic systems. Recently, Yunguang et al. [13,14,15] suggested an optimization module based on radial basis function and particle swarm optimization to develop a wheel profile fine-tuning system. Simulation results have proven that the proposed optimization algorithm can recommend an optimal wheel profile according to train operators' needs.

Training the RBFNN includes calculating the number of hidden neurons, centers of the Radial Basis Function (RBF), widths of the hidden layers, and the connection weights. Therefore, determining the optimal values for these parameters is a crucial factor for the RBFNN network performance. To address this concern, an optimization algorithm shall be used to enhance the training performance and then the modeling accuracy. Recently, different optimization algorithms have revealed promising performance. When compared to other optimization approaches, Particle Swarm Optimization (PSO) has a robust search ability, fast computation, and is inexpensive in terms of speed and memory [16]. However, it was proven that PSO is certifiably not a global optimization algorithm [17]. Therefore, numerous variants of PSO have been proposed to work on the performance of PSO [18,19,20,21,22,23]. Quantum Behaved Particle Swarm Optimization (QPSO) algorithm is another adaptation of the conventional PSO that was presented by Sun [24]. It had been started by quantum mechanics and the analysis of PSO dynamic behavior. Besides, QPSO is a sort of stochastic algorithm that has iterative equations, which differ from that of the conventional PSO. Moreover, there are limited QPSO parameters that should be adapted compared with conventional PSO. Hence, experimental results showed that QPSO has a superior performance compared with the standard PSO on various benchmark functions [25]. In any case, QPSO is a proper algorithm for global optimization issues, yet it suffers from premature convergence. Consequently, this premature convergence

enables performance degradation and inefficiency for solving optimization problems. This convergence is caused because of catching in local optimal. Nevertheless, premature convergence happens because of the consistent declination of particles' diversity [26].

This research proposes a high accuracy modeling scheme for spacecraft reaction wheel utilizing RBFNN and a further enhanced version of the QPSO algorithm. As an improvement, firstly, two progressive mutations were applied to further improve the exploitation process. Besides, a diversity control strategy is applied to enhance the particles' diversity and overcome the premature convergence. Subsequently, expanding the chance of the swarm to leap out the local minima and discovering new encouraging solutions further improve the algorithm performance. Accordingly, an improved QPSO algorithm, signified by Enhanced Quantum Particle Swarm Optimization – 2 Mutation (EQPSO-2M), is proposed for the training of the RBFNNbased reaction wheel model. The enhancement aims to improve the search abilities of QPSO and trying not to stick at local optimal. Moreover, the proposed reaction wheel mathematical model that was proposed in [27], has been implemented to create the dataset that is needed for the testing of the RBFNN-based reaction wheel model. The effectiveness of the proposed EQPSO-2M algorithm is investigated using convergence speed, mean best fitness value, stability, and the number of successful runs. The obtained results indicate the superior performance of the EQPSO-2M method. Once the optimal parameters of RBFNN are obtained, the performance of the proposed reaction wheel model has been tested using the simulation results.

METHODS

Spacecraft Dynamic Model

Attitude Control Subsystem (ACS) is one of the vital systems in the spacecraft that provides the in-orbit attitude control and determination functions. ACS is conceptually composed of three main parts: attitude sensors, feedback control system, and actuators [28]. Figure 1 illustrates the simplified block diagram of the ACS subsystem. Spacecraft can be represented as a rigid body where the dynamics can be obtained using Euler's dynamical formulas.

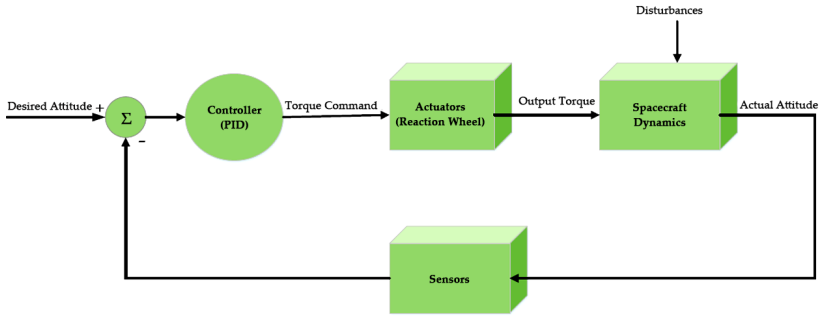


Figure 1: Attitude control subsystem block diagram.

Euler's equation is equivalent to Newton's second law for rotation about the center of mass. Thus, the body motion equations about its center of mass using reaction wheels as actuators are described by Euler equations as in [28] as follows:

$$\dot{\omega}_x = \frac{1}{I_{xx}} \left(\tau_x + \tau_{dx} - \omega_z \omega_y \left(I_{zz} - I_{yy} \right) \right) \quad (1)$$

$$\dot{\omega}_y = \frac{1}{I_{yy}} \left(\tau_y + \tau_{dy} - \omega_x \omega_z \left(I_{xx} - I_{zz} \right) \right) \quad (2)$$

$$\dot{\omega}_z = \frac{1}{I_{zz}} \left(\tau_z + \tau_{dz} - \omega_y \omega_x \left(I_{yy} - I_{xx} \right) \right) \quad (3)$$

where I_{xx} , I_{yy} and I_{zz} represent the spacecraft moment of inertia. ω_x, ω_y, and ω_z are the spacecraft's angular velocities in body-fixed axes toward inertial coordinate system along x, y, and z axes, respectively. τ_{dx}, τ_{dy}, and τ_{dz} represent the disturbances torques, which act on the spacecraft about roll, pitch, and yaw axis respectively. τ_x, τ_y, and τ_z represent the torque due to the motion of the wheel on each axis. To get the spacecraft's actual attitude, which are the Euler angles roll, pitch, and yaw, the Eqs. 1, 2, and 3 shall be integrated twice.

Reaction Wheel Mathematical Model

Reaction wheels are the common actuators for three axes stabilized spacecraft ACS, specifically for unmanned spacecraft. They are simply flywheels mounted to an electric direct current (DC) motor that can rotate in the desired direction to establish one axis control for each RW [29]. Furthermore, the reaction wheel is a nonlinear ACS component, which consists of several internal loops. Thus, these loops should be considered to

ensure accurate mathematical modeling. Figure 2 illustrates the RW internal loops, which are described in [27]. The block diagram in Fig. 2 can be described mathematically as in [30] by Eqs. 4 and 5 as follows :

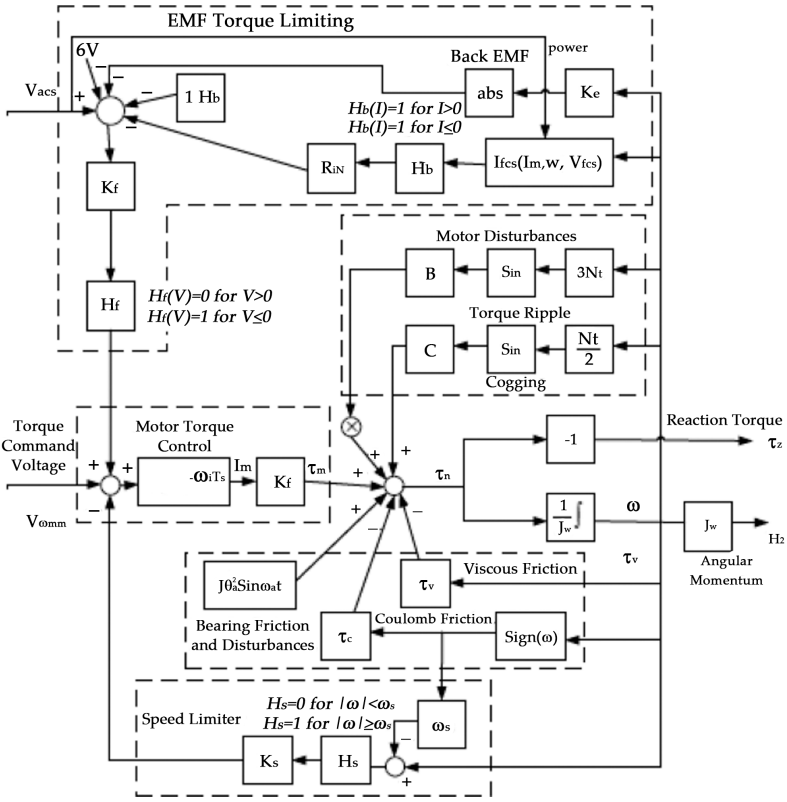


Figure 2: Reaction wheel mathematical model.

$$\begin{bmatrix} \dot{I}_m \\ \dot{w}_m \end{bmatrix} = \begin{bmatrix} G_d w_d [\Psi_1(I_m, w_m) - \Psi_3(w_m) - w_d I_m] \\ \frac{1}{J_\omega} [k_t I_{m-\tau_c} \Psi_2(w_m) - \tau_v w_m] \end{bmatrix} + \begin{bmatrix} G_d w_d \\ 0 \end{bmatrix} V_{com} \tag{4}$$

$$\tau = k_t I_m \tag{5}$$

In Eq. 4, I_m represents the motor current, k_t is the motor torque constant, w_m is the motor angular velocity, G_d is the driver gain, w_d is the driver bandwidth, and Ψ_1, Ψ_2, and Ψ_3 represent the nonlinearities for back-EMF limiting torque, Coulomb friction, and speed limiter circuit. This research proposes the use of the ITHACO type-A reaction wheel, which is produced by Goodrich Corporation. Table 1 shows the parameters of the ITHACO type-A reaction wheel.

Table 1: ITHACO type-A RW main parameters

Parameter	Description	Value
G_d	Driver gain	0.19 A/V
K_t	Motor torque constant	0.029 N.m/A
K_e	Motor back-EMF	0.029 V/rad/s
K_s	Over-speed circuit gain	95 V/ rad/s
w_s	Maximum wheel speed	690 rad/s
w_d	Driver bandwidth	2000 rad/s
R_{in}	Input resistance	2 Ω
K_f	Voltage feedback gain	0.5 V/V
N	Number of motor poles	36
T_c	Torque command range	− 5 to + 5 V
τ_c	Coulomb fiction	0.002 N.m
J	Flywheel inertia	0.0077 N.m.s^2

Radial Basis Function Neural Network Architecture

The RBFNNs were firstly proposed in 1988 [31] based on the principle that the biological neuron has a local response. Moreover, RBFNN has a simple architecture, fast training time, and efficient approximation capabilities rather than other neural networks [9]. A typical architecture of RBFNN includes three layers: input layer, hidden layer, and output layer as depicted in Fig. 3. The input layer consists of input nodes that are connecting the inputs to the neural network. The hidden neurons use radial basis function such as Gaussian function $\varphi_i(x)$ as the activation function as follows:

$$\phi_i(x) = \exp\left(-\frac{\|x-c_i\|^2}{{\sigma_i}^2}\right) \tag{6}$$

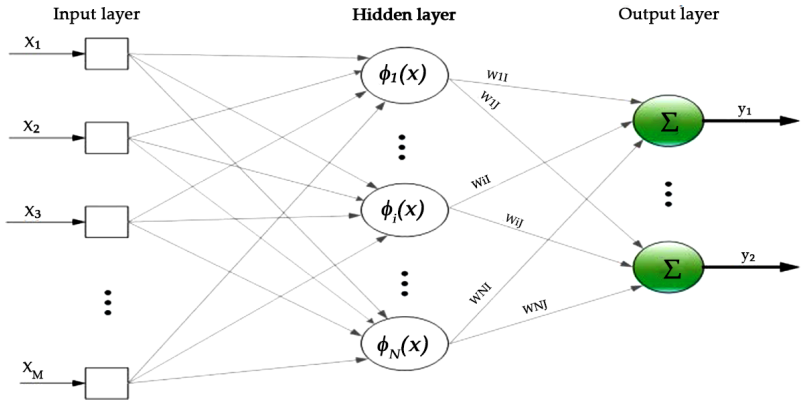


Figure 3: Typical structure of the RBFNN.

where x represents the network input, σ_i and c_i are width and center of the ith neuron, respectively, and ||•|| is the Euclidean distance between two different vectors. The output layer has a linear activation function that produces the network output corresponding to the network input [10]. Thus, the output of the network y_j can be addressed as follows:

$$y_j = \sum_{i=1}^{n} w_i \phi_i(x) + b_i \tag{7}$$

In Eq. 7, b_i and w_i , are the bias and the weight of the ith neuron respectively. Therefore, to define the proposed RBFNN-based reaction wheel model, it is mandatory to determine some critical parameters. These parameters include the number of input neurons, number of hidden neurons, output layer's neurons, and the weights of all neurons. In addition, other important parameters shall be tuned like the centers and the widths of the hidden neurons. Generally, the number of the problem inputs will determine the number of input layer neurons [32]. Thus, the input layer of the proposed RBFNN-based reaction wheel model comprises a single neuron that represents the torque command voltage. Furthermore, the number of the output layer neurons is determined corresponding to the number of model outputs. Because the reaction wheel has only one output, which is the generated torque, thus the output layer has a single neuron. The number of hidden layer neurons has a paramount impact on the RBFNN performance. Generally speaking, the more the neurons in the hidden layer, the better the network accuracy [33]. However, the addition of hidden neurons after the right number is reached will not improve the network accuracy, but increase the computational power and architectural complexity. Therefore, the optimal number of neurons in the hidden layer needs to be justified experimentally and it is based on the network designer experience [34] as will be introduced in the experimental results and discussions section. Furthermore, the centers, widths, and weights between the hidden neurons and the output layer shall be estimated. Thus, this research proposes an enhanced version of QPSO, which is EQPSO-2M to estimate the optimal values of the centers, widths, and weights.

Standard Particle Swarm Optimization

PSO was proposed by Eberhart and Kennedy [35]. In the PSO algorithm, each particle is assumed as a point in an N-dimensional Euclidian space. Moreover, at each iteration, there are three vectors, which are used to describe the behavior of the particle i that are: the current

position vector: $\boldsymbol{X_{i,n}} = (X^1_{i,n}, X^2_{i,n}, ..., X^j_{i,n}, ..., X^N_{i,n})$; the velocity vector: $\boldsymbol{V_{i,n}} = (V^1_{i,n}, V^2_{i,n}, ..., V^j_{i,n}, ..., V^N_{i,n})$, and the personal best position vector; $\boldsymbol{P_{i,n}} = (P^1_{i,n}, P^2_{i,n}, ..., P^j_{i,n}, ..., P^N_{i,n})$; where $(1 \leq j \leq N)$. Therefore, at the $(n + 1)$ iteration, the particles' velocity and position vectors are updated as the following [36]:

$$V^j_{i,n+1} = V^j_{i,n} + c_1 r^j_{i,n}\left(P^j_{i,n} - X^j_{i,n}\right) + c_2 R^j_{i,n}\left(G^j_{i,n} - X^j_{i,n}\right), \tag{8}$$

$$X^j_{i,n+1} = X^j_i + V^j_{i,n+1} \tag{9}$$

where c_1 and c_2 are the acceleration coefficients, $r^j_{i,n}$, $R^j_{i,n}$ are two different random numbers that are distributed uniformly over $(0, 1)$; therefore, $\{r^j_{i,n} : j = 1, 2, ..., N\} \in U(0, 1); \{ \quad R^j_{i,n} : j = 1, 2, ..., N\} \in U(0, 1).\boldsymbol{G^j_{i,n}}$, is the global best position vector. It is noticed that when c_1 is greater than c_2, the swarm has a higher local search ability. On the other hand, the swarm explores the search space more globally when c_2 is greater than c_1 [37]. To improve the performance of the standard PSO and minimize the probability of trapping in local optimal, many PSO variants have been proposed. For instance, Ziyu and Dingxue [21] introduced the Time-varying Adaptive PSO (TAPSO) version without using the velocity of the previous iteration. Thus, the particle's velocity update can be formulated as follows:

$$V^j_{i,n+1} = c_1 r^j_{i,n}\left(P^j_{i,n} - X^j_{i,n}\right) + c_2 R^j_{i,n}\left(G^j_{i,n} - X^j_{i,n}\right), \tag{10}$$

In TAPSO, the reinitialization criterion is based on assuming random velocity to avoid premature searching for the velocity of a particle at zero. Moreover, the authors introduced an exponential time-varying acceleration coefficient to enhance the exploration and exploitation capabilities. Therefore, the acceleration coefficients are updated according to the following equations:

$$c_1 = c_{\min} + (c_{\max} - c_{\min}).e^{-\left(\frac{4k}{G}\right)^2} \tag{11}$$

$$c_2 = c_{\min} - (c_{\max} - c_{\min}).e^{-\left(\frac{4k}{G}\right)^2} \tag{12}$$

where ***k*** represents the current iteration number, and ***G*** represents the maximum number of iterations. PSO is simple to implement, has a fast convergence, and its convergence can be controlled using a few coefficients. For this reason, it has been used for solving a wide range of optimization problems. However, standard PSO can't converge to the global optimal when it is used with complex optimization problems [38].

Quantum Behaved Particle Swarm Optimization (QPSO)

QPSO algorithm was introduced by Sun in 2004 based on quantum mechanics and computing [39]. In QPSO, the particle's state is represented by a wave function. Therefore, the probability of the particles that appear in position $\overrightarrow{\mathbf{X}}$ can be estimated from the probability density function of its position [40]. Regarding PSO convergence analysis, PSO converges when each particle converges to the local attractor $p_{i,n}^{j}$ that can be represented by:

$$p_{i,n}^{j} = \phi.P_{i,n}^{j} - (1-\phi).G_{n}^{j}, \phi \sim U(0,1) \tag{13}$$

where $p_{i,n}^{j}$ represents the *j*th dimension of the particle local attractor, $P_{i,n}^{j}$ is the particle best position, and G_{n}^{j} is the global best position. It is assumed that the particle *i* moves in *N*-Dimensional space with a δ potential well at $P_{i,n}^{j}$ to guarantee the algorithm convergence at *n* iterations. Using the Monte Carlo method, the position for the *j*th dimension of the *i*th particle at $n+1$ iteration is formulated according to Eq. 14 as follows [40]:

$$X_{i,n+1}^{j} = \begin{cases} p_{i,n}^{j} + \alpha \left| X_{i,n}^{j} - C_{n}^{j} \right| * \ln\left(1 / u_{i,n+1}^{j}\right), & \text{if } m \geq 0.5 \\ p_{i,n}^{j} - \alpha \left| X_{i,n}^{j} - C_{n}^{j} \right| * \ln\left(1 / u_{i,n+1}^{j}\right), & \text{if } m < 0.5 \end{cases} \tag{14}$$

where *u* and *m* are two random numbers that are uniformly distributed in [0,1] and $C_{n}^{j} = (C_{n}^{1}, C_{n}^{2}, ..., C_{n}^{N})$ is the average of the best positions for all particles. Thus, it can be calculated by:

$$C_{n+1}^{j} = \left(\frac{1}{M}\right) \sum_{i=1}^{M} P_{i,n}^{j} (1 \leq j \leq N) \tag{15}$$

In Eq. 14, α represents the contraction-expansion (CE) coefficient that enhances the performance of QPSO when it is properly selected [39]. Many proposed methods were introduced to control the contraction-expansion coefficient such as in [41, 42].

The proposed Enhanced Quantum Behaved Particle Swarm Optimization Algorithm

Although the QPSO algorithm has revealed a good performance to find the optimal solution for many optimization problems [40]. However, it still introduces a deteriorative performance in searching for the global optimal solution in complex optimization problems. This performance degradation in QPSO occurs due to the premature convergence. To resolve this problem for QPSO and other PSO variants, this research proposes an EQPSO-2M algorithm that has two significant improvements. First, the diversity of particles is enhanced to guarantee a healthy diversity of the particles during the search process. Therefore, to avoid the premature convergence of the algorithm. The particle diversity is calculated using the following formula:

$$D = \frac{1}{M.|A|} \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{N} \left(X_{i,n}^{j} - \overline{C}_{n}^{j} \right)^{2}} \tag{16}$$

where M represents the swarm size, N represents the dimensions of the problem, A denotes the length of the longest diagonal in the search space, $X_{i,n}^{j}$ is the jth component of the ith particle's position for the nth iteration, and $\overline{C}_{n}^{j}$ represents the particles' mean best position [42]. Meanwhile, the particles' diversity is monitored during the search process; when it is decreased below the threshold value d_{low}; the particles' mean best position will be reinitialized with values that maximize the diversity again as follows:

$$C_{n+1}^{j} = \begin{cases} X_{i,n+1}^{j} + X_{\max}, & if\ X_{i,n+1}^{j} > 0 \\ X_{i,n+1}^{j} + X_{\min}, & if\ X_{i,n+1}^{j} \leq 0 \end{cases} \tag{17}$$

where $X_{\max}$ and $X_{\min}$ represent the maximum and minimum boundaries of the search interval respectively. The main idea behind the reassignment of the mean best position vector using Eq 17 is to increase the distance between the particle's position and the mean best position as we can. Thus, the population diversity will increase monotonically and this would make the particle escapes the local optima. The other improvement of the EQPSO-2M is to overcome the premature convergence by adding two consecutive single dimension Gaussian mutations on the particle's personal best position as follows:

$$P_{i,n}^{j} = P_{i,n}^{j} r_1 + 0.01 P_{i,n}^{j} r_2, r_1, r_2 \sim U(0,1) \tag{18}$$

where r_1 and r_2 represent two different arrays of uniform distribution random numbers. The two consecutive mutations will help the particles to explore extensively different regions of the search space to find the best positions. Thus, this is to enhance the convergence speed of the QPSO and to avoid premature convergence. Applying the diversity control and the two successive single dimension Gaussian mutations will avoid the premature convergence that may occur in the conventional QPSO. Moreover, these two processes can enhance the convergence speed of QPSO and prevent the algorithm from trapping in local minima. The pseudocode for the proposed EQPSO-2M is shown in Algorithm 1 as below:

Algorithm 1: Pseudocode for the proposed EQPSO-2M algorithm

Begin
Initialize the population of size M and the dimensions of the problem N;
Set the maximum and the minimum limits of search interval $[X_{max}, X_{min}]$*;*
Set initial values for the particles' current position $X^j_{i,0}$ *and the best position* $P^j_{i,0}$*;*
Compute the fitness of the current iteration and the personal best position using equation 19. Then calculate the global best position G_0*;*
Set the value of CE coefficient α to 0.75;
Set the value of diversity threshold value d_{low}*;*
for (n = 1 to Maximum Iterations)
  for (i = 1 to population size M)
  for (j = 1 to Dimensions N)
    $\varphi^j_{i,n}$ = *rand (0,1);*
    $p^j_{i,n} = \varphi^j_{i,n} P^j_{i,n} + (1-\varphi^j_{i,n})\, G^j_n$;
    $u^j_{i,n}$ = *rand (0,1);*
    if (rand (0,1) < 0.5)

$$X^j_{i,n+1} = p^j_{i,n} + \alpha \left| X^j_{i,n} - C^j_n \right| ln\left(1/u^j_{i,n+1}\right);$$

    else

$$X^j_{i,n+1} = p^j_{i,n} - \alpha \left| X^j_{i,n} - C^j_n \right| ln\left(1/u^j_{i,n+1}\right);$$

    end if
  end for
  Compute the fitness of the particle at the position $X^j_{i,n+1}$
  if (f ($X^j_{i,n+1}$*) < f (*$X^j_{i,n}$*))*
   Update the personal best position for the current particle $P_{i,n}$
  end if
  Apply the first mutation on the personal best $P_{i,n}$ *using equation 18 and recalculate* $P_{i,n}$ *;*
  If the fitness of the new $P_{i,n}$ *is better, then update the value of* $P_{i,n}$ *and the global best Gn*
  End if
  Apply the second mutation on personal best $P_{i,n}$ *using equation 18 and recalculate* $P_{i,n}$ *;*
  If the fitness of the new $P_{i,n}$ *is better, then update the value of* $P_{i,n}$ *and the global best Gn.*
  end for
  Compute the mean best position using equation 15.
  Compute the diversity of nth iteration $D_n(s)$ *using equation 16 ;*
  if (the diversity of the nth iteration $D_n(s)$ *< Diversity threshold* d_{low}*)*
    for (j = 1 to Dimension N)
      if ($X^j_{i,n+1} < 0$*)*

$$C^j_{n+1} = X^j_{i,n+1} + X_{max}$$

      else

$$C^j_{n+1} = X^j_{i,n+1} + X_{min}$$

      end if
    end for
  end if
  Display and write the fitness of the global best position of n^{th} *iteration at this run.*
end for
end

In the above algorithm, QPSO is firstly initialized with the swarm size M and dimensions N. Therefore, the number of particles is set to 20. The swarm size selection will be discussed in the next section. The particles' positions and the personal best positions are randomly initialized. Furthermore, the initial global best and the mean best positions should be estimated. Thus, the value of α shall be set to 0.75 according to [39]. Moreover, the iterative process for updating the particle's current position $X^{j}_{i,n+1}$ should be started according to Eq. 14. Therefore, the fitness value is evaluated according to Eq. 19. When the fitness of the current particle's position is better than the previous one, the particle's best position $P_{i,n}$ should be updated. Hence, two consecutive mutations are applied on the particle's personal best position according to equation 18. After each mutation, the fitness of a new personal best position should be estimated to update the personal best position and the global best position of the particles $P_{i,n}$ and G_n, respectively. Further, the particle's mean best position C^{j}_{n+1} should be calculated using Eq. 15. Thus, the diversity of the particle should be evaluated using Eq. 16, and then compared with the threshold value d_{low}. Meanwhile, when the current diversity is below the threshold; the mean best position should be estimated according to Eq. 17. The searching process will be continued until the maximum iterations are met.

RESULTS AND DISCUSSION

In order to evaluate the performance of the proposed modeling scheme, simulation experiments should be done to benchmark the proposed RW model. Furthermore, a 3-axis ACS nonlinear model was implemented using MATLAB/SIMULINK. It includes the spacecraft dynamic model, the RW mathematical model, and the Proportional Integral Derivative (PID) controller. Therefore, the input to the ACS model is the desired attitude and the output is the actual attitude.

The RW input will be the torque command voltage, and the output is the generated torque a large number of experiments, the training dataset is suggested for the whole simulation to run with perspective angles within the range of [−5°, 5°]. Moreover, the simulation time in every iteration is three hundred seconds. Figures 4 and 5 show RW input torque command signal and output torque, respectively.

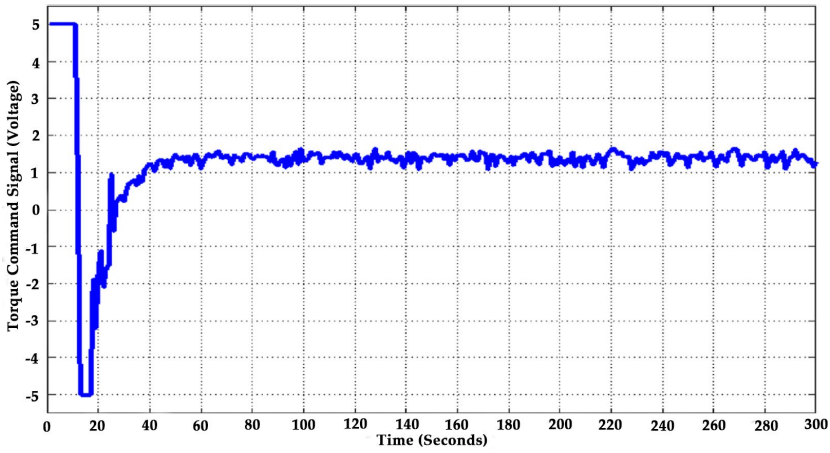


Figure 4: Reaction wheel torque command.

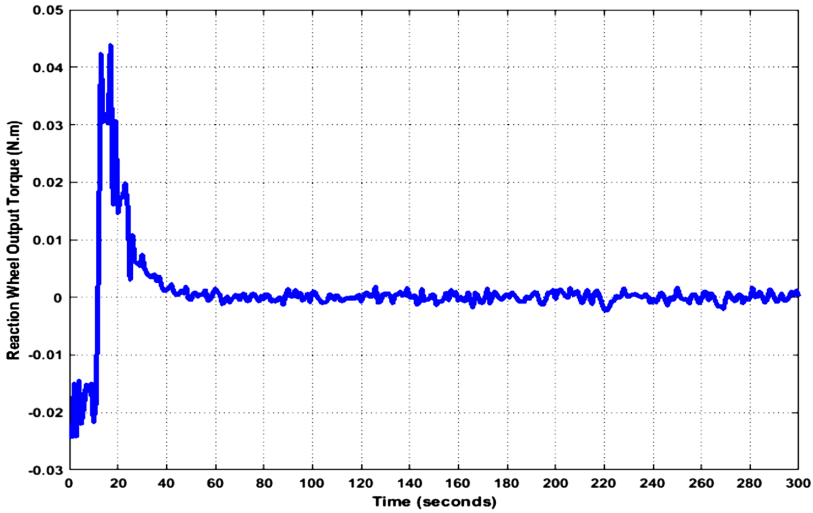


Figure 5: Reaction wheel output torque.

RBFNN Hidden Layer Size Analysis

The determination of the suitable number of hidden neurons significantly affects the RBFNN performance. In this research, the number of hidden neurons is chosen on the basis that to get the best performance from RBFNN and keep the design of RBFNN as simple as could be expected. To choose the number of the hidden neurons, we began according to [43] with one

hidden neuron and increment the number of neurons progressively by one neuron. Table 2 shows the results of this study.

Table 2: QPSO-trained RBFNN performance at different hidden layer neurons

Hidden layer neurons	Performance, mean square error (MSE)
1	1.87E−05
2	**6.54E−07**
3	6.44E−07
4	6.32E−07
5	6.44E−07

It can be observed from Table 2 that the performance of the RNFNN model is improved when the number of hidden layer neurons increased. The RBFNN with only one hidden layer neuron has MSE ≈ 1.87E-05. As seen from Table 2, we can notice that RBFNN with two hidden neurons decreases the mean square error (MSE) to be 6.54E−07. However, increasing the number of hidden neurons to more than two neurons has no significant improvement in the model performance. Therefore, it is recommended for the number of RBFNN hidden layer neurons for the spacecraft reaction wheel model to be two neurons.

Fitness Function

The problem of the RBFNN model training has been defined as a nonlinearly constrained optimization problem, which is settled utilizing the proposed EQPSO-2M. This optimization problem aims to find the optimal values of the RBFNN parameters that minimize the error between the RBFNN model output and the target output. Accordingly, to utilize the proposed EQPSO-2M algorithm for the training of the RBFNN-based reaction wheel model, a fitness function ought to be carried out. In this research, the well-known MSE has been chosen as the objective function. This function takes the difference between the RBFNN output and the actual reaction wheel output to compute the mean of the square errors as follows:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left\| t_i - \left(w_1 \exp\left(-\frac{\|x_i - c_1\|^2}{{\sigma_1}^2} \right) + w_2 \exp\left(-\frac{\|x_i - c_2\|^2}{{\sigma_2}^2} \right) + b \right) \right\|^2 \tag{19}$$

In Eq. 19, N is the number of the training patterns, t_i is the target reaction wheel output torque, w_1 is the weight among the first hidden neuron and the

output neuron, w_2 represents weight among the second hidden neuron and the output neuron c_1 and c_2 are the centers of the first and the second hidden neuron RBF respectively, σ_1 and σ_2 are the widths of first and second hidden neurons, respectively, and b is the bias. These parameters can be obtained when the fitness function in Eq. 19 is minimized.

Swarm Size Selection Assessment

Picking the fitting population size of the QPSO algorithm is a principal factor that influences its performance. As a rule, the optimal swarm size relies upon the complexity of the optimization problem to be addressed. However, increasing the population size might increase the algorithm's performance, but it will increase the computational time. Then again, decreasing the number of particles to a specific limit might cause the optimization process to fail. Danial J et al. [44] suggested that the population size is regularly changed from 20 to 50 particles. In addition, to choose the proper population size, five experiments have been done. Each experiment was executed 20 times with a most extreme number of cycles up to 3000. Further, the five experiments were analyzed in terms of standard deviation (STD) of the fitness values, mean of best fitness values, and success rate (SR)

The SR is computed as follows:

$$\mathrm{SR} = \frac{\mathrm{NSR}}{\mathrm{TNR}} * 100\% \tag{20}$$

where NSR addresses the number of successful runs and TNR is the total number of runs, which are 20 runs in runs in all the experiments. Besides, it is considered for the single run to be effective at the end of 3000 iterations in a manner that MSE ≈6.5E−7. The results acquired from these trials are given in Table 3 and Fig. 6. As per the results in Table 3, it can be observed that the QPSO algorithm with 20 particles has a 50% success rate. The STD of the best fitness value for the four cases is around something similar. It can be observed from Fig. 6 that the four cases have approximately the same average fitness values. Although, QPSO with 50 particles has a slightly fast convergence speed, but expanding the population size will increment the computational time and the calculation intricacy. Consequently, we chose the population size to be 20 particles.

Table 3: Effect of swarm size change on QPSO-RBFNN performance

Swarm size	Iterations	Mean best fitness value	STD	Success rate, %
20	**3000**	**4.1130E−06**	**4.8996E−06**	**50%**
30	3000	3.6148E−06	4.1460E−06	30%
40	3000	3.4110E−06	4.5874E−06	25 %
50	3000	3.0094E−06	4.6556E−06	30%

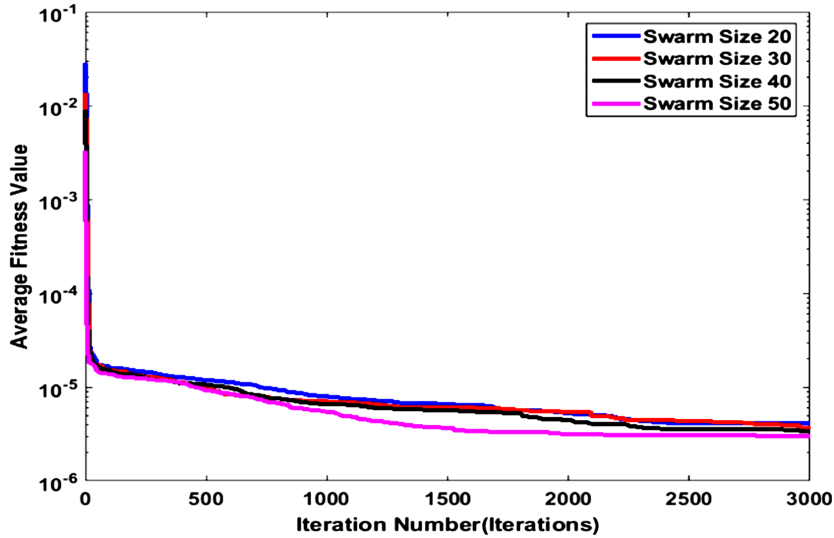


Figure 6: Convergence speed of QPSO algorithm at different population size.

Experiments and Analysis

Evaluation criteria

In this subsection, the benchmark results of various algorithms including EQPSO-2M were compared in terms of convergence speed, mean best fitness, STD, SR, Minimum best fitness, and Maximum best fitness. Moreover, every one of the outcomes is tested with a nonparametric statistical investigation utilizing Wilcoxon rank-sum test. To investigate the efficiency of the proposed approach for optimizing the RBFNN parameters, it is compared with other optimization algorithms. These algorithms incorporate the TAPSO [21], Modified PSO (MPSO) [20], Autonomous Groups PSO (AGPSO) [22], enhanced leader PSO (ELPSO) ELPSO [23], modified PSO with inertia weight coefficient (PSO-In) [45], and the traditional QPSO algorithm [39]. Moreover, the proposed scheme

is compared with an enhanced QPSO algorithm (EQPSO-1M) that was developed during this research dependent on diversity control and just a one mutation strategy. To ensure the fairness of the comparison, every one of the outcomes is gotten dependent on the results of 30 free experiments through 2000 cycles. Meanwhile, all tests are done utilizing a similar PC and with similar conditions. MATLAB 2019B software is utilized for creating and testing during every one of the investigations.

Parameter settings

The problem dimension is set to seven variables that represent the proposed RBFNN architecture. All the control parameters of the algorithms are chosen by the suggestions from the original literature. Concerning the TAPSO algorithm [21], the acceleration coefficients c_1 and c_2 are refreshed by Eqs. 11 and 12. The values of c_{max} and c_{min} are set to 2.5 and 0.5, respectively. In the MPSO algorithm [20], c_1 and c_2 are updated during the search process utilizing Eqs. 14 and 15. c_{1max} is set to 2.25, and c_{1max} = 1.25, c_{2max} is set to 2.55, and c_{2min} is set to 0.5. The inertia weight coefficient w is diminished linearly from 1 to 0.4. As to the AGPSO algorithm [22], the particles are divided into groups, where c_1 and c_2 for each group are refreshed by Table 4: In Table 4, T represents the greatest number of iterations, and t shows the current iteration.

Table 4: AGPSO coefficients updating strategies

Group	c_1	c_2
Group 1	$1.95-2t^{1/3}/T^{1/3}$	$2t^{1/3}/T^{1/3}+0.05$
Group 2	$(-2t^3/T^3)+2.5$	$(2t^3/T^3)+0.5$
Group 3	$1.95-2t^{1/3}/T^{1/3}$	$(2t^3/T^3)+0.5$
Group 4	$(-2t^3/T^3)+2.5$	$2t^{1/3}/T^{1/3}+0.05$

Besides, the inertia weight parameter w is diminished step by step from 0.9 to 0.4. In the ELPSO calculation $c_1 = c_2 = 2$, w is decreased linearly from 0.9 to 0.4. the STD of the gaussian mutation is set to 1, and the scale parameter of Cauchy mutation is 2. In PSO with inertia weight coefficient, the inertia weight coefficient is set to 0.5, $c_1 = c_2 = 1.5$. Therefore, the values of the coefficients are updated according to the following equation [45]:

$$c_1 + c_2 < \frac{24\,(1-\omega^2)}{7-5\omega} \tag{21}$$

For the QPSO algorithm, the upper and lower limits of the search interval are [-5, 5], and the threshold of the diversity is set to 1E-05 as experimentally seen from the simulation.

Discussion

The statistical results acquired by the proposed EQPSO-2M and different algorithms are introduced in Table 5. As can be acquired from the results in Table 5, the proposed EQPSO-2M outperforms all other peers in terms of Max best fitness value, mean best fitness value, STD, and SR. In the meantime, the SR of 96.7% at the proposed algorithm shows that the algorithm converges to the global minima at 29 of the 30 experiments. Besides, the STD results demonstrate the higher stability of the proposed algorithm in optimizing the RBFNN parameters. It very well may be seen from Table 5 that the proposed algorithm outperforms any remaining peers by 1.6E-07 of STD. The second-best outcomes in terms of STD, SR, mean best fitness, and the Max best fitness are acquired by the EQPSO-1M algorithm. It is obvious that applying the diversity control and a single mutation has further improved the SR of the EQPSO-1M by 43% compared with conventional QPSO. However, using the diversity control and two progressive mutations have enhanced the SR of the EQPSO-2M by 53%. The third best outcomes in terms of SR and mean best fitness are accomplished by PSO-In, TAPSO, and QPSO algorithms, respectively. Therefore, in light of STD, EQPSO-2M is positioned one followed by EQPSO-1M, AGPSO, MPSO, ELSPO, TAPSO, PSO-In, and QPSO, respectively. In terms of the mean best fitness, the best four outcomes are accomplished by EQPSO-2M, EQPSO-1M, PSO-In, and TAPSO, individually. The ELPSO and MPSO algorithms have a similar SR of ≈13 %. The improvement in the results of the PSO-In is because of the legitimate determination of the PSO control parameters ω, c_1, and c_2. Besides, the TAPSO algorithm profits by the dramatic time-fluctuating acceleration coefficients that enhanced the exploration in the beginning stage and the exploitation in the later period of the search cycle. The fundamental justification of the bad outcomes got by the ELPSO algorithm is the utilization of constant values for $c_1 = c_2$. Hence, the algorithm fails to make a balance between global and local searching stages, and it is caught in local minima at 26 experiments. Moreover, the observed outcomes reveal that the AGPSO and MPSO algorithms failed to accomplish a decent harmony among exploration and exploitation. In this way, the algorithms get trapped in most of the experiments and have the most exceedingly awful mean best fitness.

Table 5: Comparison results between EQPSO-2M and other algorithms

Algorithm	Minimum best fitness	Maximum best fitness	Mean best fitness	STD	Success rate, %
TAPSO	6.7E−07	0.3229	4.4E−06	6.1E−06	63.3
PSO-In	6.5E−07	0.4624	3.9E−06	6.4E−06	73.3
MPSO	6.6E−07	0.6003	1.0E−05	4.8E−06	13.3
AGPSO	6.7E−07	0.4677	8.4E−06	3.9E−06	20
ELPSO	7.0E−07	0.1998	1.1E−05	5.8E−06	13.1
QPSO	6.6E−07	0.3977	6.2E−06	6.5E−06	43.3
EQPSO-1M	6.5E−07	0.0393	3.2E−06	1.8E−06	86.7
EQPSO-2M	**6.5E−07**	**0.0088**	**6.9E−07**	**1.6E−07**	**96.7**

The convergence speed is a significant factor that can be utilized to assess optimization problems. Therefore, to additionally assess the performance of the proposed EQPSO-2M strategy, its convergence speed is compared with different algorithms. Figure 7 shows the average convergence curves that are plotted in a logarithmic scale for the proposed EQPSO-2M and different algorithms. As can be seen from Fig. 7, that the average fitness of the EQPSO-2 M algorithm is the minimum. In addition, the EQPSO-2M algorithm shows the best convergence speed compared with different peers. It converges to an average fitness value of ≈ 6.5E-07 at around 300 iterations. Besides, EQPSO-1M has the convergence speed after the proposed algorithm. Thanks for applying the diversity control and the single mutation. Based on the convergence speed, the second-best result is obtained by the PSO-In algorithm. This improvement in the results of the PSO-In ensures strong relation between the PSO convergence behavior and the control parameters selection. The acquired results demonstrate that the other algorithms (ELSPO, AGPSO, QPSO, MPSO, and TAPSO) show less performance and they have a slow convergence to the optimal minimum value. These algorithms have a delay in converging to the minimum best values due to stagnation conditions. The main reason for the high convergence rate of the proposed EQPSO-2M is that the utilization of the two successive mutations mechanism helps the swarm in each iteration to explore the search space extensively near the personal best position to find more best positions. In addition, the utilization of diversity control improves the diversity of the particles. Consequently, it reveals a better performance and more efficacy in jumping out from local minima in case of stagnation, and hence obtaining more high-quality positions. Consequently, the proposed EQPSO-2M achieves a better performance in terms of exploration and exploitation.

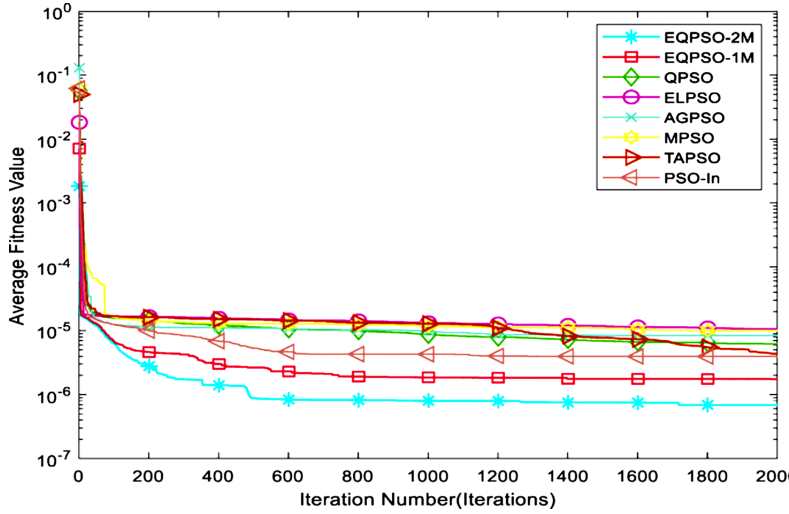


Figure 7: Average fitness curves for different algorithms.

From Table 5 and Fig. 7, the proposed EQPSO-2M algorithm has the best results compared to all other algorithms in terms of convergence accuracy and speed. The EQPSO-2M benefits from the diversity control and mutation strategies that allow the QPSO algorithm to generate a better global search ability and converge faster than the other algorithms. Moreover, the proposed algorithm achieves a good balance between exploration and exploitation. Thus, it has an efficient performance that allows escaping from the local minimum for finding the optimal parameters of the RBFNN-based reaction wheel model.

Wilcoxon rank-sum test analysis

Wilcoxon rank-sum test is a non-parametric test strategy of the t-test for two independent samples. It is utilized primarily to test that there are differences between two groups of samples. Moreover, it is utilized to test the invalid speculation that two samples are procured from a continuous distribution with equivalent means [46]. Additionally, utilizing the mean and STD values for assessing the performance of the proposed algorithm compared with different algorithms might be questionable. To determine this issue, the Wilcoxon rank-sum test as a nonparametric test method is utilized as

evidence that the results of the EQPSO-2M mechanism are not the same as those of different mechanisms. The significance level α is set to 0.05. In the interim, if the p-value is greater than 0.05, this implies that there is no huge distinction between the results of the two algorithms [47]. Something else, if the *p* value is lower than the significance level α, it implies that there is a huge contrast between the results of the compared algorithms. Table 6 shows the *p* values got by Wilcoxon rank total test at 0.05 significance level of EQPSO-2M outcomes against the consequences of QPSO-1M, QPSO, ELSPO, AGPSO, MPSO, TAPSO, and PSO-In.

Table 6: *p*-values of Wilcoxon rank sum test comparison between the results of EQPSO-2M and other algorithms

EGPSO-2M Vs.	*p* value
EGPSO-1M	0.0138
QPSO	6.11E−10
ELSPO	4.50E−11
AGPSO	7.39E−11
MPSO	8.99E−11
TAPSO	6.71E−10
PSO-In	0.0574

As displayed in Table 6, the *p* values that are lower than 0.05 show the predominance of the proposed algorithm. Notwithstanding, there is no huge contrast between the proposed algorithm and the PSO-In with coefficients controlled by [45]. However, the proposed EQPSO-2M algorithm reveals better performance in terms of convergence speed, SR, stability, and the mean best fitness.

Modeling Scheme Performance Evaluation

Based on the results of the proposed EQPSO-2M algorithm, three RBF-NN-based Reaction Wheel models have been created for the spacecraft roll, pitch, and yaw axes. In the meantime, the global best positions of EQPSO-2M represent the optimal values of the RBFNN coefficients. To evaluate the performance of the proposed models, they were tested for various tilting angles. Figure 8 shows the outputs of the developed RBFNN-based RW models compared with the actual reaction wheel outputs for 10° roll, pitch,

and yaw tilting angles. Furthermore, from Fig. 8, it very well may be seen that there is a good agreement between the models' outputs and the actual RW outputs. Moreover, the MSE error is about 3.9E−07 for roll, 3.5E−07 for pitch, and 5.9E−07 for yaw angles. Moreover, Fig. 8 reveals the superior matching between the models› outputs and the actual RW outputs at various working conditions.

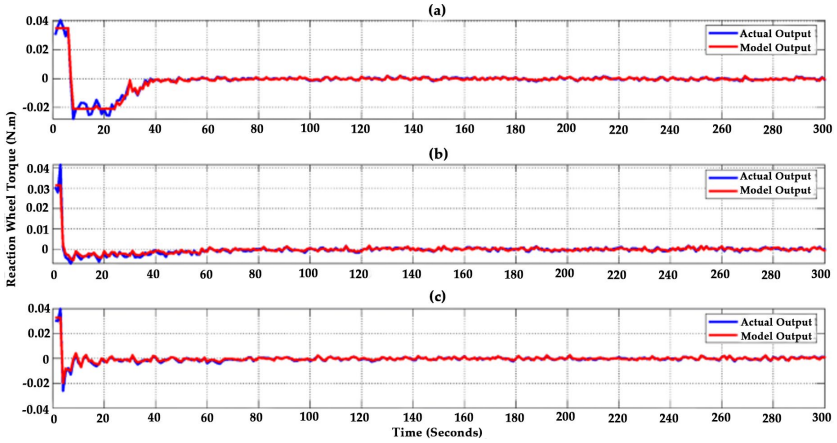


Figure 8: Output of reaction wheel model compared to the actual output at 10° (a) Roll output, (b) Pitch output, (c) Yaw output.

Modeling Scheme Generalization Evaluation

To explore the generalization of the proposed modeling scheme, the performance of the developed models was tested for tilting angles in the scope of [− 90°, 90°] for roll, pitch, and yaw. Figure 9 shows how the mean square errors between the models› outputs and the actual outputs change with the tilting angles. The results show that the three models can foresee the RW output torque with high precision. The MSE during the interval of [− 20°,20°] is about 6E−7 for roll, pitch, and yaw. It increases to arrive at 4E−6 at the interval limits. These tiny MSE values show the predominant presentation of the proposed RW models. Furthermore, Fig. 9 shows the generalization ability of the proposed modeling scheme that has demonstrated the ability of the models to work in a wide working scope of tilting angles with a high pointing accuracy.

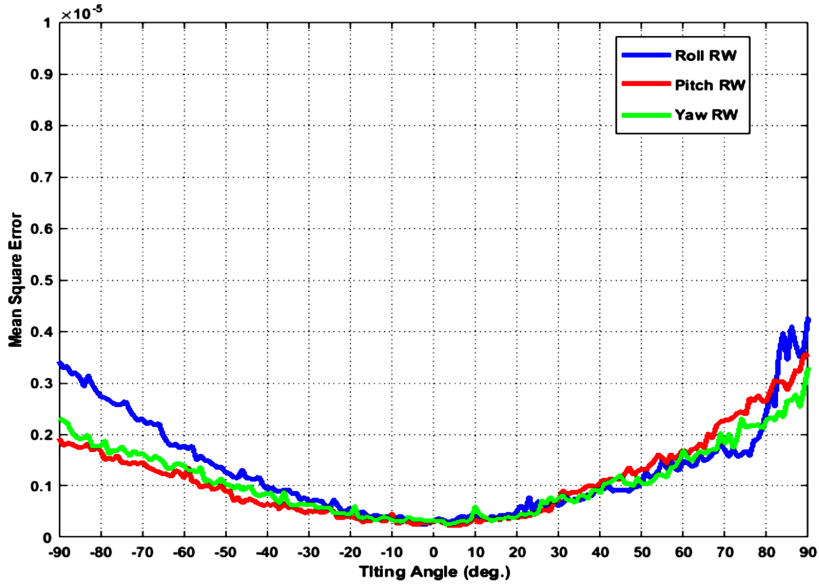


Figure 9: Performance of the developed RW models measured at tilting angle from − 90° to 90° roll, pitch, and yaw.

CONCLUSIONS

This research proposes another modeling scheme for the spacecraft reaction wheel utilizing RBFNN with an enhanced version of QPSO. In light of the principles of the diversity control and mutation strategy, EQPSO-2M is proposed to ameliorate the RBFNN parameters. In this way, the estimation of the RBFNN parameters is demonstrated as an optimization problem that was settled in terms of the EQPSO-2M algorithm. Additionally, the performance of the proposed algorithm was compared with other strategies like ELPSO, AGPSO, PSO-In, MPSO, TAPSO, and the conventional QPSO algorithm. Statistical benchmark rules dependent on the SR, convergence speed, and stability have shown the superiority and effectiveness of the proposed EQPSO-2M. Thanks to the EQPSO-2M algorithm for efficient performance and to accurately find the best particles' positions. Consequently, further improving the global search ability, helps the particles from stagnation in nearby optima and overcomes the premature convergence of the conventional QPSO.

In addition, the simulation results revealed that the proposed EQPSO-2M has a superior performance in terms of stability, mean best fitness value,

SR, and convergence speed. Moreover, three RBFNN-based reaction wheel models that are roll, pitch, and yaw were developed and then validated with MATLAB mathematical model. Extensive simulation has been done to evaluate the models' performance. Therefore, the very small value of MSE, which is close to 6.5E−7 indicates a distinct performance and stability of the proposed modeling scheme.

To further investigate the generalization of the proposed reaction wheels' models, they were tested for roll, pitch, and yaw angles in the range of [− 90°, 90°]. The superiority of the proposed approach additionally emanates from the MSE value, which is approximately proximate to 4E−6. Thus, the efficiency of testing results proves the capability of the proposed RBFNN modeling scheme. In fact, the EQPSO-2M algorithm is an efficient mechanism for optimizing the RBFNN parameters. Furthermore, the proposed modeling scheme is considered to be superior for modeling dynamic systems like spacecraft reaction wheels. It is recommended for the future work to utilize the developed model for the detection and identification of reaction wheel faults.

REFERENCES

1. Rahimi A, Saadat A (2020) Fault isolation of reaction wheels onboard three-axis controlled in-orbit satellite using ensemble machine learning. Aerosp Syst 3(2):119–126. https://doi.org/10.1007/s42401-020-00046-x
2. Afram A, Farrokh J (2015) Black-box modeling of residential HVAC system and comparison of gray-box and black-box modeling methods. Energ Buildings 94:121–149. https://doi.org/10.1016/j.enbuild.2015.02.045
3. Al-Zyoud I, Khorasani K (2006) Neural network-based actuator fault diagnosis for attitude control subsystem of an unmanned space vehicle. In: Proceeding Of 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC Canada, pp 3686–3693. https://doi.org/10.1109/IJCNN.2006.247383
4. Li Z, Ma L, Khorasani K (2006) dynamic neural network-based reaction wheel fault diagnosis for satellites. In: Proceeding Of 2006 International Joint Conference on Neural Networks Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, pp 3714–3721. https://doi.org/10.1109/IJCNN.2006.247387
5. Mousavi S, Khorasani K (2014) Fault detection of reaction wheels in attitude control subsystem of formation flying satellites: A dynamic neural network-based approach. Int J Intell Unmanned Syst 2(1):2–26. https://doi.org/10.1108/IJIUS-02-2013-0011
6. Vahid I et al (2017) Supervisory algorithm based on reaction wheel modelling and spectrum analysis for detection and classification of electromechanical faults. IET Sci Meas Technol 11(8):1085–1093
7. Mba CU et al (2017) Fault Diagnosis in Flywheels: Case Study of a Reaction Wheel Dynamic System with Bearing Imperfections. Int J Performability Eng 13(4):362–373
8. Franceso S, Daniel A et al (2018) A novel Dynamic Model of a Reaction Wheel Assembly for High Accuracy Pointing Space Missions. In: Proceedings of ASME Dynamic Systems and Control Conference, Atlanta, Georgia, USA. https://doi.org/10.1115/DSCC2018-8918
9. Rajesh K, Smriti S, Gupta1 (2016) Modeling and adaptive control of nonlinear dynamical systems using radial basis function network. Soft Comput 21(15):4447–4463. https://doi.org/10.1007/s00500-016-2447-9

10. Xie Y, Yu J, Xie S, Huang T, Gui W (2019) On-line prediction of ferrous ion concentration in goethite process based on self-adjusting structure RBF neural network. Neural Netw 116:1–10. https://doi.org/10.1016/j.neunet.2019.03.007
11. Ali N, Noor M, Mohammed F, Ahmed E (2018) RBF-NN-based model for prediction of weld bead geometry in Shielded Metal Arc Welding (SMAW). Neural Comput Applic 29:889–899. https://doi.org/10.1007/s00521-016-2496-0
12. Linag et al (2020) Radial Basis Function Neural Network for prediction of medium frequency sound absorption coefficient of composite structure open-cell aluminum foam. Appl Acoust 170:107505. https://doi.org/10.1016/j.apacoust.2020.107505
13. Ye Y, Qi Y, Shi D, Sun Y, Zhou Y, Hecht M (2020) Rotary-scaling fine-tuning (RSFT) method for Optimizing RAILWAY WHEEL profiles and its application to a locomotive. Railw Eng Sci 28(2):160–183. https://doi.org/10.1007/s40534-020-00212-z
14. Ye Y, Vuitton J, Sun Y, Hecht M (2021) Railway wheel profile fine-tuning system for profile recommendation. Railw Eng Sci 29(1):74–93. https://doi.org/10.1007/s40534-021-00234-1
15. Qi Y, Dai H, Wu P, Gan F, Ye Y (2021) RSFT-RBF-PSO: A RAILWAY WHEEL Profile optimisation procedure and its application to a metro vehicle. Veh Syst Dyn:1–21. https://doi.org/10.1080/00423114.2021.1955135
16. Shanshan TU et al (2020) A novel quantum inspired particle swarm optimization algorithm for electromagnetic applications. IEEE Access 8:21909–21916. https://doi.org/10.1109/ACCESS.2020.2968980
17. Xin-gang Z, Liang J et al (2020) An improved quantum particle swarm optimization algorithm for environmental economic dispatch. Expert Syst Appl 152:113370. https://doi.org/10.1016/j.eswa.2020.113370
18. Shi Y, Eberhart R (1998) A modified Particle Swarm Optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence, Anchorage, AK, USA, 69-73
19. Z. Cui, J. Zeng and Y. Yin (2008) An Improved PSO with Time-Varying Accelerator Coefficients. 2008 Eighth International Conference on Intelligent Systems Design and Applications, Kaohsuing, Taiwan,638-643 https://doi.org/10.1109/ISDA.2008.86.

20. G. Q. Bao and K. F. Mao (2009) Particle swarm optimization algorithm with asymmetric time varying acceleration coefficients. 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guilin, China, 2134-2139 https://doi.org/10.1109/ROBIO.2009.5420504.
21. T. Ziyu and Z. Dingxue, et al (2009) A Modified Particle Swarm Optimization with an Adaptive Acceleration Coefficients. 2009 Asia-Pacific Conference on Information Processing, Shenzhen, China, 330-332 https://doi.org/10.1109/APCIP.2009.217.
22. Mirjalili S et al (2014) Autonomous Particles Groups for Particle Swarm Optimization. Arab J Sci Eng 39(6):4683–4697. https://doi.org/10.1007/s13369-014-1156-x
23. Jordehi AR (2014) Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimization problems. Appl Soft Comput 26:401–417. https://doi.org/10.1016/j.asoc.2014.10.026
24. Sun J, Feng B, Xu W (2004) Particle Swam Optimization with Particles Having Quantum Behavior. Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), Portland, OR, USA, USA, 325-331.
25. Kun Y (2018) Quantum-Behaved particle swarm optimization for far-distance rapid cooperative rendezvous between spacecraft. Adv Space Res 62(11):2998–3011
26. Tianyu L (2016) Cultural quantum-behaved particle swarm optimization for environmental/economic dispatch. Appl Soft Comput 48:597–611. https://doi.org/10.1016/j.asoc.2016.04.021
27. Bialke B (1998) High fidelity mathematical modeling of reaction wheel performance, In: Annual Rocky Mountain guidance and control conference; 21st, Guidance and control, Breckenridge.
28. Yaguang Y (2019) Spacecraft modeling, Attitude determination, and control quaternion-based approach. CRC Press, Taylor & Francis Group, U.S.Ahttps://doi.org/10.1201/9780429446580
29. Omran EA, Murtada W (2019) An Efficient anomaly classification for spacecraft reaction wheels. Neural Comput Applic 31(7):2741–2747. https://doi.org/10.1007/s00521-017-3226-y
30. Afshin R, Krishna D et al (2020) Fault Isolation of Reaction Wheels for Satellite Attitude Control. IEEE Trans Aerosp Electron Syst 56(1):610–629. https://doi.org/10.1109/TAES.2019.2946665

31. Broomhead D S, David L (1988) Radial Basis Functions Multi-Variable Functional Interpolation and Adaptive Networks. Complex Systems. 2:321-355
32. Ortombina L, Tinazzi F, Zigliotto M (2017) Magnetic Modelling of Synchronous Reluctance and Internal Permanent Magnet Motors Using Radial Basis Function Networks. IEEE Trans Ind Electron 65(2):1140–1148. https://doi.org/10.1109/TIE.2017.2733502
33. Yadav AK (2017) Daily array yield prediction of grid-interactive photovoltaic plant using relief attribute evaluator based Radial Basis Function Neural Network. Renew Sustain Energy Rev 18:2115–2127. https://doi.org/10.1016/j.rser.2017.06.023
34. Alexandridis A, Chondrodima E, Sarimveis H (2013) Radial Basis Function Network Training Using a Nonsymmetric Partition of the Input Space and Particle Swarm Optimization. IEEE Trans Neural Netw Learn Syst 24:219–230. https://doi.org/10.1109/TNNLS.2012.2227794
35. Kennedy J, Eberhart R (1995) Particle Swarm Optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1942-1948
36. Najjarzadeh M, Sadjedi H (2020) Implementation of particle swarm optimization algorithm for estimating the innovative parameters of a spike sequence from noisy samples via maximum likelihood method 106: 102799
37. Chen K, Fengyu Z et al (2017) A hyprid particle swam optimization with sine cosine acceleration coefficients. Inform Sci 422:218–241. https://doi.org/10.1016/j.ins.2017.09.015
38. Khan S, Yang S, Ur Rehman, Obaid (2017) A Global Particle Swarm Optimization Algorithm Applied to Electromagnetic Design Problem. Int J Appl Electromagn Mech 53(3):451–467. https://doi.org/10.3233/JAE-160063
39. Sun J, Wu X, Palade V, Fang W, Lai C-H, Xu W (2012) Convergence analysis and improvements of quantum-behaved particle swarm optimization. Inform Sci 139:81–103. https://doi.org/10.1016/j.ins.2012.01.005
40. Amandeep S, Mandeep K et al (2020) QPSO-CD: quantum-behaved particle swarm optimization algorithm with Cauchy distribution. Quantum Inf Process 19(10):345. https://doi.org/10.1007/s11128-020-02842-y

41. Liu W, He J, Hongbo S (2017) A cooperative quantum particle swarm optimization based on multiple groups. In: Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Canada
42. Xie Y, Peng L (2021) Differential privacy distributed learning under chaotic quantum particle swarm optimization. Computing 103(3):449–472. https://doi.org/10.1007/s00607-020-00853-2
43. Zurada JM (1992) Introduction to artificial neural systems. St. Paul: West Publishing Company Los Angeles USA.
44. Danial J, Raja S, Koohyar F, Ahmad S (2017) Developing a hybrid PSO–ANN model for estimating the ultimate bearing capacity of rock-socketed piles. Neural Comput Applic 28(2):391–405. https://doi.org/10.1007/s00521-015-2072-z
45. C. W. Cleghorn and A. P. Engelbrecht (2014) Particle swarm convergence: An empirical investigation," 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, 2524-2530, https://doi.org/10.1109/CEC.2014.6900439.
46. Mohammadi D, Abd Elaziz M, Moghdani R, Demir E, Mirjalili S (2021) Quantum Henry gas solubility optimization algorithm for global optimization. Eng Comput. https://doi.org/10.1007/s00366-021-01347-1
47. Joaquin D, Garcia S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18. https://doi.org/10.1016/j.swevo.2011.02.002

Chapter

3

MATHEMATICAL MODELING OF VAPORIZATION DURING LASER-INDUCED THERMOTHERAPY IN LIVER TISSUE

Sebastian Blauth[1,2] , Frank Hübner[3] , Christian Leithäuser[1] , Norbert Siedow[1] and Thomas J. Vogl[3]

[1]Fraunhofer Institute for Industrial Mathematics ITWM, Kaiserslautern, Germany
[2]TU Kaiserslautern, Kaiserslautern, Germany
[3]Institute for Diagnostic and Interventional Radiology of the J.W. Goethe University Hospital, Frankfurt/Main, Germany

ABSTRACT

Laser-induced thermotherapy (LITT) is a minimally invasive method causing tumor destruction due to heat ablation and coagulative effects. Computer simulations can play an important role to assist physicians with the planning and monitoring of the treatment. Our recent study with ex-vivo

Citation: (APA): Blauth, S., Hübner, F., Leithäuser, C., Siedow, N., & Vogl, T. J. (2020). Mathematical modeling of vaporization during laser-induced thermotherapy in liver tissue. Journal of Mathematics in Industry, 10(1), 1-16. (16 pages).

porcine livers has shown that the vaporization of the water in the tissue must be taken into account when modeling LITT. We extend the model used for simulating LITT to account for vaporization using two different approaches. Results obtained with these new models are then compared with the measurements from the original study.

INTRODUCTION

Thermal ablation methods briefly generate cytotoxic temperatures in tumorous tissue in order to destroy it. These minimally invasive methods are used for treating cancer, e.g., in lung, liver, or prostate, when surgical resection is either not possible or too dangerous for the patient. All of these methods utilize the fact that tumorous tissue is more susceptible to heat than healthy tissue to destroy as little healthy tissue as possible. Among the most common thermal ablation methods are LITT, radio-frequency ablation, and microwave ablation.

The principle of LITT [1] is based on the local supply of energy via an optical fiber, located in a water-cooled applicator. This applicator is placed directly into the tumorous tissue. The LITT treatment can take place under MRI control because the laser applicator is sourced by an optical fiber and does not include any metal parts. Therefore the patient is not exposed to radiation, in contrast to other treatments that can only be carried out under CT control.

For the therapy planning of LITT, accurate numerical simulations are needed to guide the practitioner in deciding when to stop the treatment. Mathematical models for this have been proposed, e.g., in [2, 3]. The liver consists of about 80% water which vaporizes if the temperatures during the treatment become sufficiently large. The vaporization of this water is currently not included in these models but our study in [4, 5] suggests that this effect is relevant for an accurate simulation. In this study the ex-vivo experiments with a larger power of 34 W show a good agreement between measured and simulated temperature until the temperature reaches approximately 100°C. Then, the measured temperature stagnates while the simulated one rises further (cf. [4], Fig. 3). We presume that this happens due to phase change of water which was not included in the model we used.

In this paper we use the measurements from [4] and compare two models for the vaporization. One of them is the effective specific heat (ESH) model

introduced in [6] which modifies the specific heat coefficient to account for the phase change. The other one is the enthalpy model which uses an additional state equation to model the phase transition. We compare the models to experimental data with ex-vivo porcine livers from [4].

Of course the presence of vapor makes the situation far more complicated. The vapor expands, pressure builds up and the vapor has its own dynamics within the tissue. Once the vapor reaches a cooler region it may condensate again. There are many approaches studying this in detail [7, 8]. The drawback of course is that such a detailed approach makes the model far more complex, at the costs of computational time, and introduces new tissue dependent parameters, which may not be easily available. Therefore, in this study we use and extremly simplified approach to model the vapor which was proposed by [6] and does not include any physically motivated transport mechanism for the vapor. One purpose of this study is to investigate if this simplified approach may be sufficient for modeling LITT or if more advanced models are necessary to account for the vapor (see also Remark 2).

This paper is structured as follows. Our existing mathematical model for simulating LITT including heat and radiative transfer is described in Sect. 2. This model is based on the work of [2] and we have also used it in [4]. In Sect. 3 we modify and extend this model in such a way that it also covers the effect of vaporization during the treatment. Therefore, we consider both the ESH model of [6] as well as an enthalpy model for vaporization. Afterwards, we present the details of the numerical solution of our models in Sect. 4. Finally, the models are validated with measurement data obtained from experiments made with ex-vivo porcine liver tissue (cf. [4]) in Sect. 5.

MATHEMATICAL MODEL

We denote by $\Omega \subset \mathbb{R}^3$ the geometry of the liver and by $\Gamma=\partial\Omega$ its boundary. The latter consists of the radiating surface of the adjacent applicator Γ_{rad}, the cooled surface of the applicator Γ_{cool}, and the ambient surface of the liver Γ_{amb} (see Fig. 1). The mathematical model is described by a system of partial differential equations (PDEs) for the heat transfer inside the liver, the radiative transfer from the applicator into the liver tissue, and a model for tissue damage (cf. [2–4]).

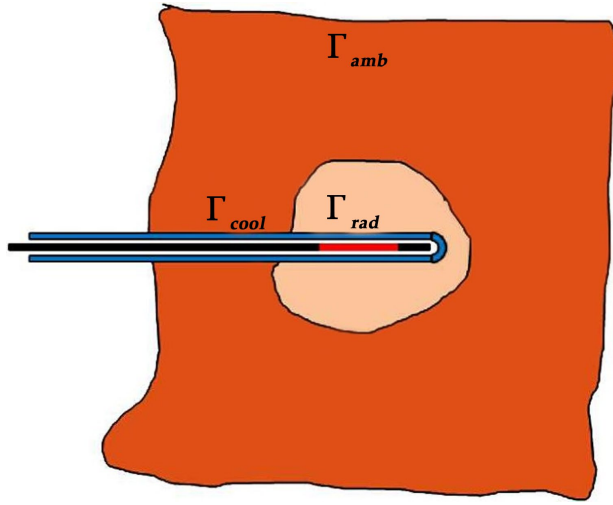


Figure 1: Sketch of the geometry including the water-cooled applicator with radiating laser fiber.

Heat Transfer

The heat transfer in the liver tissue is modeled by the well-known *bio-heat equation* (cf. [9])

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (\kappa \nabla T) + \xi_b (T - T_b) = Q_{rad} \quad \text{in } (0,\tau) \times \Omega,$$
$$T(0,\cdot) = T_{init} \quad \text{in } \Omega, \tag{1}$$

where T=T(x,t) denotes the temperature of the tissue, depending on the position x∈Ω and the time t∈(0,τ). Here, the end time of the simulation is denoted by τ>0. Further, C_p is the specific heat capacity, ρ the density of the tissue, and κ the thermal conductivity. The perfusion rate due to blood flow is denoted by ξ_b and the blood temperature by T_b. Note that in the current ex-vivo study the perfusion rate ξ_b is set to zero. Finally, Q_{rad} is the energy source term due to the irradiation of the laser fiber and the initial tissue temperature distribution is given by T_{init}.

For the heat transfer between the tissue and its surroundings, given by the ambient surface and the applicator, the following Robin type boundary conditions are used

$$\kappa\,\partial_n T = \alpha_{\text{cool}}(T_{\text{cool}} - T) \quad \text{on } (0,\tau)\times(\Gamma_{\text{rad}}\cup\Gamma_{\text{cool}}),$$

$$\kappa\,\partial_n T = \alpha_{\text{amb}}(T_{\text{amb}} - T) \quad \text{on } (0,\tau)\times\Gamma_{\text{amb}}.$$

Here, n is the outer unit normal vector on Γ. Additionally, α_{cool} and α_{amb} are the heat transfer coefficients for the water-cooled part of the applicator and the surroundings of the liver, respectively. The temperature of the cooling water is denoted by T_{cool} and T_{amb} is the ambient temperature.

Remark 1

Please note that the temperature T_{cool} of the water coolant is assumed to be known and constant in this study. This is of course a simplification because the cooling water is heated up on its way through the applicator. However, measurements of the cooling temperature before and after the applicator in [4, Fig. 2] show that the temperature of the coolant does not increase by more than 5°C. Therefore, setting T_{cool} to the measured inlet coolant temperature should approximate the problem. Of course it is also possible to model the flow through the applicator in detail as done in [2].

We come back to this bio-heat equation in Sect. 3, where we modify it such that it also covers the effect of vaporization of water in the tissue. The radiative source term Q_{rad} is defined in the next section by (5).

Radiative Transfer

The irradiation of laser light is modeled by the *radiative transfer equation*

$$s\cdot\nabla I + (\mu_a+\mu_s)I = \frac{\mu_s}{4\pi}\int_{S^2} P(s\cdot s')I(s',x)\,\mathrm{d}s' \quad \text{in } S^2\times\Omega, \tag{2}$$

where the radiative intensity $I=I(s,x)$ depends on a direction $s\in S^2$ on the (unit) sphere and the position $x\in\Omega$, and μ_a and μ_s are the absorption and scattering coefficients, respectively. In particular, as that radiative transfer happens significantly faster than temperature transfer, we neglect the time-dependence and use this quasi-stationary model. The scattering phase function $P(s\cdot s')$ is given by the Henyey-Greenstein term which reads (cf. [10])

$$P(s\cdot s') = \frac{1-g^2}{(1+g^2-2g(s\cdot s'))^{3/2}}.$$

Here, g∈[−1,1] is the so-called anisotropy factor that describes backward scattering for g=−1, isotropic scattering in case g=0 and forward scattering for g=1.

Due to the high dimensionality of the radiative transfer equation (2), we use the so-called P_1-approximation to model the radiative energy, the details of which can be found, e.g., in [11]. Introducing the ansatz

$$I(s,x) = \phi(x) + 3s \cdot q(x),$$

where $q(x) = \frac{1}{4\pi} \int_{S^2} I(s,x)s \, \mathrm{d}s$ is radiative flux vector, one obtains the much simpler three-dimensional diffusion equation

$$-\nabla \cdot (D\nabla\varphi) + \mu_a\varphi = 0 \quad \text{in } \Omega, \tag{3}$$

where φ=φ(x) is the radiative energy and the diffusion coefficient D is given by

$$D = \frac{1}{3(\mu_a + (1-g)\mu_s)}.$$

To derive the boundary conditions we use Marshak's procedure as described in, e.g., [11]. We obtain Robin type boundary conditions

$$D\frac{\partial\varphi}{\partial n} = \frac{q_{\text{app}}}{A_{\Gamma_{\text{rad}}}} \quad \text{on } \Gamma_{\text{rad}}, \qquad D\frac{\partial\varphi}{\partial n} + b\varphi = 0 \quad \text{on } (\Gamma_{\text{cool}} \cup \Gamma_{\text{amb}}), \tag{4}$$

where q_{app} is the laser power entering the tissue and $A_{\Gamma_{\text{rad}}}$ the surface area of the radiating part of the applicator. The former can be written as

$$q_{\text{app}}(t) = \begin{cases} (1-\beta_q)\hat{q} & \text{if } t_{\text{on}} \le t \le t_{\text{off}}, \\ 0 & \text{otherwise,} \end{cases}$$

where $\hat{q}$ is the configured laser power and the factor (1−βq) models the absorption of energy by the coolant (cf. [4]). Moreover, the parameter b in (4) is given as b=0.5 on Γ_{amb} and b=0 on Γ_{cool}. From the numerical point of view the system given by (3) and (4) is much easier to solve than the original system given by (2). Finally, the radiative energy is used to define the source term for the bio-heat equation in the following way

$$Q_{\text{rad}}(x) = \mu_a\varphi(x). \tag{5}$$

Tissue Damage and Its Influence on Optical Parameters

The optical parameters μ_a, μ_s and g are very sensitive to changes of tissue's state. In particular, once the coagulation of cells starts, their optical parameters change and, as a result, the radiation cannot enter the tissue as deeply as before. Therefore, we model the damage of the tissue as in, e.g., [2, 3] with the help of the *Arrhenius law*, which is given by

$$\omega(t,x) = \int_0^t A \exp\left(-\frac{E_a}{RT(s,x)}\right) \mathrm{d}s, \tag{6}$$

with so-called frequency factor A, activation energy E_a, and universal gas constant R. This describes the change of optical parameters due to coagulation in the following way

$$\mu_a = \mu_{an} + \left(1 - e^{-\omega}\right)(\mu_{ac} - \mu_{an}),$$

$$\mu_s = \mu_{sn} + \left(1 - e^{-\omega}\right)(\mu_{sc} - \mu_{sn}),$$

$$g = g_n + \left(1 - e^{-\omega}\right)(g_c - g_n),$$

where the subscripts n and c indicate properties of native and coagulated tissue, respectively (cf. [2]).

MATHEMATICAL MODELING OF VAPORIZATION

Vaporization of water inside organic materials plays an important role in many different fields, e.g., in medicine or the food industry. To model the temperature distribution in such materials correctly, it is important to take the vaporization into account as a significant amount of energy is necessary for the phase transition from water to vapor. The basic principle is the following (see, e.g., [12]). If energy in the form of heat is added to water (under constant pressure), the water's temperature increases as long as it is below the vaporization temperature, i.e., below 100°C. However, as soon as the water reaches this temperature, the temperature does not increase further, although heat is still added to the water. At this point, the water starts to boil and eventually vaporizes after a sufficient amount of energy was added to it. Finally, the temperature of the emerging water vapor increases again after all water has been vaporized. This happens due to the fact that the energy added to the water at its boiling point is used to change its phase and not to increase its temperature, until all water is vaporized.

In the following, we discuss two vaporization models. First, we take a look at the effective specific heat (ESH) model introduced in [6] which uses a varying specific heat capacity to model the phase change. In this model the phase transition is spread over a reasonably small interval around 100°C. This simplification makes it possible to model the phase transition using a single PDE. Second, we propose an enthalpy model with an additional state equation for the enthalpy. For this model, the transition happens at a single temperature, namely at 100°C.

The Effective Specific Heat (ESH) Model

The ESH model introduced in [6] considers the following modified bio-heat equation

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (\kappa \nabla T) + \xi_b (T - T_b) = Q_{rad} - Q_{vap} + Q_{cond} \quad \text{in } (0, \tau) \times \Omega, \tag{7}$$

with the same initial and boundary conditions as (1). Here, Q_{vap} is a source term that models the vaporization of water and Q_{cond} is the source term for the condensation (see Sect. 3.2). In [6] this has the following form

$$Q_{vap} = -\lambda \frac{dW}{dt}, \tag{8}$$

where λ denotes the latent heat of water and W is the tissue water density. Using the chain rule we see that

$$\frac{dW}{dt} = \frac{\partial W}{\partial T} \frac{\partial T}{\partial t}.$$

Substituting this into (8) and (7) gives the following modified heat equation

$$\rho C_p' \frac{\partial T}{\partial t} - \nabla \cdot (\kappa \nabla T) + \xi_b (T - T_b) = Q_{rad} + Q_{cond} \quad \text{in } (0, \tau) \times \Omega,$$

where the effective specific heat capacity C_p' is given by

$$C_p' = C_p - \frac{\lambda}{\rho} \frac{\partial W}{\partial T}.$$

Since $\frac{\partial W}{\partial T} < 0$ for vaporization (the water content decreases with temperature), we have that $C_p' \geq C_p$.

Based on experiments that measured water content of bovine liver as a function of temperature in [13] the following function is used to describe the tissue water density (cf. [6, 13])

$$W(T) = 800 \cdot \begin{cases} (1 - e^{\frac{T-106}{3.42}}) & \text{if } T \leq 103°\text{C}, \\ S(T) & \text{if } 103°\text{C} < T \leq 104°\text{C}, \\ e^{-\frac{T-80}{34.37}} & \text{if } 104°\text{C} \leq T, \end{cases}$$

where S(T) is the cubic C^1 spline that interpolates between the two exponential functions, (approximately) given by

$$S(T) = 3.712982 \times 10^2\, T^3 - 11.47524\, T^2 + 1.182046 \times 10^3\, T - 4.058214 \times 10^4.$$

The function W and its derivative are depicted in Fig. 2. In particular, we get that the effective specific heat is very large in an area around 100°C. Therefore, it holds that

$$\frac{\partial T}{\partial t} \ll 1 \quad \text{for } T \text{ around } 100°\text{C},$$

which models the vaporization of the tissue water.

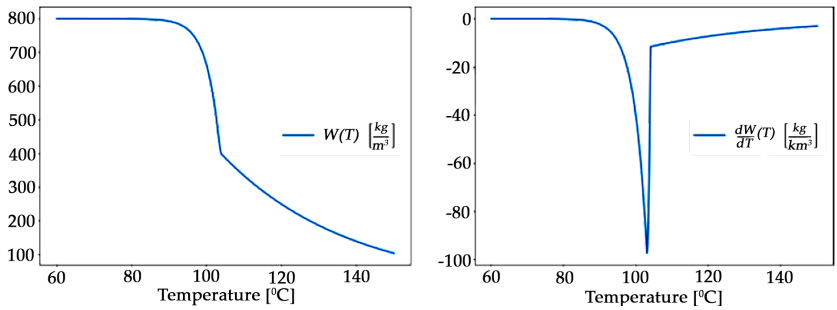


Figure 2: Function W(T)W(T) and derivative dWdT(T)dWdT(T) of tissue water density from [6].

Simple Condensation Model for ESH Model

In [6] it is discussed that, in addition to the vaporization of water, one also needs to consider the effect of condensation of the water vapor in order to obtain an accurate model. There, it was assumed that the water vapor diffuses

into a region of lesser temperatures where it condensates and releases its latent heat obtained through the vaporization. The authors of [6] describe their model for this in the following way. They say that they first calculate the total amount of water that was vaporized in the last time step. From this, the amount of latent heat generated is computed. Finally, this is added uniformly to the tissue region whose temperature is between 60°C and 80°C. We have implemented this simple condensation model in the following way. We compute the total amount of latent heat which is currently consumed through the vaporization of water by

$$\bar{Q}_{\text{vap}} = \int_{\Omega} Q_{\text{vap}} \, \mathrm{d}x,$$

where $[\bar{Q}_{\text{vap}}] = \mathrm{W}$. Additionally, we define the condensation region as

$$\Omega_{\text{cond}} := \left\{ x \in \Omega \mid T^{-} \leq T \leq T^{+} \right\},$$

for given temperature boundaries $T^{-} < T^{+} < 100°\mathrm{C}$. Uniformly distributing $\bar{Q}_{\text{vap}}$ over the condensation region then yields the condensation source term

$$Q_{\text{cond}}(x) = \begin{cases} \frac{\bar{Q}_{\text{vap}}}{\text{vol}(\Omega_{\text{cond}})} & \text{if } x \in \Omega_{\text{cond}} \text{ and } \text{vol}(\Omega_{\text{cond}}) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

In particular, this implies that our model is energy conserving. This is of course a very rough condensation model because there is no real transport mechanism for the vapor involved at all. Any vapor will instantaneously condensate in another region with lower temperature. This simple model shows promising results but there is also room for improvement as discussed in Sect. 5.4.

Remark 2

Clearly this approach for dealing with the vapor is a severe simplification of what actually happens: the expanding vapor builds up pressure and moves through the tissue, thus adding a fluid dynamical component to the problem. The vapor transport in the simple model is purely artificial and not motivated by physics. The trade-off is between a very simple model and a more accurate one which is also far more complex. One goal of this study

is to investigate if the simple model may be sufficient for LITT or if more advanced models for the vapor are needed [7, 8].

Enthalpy Model

In the section, we present the details of the second model for vaporization, which is based on an enthalpy formulation. It consists of two coupled equations, one for the temperature of the tissue and one for its enthalpy. For the temperature, we have the following, modified bio-heat equation

$$\rho C_p \frac{\partial T}{\partial t} = \begin{cases} \nabla \cdot (\kappa \nabla T) + \xi (T_b - T) + Q_{rad} + Q_{cond} & \text{if } T < 100°C \text{ or } T \geq 100°C \\ & \text{and } H = \rho\lambda_{vap}, \\ 0 & \text{if } T = 100°C \\ & \text{and } 0 \leq H < \rho\lambda_{vap}, \end{cases} \tag{9}$$

where $\lambda_{vap}=0.8\lambda$ is the proportion of the enthalpy of vaporization corresponding to the tissue's water content of 80%. Further, the (volumetric) enthalpy of the water H, $[H] = \text{J m}^{-3}$, is modeled by the following ODE

$$\frac{\partial H}{\partial t} = \begin{cases} 0 & \text{if } T < 100°C \text{ or } T \geq 100°C \\ & \text{and } H = \rho\lambda_{vap}, \\ \nabla \cdot (\kappa \nabla T) + \xi (T_b - T) + Q_{rad} & \text{if } T = 100°C \\ & \text{and } 0 \leq H < \rho\lambda_{vap}. \end{cases} \tag{10}$$

Equation (9) has the same initial and boundary conditions as (1), and the initial condition of the enthalpy is given by H=0 in Ω, i.e., no vaporization had happened before the treatment. The term Q_{cond} describes a heat source due to the condensation of water vapor in regions with temperatures below 100°C, similar to the one of the ESH model (cf. Sect. 3.2). Observe that the modified bio-heat equation (9) coincides with the classical bio-heat equation (1) and we also have H=0, i.e., no vaporization is happening, as long as we have that T<100°C everywhere. This changes as soon as T=100°C at some point $\bar{x} \in \Omega$. Then, we see that the bio-heat equation (9) gives $\frac{\partial T}{\partial t}(\bar{x}) = 0$ and, therefore, $T(\bar{x}) = 100°C$ in case $0 \leq H(\bar{x}) < \rho\lambda_{vap}$, i.e., the temperature at a point does not change until the enthalpy exceeds the enthalpy of vaporization $\rho\lambda_{vap}$. In the meantime, the energy that would usually lead to an increase in temperature now only increases the enthalpy, which models the phase change of the water in the tissue. Finally, as soon as the enthalpy reaches the enthalpy of vaporization, all water is vaporized and the bio-heat equation is valid again.

Simple Condensation Model for Enthalpy Model

Similar to Sect. 3.2 the simple condensation model suggested in [6] is used. In contrast to the ESH model, the total amount of latent heat can be computed from the change of enthalpy in the following way

$$\bar{Q}_{\text{vap}} = \int_{\Omega} \frac{\partial H}{\partial t} \, \mathrm{d}x. \tag{11}$$

Again, the condensation region is defined by

$$\Omega_{\text{cond}} = \left\{ x \in \Omega \mid T^{-} \leq T \leq T^{+} \right\},$$

and the condensation source term is

$$Q_{\text{cond}}(x) = \begin{cases} \frac{\bar{Q}_{\text{vap}}}{\text{vol}(\Omega_{\text{cond}})} & \text{if } x \in \Omega_{\text{cond}} \text{ and } \text{vol}(\Omega_{\text{cond}}) > 0, \\ 0 & \text{otherwise,} \end{cases}$$

where $\text{vol}(\Omega_{\text{cond}})$ denotes the volume of Ω_{cond}. With this, we get that the temperature increase due to condensation corresponds to the energy used to change the phase of the water, uniformly distributed over Ω_{cond}. Finally, note that the numerical discretization of this model is described in Sect. 4.2.

NUMERICAL METHODS

In this section, we detail the numerical methods used to discretize and solve the governing equations.

Numerical Solution of the Governing PDEs

The mathematical model for radiative heat transfer and the models for vaporization described above were used to simulate the behavior of ex-vivo porcine liver tissue during LITT. The computational geometry was generated using Open Cascade (Open Cascade SAS, Guyancourt, France) and the mesh was created with the help of GMSH, version 2.11.0 (cf. [14]). The governing equations were solved with the finite element method in Python, version 2.7, using the package FEniCS, version 2017.2 (cf. [15, 16]). For the numerical solution of the PDEs, we first (semi-)discretize the bio-heat equation in time using the implicit Euler method. Then, we use piecewise linear Lagrange elements for the spatial discretization of the temperature and radiative energy. The resulting sequence of linear systems was then solved with the help of PETSc (cf. [17]), where we used the conjugate gradient method with

a relative tolerance of 1×10^{-10}. Afterwards, the damage function is computed using a right-hand Riemann sum to discretize the time integral of (6).

Discretization of the Enthalpy Model

In the following we describe our discretization of the enthalpy model. In particular, to compute the temperature distribution from time t to t+Δt we proceed as follows. We first solve (3) to obtain the radiative energy at t+Δt. With this, we compute the temperature distribution at t+Δt from (1). Subsequently, we iterate over the nodes of the finite element mesh and check, whether the temperature exceeds 100°C. At these nodes, the temperature is set to 100°C and from the excess temperature we compute the corresponding increase in enthalpy. If the enthalpy surpasses the limit of $\rho\lambda_{\text{vap}}$, we return this surplus in the form of heat to the corresponding nodes. After doing so, we integrate the (local) changes in enthalpy over Ω to compute the total change of enthalpy ΔH. Therefore, we can now compute the source term $\bar{Q}_{\text{vap}}$ of (11) as follows

$$\bar{Q}_{\text{vap}} = \frac{\Delta H}{\Delta t},$$

which is then used as the source term for the next time step, simulating the release of enthalpy by the condensation of the water vapor. Then, the new tissue damage is computed from (6) and the procedure is continued until we reach the end time τ.

RESULTS AND DISCUSSION

We use the experiments from the study of [4] to test the vaporization models. In this study LITT was applied to ex-vivo porcine livers and the resulting temperature was measured with a probe. The experiment was repeated nine times with different laser powers and different flow rates for the applicator cooling system. For the study in [4], the authors used the mathematical model introduced in Sect. 2 which was derived from the one presented in [2]. However, the model did not take into account the vaporization of water in the tissue. While the general agreement between experiment and simulation was good, there were notably two outliers, namely the cases P34F47 and P34F70, for which the highest laser power was used. For these cases, the simulated probe temperature would rise to well above 100°C, while the measured probe temperature would reach a plateau below 100°C. Therefore,

in [4] the authors suspected that the missing vaporization model was the reason for this discrepancy. Now, we test this hypothesis by repeating the simulations with the previously introduced modified models that now include vaporization and condensation effects.

Experimental Setting

For the validation of our models, we use the measurements from the experiments made in [4]. For these, livers were obtained from pigs which had been slaughtered approximately 6 hours prior to the experiment. The temperature of the samples was room temperature at the beginning of the experiments. A laser applicator from Somatex® Medical Technologies (Teltow, Germany) was placed into the middle of the liver sample. An optical fiber from the same company with a diffuser part of 3 cm at its tip was inserted into the applicator for delivering the laser energy from a Nd:YAG laser device (MY30; Martin Medizintechnik, Tuttlingen, Germany; wavelength 1064 nm) to the tissue. The applicator was equipped with a cooling water circulation system to protect the optical fiber and prevent the burning of tissue in close proximity to the applicator. A temperature probe was introduced into the porcine liver and placed close to the applicator in order to generate temperature measurements for validating the models of LITT.

The setup for the nine test cases is shown in Table 1. The laser was applied with different powers, namely 22, 28, and 34 W, and different flow rates of the applicator cooling system. However, it is assumed that the effect of the coolant flow rate is negligible (cf. [4]). Furthermore, the position of the temperature probe is characterized by its radial distance d_r to the applicator axis as well as its distance d_z from the applicator tip, where the applicator itself is contained in the half space with $z \geq 0$. We now simulate this experiment again using the vaporization models introduced previously, and compare the results with the measurement data as well as with the results obtained by the original model which does not consider vaporization. The optical, thermal, and damage parameters used for the simulation are listed in Table 2. They are taken from [4] and the references therein (cf. [18–21]). For the condensation region Ω_{cond} we have chosen the points where the temperature was between $T^- = 60°C$ and $T^+ = 80°C$, as proposed in [6].

Table 1: Experimental setup for nine test cases (from [4])

Case label	P22F47	P22F70	P22F92	P28F47	P28F70	P28F92	P34F47	P34F70	P34F92
Laser Power [W]									
-Measured $\hat{q}_{app}$	22.1	22.1	22.1	28.0	28.0	28.0	33.8	33.8	33.8
Coolant $\dot{V}$ [ml/min]	47.2	69.9	91.7	47.5	70.3	91.8	47.2	70.4	92.2
Time [s]									
-Laser on t_{on}	24	30	36	18	30	60	18	24	48
-Laser off t_{off}	1266	1236	684	942	1722	1098	1206	948	1182
-End t_{end}	1284	1248	702	954	1734	1116	1218	972	1206
Probe Position [mm]									
-Radial d_r	10.1	11.4	9.2	13.5	13.7	11.1	11.2	9.9	9.6
-Axis-direction d_z	12.6	25.7	20.9	21.0	7.5	10.1	23.8	26.3	35.3

Table 2: Physical parameters for LITT in ex-vivo porcine liver tissue

Parameter	Value
Optical (native):	
Absorption coefficient μ_{an} [m^{-1}]	50
Scattering coefficient μ_{sn} [m^{-1}]	8000
Anisotropy factor g_n	0.97
Optical (coagulated):	
Absorption coefficient μ_{ac} [m^{-1}]	60
Scattering coefficient μ_{sc} [m^{-1}]	30,000
Anisotropy factor g_c	0.95
Thermal:	
Heat conductivity κ [W m^{-1} K^{-1}]	0.518
Heat capacity C_p [J kg^{-1} K^{-1}]	3640
Tissue density ρ [kg m^{-3}]	1137
Heat transfer coefficient α_{cool} [W m^{-2} K^{-1}]	250
Heat transfer coefficient α_{amb} [W m^{-2} K^{-1}]	44
Damage:	
Damage rate constant A [s^{-1}]	3.1×10^{98}
Damage activation energy E_a [J mol^{-1} K^{-1}]	6.3×10^{5}
Universal gas constant R [J mol^{-1} K^{-1}]	8.31
Vaporization:	
Latent heat of water λ [J kg^{-1}]	2257×10^{3}

The Case P34F47

Let us start the investigation of the vaporization models with the test case P34F47 of [4], where a laser power of 34 W was used. The results for this case are shown in Fig. 3, where the measurement from the temperature probe, the results for the model of [4] and the results for both vaporization models of Sect. 3 are shown. For this specific case, the probe temperature simulated without a vaporization model rises well above 100°C while the measured temperature reaches a plateau below 100°C (see Fig. 3). In contrast, both vaporization models do not overestimate the temperature to the end of the treatment and predict the occurring plateau correctly. This is further visualized in Fig. 4, where the difference of the models from the measurement over the entire treatment is depicted. These results show that all models are reasonably close to the measured temperature until up to about 80°C. After that point the model without vaporization overestimates the temperature significantly. The models that include vaporization give considerably better results since their predicted temperatures match the measured ones more closely throughout the whole treatment.

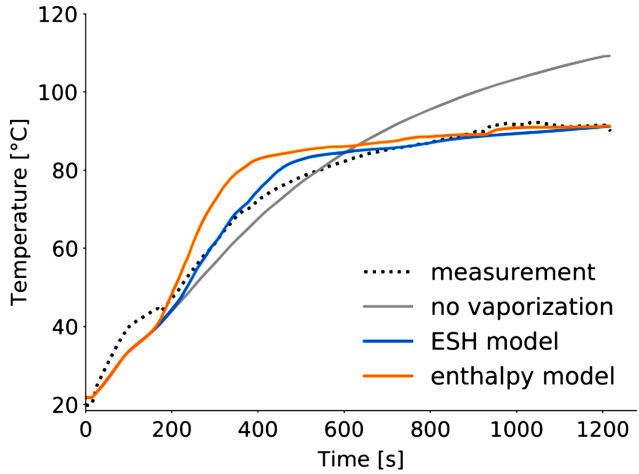


Figure 3: Comparison of the models for the case P34F47.

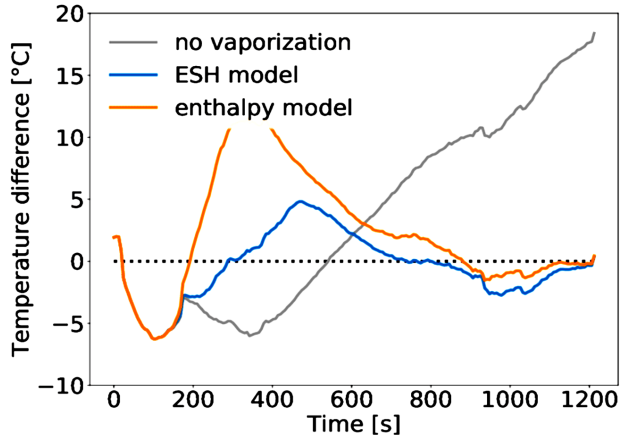


Figure 4: Difference between simulated and measured temperature for the case P34F47.

All Nine Test Cases

After investigating the vaporization models in the context of the previous test case, where the original model without vaporization of [2, 4] failed to predict the correct temperatures, we now investigate the other test cases from the study of [4]. The corresponding results are shown in Fig. 5, where the measured and simulated temperature at the probe is shown, and Fig. 6, which visualizes the difference of the simulated temperatures to the measurement. In general, the vaporization models are good in estimating the final temperature of the experiment. Especially for the cases P34F47 and P34F70, which could not be simulated accurately in [4], the vaporization model performs significantly better and does not overestimate the temperature to the end of the treatment. However, during the middle of the experiment the vaporization models tend to overestimate the temperatures. This can be seen, e.g., for the cases P22F70 and P28F70 (cf. Fig. 5). We suspect that the simple condensation model is responsible for this discrepancy as we explain in Sect. 5.4.

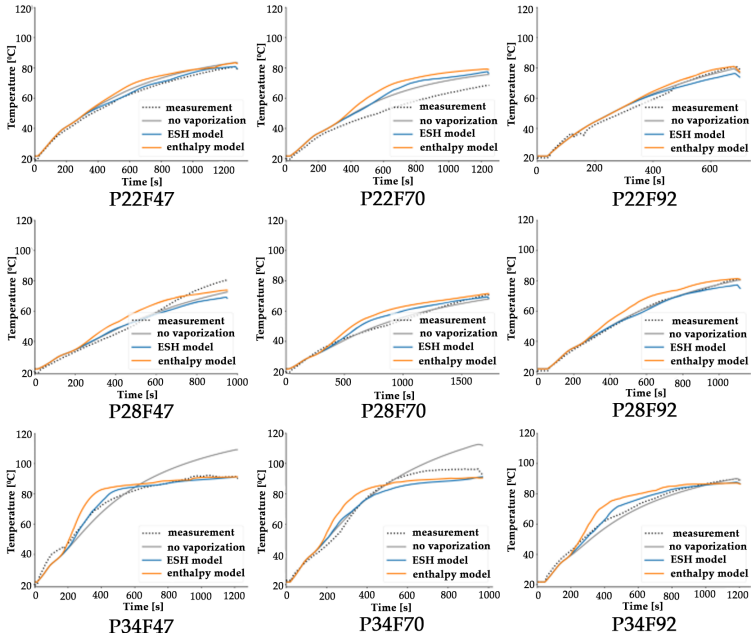


Figure 5: Comparison of the models with temperature measurements.

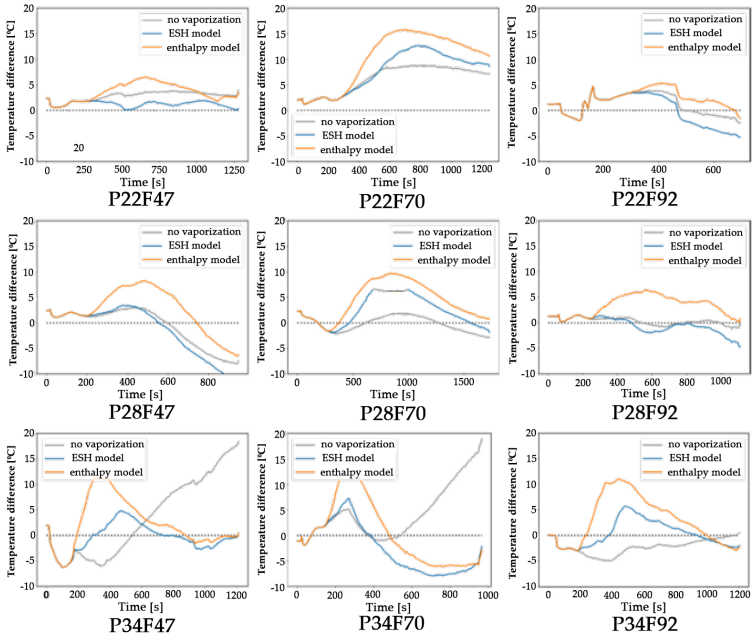
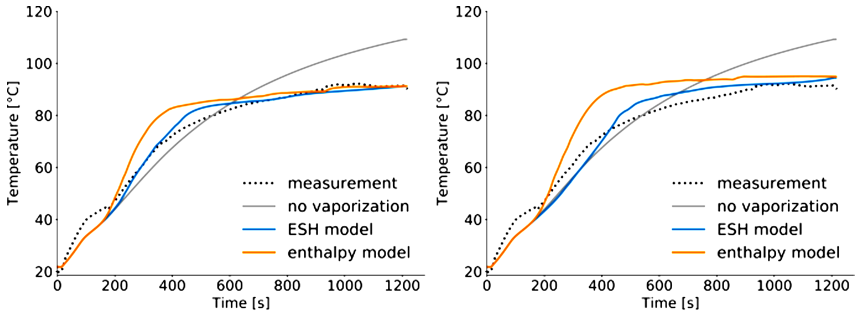


Figure 6: Difference between simulated and measured temperature.

Altogether, the ESH and the enthalpy model both show comparable but slightly different temperature curves. Especially the overestimation of the temperature during the middle of the experiment is usually higher for the enthalpy model. To compensate one could think about adjusting parameters, like the exact amount of water in the liver tissue. However, a first step should be to improve the simple condensation model.

Discussion of the Simple Condensation Model

The consideration behind the simple condensation model described in Sects. 3.2 and 3.4 is solely to preserve the conservation of energy. Therefore, all the water which was vaporized at a certain time is assumed to instantaneously condensate in the condensation region $\Omega_{\text{cond}} = \{x \in \Omega \mid T^- \leq T \leq T^+\}$. This consideration is strictly global and does not involve any form of transport mechanism for the vapor. Hence, it is possible that vapor which was generated in one region can instantaneously condensate in another region. Through this mechanism temperature can be shifted from one region to another without any delay. This effect is possibly the reason for the overestimated temperatures during the middle of the experiment. This can be seen, e.g., for the case P28F70 (cf. Fig. 6), where all simulated temperatures are the same until about 400 s into the experiment. At that point the simulated temperatures rise much faster for the models that include vaporization than for the one without it. We suspect that at this point of the experiment, vaporization occurs in tissue close to the applicator. Due to the instantaneous transport mechanism of the simplified condensation model heat is then added to regions further away from the applicator, where the applicator is placed. This results in the non-physical temperature increase that can be seen in this case. Additionally the simple condensation model is also rather sensitive with respect to the choice of the condensation region as can be seen in Fig. 7, where the temperature at the probe for the case P34F47 is shown for the condensation region given by T^-=60°C and T^+=80°C in Fig. 7(a) as well as for T^-=70°C and T^+=90°C in Fig. 7(b).



(a) $T^- = 60$ °C and $T^+ = 80$ °C. (b) $T^- = 70$ °C and $T^+ = 90$ °C.

Figure 7: Sensitivities with respect to the choice of the condensation region $\Omega_{\text{cond}} = \{x \in \Omega \mid T^- \leq T \leq T^+\}$.

To resolve this issue, the transport of vapor within the tissue must be taken into account. This could for example be done by adding an additional diffusion equation similar to the bio-heat equation to the state system. Therefore, an effective diffusion coefficient for the vapor must be known or estimated from measurements. Alternatively, a more complex solution would be to model the tissue as porous medium and to use a pressure based formulation for the vapor transport.

CONCLUSION

LITT is a minimally-invasive method in the field of interventional oncology used for treating liver cancer. Mathematical modeling and computer simulation are important features for treatment planning and simulating the necrosis of the tissue. The numerical simulation is in good agreement with temperature measurements for ex-vivo porcine liver. In particular, the incorporation of vaporization of water in liver tissue improves the simulation. Still a refinement of the simple and artificial model for the vapor might be necessary. Due to its global nature, this model allows for an undelayed flow of temperature from a hot region to a colder one. This is probably the reason for the overestimated temperatures during the middle of the experiments. An additional physically motivated transport meachanism for the vapor might be necessary. In order to use simulations for the monitoring and assistance during the treatment of humans it is important to model the blood perfusion, because blood vessels have a significant cooling effect. One approach can be to identify the blood perfusion rate from MR thermometry during the beginning of the treatment and use this information to correctly simulate the remaining treatment (cf. [22]).

REFERENCES

1. Müller GJ, Roggan A. Laser-induced interstitial thermotherapy. Bellingham: SPIE; 1995.
2. Fasano A, Hömberg D, Naumov D. On a mathematical model for laser-induced thermotherapy. Appl Math Model. 2010;34(12):3831–40. https://doi.org/10.1016/j.apm.2010.03.023.
3. Mohammed Y, Verhey JF. A finite element method model to simulate laser interstitial thermo therapy in anatomical inhomogeneous regions. Biomed Eng Online. 2005;4(1):2. https://doi.org/10.1186/1475-925X-4-2.
4. Hübner F, Leithäuser C, Bazrafshan B, Siedow N, Vogl TJ. Validation of a mathematical model for laser-induced thermotherapy in liver tissue. Lasers Med Sci. 2017;32(6):1399–409. https://doi.org/10.1007/s10103-017-2260-4.
5. Leithäuser C, Hübner F, Bazrafshan B, Siedow N, Vogl TJ. Experimental validation of a mathematical model for laser-induced thermotherapy. In: European consortium for mathematics in industry. Heidelberg: Springer; 2018.
6. Yang D, Converse MC, Mahvi DM, Webster JG. Expanding the bioheat equation to include tissue internal water evaporation during heating. IEEE Trans Biomed Eng. 2007;54(8):1382–8. https://doi.org/10.1109/TBME.2007.890740.
7. Suárez RB, Campanone L, Garcia M, Zaritzky N. Comparison of the deep frying process in coated and uncoated dough systems. J Food Eng. 2008;84(3):383–93.
8. Carey VP. Liquid-vapor phase-change phenomena: an introduction to the thermophysics of vaporization and condensation processes in heat transfer equipment. Boca Raton: CRC Press; 2020.
9. Pennes HH. Analysis of tissue and arterial blood temperatures in the resting human forearm. J Appl Physiol. 1948;1(2):93–122. https://doi.org/10.1152/jappl.1948.1.2.93.
10. Niemz MH et al.. Laser-tissue interactions. Berlin: Springer; 2007. https://doi.org/10.1007/978-3-030-11917-1.
11. Modest MF. Radiative heat transfer. San Diego: Academic Press; 2003.
12. Demtröder W. Experimentalphysik 1. Berlin: Springer; 2018. https://doi.org/10.1007/978-3-662-54847-9.

13. Yang D, Converse MC, Mahvi DM, Webster JG. Measurement and analysis of tissue temperature during microwave liver ablation. IEEE Trans Biomed Eng. 2007;54(1):150–5. https://doi.org/10.1109/TBME.2006.884647.
14. Geuzaine C, Remacle J-F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. Int J Numer Methods Eng. 2009;79(11):1309–31. https://doi.org/10.1002/nme.2579.
15. Alnæs MS, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, Richardson C, Ring J, Rognes ME, Wells GN. The fenics project version 1.5. Arch Numer Softw. 2015;3(100). https://doi.org/10.11588/ans.2015.100.20553.
16. Logg A, Mardal K-A, Wells GN et al.. Automated solution of differential equations by the finite element method. Heidelberg: Springer; 2012. https://doi.org/10.1007/978-3-642-23099-8.
17. Balay S, Abhyankar S, Adams MF, Brown J, Brune P, Buschelman K, Dalcin L, Dener A, Eijkhout V, Gropp WD, Karpeyev D, Kaushik D, Knepley MG, May DA, McInnes LC, Mills RT, Munson T, Rupp K, Sanan P, Smith BF, Zampini S, Zhang H, Zhang H. PETSc users manual. Technical report ANL-95/11—Revision 3.11. Argonne National Laboratory; 2019. https://www.mcs.anl.gov/petsc.
18. Puccini S, Bär N-K, Bublat M, Kahn T, Busse H. Simulations of thermal tissue coagulation and their value for the planning and monitoring of laser-induced interstitial thermotherapy (LITT). Magn Reson Med. 2003;49(2):351–62. https://doi.org/10.1002/mrm.10357.
19. Roggan A, Dorschel K, Minet O, Wolff D, Muller G. The optical properties of biological tissue in the near infrared wavelength range. In: Laser-induced interstitial therapy. Bellingham: SPIE; 1995. p. 10–44.
20. Giering K, Minet O, Lamprecht I, Müller G. Review of thermal properties of biological tissues. In: Laser-induced interstitial therapy. Bellingham: SPIE; 1995. p. 45–65.
21. Schwarzmaier H-J, Yaroslavsky IV, Yaroslavsky AN, Fiedler V, Ulrich F, Kahn T. Treatment planning for MRI-guided laser-induced interstitial thermotherapy of brain tumors—the role of blood perfusion. J Magn Reson Imaging. 1998;8(1):121–7. https://doi.org/10.1002/jmri.1880080124.
22. Andres M, Blauth S, Leithäuser C, Siedow N. Identification of the blood perfusion rate for laser-induced thermotherapy in the liver. 2019. arXiv:1910.09199.

Chapter

4

A NOVEL MATHEMATICAL MODELING WITH SOLUTION FOR MOVEMENT OF FLUID THROUGH CILIARY CAUSED METACHRONAL WAVES IN A CHANNEL

Wasim Ullah Khan[1], Ali Imran[2] , MuhammadAsif Zahoor Raja[3] , Muhammad Shoaib[2] , Saeed EhsanAwan[4] , Khadija Kausar[2] & Yigang He[1]

[1] School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China

[2] Department of Mathematics, COMSATS University Islamabad, Attock Campus, Kamra Road, Attock, Pakistan

[3] Future Technology Research center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan, ROC

[4] Department of Electrical and Computer Engineering, COMSATS University Islamabad, Attock Campus, Kamra road, Attock, Pakistan

Citation: (APA): Khan, W. U., Imran, A., Raja, M. A. Z., Shoaib, M., Awan, S. E., Kausar, K., & He, Y. (2021). A novel mathematical modeling with solution for movement of fluid through ciliary caused metachronal waves in a channel. Scientific reports, 11(1), 1-12. (12 pages).

ABSTRACT

In the present research, a novel mathematical model for the motion of cilia using non-linear rheological fluid in a symmetric channel is developed. The strength of analytical perturbation technique is employed for the solution of proposed physical process using mectachoronal rhythm based on Cilia induced flow for pseudo plastic nano fluid model by considering the low Reynolds number and long wave length approximation phenomena. The role of ciliary motion for the fluid transport in various animals is explained. Analytical expressions are gathered for stream function, concentration, temperature profiles, axial velocity, and pressure gradient. Whereas, transverse velocity, pressure rise per wave length, and frictional force on the wall of the tubule are investigated with aid of numerical computations and their outcomes are demonstrated graphically. A comprehensive analysis for comparison of Perturb and numerical solution is done. This analysis validates the analytical solution.

INTRODUCTION

Three types of cells movement in human beings and in various animals have been observed, namely: (i) Amoeboid movement (ii) Muscular movement (iii) Ciliary movement. (i) Amoeboid movement: Movement by pseudopodia (pseudo means false and podia means feet). Cells in human body which exhibit amoeboid movement are: Leucocytes (White blood cells), Macrophages (immune system). (ii) Muscular movement: In the human body, this movement is shown by movement of limbs and movement of jaws, tongue, eyelids etc. (iii) Ciliary movement: Movement by numerous hair-like structure. The regions in human body which exhibit this type of movement are: Respiratory tract lined by ciliated epithelium and reproductive system for the movement of fluid. In respiratory tract, cilia are present in trachea which helps to inhale oxygen inside and stop dust and other harmful particles and remove them outside. Cilium is a short microscopic hairlike vibrating structure, found in large numbers on the surface of certain cells, or in some protozoans and other small organisms, providing propulsion. Cilia consist of plasma membrane, peripheral microtubules, central microtubule, radial spoke, and liner and they contain basal body base. They are found in almost all animals, and they provide locomotion to moving fluid along internal epithelial tissue and ciliated protozoans. In some animals, many cilia may fuse together to form cirri. “The cirri are stiff structures and are used as something like legs”. According to Lardner and Shack1, movement of

cilia plays an important role in many physical procedures i.e., reproduction, rotation, inhalation, alimentation and locomotion. The rehological fluid motion due to ciliary caused metachoronal wave is exhibited in Fig. 1.

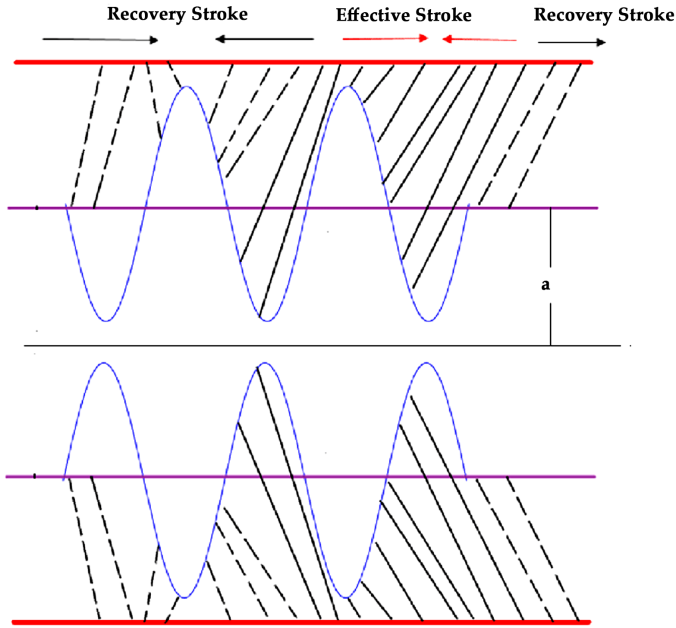


Figure 1: Metachronal wave pattern are exhibited due to ciliary wave motion.

The physiological aspects of ciliary transport has been studied by Lodish et al.2. Akbar et al.3 presented a non-Newtonian physiological fluid motion in a channel consisting of two parallel oscillating walls. Sadaf and Nadeem4 investigated fluid motion produced by cilia and pressure gradient through a curved channel along with heat transfer and radial magnetic field effects. Akram et al.5 examined the combined effects of peristalsis along with electroosmosis induce flow of silver-water nanofluid and silicon dioxide–water nanofluid for a permeable channel. Riaz et al.6 carried out a computational investigation which is applied on the peristaltic propulsion of nanofluid flow for a permeable rectangular duct. The impact of Hall effect on the peristaltic motion of Johnson-Segalman fluid in a heated channel with elastic boundaries has been investigated by Javed7. Bhatti et al.8 focused on transport phenomena of particle-fluid motion through an annular gape region. There are two groups in which cilia are divided, namely motile and non-motile cilia. Non-motile are identified as primary cilia. Single non motile cilium is found in nearly all cells, and plays a role in sensory

functions. We will discuss importance of motile cilia here, which don't beat casually(randomly), but through a synchronized way. The behaviour of cilia holds some vital features of ciliated epithelium. Mucus layer is present on the top of motile cilia. Motile cilia are rarely exists, and they are found in respiratory and reproductive system as well as in brain and spinal cord. Rivera9 narrated several explanations and implications about cilia gill (respiratory organ) for aquatic species, which may be enlisted as follow: (i) The beating rate of all the cilia is fairly unvarying, in any given tissue. (ii) The flagellation of cilium and cilium on the adjusted cells stay greatly synchronized. (iii) Certain movements from one to another place are created, whereas a movement in a given row of cells can be defined as a movement involving a beat arrangement, from one line to another line of cilia and so on.

As it well addressed metachronal rhythm provides concise flow of water with time through the surface of cilia, or probable it is unrealistic to arise synchronous beat over large area, it's thought that cilia do not beat in a synchronous way, but in a systematic way. Nevertheless, along the surface of cilia metachronal rhythm may vary their shape, and this variation depending on whether the metachronal rhythm is accelerated toward the operative lash of the ciliary beat, or cilia beat is perpendicular to the lash of wave movement or may pass in the reverse direction of the of actual lash of beat and then in opposite way of flow. Very limited data is available about metachronal rhythm velocities, frequencies and wave-lengths. A 2-dimensional viscous fluid transport of nanoparticles past a channel along ciliated walls is investigated by Nadeem and Hina10. According to previous observations which revealed through experiments, many biological fluids exhibit non-Newtonian behavior11,12,13,14,15,16,17,18,19,20,21. For the simple Newtonian fluid non-satisfactory outcomes are analyzed. For rheological fluid transport, some of the researchers used Power-Law Model14,15,16,17,22. This model mostly rely on the fluid behavior index n due to its rheological nature.

Lauga and Powers23, Cordero and Lauga24 emphasized on biophysical and mechanical aspect for locomotion of microorganisms. They mathematically explored the importance of shear-dependent viscosities for the locomotion of flagella and cilia induced motion because of metachronal rhythm. Ciliary motion has been studied by the researchers by utilizing two models i.e. (i) Envelop model (ii) Sublayer model. The envelope model

approach has edge over the other model because of metachronal beats on cilia layer by overlooking the particulars of sub-layer dynamic forces. Furthermore, the envelope model can be used for quantitative analysis e.g., for comparing swimming velocities that are mathematically gauged with available data, which are recorded in water for numerous microorganisms24. In addition, the perturbation method can be used for analysis and systematic study of non-Newtonian effects. Recently, the power-law fluid because of ciliary motion has been studied by Siddique et al.25, in the unlimited channel. They showed that power-law fluid provides outcomes, which are nearer to estimated value 6×10^{-3} ml/h. One can find some of recent work27,28,29,30.

A problem (non-linear) of Pseudo plastic fluid transference produced through cilia beating sequence of cilia in a given row of cells from row-to-row and metachronal wave movement is of great importance. On basis of mathematical study, rate of flow, velocity and pressure change will be calculated. Han et al.31 demonstrated that ultracold atomic may initiate a super solid phase while interacted with spin-orbit and a spin-dependent array of potential. Li et al.32 studied multi variant solutions for the polar and ferromagnetic of the universal model of a spinor model of the Bose-Einstein condensate. Wen et al.33 emphasized on matter rogue wave in Bose-Einstein condensates with dynamic inter-atomic contact with the aid of approximate and computation techniques. Transport of Non-Newtonian fluid in the ductus efferentes has been extensively studied by34,35,36.

There are some useful applications of artificial cilia in microfluidics (a) closed-loop channel and (b) open-loop channel with artificial magnetic cilia used in microfluidic pumping and also for flow control in tiny bio-sensors26. Cilia in the following study will not be assumed as flagella but ciliated epithelium. The key aspects of current investigation may me expressed in term salient features as:

- To emphasis on the motion of cilia induce mechanism using non-linear rheological fluidic system past a symmetric channel.
- Viewing the physiology of the problem, a mathematical model is developed using low Reynolds number and long wave length approximation.
- Analytical expressions for stream function, axial velocity, concentration and temperature profiles and pressure gradient are explored, whereas an attempt is made to numerically compute

transverse velocity, pressure rise and friction force.

- Physical impact of crucial flow parameters are examined on the stream function, velocity profile, concentration, temperature profile, pressure gradient, pressure rise, frictional force on the walls the channel.

Further the paper is designed in following systematic manner.

MODELING OF THE RHEOLOGICAL PROBLEM

A two dimensional, incompressible, rheological Pseudo plastic nanofluid in a symmetric channel is analyzed. A cilia induced flow in a channel having infinite length is considered. The inside walls of the system based channel are assumed to be populated with a ciliated carpet. Further, it is assumed that flow is initiated by systematic beating of cilia which creates a metachronal wave, at the right side of the channel. We can identify a reference frame $(\overline{X}, \overline{Y})$, in a manner that $\overline{X}$-axis lies along the center of the channel and $\overline{Y}$ -axis is in the transverse direction. Both the plates are at 2 h apart.

For physiological problem we take velocity profile in the following form

$$\mathbf{V} = \left[\overline{U}\left(\overline{X}, \overline{Y}, \overline{t}\right), \overline{V}\left(\overline{X}, \overline{Y}, \overline{t}\right), 0\right], \tag{1}$$

where $\overline{U}$ and $\overline{V}$ components of fluidic velocity profile in the axial and transverse direction, respectively.

Considering $\overline{S}$ the stress tensor for pseudo plastic fluid model is given,

$$\overline{\mathbf{S}} + \overline{\lambda}_1 . \overline{\mathbf{S}}^{\nabla} + \frac{1}{2}(\overline{\lambda}_1 - \overline{\mu}_1)(\overline{\mathbf{A}}_1 \overline{\mathbf{S}} + \overline{\mathbf{S}}\,\overline{\mathbf{A}}_1) = \mu \overline{\mathbf{A}}_1 \tag{2}$$

$$\overline{\mathbf{S}}^{\nabla} = \frac{d\overline{\mathbf{S}}}{dt} - \overline{\mathbf{S}}\,\overline{\mathbf{L}}^{T} - \overline{\mathbf{L}}\,\overline{S} \tag{3}$$

$$\overline{\mathbf{A}}_1 = \overline{\mathbf{L}} + \overline{\mathbf{L}}^{*}, \tag{4}$$

where

$$\overline{\mathbf{L}} = \nabla \overline{\mathbf{V}}, \tag{5}$$

where μμ represents viscosity of the fluid, $\overline{\mathbf{S}}^{\nabla}$ is upper-convected derivative, $\frac{d}{\overline{dt}}$ A_1 is used for Rivlin-Ericksen tensor of first type, $\frac{d}{\overline{dt}}$ is material derivative and $\overline{\mu}_1$ and $\overline{\lambda}_1$ are the relaxation times. Continuity, momentum, energy and concentration equations may narrated in the vector form as:

$$\nabla.\overline{\mathbf{V}} = 0, \tag{6}$$

$$\rho_f \frac{d\overline{\mathbf{V}}}{dt} = -\nabla\overline{P} + \nabla\overline{\mathbf{S}}, \tag{7}$$

$$\frac{d\overline{\mathbf{T}}}{dt} = \overline{\mathbf{S}}.\overline{\mathbf{L}} + \alpha_1\nabla^2\overline{\mathbf{T}} + \tau\left[D_B\nabla\overline{\mathbf{C}}.\nabla\overline{\mathbf{T}} + \frac{D_T}{T_m}\nabla\overline{\mathbf{T}}.\nabla\overline{\mathbf{T}}\right] \tag{8}$$

$$\frac{d\overline{\mathbf{C}}}{dt} = \frac{D_T}{T_m}\nabla^2\overline{\mathbf{T}} + D_B\nabla^2\overline{\mathbf{C}} \tag{9}$$

where ρ_f is the density of fluid, $\overline{P}$ is the pressure, τ is the ratio of heat capacity of nano particle material to fluid, $\overline{S}$ is the extra stress tensor, D_B is the Brownian diffusion coefficient, D_T is the thermophoretic diffusion coefficient, T_m is the fluid mean temperature and α_1 is the thermal diffusivity.

In order to investigate the problem in a better and simple method, laboratory frame is shifted to wave frame, the transformations from moving frame of wave frame are $(\overline{X}, \overline{Y})$,

$$\bar{u}(\bar{x},\bar{y}) = \bar{U} - c, \bar{v}(\bar{x},\bar{y}) = \bar{V}, \quad \bar{x} = \bar{X} - c\bar{t}, \bar{y} = \bar{Y}, \bar{p}(\bar{x},\bar{y}) = \bar{P}(\bar{X},\bar{Y},\bar{t}).$$

$$\lambda_1 = \frac{\bar{\lambda}_1 c}{d_1}, x = \frac{\bar{x}}{\lambda}, y = \frac{\bar{y}}{a}, \quad \delta = \frac{a}{\lambda}, h = \frac{H}{a}, t = \frac{c\bar{t}}{\lambda}, v = \frac{\bar{v}}{c}, \quad Re = \frac{\rho ca}{\mu},$$

$$\sigma = \frac{\bar{C} - C_0}{\bar{C}_1 - \bar{C}_0}, \quad \Pr = \frac{v}{\alpha}, u = \frac{\bar{u}}{c}, \theta = \frac{\bar{T} - T_0}{T_1 - T_0}, N_t = \frac{\tau D_T(T_1 - T_0)}{\alpha T_m},$$

$$N_b = \frac{\tau D_B(C_1 - C_0)}{\alpha}, \quad S_{ij} = \frac{a\bar{S}_{ij}}{\mu c}, \quad \mu_1 = \frac{\bar{\mu}_1 c}{a}, u = \frac{\partial\psi}{\partial y}, v = -\delta\frac{\partial\psi}{\partial x},$$

where *Re* is the Reynold number, N_t is thermophoresis number, N_b is Brownian motion and *Pr* is the Prandtl number.

In moving wave frame, by capitalizing the non-dimensional parameter along with low Reynolds number and long wavelength approximation the equations of motion38·39, take the form as:

$$\frac{\partial^4\psi}{\partial y^4} - \zeta\frac{\partial^2}{\partial y^2}\left(\frac{\partial^2\psi}{\partial y^2}\right)^3 = 0, \tag{10}$$

$$\frac{\partial p}{\partial x} = \frac{\partial}{\partial y}\left[\psi_{yy}\left\{1 - \zeta\left(\frac{\partial^2\psi}{\partial y^2}\right)^2\right\}\right], \tag{11}$$

$$\frac{\partial^2\theta}{\partial y^2} + PrN_b\frac{\partial\theta}{\partial y}\frac{\partial\sigma}{\partial y} + PrN_t\left(\frac{\partial\theta}{\partial y}\right)^2 + PrEc\left(\frac{\partial^2\psi}{\partial y^2}\right)^2 = 0, \tag{12}$$

$$\frac{N_t}{N_b}\frac{\partial^2\theta}{\partial y^2}+\frac{\partial^2\sigma}{\partial y^2}=0, \tag{13}$$

with the relaxation time $(\lambda_1^2-\mu_1^2)=\zeta$ and the corresponding dimensionless boundary conditions[40,41] for cilia induced rheological fluid model are

$$\frac{\partial^2\psi}{\partial y^2}=0,\psi=0,\theta=0,\sigma=0, \tag{14}$$

at y=0

$$\frac{\partial\psi}{\partial y}=u_h=-1-2\alpha\delta\epsilon cos(2\pi x),\psi=\frac{F}{2},\theta=1,\sigma=1, \tag{15}$$

at h=1+cos(2πx).

SOLUTION METHODOLOGY

It is hard to get the exact solution of the Eqs. (10–13), so we will employ perturbation method for small parameter ζ

$$\psi=\psi_0+\zeta\psi_1+\zeta^2\psi_2^2+... \tag{16}$$

$$\sigma=\sigma_0+\zeta\sigma_1+\zeta^2\sigma_2^2+... \tag{17}$$

$$\theta=\theta_0+\zeta\theta_1+\zeta^2\theta_2^2+... \tag{18}$$

Expressions using perturb solution up to second order for stream function, concentration, and temperature are

$$\psi=\frac{3Fh^2y-2h^3u_hy-Fy^3+2hu_hy^3}{4h^3}+\zeta\frac{27(-F+hu_h)^3y(h^2-y^2)^2}{20h^9}+\zeta^2\frac{1}{385000h^{27}}19683(-F+hu_h)^9y(h^2-y^2)^2(767h^6+1150h^4y^2-2625h^2y^4+3500y^6), \tag{19}$$

$$\sigma=\gamma_1-\frac{\gamma_4N_t}{N_b}+\gamma_2y+\frac{N_t\left(4\gamma_2{}^2\gamma_3e^{-\gamma_2N_b\Pr y}N_b{}^2\Pr^2+\frac{3Ec(F-2hu_h)^2y(6+\gamma_2N_b\Pr y(-3+\gamma_2N_b\Pr y))}{h^6}\right)}{4\gamma_2{}^3N_b{}^4\Pr^3}+\zeta\left(\delta_1+\delta_2y-\frac{1}{175\delta_2{}^7h^{18}N_b{}^8\Pr^7}N_t\left(-175\delta_2{}^6\delta_3e^{-\delta_2N_b\Pr y}h^{18}N_b{}^6\overset{6}{\Pr}+175\delta_2{}^7\delta_4h^{18}N_b{}^7\overset{7}{\Pr}-91854Ec\left(1000-40\delta_2{}^2h^2N_b{}^2\overset{2}{\Pr}+\delta_2{}^4h^4N_b{}^4\overset{4}{\Pr}\right)(F-hu_h)^6y+45927\delta_2EcN_b\Pr\left(1000-40\delta_2{}^2h^2N_b{}^2\overset{2}{\Pr}+\delta_2{}^4h^4N_b{}^4\overset{4}{\Pr}\right)(F-hu_h)^6y^2-15309\delta_2{}^2EcN_b{}^2\overset{2}{\Pr}\left(1000-40\delta_2{}^2h^2N_b{}^2\overset{2}{\Pr}+\delta_2{}^4h^4N_b{}^4\overset{4}{\Pr}\right)(F-hu_h)^6y^3-153090\delta_2{}^3EcN_b{}^3\overset{3}{\Pr}\left(-25+\delta_2{}^2h^2N_b{}^2\overset{2}{\Pr}\right)(F-hu_h)^6y^4+30618\delta_2{}^4EcN_b{}^4\overset{4}{\Pr}\left(-25+\delta_2{}^2h^2N_b{}^2\overset{2}{\Pr}\right)(F-hu_h)^6y^5+127575\delta_2{}^5EcN_b{}^5\overset{5}{\Pr}(F-hu_h)^6y^6-18225\delta_2{}^6EcN_b{}^6\overset{6}{\Pr}(F-hu_h)^6y^7\right)\right) \tag{20}$$

$$\theta = \gamma_4 - \frac{4\gamma_2{}^2\gamma_3 e^{-\gamma_2 N_b \Pr y} N_b{}^2 \Pr^2 + \frac{3Ec(F-2hu_h)^2 y(6-3\gamma_2 N_b \Pr y+\gamma_2{}^2 N_b{}^2 \Pr^2 y^2)}{h^6}}{4\gamma_2{}^3 N_b{}^3 \Pr^3}$$
$$\zeta\left(\frac{1}{175\delta_2{}^7 h^{18} N_b{}^7 \Pr^7}\left(-175\delta_2{}^6\delta_3 e^{-\delta_2 N_b \Pr y} h^{18} N_b{}^6 \overset{6}{\Pr} + 175\delta_2{}^7\delta_4 h^{18} N_b{}^7 \overset{7}{\Pr} - 91854Ec\right.\right.$$
$$\left(1000 - 40\delta_2{}^2 h^2 N_b{}^2 \overset{2}{\Pr} + \delta_2{}^4 h^4 N_b{}^4 \overset{4}{\Pr}\right)(F-hu_h)^6 y + 45927\delta_2 Ec N_b \Pr$$
$$\left(1000 - 40\delta_2{}^2 h^2 N_b{}^2 \overset{2}{\Pr} + \delta_2{}^4 h^4 N_b{}^4 \overset{4}{\Pr}\right)(F-hu_h)^6 y^2 - 15309\delta_2{}^2 Ec N_b{}^2 \overset{2}{\Pr}$$
$$\left(1000 - 40\delta_2{}^2 h^2 N_b{}^2 \overset{2}{\Pr} + \delta_2{}^4 h^4 N_b{}^4 \overset{4}{\Pr}\right)(F-hu_h)^6 y^3 - 153090\delta_2{}^3 Ec N_b{}^3 \overset{3}{\Pr}$$
$$\left(-25 + \delta_2{}^2 h^2 N_b{}^2 \overset{2}{\Pr}\right)(F-hu_h)^6 y^4 + 30618\delta_2{}^4 Ec N_b{}^4 \overset{4}{\Pr}\left(-25 + \delta_2{}^2 h^2 N_b{}^2 \overset{2}{\Pr}\right)$$
$$\left.\left.(F-hu_h)^6 y^5 + 127575\delta_2{}^5 Ec N_b{}^5 \overset{5}{\Pr}(F-hu_h)^6 y^6 - 18225\delta_2{}^6 Ec N_b{}^6 \overset{6}{\Pr}(F-hu_h)^6 y^7\right)\right) \tag{21}$$

In Eqs. (20 and 21) $\gamma_1,...,\gamma_4$ and $\delta_2,...,\delta_4$ are variable expressions, and their values are incorporated in the "Appendix".

VELOCITY PROFILE

Using the relation $u = \frac{\partial \psi}{\partial y}$ one may obtain expression for axial component of velocity from Eq. (19) as:

$$u = -\frac{u_h}{2} + \frac{3\left(2hu_h y^2 + F(h-y)(h+y)\right)}{4h^3} + \frac{27(-F+hu_h)^3\left(h^4 - 6h^2y^2 + 5y^4\right)\zeta}{20h^9}$$
$$+ \frac{19683(-F+hu_h)^9(h-y)(h+y)\left(767h^8 - 385h^6y^2 - 21175h^4y^4 + 48125h^2y^6 - 38500y^8\right)\zeta^2}{385000h^{27}} \tag{22}$$

Pressure gradient is gathered as

$$\frac{dp}{dx} = -\frac{3(F_0 - 2hu_h)}{2h^3} - \zeta\left(\frac{3\left(-27F_0{}^3 + 20F_1h^4 + 162F_0{}^2hu_h - 324F_0h^2u_h{}^2 + 216h^3u_h{}^3\right)}{40h^7}\right) \tag{23}$$

One may obtain the expression by integration the continuity equation

$$v = -\int \frac{\partial u}{\partial x} dy + c, \tag{24}$$

The pressure rise per Wavelength is explored as

$$\Delta p_\lambda = \int_0^1 \frac{dp}{dx} dx. \tag{25}$$

F_λ is the frictional force which can be obtained as

$$F_\lambda = \int_0^1 h\left(-\frac{dp}{dx}\right) dx. \tag{26}$$

It is hard to get the analytical expression for velocity component in the transverse direction, pressure rise, and frictional force. So, they are computed numerically and their results are exhibited graphically.

ANALYSIS OF THE PHYSICAL PROBLEM

Cilia has numerous applications, it has been investigated by various researchers that cilia are responsible for fluid locomotion in ductus efferentes. Ductus efferentes are various small tubes which establishes important relation between testis and epididymis. Composition of these tubes is that these tubes are consists of single layer epithelium, this structure is strengthened by layer of uniform muscle and adjoining tissue[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17]. These tubes transport sperm via rate testis to epididymis and recollect large quantity of fluid arising from rete testis. Ductus efferentes epithelium consists of both ciliated non ciliated cells. Besides this ciliary activity has great significance in the transport of protozoa in which locomotion is done via cilia. Outcomes of current investigation may be significant cilia dependent actuator in the function of biosensors and in drug delivery systems. It is important to mention here that not too much information is available about rate to due ciliary caused flo ws[1,2,3,4,5,6,7,8,9,10,11,12,13,14, 15,16,17,18,19,20, 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37]. For the purpose of quantitative investigation, we provide estimate of different physical quantities related to physical study of fluid rheology of cilia induce flows. We have used following data to study rheological fluid motion. ε=0.1 to 0.2, α=0.2 to 1, δ=0.1 to 0.1, Q=0.1 to 0.5.

Cilia induced flow for pseudo plastic nano fluid model is investigated. Flow is modelled by considering the long wave length theory and low Reynolds number. Solution for the proposed physical phenomenon is obtained by capitalizing the strength of perturbation technique. Analytical expressions are gathered for stream function, concentration, temperature profiles, axial velocity, and pressure gradient. Whereas, transverse velocity, wave length for pressure rise, and frictional force on the walls of the tubule are investigated with aid of numerical computations and their outcome are demonstrated graphically. Here in this section impacts of ζ relaxation time, thermophoresis parameter N_t, Prandtl number *Pr* on velocity distribution, concentration, temperature profiles, pressure gradient, wave length for pressure rise and frictional force are investigated. A comprehensive investigation in the form of numerical data has been exhibited in the Tables 1 and 2. In the first table analysis of perturb and numerical solution for axial velocity *u* is made, almost similar values of perturb and numerical solution with very small difference is recorded. In the Table 2 comparison of perturb and numerical solution for stream function is done and almost identical numerical data is obtained. Both the tables prove the authenticity of our

analytical solution. Graph of axial component of velocity, for perturb and numerical solution is exhibited in Fig. 2, it is worth to mention here that we have obtained all most similar curves for both solution, which validates our analytical solution. Trapping phenomenon is exhibited in Figs. 3 and 4, it is quite evident that as relaxation time is enhanced, the number of circulating streamlines increases and some fluctuations occurs in the size of the trapped bolus. Variations on the velocity profile with enhancement in the relaxation time demonstrated in Fig. 5, it is seen that initially longitudinal component of velocity depreciates as relaxation time ζ is increased and surge is observed in the velocity because of the no slip phenomenon. Where as, mere a sharp decline is seen in the transverse component of velocity with rising values of ζ. Actually relaxation time ζ is the measure of fluid inertia, because of this factor retardation in the velocity profile is recorded.

Figure 6 portrays the impact of relaxation time on concentration and temperature profiles, it is concluded from the plots that concentration falls as enhancement in the value of the ζ is made, while opposite behavior is observed for the temperature distribution. Figure 7 elucidate that the as thermophoresis parameter N_t is enhanced, yields decrease in concentration and increase in temperature profile. Which happen due to fact that thermophoresis mechanism give rise to the motion of fluid elements, they collides with each other due to which energy of fluid element increases, which results increase in temperature and decline in the concentration profile.

Effect of Prandtl number is investigated in Fig. 8, it is concluded from the first figure that concentration profile declines as Prandtle number is enhanced, which means that momentum diffusivity become weak and thermal diffusivity has dominant role. From Fig. 8b it noted that temperature profile become strong as Prandtle number is enhanced.

Plots of pressure gradient as function of relaxation time ζ and wave number δ are portrayed in Fig. 9, it is seen from the first figure that as the measure of fluid is raised, the pressure gradient profile enhances, on other when the wave number δ is extended, then the reverse behavior is seen. Figure 10 demonstrates the impact of ζ and δ on pressure rise based on wave length ΔP_λ, remarkable rise is seen in the value of ΔP_λ with the increase in relaxation time while opposite trend is recorded for enhancing the wave number δ. Impact of ζ and wall contraction/length ε on friction force F_λ over the wall is exhibited in Fig. 11, it is observed that friction force F_λ significantly declines with the enhancement in measure of fluid inertia, and quit opposite behavior is noted for increasing wall contraction F_λ.

Table 1: Shows analysis of Perturb solution with numerical solution with ζ=0.01,α=0.2, ε=0.15, δ=0.01, F=0.1, x=1

y	Perturb solution for *u*	Numerical solution for *u*	Difference
0.	0.564385	0.564638	-2.53724×10^{-4}
0.1	0.552597	0.552843	-2.4605×10^{-4}
0.2	0.517232	0.517455	-2.23174×10^{-4}
0.3	0.458275	0.458461	-1.8578×10^{-4}
0.4	0.375705	0.375841	-1.35904×10^{-4}
0.5	0.269491	0.269569	-7.76974×10^{-5}
0.6	0.13959	0.139607	-1.75488×10^{-5}
0.7	− 0.0140506	− 0.0140875	3.68592×10^{-5}
0.8	− 0.191489	− 0.191567	7.83305×10^{-5}
0.9	− 0.392793	− 0.392894	1.01432×10^{-4}
1.	− 0.618037	− 0.618137	1.00026×10^{-4}
1.1	− 0.867319	− 0.867374	5.51864×10^{-5}

Table 2: Shows analysis of Perturb solution with numerical solution for stream function with α=0.2,ε=0.15,δ=0.01,F=0.1,x=1

y	Perturb solution	Numerical solution	Difference
0.	0.	0.	0.
0.1	0.0560456	0.0560623	-1.67013×10^{-5}
0.2	0.109734	0.109766	-3.20222×10^{-5}
0.3	0.158706	0.15875	-4.45712×10^{-5}
0.4	0.200601	0.200655	-5.31095×10^{-5}
0.5	0.233058	0.233115	-5.67039×10^{-5}
0.6	0.25371	0.253765	-5.49514×10^{-5}
0.7	0.260185	0.260233	-4.81658×10^{-5}
0.8	0.250107	0.250144	-3.74213×10^{-5}
0.9	0.221092	0.221116	-2.44263×10^{-5}
1.	0.17075	0.170762	-1.14196×10^{-5}
1.1	0.0966832	0.0966849	-1.73212×10^{-6}

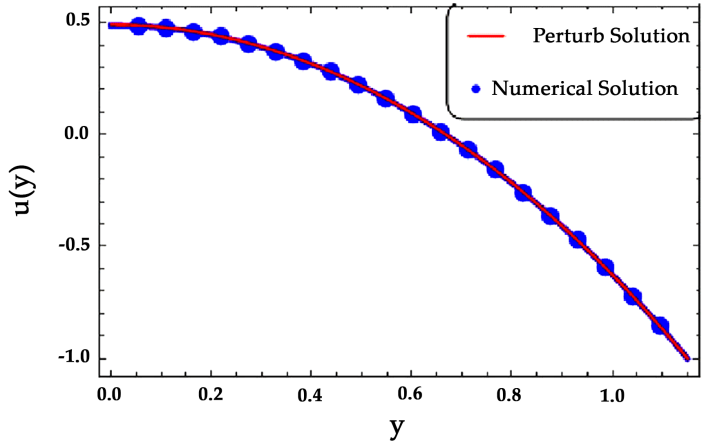


Figure 2: Plot of Perturb with numerical solution with ζ=0.01,α=0.2, ε=0.15, δ=0.01, F=0, x=1.

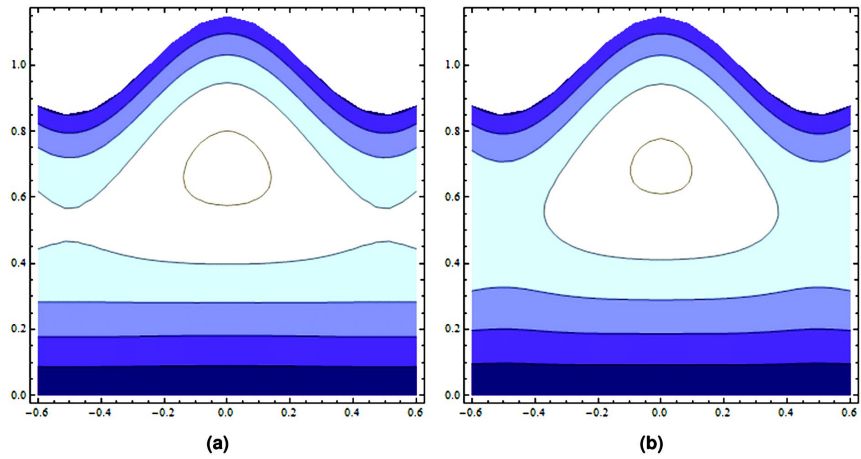


Figure 3: Plots of several stream lines with α=0.2, ε=0.15, F=0.1, δ=0.05, (a) ζ=0, (b) ζ=0.01.

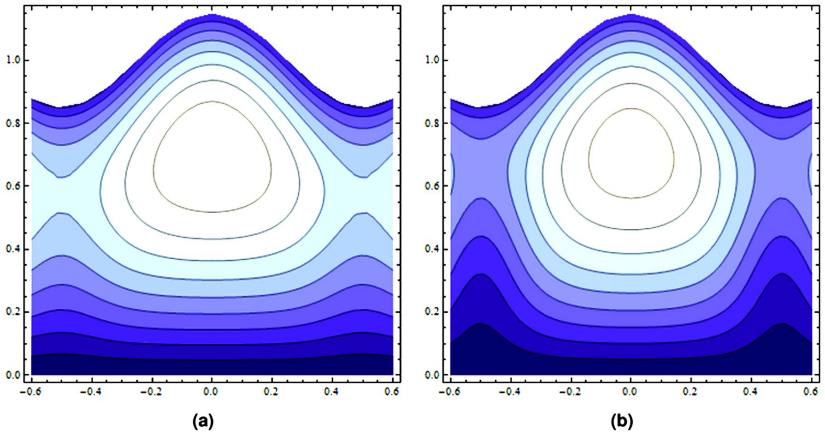


Figure 4: Plots of several stream lines with α=0.2, ε=0.15, F=0.1, δ=0.05, (a) ζ=0.02, (b) ζ=0.03.

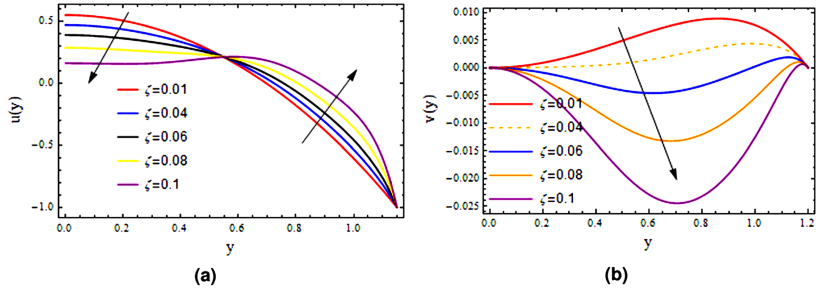


Figure 5: Velocity profile variations with (a) and (b) α=0.2, ε=0.15, δ=0.01, F=0.1, x=1.

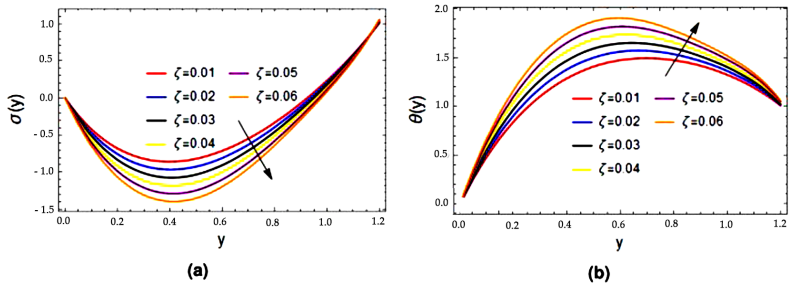


Figure 6: Plots of (a) concentration σ and (b) temperature θ for variations of relaxation time, with α=1, ε=0.2, δ=0.1, F=0.5, x=1, Pr=2, N_b=0.8, N_t=1, E_c=1.

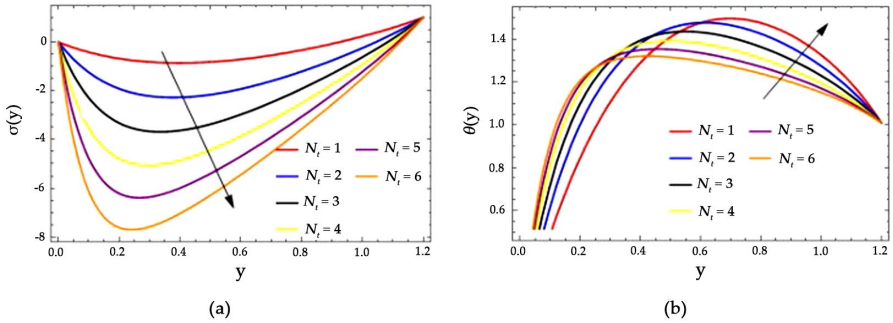


Figure 7: Plots of (a) concentration σσ and (b) temperature θθ for different values of thermophoresis parameter with α=1, ε=0.2, δ=0.1, F=0.5, x=1, Pr=2, N_b=0.8, ζ=0.01, E_c=1.

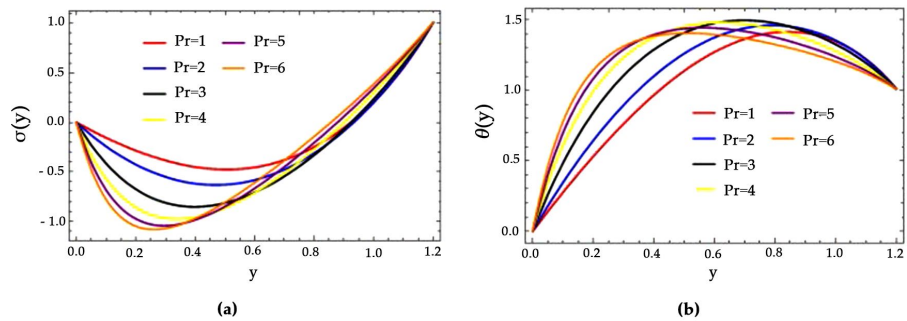


Figure 8: Plots of (a) concentration σσ and (b) temperature θ for Prandtle number with α=1, ε=0.2, δ=0.1, F=0.5, x=1, E_c=1, N_t=1, ζ=0.01, N_b=0.8.

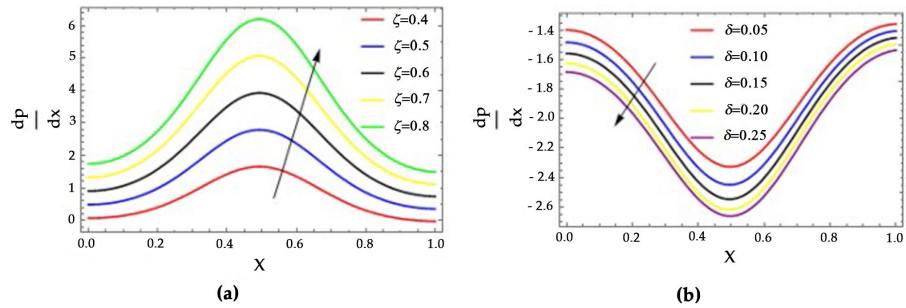


Figure 9: Plots of pressure gradient, impacts of ζ and δ are studied with α=1,ε=0.2,Q=0.5 (a) δ=0.05, (b) ζ=0.05.

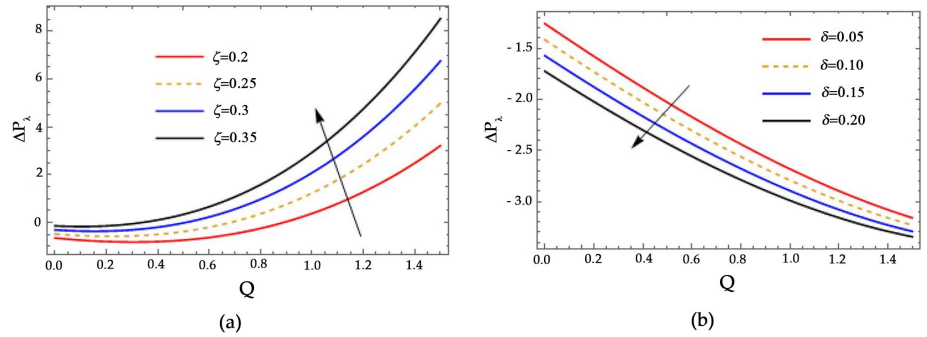


Figure 10: Plots of pressure rise per wave length, variations of ζ and δ are exhibited with α=1,ε=0.2,Q=0.5 (a) δ=0.05, (b) ζ=0.02.

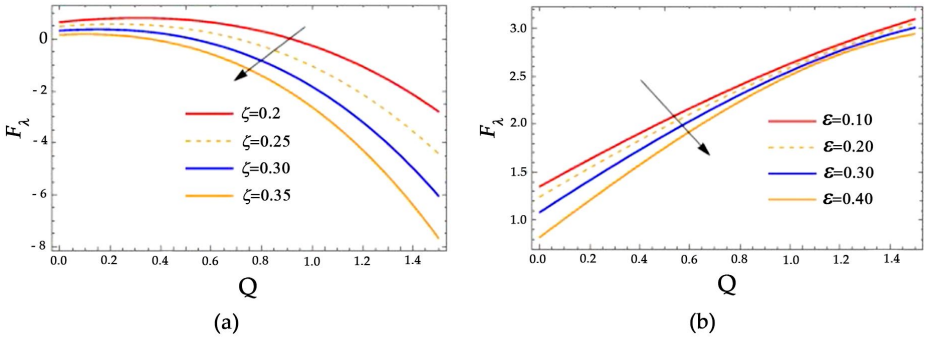


Figure 11: Plots of frictional force on wall of tubules, variations of ζζ and εε are exhibited with α=1,δ=0.2,Q=0.5 (a) ε=0.2, (b) ζ=0.02.

CONCLUSION

In this investigation an effort is made to explore the Cilia induced flow for pseudo plastic nano fluid model which is applicable to ductus efferent of human male reproductive tract. For physiological problem, flow is modeled by employing low Reynolds number and long wave length approximation. A novel solution for the proposed physical phenomenon is obtained by capitalizing the strength of perturbation technique. Analytical expressions are gathered for stream function, concentration, temperature profiles, axial velocity, and pressure gradient. Whereas, transverse velocity, pressure rise per wave length, and frictional force on the wall of the tubule are investigated with aid of numerical computations. Key finding of the current investigation may be elaborated as:

- Circulating stream lines are remarkably increased with the enhancement in the value of fluid inertia ζ.
- Velocity profile deteriorates with increasing relaxation time.
- It is studied that as value of relaxation is enhanced, the concentration profile declines and temperature profile become strong.
- Concentration profile deteriorates with thermophoresis parameter N_t and Brownian motion parameter N_b, whereas temperature profile significantly enhances.
- Pressure rise per wave length ΔP_λ enhances appreciatively with relaxation time and decline with wave number δ.
- Frictional force on the wall of the channel decreases with increasing relaxation time and contraction/length ε.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 51977153, 51977161, 51577046.

REFERENCES

1. Lardner, T. J. & Shack, W. J. Cilia transport. *Bull. Math. Biophys.* 34(3), 325–335 (1972).
2. Lodish, H., Berk, A., Zipursky, L. S., Matsudaira, P., Baltimore, D. & Darnell, J. Cilia and flagella: Structure and movement. Mol. Cell Biol. (2000).
3. Akbar, N. S., Tripathi, D., Khan, Z. H. & Beg, O. A. Mathematical modeling of pressure-driven micropolar biological flow due to metachronal wave propulsion of beating cilia, Prof. E. O. Voit (Wallace H. Coulter Dept. of Biomedical, Engineering, Georgia Tech) (2018).
4. Sadaf, H. & Nadeem, S. Fluid flow analysis of cilia beating in a curved channel in the presence of magnetic field and heat transfer. *Can. J. Phys.* 98(2), 191–197 (2020).
5. Akram, J., Akbar, N. S. & Maraj, E. N. A comparative study on the role of nanoparticle dispersion in electroosmosis regulated peristaltic flow of water. *Alex. Eng. J.* 59(2), 943–956 (2020).
6. Riaz, A., Zeeshan, A., Bhatti, M. M. & Ellahi, R. Peristaltic propulsion of Jeffrey nano-liquid and heat transfer through a symmetrical duct with moving walls in a porous medium. *Physica A* 545, 123788 (2020).
7. Javed M. A mathematical framework for peristaltic mechanism of non-Newtonian fluid in an elastic heated channel with Hall effect. Multidiscip. Model. Mater. Struct. (2020).
8. Bhatti, M. M., Elelamy, A. F., Sait, S. & Ellahi, R. Hydrodynamics interactions of metachronal waves on particulate-liquid motion through a ciliated annulus: Application of bio-engineering in blood clotting and endoscopy. *Symmetry* 12(4), 532 (2020).
9. Rivera, J. A. *Cilia, Ciliated Epithelium and Ciliary Activity.* (Pergamon Press, 1962).
10. Nadeem, S. & Sadaf, H. Trapping study of nanofluids in an annulus with cilia. *AIP Adv.* 5(12), 127204 (2015).
11. Malek, J., Necas, J. & Rajagopal, K. R. Global existence of solutions for fluids with pressure and shear dependent viscosities. *Appl. Math. Lett.* 15, 961–967 (2002).
12. Mendeluk, G., Flecha, F. L. G., Castello, P. R. & Bregni, C. Factors involved in the biochemical etiology of human seminal plasma hyperviscosity. *J. Androl.* 21, 262–267 (2000).

13. Xue, H. The modified Casson's equation and its application to pipe flows of shear thickening fluid. *Acta Mech. Sin.* 21, 243–248 (2005).
14. Misra, J. C. & Maiti, S. Peristaltic transport of rheological fluid: Model for movement of food bolus through esophagus. *Appl. Math. Mech.* 33, 15–32 (2012).
15. Misra, J. C. & Maiti, S. Peristaltic pumping of blood through small vessels of varying cross-section. *J. Appl. Mech. Trans. ASME* 22, 061003 (2012).
16. Misra, J. C. & Pandey, S. K. Peristaltic flow of a multi-layered power-law fluid through a cylindrical tube. *Int. J. Eng. Sci.* 39, 387–402 (2001).
17. Maiti, S. & Misra, J. C. Peristaltic transport of a couple stress fluid: Some applications to hemodynamics. *J. Mech. Med. Biol.* 12, 1250048 (2012).
18. Liu, Y. & Boling, G. Coupling model for unsteady MHD flow of generalized Maxwell fluid with radiation thermal transform. *Appl. Math. Mech.* 37, 137–150 (2016).
19. Hayat, T., Asad, S. & Alsaedi, A. Flow of Casson fluid with nanoparticles. *Appl. Math. Mech.* 37, 459–470 (2016).
20. Siddiqui, A. M., Ashraf, H., Walait, A. & Haroon, T. On study of horizontal thin lm flow of Sisko fluid due to surface tension gradient. *Appl. Math. Mech.* 36, 847–862 (2015).
21. Ding, Z., Jian, Y. & Yang, L. Time periodic electroosmotic flow of micropolar fluids through microparallel channel. *Appl. Math. Mech.* 36, 769–786 (2016).
22. Rao, A. R. & Mishra, M. Peristaltic transport of a power-law fluid in a porous tube. *J. Non-Newtonian Fluid Mech.* 121, 163–174 (2004).
23. Lauga, E. & Powers, T. R. The hydrodynamics of swimming microorganisms. *Rep. Progr. Phys.* 72, 096601 (2009).
24. Vlez-Cordero, J. R. & Lauga, E. Waving transport and propulsion in a generalized Newtonian fluid. *J. Non-Newtonian Fluid Mech.* 199, 37–50 (2013).
25. Siddiqui, A. M., Haroon, T., Rani, R. & Ansari, A. R. An analysis of the flow of a power law fluid due to ciliary motion in an infinite channel. *J. Biorheol.* 24, 56–69 (2010).

26. Maiti, S. & Pandey, S. K. Rheological fluid motion in tube by metachronal waves of cilia. *Appl. Math. Mech.* 38(3), 393–410 (2017).
27. Waini, I., Ishak, A. & Pop, I. Hybrid nanofluid flow towards a stagnation point on a stretching/shrinking cylinder. *Sci. Rep.* 10(1), 1–12 (2020).
28. Gsell, S., Loiseau, E., D'ortona, U., Viallat, A. & Favier, J. Hydrodynamic model of directional ciliary-beat organization in human airways. *Sci. Rep.* 10(1), 1–12 (2020).
29. Pacherres, C. O., Ahmerkamp, S., Schmidt-Grieb, G. M., Holtappels, M. & Richter, C. Ciliary vortex flows and oxygen dynamics in the coral boundary layer. *Sci. Rep.* 10(1), 1–10 (2020).
30. Shah, Z., Kumam, P. & Deebani, W. Radiative MHD Casson nanofluid flow with activation energy and chemical reaction over past nonlinearly stretching surface through entropy generation. *Sci. Rep.* 10(1), 1–14 (2020).
31. Han, W., Juzeliũnas, G., Zhang, W. & Liu, W. M. Supersolid with nontrivial topological spin textures in spin-orbit-coupled Bose gases. *Phys. Rev. A* 91(1), 013607 (2015).
32. Li, L., Li, Z., Malomed, B. A., Mihalache, D. & Liu, W. M. Exact soliton solutions and nonlinear modulation instability in spinor Bose-Einstein condensates. *Phys. Rev. A* 72(3), 033611 (2005).
33. Wen, L. *et al.* Matter rogue wave in Bose–Einstein condensates with attractive atomic interaction. *Eur. Phys. J. D* 64(2), 473–478 (2011).
34. Srivastava, L. M. & Srivastava, V. P. Peristaltic transport of a power-law fluid: Application to the ductus efferentes of the reproductive tract. *Rheol. Acta* 27, 428–433 (1988).
35. Usha, S. & Rao, A. R. Peristaltic transport of two-layered power-law fluids. *J. Biomech. Eng.* 119, 483–488 (1997).
36. Maiti, S. & Misra, J. C. Non-Newtonian characteristics of peristaltic flow of blood in micro-vessels. *Commun. Nonlinear Sci. Numer. Simul.* 18, 1970–1988 (2013).
37. Blake, J. R. On the movement of mucus in the lungs. *J. Biomech.* 8, 179–190 (1975).
38. Noreen, S. Peristaltically assisted nanofluid transport in an asymmetric channel. *Karbala Int. J. Mod. Sci.* 4(1), 35–49 (2018).
39. Mustafa, M., Hina, S., Hayat, T. & Alsaedi, A. Influence of wall properties on the peristaltic flow of a nanofluid: Analytic and numerical solutions. *Int. J. Heat Mass Transf.* 55(17–18), 4871–4877 (2012).

40. Imran, A., Akhtar, R., Zhiyu, Z., Shoaib, M. & Raja, M. A. Z. Analysis of MHD and heat transfer effects with variable viscosity through ductus efferentes. *AIP Adv.* 9(8), 085320 (2019).
41. Imran, A., Akhtar, R., Zhiyu, Z., Shoaib, M. & Raja, M. A. Z. Heat transfer analysis of biological nanofluid flow through ductus efferentes. *AIP Adv.* 10(3), 035029 (2020).

Chapter

5

A NEW MATHEMATICAL MODELING APPROACH FOR THERMAL EXPLORATION EFFICIENCY UNDER DIFFERENT GEOTHERMAL WELL LAYOUT CONDITIONS

JunyiGao[1,2] & Qipeng Shi[2,3]

[1] School of Architecture and Civil Engineering, Yan'an University, Yan'an 716000, China
[2] Shandong Provincial Lunan Geology and Exploration Institute, Jining 272100, China
[3] Shandong Geothermal Clean Energy Exploration and Development Engineering Research Center, Jining 272100, China

ABSTRACT

The water temperature at the outlet of the production well is an important index for evaluating efficient geothermal exploration. The arrangement mode of injection wells and production wells directly affects the temperature

Citation: (APA): Gao, J., & Shi, Q. (2021). A new mathematical modeling approach for thermal exploration efficiency under different geothermal well layout conditions. Scientific Reports, 11(1), 1-14. (14 pages).

distribution of the production wells. However, there is little information about the effect of different injection and production wells on the temperature field of production wells and rock mass, so it is critical to solve this problem. To study the influence mechanism of geothermal well arrangement mode on thermal exploration efficiency, the conceptual model of four geothermal wells is constructed by using discrete element software, and the influence law of different arrangement modes of four geothermal wells on rock mass temperature distribution is calculated and analyzed. The results indicated that the maximum water temperature at the outlet of the production well was 84.0 °C due to the thermal superposition effect of the rock mass between the adjacent injection wells and between the adjacent production wells. Inversely, the minimum water temperature at the outlet of the production well was 50.4 °C, which was determined by the convection heat transfer between the water flow and the rock between the interval injection wells and the interval production wells. When the position of the model injection well and production well was adjusted, the isothermal number line of rock mass was almost the same in value, but the direction of water flow and heat transfer was opposite. The study presented a novel mathematical modeling approach for calculating thermal exploration efficiency under various geothermal well layout conditions.

INTRODUCTION

In the process of geothermal exploration, if the limited groundwater resources around the geothermal well cannot replenish pumping capacity through runoff, it is then necessary to consider the injection well. This is replenish production well-pumping capacity in time to achieve the dynamic balance between pumping capacity and injection capacity, allowing for long-term geothermal exploration. Underground hot water can be used for heating and generating power after being pumped to the ground. The geothermal water extraction system is affected not only by the groundwater flow field and temperature field but also by the layout of geothermal wells and many other factors. Under the combined effect of these factors, how injection wells and production wells are scientifically and reasonably arranged has a significant impact on the temperature field of the rock mass near the production wells and well groups. Therefore it is of great engineering significance to study the wellbore temperature field in the exploration and development of geothermal resources[1,2,3].

At present, research on geothermal well temperatures primarily focuses on numerical simulation analysis. Many scholars have researched the influencing factors of fluid, rock temperature field and wellbore temperature4, the influence of groundwater flow velocity in sandy aquifer on the thermal performance of borehole heat exchanger5, three-dimensional thermoporoelastic modeling and analysis of flow, heat transport and deformation in fractured rock with applications to a lab-scale geothermal system6 and numerical simulation analysis on the influence of different factors on the thermal distribution around wellbore7. Groundwater flow estimation for temperature monitoring in borehole heat exchangers during thermal response tests8, heat extraction analysis of a novel multilateral-well coaxial closed-loop geothermal system9 and research on the influence of borehole heat-water exchanger characteristics on the performance of vertical closed-loop ground heat pump systems were carried out10. Gao11,12 studied the influence mechanism of geothermal well spacing, geothermal temperature and production well depth on the water flow and heat transfer temperature of rock masses, but the literature did not consider the influence of the interaction of injection wells and production wells on the temperature field of production wells and rock masses. Research on outlet temperature and temperature field of geothermal well13,14,15,16, sensitivity analysis of influencing factors for heat loss of geothermal wells17 and wellbore temperature loss model and application for heating geothermal mining18. However, the research contents of these scholars did not involve the comparative study of the water temperature and temperature field at the outlet of geothermal wells under different conditions of the water inlet and water outlet. Scholars have carried out researches on the influence of pumping and irrigation well layout on the groundwater flow field and temperature field19, the influence of pumping and irrigation well distribution mode, and pumping and irrigation well water quantity on heat transfer characteristics of underground heat exchanger well20,21, and the application of numerical simulation of water and heat transport to optimize pumping and irrigation well the layout of groundwater source heat pump system22, numerical simulation of water-heat coupling of single well ground water source heat pump in T2Well23 and optimization of reasonable well spacing and layout of shallow source heat pump simulated by sand tank-taking Jiuxi in Fenglin as an example24. Sustainable electricity generation from an enhanced geothermal system were carried out considering reservoir heterogeneity and water losses with a discrete fracture model25 and enhanced geothermal systems (EGS): hydraulic fracturing in a thermoporoelastic framework26 and

modified zipper fracturing in an enhanced geothermal system reservoir and heat extraction optimization via orthogonal design27. Again, Xu et al.28 Studied on optimal arrangement of pumping and irrigation systems for a groundwater heat pump. Deng et al.29 conducted a simulation study on the optimization of middle-deep geothermal recharge wells based on optimal recharge efficiency. Olabi et al.30 thought that geothermal-based hybrid energy systems are an energy method towards eco-friendliness. Rezaei et al.31 researched an enviro-economic optimization of a hybrid energy system from biomass and geothermal resources for low-enthalpy areas. The system off-design evaluation of geothermal-solar hybrid power and operational strategies for its heat pump system was studied32,33. Tian et al.34 studied Carbon–neutral hybrid energy systems with deep water source cooling, biomass heating, and geothermal heat and power. Chen et al.35 carried out Thermodynamic performance analysis and multi-criteria optimization of a hybrid combined heat and power system coupled with geothermal energy. In summary, although some achievements have been made in the study of geothermal well temperature, there are few reports on the complex model of thermal recovery efficiency optimization under different geothermal well layout conditions. The actual geothermal mining process is closely related to the scientific and reasonable layout of geothermal wells. The influence of different geothermal well layout conditions on the temperature field of production wells and rock masses is directly related to the safety and efficiency of geothermal mining. Given this, it is necessary to research the optimization of thermal mining efficiency under different geothermal well layout conditions.

In this paper, first, the fractured rock mass models of four injection wells and production wells are constructed by 3DEC discrete element software. The effect of different water inlets and outlets on the temperature field of the production well and rock mass, as well as the water temperature of the production well outlet, is then calculated under various geothermal well layout conditions. Finally, through comparative analysis, the law of the influence of different geothermal well layouts on the rock mass water flow and heat transfer temperature is revealed.

CONCEPTUAL MODEL OF GEOTHERMAL EXPLOITATION

Figure 1 shows a schematic diagram of geothermal resource exploitation. Four water injection wells and water output wells were drilled from the

ground by using mechanical drills. The hot rock area at the bottom of the water injection wells and water output wells was mechanically fractured to form a microjoint system to open its fractured channel. The ground injected low-temperature water into the water injection well, and the water flowed into the well's bottom. Hot water is stored in the artificial heat reservoir area through convection and heat transfer with high-temperature hot rock, and high-temperature water is pumped out to the ground through the well for comprehensive utilization, such as power generation and heat. In this paper, only four injection wells and production wells are considered, and engineering fracture systems are ignored.

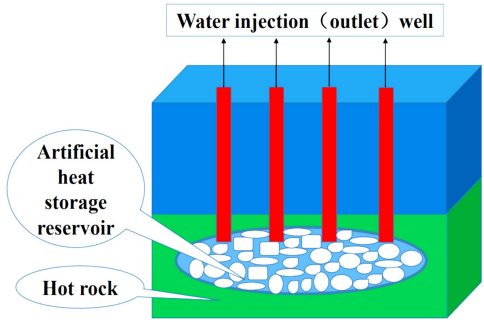


Figure 1: Schematic diagram of geothermal resource exploitation.

BASIC ASSUMPTIONS OF THE MODEL

The variables involved in heat conduction in 3DEC are temperature and the three components of the heat flux. The energy balance equation and Fourier law of heat conduction are related to these variables. The differential equation of heat conduction is obtained by combining the Fourier law with the energy balance equation. The differential equation can be solved under specific boundary and initial conditions based on specific geometry and properties. The following dimensionless numbers are used to characterize transient heat conduction.

Characteristic length:

$$Lc = \frac{V_s}{A_s} \tag{1}$$

where the characteristic length of the solid is expressed by Lc[m]; the volume of the solid is expressed by V_s[m^3], and the surface area of heat exchange is expressed by A_s[m^2].

Thermal diffusivity:

$$\kappa = \frac{k}{\rho C_v} \tag{2}$$

where κ is the thermal diffusivity in [m²/s]; k is the thermal conductivity in [W/(m·°C)]; ρ is the density in [kg/m³]; and C_v is the specific heat at constant volume in [J/kg·°C].

Characteristic time:

$$t_c = \frac{L_C^2}{\kappa} \tag{3}$$

where the characteristic time of the solid is expressed by $t_c[s]$.

The differential expression of the energy balance has the following form:

$$-q_{i,i} + q_v = \frac{\partial \zeta}{\partial t} \tag{4}$$

where $q_{i,i}$ is the heat-flux vector in [W/m³]; q_v is the volumetric heat-source intensity in [W/m³], and ζ is the heat stored per unit volume in [J/m³].

In general, the temperature change may be caused by variations in both energy storage and volumetric strain ε. The constitutive thermal law relating those parameters may be expressed.

as:

$$\frac{\partial T}{\partial t} = M_{th}\left(\frac{\partial \zeta}{\partial t} - \beta_{th}\frac{\partial \varepsilon}{\partial t}\right) \tag{5}$$

where M_{th} and β_{th} are material constants and T represents the temperature.

In this law, a particular case of $\beta_{th} = 0$ and $M_{th} = \frac{1}{\rho C_v}$ is considered, in which ρ is the mass density of the medium in [kg/m³] and C_v is the specific heat at constant volume in [J/kg·°C]. The change in strain is assumed to play a minor role in influencing the temperature validity for quasistatic mechanical problems involving solids and liquids.

$$\frac{\partial \zeta}{\partial t} = \rho C_v \frac{\partial T}{\partial t} \tag{6}$$

By substituting Eq. (6) for Eq. (4), the energy-balance equation was yielded.

$$-q_{i,i} + q_v = \rho C_v \frac{\partial T}{\partial t} \tag{7}$$

For all solids and liquids, the specific heats at constant pressure and constant volume are principally equivalent. Accordingly, C_v and C_p can be used by each other.

According to the finite-difference approximation principle of spatial derivatives, the numbers from 1 to 4 represent each node of the tetrahedron, the opposite side of node n is face n, and the value of the superscript (f) is related to the relevant variable on the face f.

The temperature changes linearly in the tetrahedron. The temperature gradient is represented by the node value of temperature according to the Gauss divergence theorem:

$$T_{,j} = -\frac{1}{3V} \sum_{l=1}^{4} T^l n_j^{(l)} S^{(l)} \tag{8}$$

where the external unit vector perpendicular to surface l is denoted by $[n]^{(l)}$, the surface area is denoted by S, and the tetrahedral volume is denoted by V.

Energy-balance equation formula of nodes. The energy-balance Eq. (7) may be expressed as:

$$q_{i,i} + b* = 0 \tag{9}$$

where

$$b* = \rho C_v \frac{\partial T}{\partial t} - q_v \tag{10}$$

is the instantaneous "physical strength" in the mechanical node formula. Using a tetrahedron analogy, the nodal heat $Q_e^n[w]\,\mathrm{n} = 1,4$ in equilibrium with its heat flux and body force can be expressed as:

$$Q_e^n = Q_t^n - \frac{q_v V}{4} + m^n C_v^n \frac{\mathrm{d}T^n}{dt} \tag{11}$$

where

$$Q_t^n = \frac{q_i n_i^{(n)} S^{(n)}}{3} \tag{12}$$

and

$$m^n = \frac{\rho V}{4} \tag{13}$$

In this theory, the node form of the energy-balance equation is required at each global node, in which the sum of equivalent node heat $(-Q_e^n)$ of all tetrahedrons and the node contribution $(-Q_w^n)$ of the applied boundary flux and source are zero.

In heat convection, it is presumed that fluid flow occurs within saturated fractures while the rock matrix is impermeable. As described in the previous section, heat can be transported by fluid convection, conducting in itself, and the rock mass. The fluid temperature generally varies in different rocks. Therefore, between the fracture fluid and the contacting rock (fluid-thermal coupling), heat transfer may occur, according to Newton's law of cooling. Coupling to heat transfer within the rock and the logic for heat transfer within the fluid is presented as follows.

Heat convection in the flow planes is described by the following equations. Heat is transported.

by conduction in the fracture fluid, according to Fourier's law:

$$q_f^T = -k_f^T \Delta T \tag{14}$$

where q_f^T is the specific heat flux in the fluid in [W/s^2] and k_f^T is the fluid thermal conductivity in [W/(m·°C)]. The energy-balance equation for the fluid obeys the equation.

$$\rho_f c_f \frac{\partial T_f}{\partial t} + \nabla \cdot q_f^T + \rho_f c_f q^f \cdot \nabla T_f + A_f h(T_f - T_s) = 0 \tag{15}$$

where $\rho_f c_f$ is the fluid density [kg/m^3] times the specific *heat [*J/(g·°C*)*]; q_f is the specific fluid discharge.

in [m^2/s]; A_f is the contact area per unit fluid volume in [m^2]; h is the fluid/rock heat transfer coefficient in [W/(m^2·°C)]; and T_f and T_s are the temperatures of the fluid and solid block, respectively.

For the solid blocks, the fluid flow was neglected; the transport of heat obeys Fourier's law as follows:

$$q^T = -k^T \Delta T \tag{16}$$

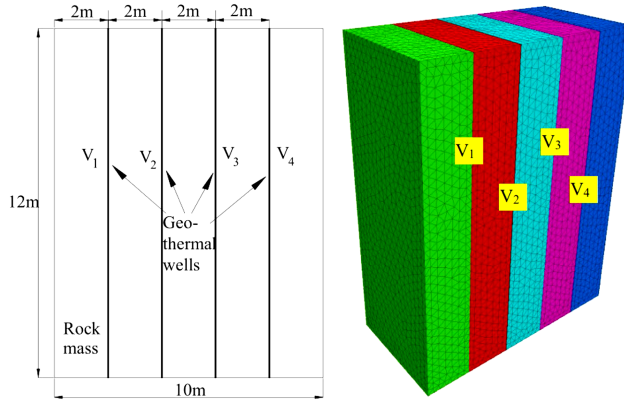
where q^T is the specific heat flux in [W/s^2] and k^T is the rock thermal conductivity in [W/(m·°C)]. The energy balance is

$$\rho_s c_s \frac{\partial T_s}{\partial t} + \nabla \cdot q_s^T - A_s h(T_f - T_s) = 0 \quad (17)$$

where $\rho_s c_s$ is the solid density [kg/m^3] times the specific *heat* [J/(g·°C)] and A_s is the contact area per unit volume of solid (from the aspect of fluid, there is contact on two sides: A_s^+, A_s^-, and $A_s = A_s^+ + A_s^-$).

EXAMPLE MODEL

In this paper, it is assumed that there is a hot rock with well-developed fractures in Northwest China, which has a huge heat reserve but is relatively deficient in groundwater resources. As a result, it proposed to inject water and effluent to ensure the long-term viability of geothermal exploration and provide stable expedition for local businesses. Considering the hydrothermal heat storage at approximately 100 m underground, low-temperature geothermal resources less than 90 °C are used for heating and technological processes. In the process of geothermal exploration, the interaction between the injection well and production well affects the water temperature distribution at the outlet of the production well and the temperature of the rock mass. The geothermal expedition process involves the interaction of injection wells, injection wells and production wells, and production wells on the outlet water temperature of production wells and rock mass temperature. In this paper, it is assumed that there are four geothermal wells in the model, and the optimization mechanism of the thermal recovery efficiency under different geothermal well layout conditions was studied. The model size was 10 m [length] × 5 m [width] × 12 m [height], the spacing between geothermal wells was set at 2 m, and the distance between the geothermal well and model boundary was also set as 2 m. The boundary conditions were as follows: the inlet unit temperature of the production well was set as the geothermal temperature, the outlet unit was set as the free temperature, then the inlet unit temperature of the injection well was set as the normal temperature. The outlet unit was set as the free temperature, and the other sides were adiabatic. The surrounding rock temperature was approximately 20 °C at −100 m above the ground, and the model assumed that the initial water temperature of the rock and injection well was 20 °C. The numerical model size and grid division of the optimization study on the thermal recovery efficiency under different geothermal well layout conditions are shown in Fig. 2. Here fractures V_1, V_2, V_3 and V_4 were simulated in four geothermal wells, with different water injection and water outlets.



(a) Geothermal well model size (unit: m); **(b)** Model mesh generation.

Figure 2: Geothermal well model size and mesh generation.

PARAMETERS AND CONTENT

Under conventional conditions, the thermophysical parameters of rock and water are listed in Table 1, in which the heat convection coefficient of rock and water was 30 W/(m^2·°C).

Table 1: Thermophysical parameters of the rock and water

Material	Density/(kg/m^3)	Specific heat/(J/(g·°C))	Coefficient of thermal conductivity/(W/(m·°C))
Rock	2700	0.8	2.3
Water	1000	4.2	0.6

The calculation conditions of the model are shown in Fig. 3. The calculation was carried out per the principle of establishing the same opening of geothermal wells, ensuring the same flow rate of injection wells and production wells and the same water flow velocity of injection wells and production wells. The calculation contents of the model are shown in Table 2. Here the inlet water temperature of the production well was 90 °C, the fracture opening (production well) was 2.5 mm. Fractures V_1 and V_4 were set to inject water, whereas V_2 and V_3 were set to outlet water, and the water flow speed was 2 mm/s. The fractures V_1 and V_4 were used to outlet water, while V_2 and V_3 were used to inject water, and the water flow speed was 2 mm/s. Set fractures V_1 and V_2 to inject water and V_3 and V_4 to outlet water, with a water flow rate of 2 mm/s. Water was injected into fractures

V_1 and V_3, and a water flow velocity of 2 mm/s was applied to fractures V_2 and V_4. Under these four working conditions, the influence of different water injections and water flows on the heat transfer temperature of the rock mass was simulated, calculated, and analyzed. The data obtained under each working condition were processed by postprocessing software into the rock mass temperature field and water temperature–time curve at the outlet of the production well for comparative analysis.

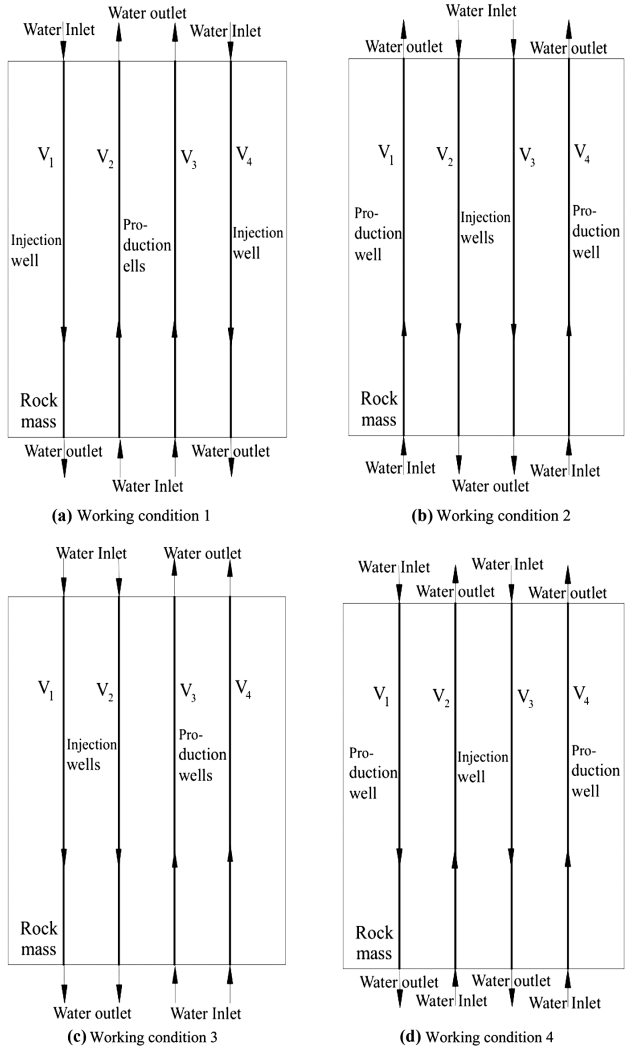


Figure 3: Model calculation conditions.

Table 2: Numerical simulation conditions

Calculation content	Water injection mode	Water velocity/(mm/s)	Inlet water temperature of production well/(°C)	Fracture (geothermal well) opening/(mm)
1	V_1, V_4 water injection, V_2, V_3 water outlet	2	90	2.5
2	V_1, V_4 water outlet, V_2, V_3 water injection			
3	V_1 V_2 water injection, V_3, V_4 water outlet			
4	V_1, V_3 water injection, V_2, V_4 water outlet			

RESULTS AND DISCUSSION

Influence of Different Injection Wells and Outlet Wells on the Temperature Field of the Rock Mass

The temperature field of the rock mass is shown in Fig. 4 under four working conditions, when the model reached a steady state.

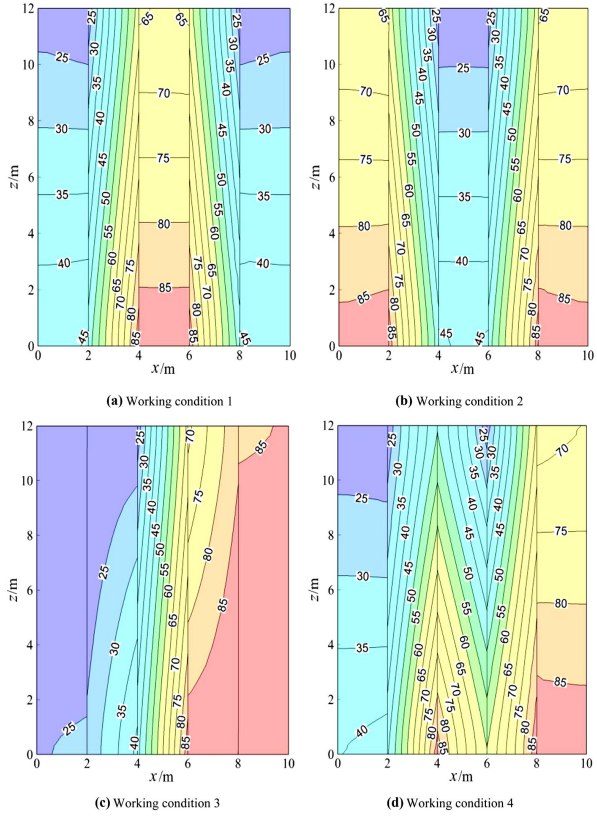


Figure 4: Temperature field of rock mass.

Figure 4a, b revealed that, in the initial state, low-temperature water (20 °C) was injected into the ground along with injection wells V_1 (V_2) and V_4 (V_3), while high-temperature water (90 °C) was pumped out from production wells V_2 (V_1) and V_3 (V_4). When the high-temperature water of production wells V_2 (V_1) and V_3 (V_4) entered the production well, it convected heat transfer with the rock mass on both sides of the production well (initial 20 °C); that is, the heat absorption temperature of the rock mass on both sides of the production well gradually increased, and the heat release temperature of the water flow of the production well decreased steadily. As the temperature of the rock mass on both sides of the production well increases, the low-temperature water flow of injection wells V_1 (V_2) and V_4 (V_3) passes through the rock mass with elevated temperature on one side and heat convection occurs between them. The three water flow heat release processes of production wells (heat convection between water flow of production well and its rock mass wall), the production wells water absorbed heat by contacting the rock mass wall (respective heat conduction of production well water and rock mass wall) and water flow heat absorption of injection wells (heat convection between water flow of injection well and its rock mass wall) were accompanied by water injection and water pump until the model reached a uniform state. At this time, the total amount of heat provided by the inlet water of the production wells was equal to the heat absorbed by the rock mass at its sidewall. The heat was absorbed by the water flow of the injection well, and they reached dynamic equilibrium. In addition, the temperature gradient at the edge (middle) of rock under working condition 1 was similar to that at the middle (edge) of rock under working condition 2. After the injection wells and production wells under two working conditions were switched, their temperature gradients were very similar, which constituted axial symmetry. The rock temperature gradients on both sides of the injection well (production well) were about 1.67 °C/m and 4.93 °C/m respectively, and the values of the rock temperature gradients were the same, but the temperature gradients' direction was opposite, which was caused by the same boundary conditions.

Comparison Fig. 4a, c showed that after the middle production well and edge injection well in Fig. 4a were changed to the left adjacent production well and the right adjacent injection well in Fig. 4c, the temperature field of rock mass of both sides of the edge formed a central symmetry, and the temperature gradient from the middle to both sides of the rock mass became smaller and smaller. Also, it showed that the water temperature at

the outlet of injection well V_1 decreased significantly and that at the outlet of production well V_4 increased significantly, the water temperature at the outlet of injection well V_2 decreased slightly, while that at the outlet of production well V_3 increased slightly. This is due to the thermal superposition effect of the adjacent injection well and the production well through the rock mass, which led to the higher temperature of the production well. A comparison between Fig. 4a,d indicated that from the middle production well in Fig. 4a, the edge injection well was changed to the interval between the injection well and production well in Fig. 4d. The temperature field of the rock mass on both sides of the edge formed a central symmetry, and the temperature gradient from the middle to both sides of the rock mass decreased, showing that the water temperature at the outlet of injection well V_1 decreased slightly and that at the outlet of injection well V_3 increased significantly. Also, the water temperature at the outlet of production well V_2 decreased significantly, while the water temperature at the outlet of production well V_4 changed a little, whereas the temperature gradient mainly showed a large difference between x (4–6 m). This was because in Fig. 4a, the heat superposition effect occurred in the middle production well through the rock mass, forming a temperature gradient from the bottom to the top, while in Fig. 4d, there was no heat superposition effect between the water injection wells and the production wells, but the heat convection between injection wells and the production wells was dominant, and the temperature gradient was mainly formed from left to right in a steady state.

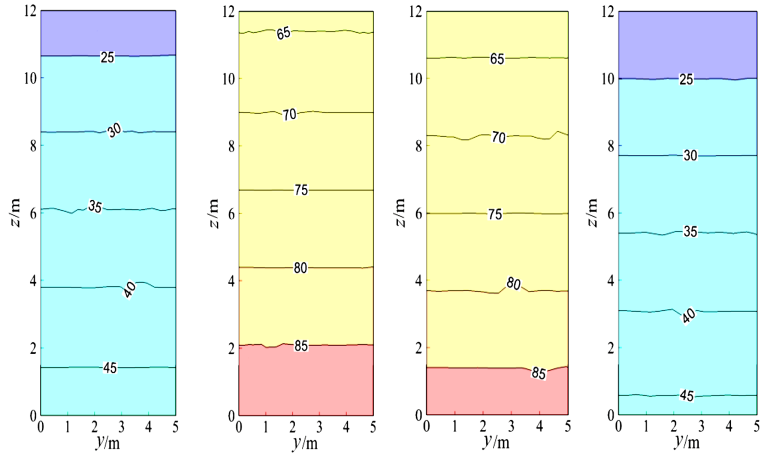
By comparing Fig. 4b with Fig. 4c, after the production well V_1 and injection well V_3 in Fig. 4b were changed to injection well V_1 and production well V_3 in Fig. 4c, only the temperature field within the range of [x (6–8 m) and z (0–12 m)] in Fig. 4b and [x (4–6 m) and z (0–12 m)] in Fig. 4c was the same. This is due to the "reverse direction" heat superposition of rocks between the injection well and production well under working conditions 2 and 3, resulting in a large temperature gradient of 4.93 °C/m. The temperature gradient of rocks between the injection wells in Fig. 4b was about 1.67 °C/m, and that between injection wells in Fig. 4c was 1.23 °C/m, which indicated that the heat conduction rate of rocks between injection wells in condition 3 was lower than that in condition 2. This is due to the different boundary

conditions outside the injection wells. According to the comparison between Fig. 4b,d, after changing from the production well V_1 and injection well V_2 in Fig. 4b to injection well V_1 and production well V_2 in Fig. 4d, only [x (2–4 m), z (0–12 m)], [x (6–8 m) and z (0–12 m)] in Fig. 4b were the same as those in [x (4–6 m), z (0–12 m)], [x (6–8 m) and z (0–12 m)] in Fig. 4d, which is due to the "opposite direction" heat superposition of rocks between injection wells and production wells in working conditions 2 and 4. In Fig. 4b, the temperature gradients of rocks from the outside to the middle were about 1.67 °C/m、4.93 °C/m and 1.67 °C/m respectively. In Fig. 4d, the temperature gradients of rocks from the outside to the middle were about 1.25 °C/m、4.93 °C/m and 4.93 °C/m respectively. The average temperature gradients under working condition 2 and 4 were about 2.76 °C/m and 3.7 °C/m respectively, indicating that the heat conduction rate of rocks in working condition 4 was higher than that in working condition 2. This is because the rock heat superposition effect between injection wells was less than the rock heat conduction effect between injection wells and production wells. That is, the rock temperature between injection wells was less than that between injection wells and production wells.

Comparing Fig. 4c with Fig. 4d, it can be seen that water injection wells V_2 and V_3 in Fig. 4c were changed into production wells V_2 and V_3 in Fig. 4d; that is, adjacent injection wells and adjacent production wells were changed into spaced injection wells and production wells. In Fig. 4c, the heat superposition effect between adjacent production wells was dominant, which made the water temperature at the production well outlet greatly increase, while in Fig. 4d, heat convection was dominant in the injection wells and production wells, which greatly decreased the water temperature at the outlet of the production well.

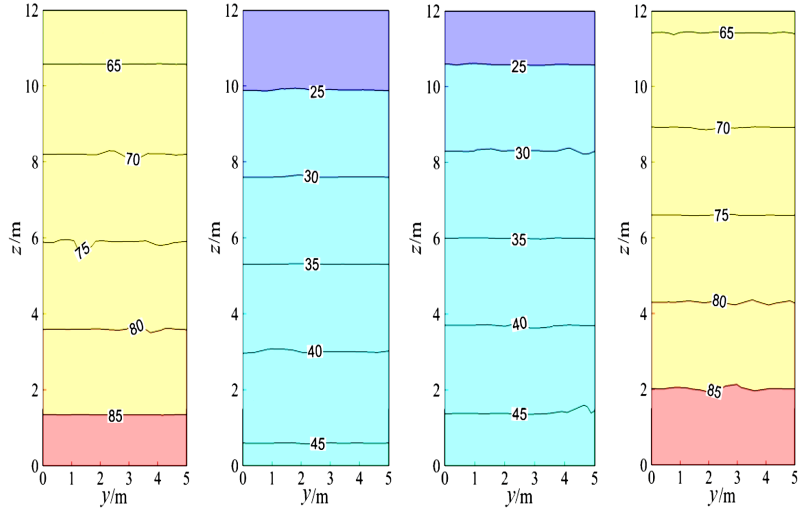
Temperature Field Analysis of the Water Injection Well and Water Outlet Well

The temperature fields of the injection wells and water outlet wells are shown in Figs. 5, 6, 7, and 8 when the model reached a steady-state under four working conditions.



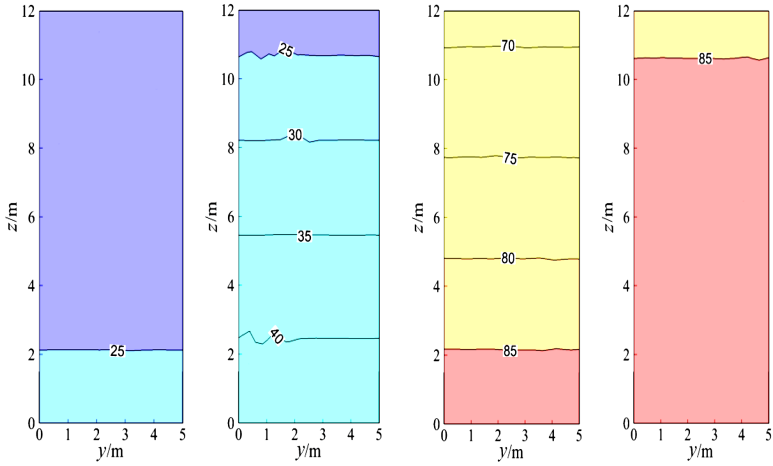
(a) water injection well V_1 **(b)** production well V_2 **(c)** production well V_3 **(d)** water injection well V_4

Figure 5: Temperature field of the geothermal well plane (working condition 1).



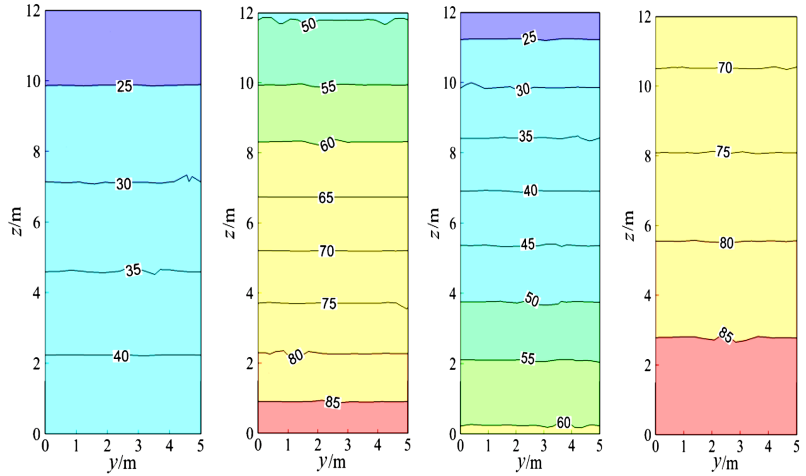
(a) production well V_1 **(b)** water injection well V_2 **(c)** water injection well V_3 **(d)** production well V_4

Figure 6: Temperature field ofthegeothermal well plane (working condition 2).



(a) water injection well V_1 **(b)** water injection well V_2 **(c)** production well V_3 **(d)** production well V_4

Figure 7: Temperature field of the geothermal well plane (working condition 3).



(a) water injection well V_1 **(b)** production well V_2 **(c)** water injection well V_3 **(d)** production well V_4

Figure 8: Temperature field of the geothermal well plane (working condition 4).

When the model reached a steady state, the production wells V_1 and V_4 and injection wells V_2 and V_3 were symmetrical, and the theoretical isotherms were the same, as shown in Fig. 5. The slight difference between the temperature fields of the production well and injection well was due to the random distribution of the model calculation grid, which had certain errors. The temperature gradients of the overall injection well and the production well were similar, and the temperature gradients of production wells V_1 and V_4 and injection wells V_2 and V_3 were also almost the same (approximately 2.13 °C/m). Figure 5 and Fig. 6 indicated that after the switch between the production well and water injection well, the temperature gradients of production wells V_1 and V_4 and injection wells V_2 and V_3 were almost the same (approximately 2.15 °C/m). The temperature gradients of the production well and injection well under the two working conditions were almost the same in numerical terms, but the difference was that the temperature gradients were in opposite directions. Again, Fig. 5 and Fig. 7 showed that after changing from the middle production well and edge injection well in Fig. 5 to the injection well on the left and the production well on the right, the temperature gradients of V_1 and V_2 of the injection well were approximately 0.51 °C/m and 1.85 °C/m, respectively. The temperature gradients of production wells V_3 and V_4 were approximately 1.69 °C/m and 0.47 °C/m, respectively, and the temperature gradient decreased. The reason for this was that the boundary conditions of injection and production wells had been altered. A comparison between Fig. 5 and Fig. 8 showed that the middle production well and marginal injection well in Fig. 5 were changed into interval injection wells and production wells in Fig. 8, whereas in Fig. 8, the temperature gradients of injection wells V_1 and V_3 were approximately 1.95 °C/m and 3.13 °C/m, respectively, and those of production wells V_2 and V_4 were approximately 3.18 °C/m and 1.97 °C/m, respectively. The temperature gradients of water injection wells V_1 and V_4 and V_3 and V_2 were indistinguishable. This is due to the similar boundary conditions between interval injection wells and production wells.

Comparison Figs. 6 and 7 revealed that after the production well V_1 and injection well V_3 in Fig. 6 were changed to injection well V_1 and production well V_3 in Fig. 7, the temperature field of injection well and production well plane in Fig. 7 changed greatly. The temperature gradient of the injection well and production well in Fig. 6 was 1.67 °C/m whiles the temperature gradient of injection well V_1 and production well V_4 in Fig. 7 became 0. The temperature gradient of injection well V_2 and production well V_3 was about 1.25 °C/m, indicating that after the two injection wells adjacent to the middle of Fig. 6 became the injection well adjacent to the left and the production well adjacent to the right of Fig. 7, the temperature gradient of the geothermal well decreased, that is, the water flow and heat transfer rate of the geothermal well decreased. When comparing Figs. 6 and 8, the temperature gradient of both injection well and production well in Fig. 6 was 1.67 °C/m, while that of injection well V_1 and production well V_4 in Fig. 8 was around 1.25 °C/m. The temperature gradient of production well V_2 and injection well V_3 was about 2.92 °C/m, and the average temperature gradient was about 2.09 °C/m, indicating that the temperature gradient of the geothermal well increased after the two injection wells adjacent to the middle part of Fig. 6. The central injection wells adjacent became the interval injection well and production well respectively, implying that the heat transfer rate of water flow increased.

The injection wells V_2 and V_3 in Fig. 7 were changed into production wells V_2 and V_3 in Fig. 8 based on the comparison of Figs. 7 and 8. That is, adjacent injection wells and adjacent production wells were modified into spaced injection wells and production wells, and the temperature gradient of water flow in injection wells and production wells in Fig. 7 was much smaller than that in Fig. 8. This is because heat convection between spaced injection wells and production wells was dominant and the water flow and the heat transfer speed was faster under the assumption of constant thermal resistance between the rock mass and the contact surface of the water flow.

Water Temperature–time Analysis of Geothermal Well Outlet

The temperature–time curve of the geothermal well outlet is shown in Fig. 9 under four conditions.

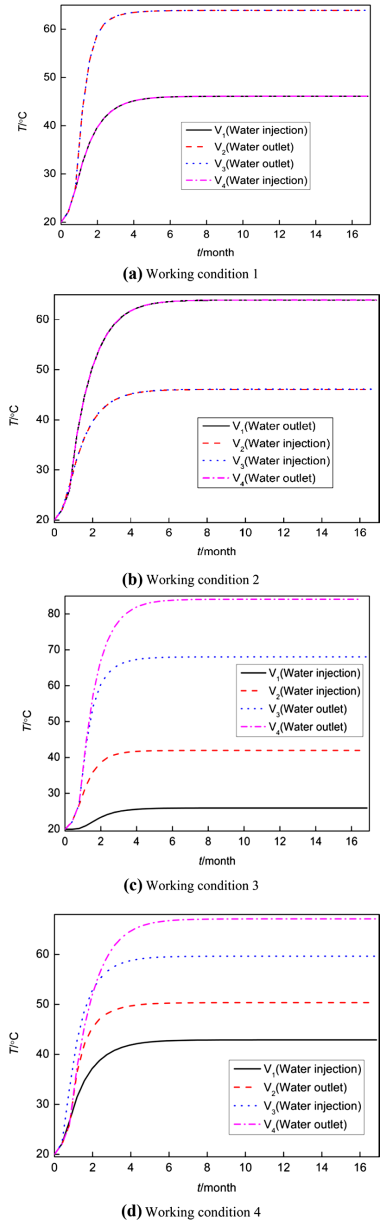


Figure 9: Temperature–time curve of geothermal well outlet.

As shown in Fig. 9a, under the condition that the middle part was production wells and the edge was injection wells, due to the symmetry of the model, the water temperature–time curves of production wells V_1 and V_4 and injection wells V_2 and V_3 coincided from the beginning to the end. It took approximately 7 months for the model to reach a steady state. At this time, the water temperature at the outlet of the production well reached 63.9 °C, the water temperature at the outlet of the injection well reached 46.1 °C, and the water temperature of the production wells was 38.61% higher than that of injection wells. According to the comparison in Fig. 9a, b, the model under working condition 2 took approximately 7 months to reach a steady-state after switching between injection wells and production wells. At this time, the water temperature at the outlet of production wells V_1 and V_4 and injection wells V_2 and V_3 was the same as that at production wells V_2 and V_3 and injection wells V_1 and V_4 in working condition 1. As shown in Fig. 9c, it took approximately 6 months for the model to reach a steady state. At this time, the water temperature at the outlet of production wells V_3 and V_4 reached 68.0 °C and 84.0 °C, respectively, and the water temperature at the outlet of injection wells V_1 and V_2 reached 41.9 °C and 25.9 °C, respectively. The reason was that the thermal superposition effect of the water flow of the adjacent production well was dominant. The heat absorption capacity of the rock mass boundary on the right side of production well V_4 (outside of the model was the adiabatic boundary) was less than that on the left side of production well V_3 (heat absorption capacity of injection well V_2). Likewise, although some heat superposition effect would occur in the water flow of adjacent production wells V_1 and V_2, the heat absorption capacity of the rock mass boundary on the right side of injection well V_2 (heat release from the production well V_3) was greater than that on the left side of injection well V_1 (outside the model was the adiabatic boundary). Also, the water temperature at the outlet of the production well and the injection well changed greatly when production well V_2 and water injection well V_4 were changed modified see Fig. 9a–c. This is because the both the left injection wells and the right production wells were adjacent. The water temperature at the outlet of the production well in working condition 3 was 4.1 °C (68.0–63.9 °C) and 20.1 °C (84.0–63.9 °C) higher than that in working condition 1, respectively. The average water temperature at the outlet of the production well in working condition 3 was approximately 12.1 °C (76.0–63.9 °C) higher than that in working condition 1. As shown in Fig. 9d, it took approximately 10 months for the model to reach a steady state. At this time, the water temperature at the outlets of production wells V_4 and V_2 reached 67.1 °C and

50.4 °C, respectively, while the water temperature at the outlets of injection wells V_3 and V_1 reached 59.7 °C and 42.9 °C, respectively. Therefore, the water temperature of the injection well outlet (59.7 °C) was higher than that of the production wells (50.4 °C) under working condition 4. The reason is after the separation between the injection well and production well, one side of the boundary of the water flow on both sides of the production well V_4 was the adiabatic boundary of the rock mass, and the other side was the heat absorption boundary of the water flow on injection well V_3. Both sides of the water flow on the production well V_2 were injection wells V_3 and V_1 (the outer side of production well V_2 was the endothermic boundary of water flow), so the water temperature at the outlet of production well V_4 was higher than that of production well V_2. Both sides of injection well V_3 were heat release boundaries of production wells V_4 and V_2. Therefore, the high-temperature water flow of the two production wells provided the boundary conditions for both sides of injection well V_3 to absorb more heat. Because the water flow of injection well V_1 only absorbed heat from production well V_2 via heat convection, its temperature was the lowest. It can be seen from the comparison of Fig. 9a,d that the water temperature of production wells and injection wells outlet has changed greatly under two working conditions, and water temperature of production wells outlet in working condition 4 was 3.2 °C (67.1–63.9 °C) and −13.5 °C(50.4–63.9 °C) higher than that in working condition 1, and the average water temperature at the outlet of production well in working condition 4 was approximately 8.35 °C (67.1 °C -58.75 °C) lower than that in working condition 1, so working conditions 1 and 2 were superior to working condition 4.

Furthermore, a comparison between Fig. 9b, c indicated that after production well V_1 and injection well V_3 in Fig. 9b were changed to injection well V_1 and production well V_3 in Fig. 9c, the outlet water temperature of production well V_3 and V_4 in Fig. 9c was about 4.1 °C (68.0–63.9 °C) and 20.1 °C (84.0–63.9 °C) higher than that of production well V_1 and V_4 in Fig. 9b respectively. The water outlet temperature of working condition 3 was about 12.1 °C higher than that of working condition 2 with an average increase of outlet water temperature of production well of about 18.9%. By comparing Fig. 9b with Fig. 9d, after production well V_1 and injection well V_2 in Fig. 9b were changed to injection well V_1 and production well V_2 in Fig. 9d, the water temperature at the outlet of production well V_2 and V_4 in Fig. 9(d) was about −13.5 °C (50.4–63.9 °C) and 3.2 °C (67.1–63.9 °C) higher than that of production well V_1 and V_4 in Fig. 9b, respectively. The average outlet water temperature in working condition 4 was about −5.15 °C

higher than that in working condition 2, and the average increase of outlet water temperature in producing well was about − 8.8%. By comparing Fig. 9c with Fig. 9d, when the water temperature at the outlet of production wells V_2 and V_4 in Fig. 9d was about − 17.6 °C (50.4–68.0 °C) and − 16.9 °C (67.1–84.0 °C) higher than that of production wells V_3 and V_4 in Fig. 9c, respectively, after the water temperature at the outlet of production wells V_2 and V_4 in Fig. 9c was changed from production well V_2 and production well V_3 in Fig. 9c to production well V_2 and production well V_3 in Fig. 9d, the average water temperature at the outlet of production well in condition 4 was about − 17.25 °C higher than that in condition 3, and the average increase of water temperature at the outlet of production well was about − 29.36%. According to the comprehensive comparison of Fig. 9a–d, it can be seen that the water temperature at the outlet of the production well, the optimal order of the model was working condition 3 > working condition 1 = working condition 2 > working condition 4. Furthermore, the time required for the model of working condition 3 to reach steady state was the shortest, while the time required for the model of working condition 4 to reach a steady state was the longest.

CONCLUSION

In this paper, a new mathematical modeling approach was presented to improve the thermal exploration efficiency under different geothermal well layout conditions. Fractures V_1 and V_4 were developed as injection wells whereas V_2 and V_3 as production wells. Fractures V_1 and V_4 were taken as production wells, V_2 and V_3 as injection wells; Fractures V_1 and V_2 were constructed as injection wells, V_3 and V_4 as production wells; Fractures V_1 and V_3 were constructed as injection wells, V_2 and V_4 as production wells. Under these four working conditions, the influence of different injection wells and production wells on rock mass temperature was simulated, calculated, and analyzed by the 3DEC program. The calculations revealed that when the position of the model injection well and production well was adjusted, the isothermal number line of rock mass was almost the same in value, but the direction of the water flow and heat transfer was opposite. The maximum water temperature at the outlet of the production well was 84.0 °C due to the thermal superposition effect of the rock mass between the adjacent injection wells and between the adjacent production wells. Conversely, the minimum water temperature at the outlet of the production well was 50.4 °C under working condition 4, which was determined by the convection heat transfer

between the water flow and the rock between the interval injection wells and the interval production wells. Under these two working conditions, the isotherms of rock mass on both sides of the edge showed central symmetry, and the temperature gradient gradually decreased from the middle to both ends of the rock mass, indicating that the heat transfer velocity of rock mass gradually decreased from the middle to both ends. Working condition 3 took approximately 6 months to reach a uniform state while working condition 4 took approximately 10. Under working conditions 1 and 2, the water temperature at the outlet of production well and the time required to reach a steady state were between working conditions 3 and 4.

ACKNOWLEDGEMENTS

We extended our sincere thanks to the funding sponsore of Yan'an University and Shandong Provincial Lunan Geology and Exploration Institute.

REFERENCES

1. Wen, Q. *et al.* Review on model of wellbore temperature distribution during drilling. *West-China Exp. Eng.* 19(11), 60–63 (2007) (in Chinese).
2. Yu, J. *Research on the wellbore temperature for geothermal wells in Tibet* (China University of Geosciences, 2013) (in Chinese).
3. Wang, P., Xiang, H. & Zhou, X. Well location deployment and reasonable well spacing shallow exploration. *Chem. Enterp. Manage.* 25(4), 72 (2018) (in Chinese).
4. Wu, B., Zhang, X. & Jeffrey, R. G. A model for downhole fluid and rock temperature prediction during circulation. *Geothermics* 50(50), 202–212 (2014).
5. Angelotti, A. *et al.* Energy performance and thermal impact of a borehole heat exchanger in a sandy aquifer:influnce of the groundwater velocity. *Energy Convers. Manage.* 77, 700–708 (2014).
6. Gao, Q. & Ghassemi, A. Three-dimensional thermo-poroelastic modeling and analysis of flow, heat transport and deformation in fractured rock with applications to a lab-scale geothermal system. *Rock Mech. Rock Eng.* 53, 1565–1586 (2020).
7. Rees, S. & He, M. A three-dimensional numerical model of borehole heat exchanger heat transfer and fluid flow. *Geothermics* 46(10), 1–13 (2013).
8. Yoshioka, M., Takakura, S. & Uchida, Y. Estimation of groundwater flow from temperature monitoring in a borehole heat exchanger during a thermal response test. *Hydrogeol. J.* 26, 853–867 (2018).
9. Wang, G. *et al.* Heat extraction analysis of a novel multilateral-well coaxial closed-loop geothermal system. *Renew. Energy* 163, 974–986 (2021).
10. Dehkordi, S. E. & Schincariol, R. A. Effect of thermal-hydrogeological and borehole heat Exchanger properties on performance and impact of vertical closed-loop geothermal heat pump systems. *Hydrogeol. J.* 22, 189–203 (2014).
11. Gao, J. Study on mechanism of the influence of geothermal temperature and production well depth on water flow and heat transfer temperature in rock mass. *Prog. Geophys.* 35(05), 1659–1664 (2020) (in Chinese).

12. Gao, J. Study on geothermal well spacing based on water flow and heat transfer rock mass. *Prog. Geophys.* 35(06), 2058–2063 (2020) (in Chinese).
13. Li, W. *et al.* Borehole temperature logging and temperature field in the xiongxian geothermal field Hebei Province. *Chin. J. Geol.* 49(3), 850–863 (2014) (in Chinese).
14. Shen, X. Analysis of the influence of key processes on the water yield and temperature of geothermal wells. *Henan Water Resour. South-to-North Water Divers.* 20, 48–49 (2015) (in Chinese).
15. Wang, L. *et al.* The prediction of wellbore fluid temperature distribution of geothermal production well. *China Min. Mag.* 24(S1), 376–380 (2015) (in Chinese).
16. Wang, F. *et al.* Analysis of factors affecting fluid production temperature of porous sandstone geothermal wells in the tower. *West-China Explor. Eng.* 30(10), 45–47 (2018) (in Chinese).
17. Zhu, M. *et al.* Heat preservation suggestion and heatloss analysis of geothermal well. *Sci. Technol. Rev.* 33(22), 32–36 (2015) (in Chinese).
18. Dou, H. *et al.* A model of temperature loss in the wellbore of geothermal exploitation for heating and its application. *Geol. Explor.* 55(05), 1276–1286 (2019) (in Chinese).
19. Wang, F., Zhang, X. & Zheng, H. Effect of pumping and irrigation wells arrangement groundwater flow field and temperature field. *Build. Technol. Dev.* 43(08), 31–35 (2016) (in Chinese).
20. Ma, J. *et al.* Influence of the distribution of pumping and injection wells on heat transfer characteristic of borehole heat exchangers. *J. Basic Sci. Eng.* 27(05), 1158–1171 (2019) (in Chinese).
21. Ma, J. *et al.* Influence of quantity of pumping and injection wells on heat transfer characteristic of coupling borehole heat exchangers. *Acta Energ. Solaris Sin.* 41(03), 109–118 (2020) (in Chinese).
22. Jin, M., Tanu, Q. & Li, X. Optimum location of pumping and injection wells of groundwater heat exchange system using numerical modeling of water and heat transport. *Bull. Geol. Sci. Technol.* 31(05), 128–135 (2012).
23. Li, F. *et al.* Simulation for water-heat coupling process of single well ground source heat pump systems implemented by T2well. *Acta Energ. Solaris Sin.* 41(04), 278–286 (2020) (in Chinese).

24. Ma, Z. *et al.* Reasonable well spacing and layout optimization of shallow source heat pump using sand trough simulation:a case study in Feuglinjiuxi. *J. Water Resour. Water Eng.* 29(04), 143–149 (2018).
25. Joël, M. Z., Louis, L. & Jasmin, R. Sustainable electricity generation from an Enhanced Geothermal System considering reservoir heterogeneity and water losses with a discrete fractures model. *Appl. Therm. Eng.* 192, 116886 (2021).
26. Loret B. Enhanced geothermal systems (EGS): hydraulic fracturing in a thermo -poroelastic framework. In *Fluid Injection in Deformable Geological Formations*. (Springer, Cham, 2019).
27. Yu, L. *et al.* Modified zipper fracturing in an enhanced geothermal system reservoir and heat extraction optimization via orthogonal design. *Renew. Energy* 161, 373–385 (2020).
28. Xu, Y. *et al.* Study on optimal arrangement of pumping and irrigation system for groundwater heat pump. *Water Sci. Eng. Technol.* 06, 54–59 (2017).
29. Deng, S. *et al.* Optimization simulation research on middle-deep geothermal recharge wells based on optimal recharge efficiency. *Front. Energy Res.* 08, 598229 (2020).
30. Olabi, A. G. *et al.* Geothermal based hybrid energy systems, toward eco-friendly energy approaches. *Renew. Energy* 147, 2003–2012 (2020).
31. Rezaei, M., Sameti, M. & Nasiri, F. An enviro-economic optimization of a hybrid energy system from biomass and geothermal resources for low-enthalpy areas. *Energy Clim. Change* 2, 100040 (2021).
32. Gong, L., Zhang, Y. & Bai, Z. Geothermal-solar hybrid power with the double-pressure evaporation arrangement and the system off-design evaluation. *Energy Convers. Manage.* 244, 114501 (2021).
33. Qu, S. *et al.* Study of operational strategies for a hybrid solar-geothermal heat pump system. *Build. Simul.* 12, 697–710 (2019).
34. Tian, X. & You, F. Carbon-neutral hybrid energy systems with deep water source cooling, biomass heating, and geothermal heat and power. *Appl. Energy* 250, 413–432 (2019).
35. Chen, Y., Wang, J. & Lund, P. D. Thermodynamic performance analysis and multi-criteria optimization of a hybrid combined heat and power system coupled with geothermal energy. *Energy Convers. Manage.* 210, 112741 (2020).

Chapter

6

MATHEMATICAL MODELING AND THERMODYNAMICS OF PRANDTL–EYRING FLUID WITH RADIATION EFFECT: A NUMERICAL APPROACH

Zakir Ullah[1] , Ikram Ullah[2], Gul Zaman[1] , Hamda Khan[3] & Taseer Muhammad[4]

[1] Department of Mathematics, University of Malakand, Chakdara, Dir(L), Khyber Pakhtunkhwa 18800, Pakistan
[2] Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Peshawar, KP 25000, Pakistan
[3] Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Islamabad, Pakistan
[4] Department of Mathematics, College of Sciences, King Khalid University, Abha 61413, Saudi Arabia

Citation: (APA): Ullah, Z., Ullah, I., Zaman, G., Khan, H., & Muhammad, T. (2021). Mathematical modeling and thermodynamics of Prandtl–Eyring fluid with radiation effect: a numerical approach. Scientific Reports, 11(1), 1-11.(11 pages).

ABSTRACT

Main concern of current research is to develop a novel mathematical model for stagnation-point flow of magnetohydrodynamic (MHD) Prandtl–Eyring fluid over a stretchable cylinder. The thermal radiation and convective boundary condition are also incorporated. The modeled partial differential equations (PDEs) with associative boundary conditions are deduced into coupled non-linear ordinary differential equations (ODEs) by utilizing proper similarity transformations. The deduced dimensionless set of ODEs are solved numerically via shooting method. Behavior of controlling parameters on the fluid velocity, temperature fields as well as skin friction and Nusselt number are highlighted through graphs. Outcome declared that dimensionless fluid temperature boosts up for both the radiation parameter and Biot number. It is also revealed that the magnitude of both heat transfer rate and skin friction enhance for higher estimation of curvature parameter. Furthermore, comparative analysis between present and previous reports are provided for some specific cases to verify the obtained results.

INTRODUCTION

In fluid dynamics, the phenomenon of stagnation-point flow has got considerable attention of various researchers in the recent past due to its significant applications in natural and industrial phenomena. The former includes a flow of fluid over the tips of various objects, e.g., ships, submarines, aircrafts, rockets etc1. In biology, a blood-flow in the blood vessel at the branch/ sub-branch separates into two or more directions and corresponds to the stagnation-point flow2. Hiemenz3 in 1911, first proposed an exact solution for the stagnation-point flow in a static-rigid surface. In this study, Hiemenz utilized appropriate transformation to transform the steady two dimensional (2D) Navier-Stokes equations into non-dimensional highly ODEs. After the remarkable work of Hiemenz3, many investigators considered the stagnation-point flow phenomena by means of different physical features4,5,6,7. Recently, Vaidya et al.8 examined the steady 2D oblique stagnation-point flow on a stretching plate. They have solved analytically dimensionless highly non-linear ODEs using the Optimal Homotopy Analysis Method (OHAM). Further, it has been shown there8 that axial fluid velocity declines with a rise in the viscosity while the dual effect of viscosity is found on the transverse fluid velocity. Meanwhile, Hayat et al.9 discussed the steady 2D stagnation-point flow with both heat generation and thermal radiation. They noticed in9 that variations

in the radiation variable and Biot number improve the dimensionless fluid temperature. Further, Aly and Pop10 have obtained unique and dual solutions for a steady 2D stagnation-point flow associated with dynamic hybrid nanofluid. They showed that dual and unique solutions exist for a certain estimations of magnetic parameter and revealed that the behavior of hybrid nanofluid velocity field and temperature are different along the three regions of stability. Additionally, Wain et al.11 comprehended the analysis for incompressible stagnation-point flow in a shrinking/stretching plate, admitting growth of skin friction and heat transfer due to the melting parameter.

Non-Newtonian fluids flow phenomena plays a pivotal role in numerous natural, industrial, geophysical and engineering processes. Some common examples of these fluids are drilling mud, lubricating oils, liquid crystals, paints, silly putty, polymeric liquids, biological fluids and many others. The properties of such fluids are hard to define as a single constitutive equation but many attempts have been made by the investigators to characterize the rheological characteristics of fluids containing non-Newtonian fluid behavior. Non-Newtonian fluid models are evidently more complex and have a highly nonlinear behavior. Various investigators presented different fluid models12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27 to describe the complex nature of non-Newtonian fluids phenomena. Prandtl–Eyring model is a particular type of non-Newtonian fluid which indicates that shear stress is proportional to the sine hyperbolic function of strain rate to the fluid. Recently, Khan et al.28 proposed the combined impacts of Brownian and thermophoresis diffusion on 2D Prandtl–Eyring nanofluid with entropy generation through a heated stretchable plate. They revealed that for greater estimations of Brinkman number and material parameter, the entropy generation rate rises. Further, the influences of heat source and thermophoresis on steady incompressible MHD flow of Prandtl–Eyring nanofluid in a symmetric channel was analyzed by Akram et al.29. They analyzed that Brownian and thermophoresis parameters have opposite behavior on both the temperature gradient and heart transfer rate. Meanwhile, Uddin et al.30 examined numerically the impact of activation energy on dynamical 2D MHD Prandtl–Eyring nonofluid due to the Joule heating effect. Additionally, Rehman et al.31 studied scaling group transformation method for steady incompressible Prandtl–Eyring fluid through a 2D semi-infinite stretching sheet. With the help of scaling transformation they obtained new similarity transformations for the analysis of Prandtl–Eyring fluid flow. Abdelsalam et al.32 used the Eyring-Powell fluid model as the

base fluid to investigate the behavior of a microorganism swimming through a cervical canal. Moreover, Shankar and Naduvinamani33 carried out the numerical solution for magnetized squeezed unsteady 2D Prandtl–Eyring fluid flow through a horizontal sensor sheet. From their investigation it has been noticed that fluid velocity boosts with magnetic parameter while the fluid temperature diminishes in the flow region with magnetic parameter.

The influence of thermal radiation plays an essential role in space technology and in processes with high temperatures. The study of heat transfer characteristics on a stretched sheet with radiation was studied by a number of researchers. Smith34 was the first researcher who presented the aspect of thermal radiation on steady 2D flow. Later on, the influence of thermal radiation on fluid temperature and heat transfer in an emitting/absorbing medium flowing on a wedge was explored by Viskanta and Grosh35. Recently, Raza et al.36 numerically elaborated the impacts of MHD and thermal radiation on unsteady 2D molybdenum disulfide nanoparticle through a porous channel. They revealed that the heat transfer rises by enhancing the solid volume fraction for various shapes of nanofluids. Gireesha et al.37 analyzed the preparation process of hybrid nanomaterials on a porous longitudinal fin with thermal radiation. Wakif38 scrutinized the impact of incompressible MHD flow of Casson fluid on a horizontal stretched plate with thermal radiation and they show that with radiation parameter the nanofluid temperature increases. Additionally, the characteristics of heat transfer and MHD nanoparticle on a stretching plate with thermal radiation and Joule heating impacts was scrutinized by Dogonchi and Ganji39. They observed that with an increase in the volume of nanofluid turn out a linear rise in the Nusselt number, whereas, this number shows inverse behavior with thermal radiation. Khan and Alzahrani40 proposed the combined effects of thermal radiation and viscous dissipation on 2D nanofluid with entropy generation through a stretched surface. Raza et al.41 studied the thermal radiation impacts on the convective flow of a non-Newtonian fluid through a curved surface. Moreover, Ullah et al.42 numerically studied the flow pattern followed by hybrid nanoliquids (AA7075, AA7072) using an infinite disk in the presence of thermal radiation. Furthermore, the authors suggested that Nusselt number shows direct behavior with thermal slip and radiation parameters where reverse effect was noticed for large Eckert number.

In view of aforementioned literature survey, it is concluded that Prandtl–Eyring fluid in the cylindrical geometry is not addressed yet. Therefore our intention here is to develop a novel mathematical modeling for

incompressible MHD43,44 Prandtl–Eyring fluid flow near the stagnation-point induced by stretching cylinder. Energy expression is characterized with thermal radiation. Suitable transformations are utilized to convert the set of non-linear PDEs into a system of highly non-linear ODEs. The reduced dimensionless system is then solved by Shooting method. The influence of various controlling parameters and dimensionless numbers, like curvature, magnetic, radiation and fluid parameters, Prandtl and Biot numbers on the fluid velocity, temperature as well as skin friction and heat transfer are reported via graphs and investigated. The present results of skin friction and heat transfer rate are compared with the previous published work in the limiting cases which are found to be satisfactory.

MATHEMATICAL MODELING

We consider steady, axisymmetric and 2D MHD stagnation-point flow of incompressible Prandtl–Eyring fluid model by a stretching cylinder. Radiation is considered in the heat expression. Further, let the cylinder is being Stretchable in the xx-axis with linear velocity $u = \frac{U_0 x}{l}$. Let the respective (x, r)-coordinates are presumed in cylinder and normal to it (see Fig. 1). Moreover, heat transportation is performed under the convective surface condition. The constitutive equation for the Prandtl–Eyring fluid model45 is given as

$$T = -pI + \mu S. \tag{1}$$

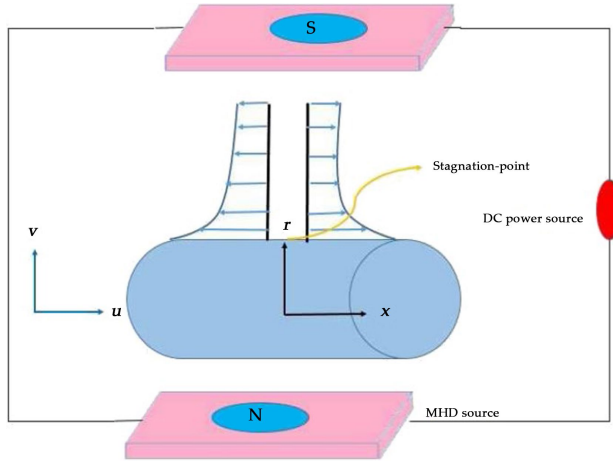


Figure 1: Flow configuration.

In Eq. (1), T, p, I and μ are fluid Cauchy stress tensor, fluid pressure, identity tensor, and dynamic viscosity respectively. Where S strands for extra stress tensor of Prandtl–Eyring fluid model and given as follows45:

$$S = \left[\frac{a_1\ arc\ \sinh\left(\frac{1}{c_1}\sqrt{\frac{1}{2}tr(A_1^2)}\right)}{\sqrt{\frac{1}{2}tr(A_1^2)}}\right]A_1. \tag{2}$$

In Eq. (2), a_1 and c_1 denotes the material parameters of fluid and $A_1=\nabla V+(\nabla V)^T$ is the first Rivlin-Ericksen tensor. The first Rivlin-Ericksen tensor A_1 for present study in cylindrical coordinates is expressed as

$$A_1 = \begin{bmatrix} 2\frac{\partial v}{\partial r} & 0 & \frac{\partial u}{\partial x}+\frac{\partial u}{\partial r} \\ 0 & 2\frac{v}{r} & 0 \\ \frac{\partial u}{\partial r}+\frac{\partial u}{\partial x} & 0 & 2\frac{\partial v}{\partial x} \end{bmatrix}. \tag{3}$$

The required component of the present model is given by

$$\tau_{rx} = \left[\frac{a_1}{\rho}\ arc\ \sinh\left(\frac{1}{c_1}\frac{\partial u}{\partial r}\right)\right], \tag{4}$$

here $\sinh^{-1}$ is presumed upto second-order estimation and is expressed by

$$\sinh^{-1}\left(\frac{1}{c_1}\frac{\partial u}{\partial r}\right) = \frac{1}{c_1}\frac{\partial u}{\partial r} - \frac{1}{6}\left(\frac{1}{c_1}\frac{\partial u}{\partial r}\right)^3. \tag{5}$$

Under the above assumption, the flow governing expressions are46,47,48

$$\frac{\partial(ru)}{\partial x}+\frac{\partial(rv)}{\partial r} = 0, \tag{6}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial r} = U_e\frac{dU_e}{dx}+\frac{a_1}{\rho c_1}\left[\frac{1}{r}\frac{\partial u}{\partial r}+\frac{\partial^2 u}{\partial r^2}\right]-\frac{a_1}{2\rho c_1^3}\left(\frac{\partial u}{\partial r}\right)^2\left(\frac{\partial^2 u}{\partial r^2}\right)-\frac{a_1}{6\rho c_1^3 r}\left(\frac{\partial u}{\partial r}\right)^3+\frac{\sigma B_0^2}{\rho}(U_e-u), \tag{7}$$

$$u\frac{\partial T}{\partial x}+v\frac{\partial T}{\partial r} = \alpha\left[\frac{\partial^2 T}{\partial r^2}+\frac{1}{r}\frac{\partial T}{\partial r}\right]+\frac{1}{\rho c_p}\frac{16\sigma^* T_\infty^3}{3k^*}\left[\frac{\partial^2 T}{\partial r^2}+\frac{1}{r}\frac{\partial T}{\partial r}\right], \tag{8}$$

along with associated boundary conditions47,48

$$u = U(x) = \frac{U_0 x}{l},\quad v = 0,\quad -k\frac{\partial T}{\partial r} = h_f(T_f - T\infty)\ \text{at}\quad r = R, \tag{9}$$

$$u \to U_e(x) = \frac{U_\infty x}{l},\quad T \to T_\infty\quad \text{as}\quad r \to \infty. \tag{10}$$

In which u and v represents the respective velocity in the xx- and rr-directions , T, T_w and T_∞, indicates fluid, boundary and free stream temperatures

respectively, the symbols ν, B_0 and σ denotes kinematic viscosity, strength of magnetic field and liquid electrical conductivity respectively. The thermal diffusivity, coefficient of mean absorption, fluid density, specific heat and Stefan-Boltzmann constant are denoted respectively by the symbols α, k^*, ρ, c_p and σ^*.

Now, considering the following similarity variables

$$\eta = \frac{r^2 - R^2}{2R}\sqrt{\frac{U_0}{l\nu}}, \quad \psi = \sqrt{\frac{U_0\nu}{l}}Rxf(\eta), \quad \theta(\eta) = \frac{T - T_\infty}{T_f - T_\infty}, \tag{11}$$

where

$$u = \frac{1}{r}\frac{\partial\psi}{\partial r} \quad \text{and} \quad v = -\frac{1}{r}\frac{\partial\psi}{\partial x}. \tag{12}$$

Using Eq. (11) along with Eq. (12) in Eqs. (6)–(10), gives

$$A\left[2Kf'' + (2K\eta + 1)f'''\right] - A\beta\left[\frac{4}{3}K(1 + 2K\eta)(f'')^3 + (1 + 2K\eta)^2(f''')(f'')^2\right] + f''f - (f')^2 + M^2(B - f') + B^2 = 0, \tag{13}$$

$$\left(1 + \frac{4}{3}R\right)\left[(1 + 2K\eta)\theta'' + 2K\theta'\right] + Prf\theta' = 0, \tag{14}$$

$$f(0) = 0, \quad f'(0) = 1, \quad \theta'(0) = -Bi(1 - \theta(0)), \tag{15}$$

$$f'(\infty) = B, \quad \theta(\infty) = 0. \tag{16}$$

In the above expressions $A = \frac{a_1}{\mu c_1}$ and $\beta = \frac{U_0^3 x^2}{2c_1 l^3 \nu}$ denoted fluid parameters, $K = \frac{1}{R}\sqrt{\frac{l\nu}{U_0}}$ indicates the curvature parameter, $M = \sqrt{\frac{l\sigma B_0^2}{U_0\rho}}$ means a magnetic field parameter, $B = \frac{U_\infty}{U_0}$ is the ratio of velocities, $Pr = \frac{\nu}{\alpha}$ denotes the Prandtl number, $R = \frac{4\sigma^* T_\infty^3}{k*k}$ denotes radiation parameter and $Bi = \sqrt{\frac{l\nu}{U_0}}\frac{h_f}{k}$ is the Biot number.

Finally, the mathematical expressions for the important aspects i.e., skin friction coefficient (Cfx)(Cfx) and the Nusselt number NuxNux are given by

$$C_{f_x} = \frac{\tau_w}{\frac{\rho U_0^2 x^2}{2l^2}}, \quad Nu_x = \frac{xq_w}{k(T_f - T_\infty)}. \tag{17}$$

In Eq. (17), the wall shear stress and heat flux respectively are

$$\tau_w = a_1\left[\frac{1}{c_1}\frac{\partial u}{\partial r} - \frac{1}{6c_1^3}\left(\frac{\partial u}{\partial r}\right)^3\right]_{r=R} \quad q_w = \left(k + \frac{16\sigma^* T_\infty^3}{3k^*}\right)\left(\frac{\partial T}{\partial r}\right)_{r=R}. \tag{18}$$

Inserting Eq. (11) along with Eq. (12) into Eq. (17) we obtain

$$\frac{Re^{\frac{1}{2}} C_{f_x}}{2} = Af''(0) - \frac{1}{3}A\beta f''^3(0), \quad Re^{-\frac{1}{2}} Nu_x = -\left(1 + \frac{4}{3}R\right)\theta'(0), \tag{19}$$

where represents the local Reynolds number and can be expressed as $Re = \frac{U_0 x^2}{l\nu}$.

NUMERICAL SCHEME

The obtained dimensionless system of ODEs and validation analysis together with the appropriate conditions cannot be simulated directly or analytically due to highly non-linear nature . Therefore, these non-linear ODEs are solved numerically by implementing Shooting iterative technique via Mathematica software. Here, in this numerical procedure first higher order ODEs in Eqs. (13) and (14) are altered into a set of first order ODEs. In this numerical procedure, it is also very significant to assume an appropriate finite value for $\eta \to \infty$. Furthermore, we also choose suitable initial guesses of $f''(0)$ and $\theta'(0)$ and obtain the solution by adopting Runge-Kutta Fehlberg fifth order technique as an initial value problem which has truncation error of order 5. The accuracy of the current results has been verified and are given in Tables 1, 2, 3 by comparing with the existing solutions of[49,50,51,52,53,54] for some particular cases, where it is revealed that the current results and their solutions are approximately identical.

Table 1: Comparative values of skin friction $Re^{\frac{1}{2}} C_{f_x}$ **against variations in** M **when** A=1,K=β=B=Pr=R=Bi=0.0

M	Ref.[49]	Ref.[50]	HAM results	Present results
0	−1	−1	−1	−1
0.5	−1.11803	−1.1180	−1.11852	−1.11803
1.0	−1.41421	−1.4140	−1.41620	−1.41421
5.0	−2.44949	−2.4493	−2.44830	−2.44949

Table 2: Comparison of f″(0) **when** A=1,M=K=β=Pr=R=Bi=0.0 **for some particular values of** *B*

B	Ref.[51]	Ref.[52]	Present results
0.01	−0.9980	−0.9980	−0.9980
0.1	−0.9694	−0.9694	−0.9694
0.2	−0.9181	−0.9181	−0.9181
0.5	−0.6673	−0.6673	−0.6673
2.0	2.0175	2.0175	2.0175
3.0	4.7293	4.7293	4.7293

Table 3: Comparison of θ′(0) **when** A=1,Pr=10,M=K=β=R=B=0.0 **for various values of** *Bi*

Bi	Ref.[53]	Ref.[54]	Present results
0.05	0.0468	0.04679	0.04679
0.10	0.0879	0.08793	0.08794
0.20	0.1569	0.15690	0.15690
0.40	0.2582	0.25818	0.25817
0.60	0.3289	0.32895	0.32895
0.80	0.3812	0.38119	0.38118
1.0	0.4213	0.42134	0.42134
5.0	0.6356	0.63556	0.63556
10.0	0.6787	0.67872	0.67872
20.0	0.7026	0.70256	0.70255

DISCUSSION ON GRAPHICAL OUTCOMES

Here significance of different control physical parameters of the projected problem on the flow velocity (f′(η)), Skin friction $(Re^{\frac{1}{2}} C_{f_x})$, temperature (θ(η)) and heat transfer $(Re^{-\frac{1}{2}} Nu_x)$ are discussed and presented through graphs.

Figures 2, 3, 4 demonstrated the influences of distinct values of fluid parameters A and B_1, magnetic parameter M, curvature parameter K and ratio of velocities B over velocity gradients. Figure 2a portrays the features of fluid parameter A on the fluid velocity for both cases (M=0, and M=1), while remaining parameters are kept fixed. It is concluded from this graph that a rise in values of A causes boosts up $f'(\eta)$ and momentum boundary layer thickness. Because the higher values of A tend to diminish the viscosity and this overcomes the resistance offered to the liquid. Therefore, boundary layer thickness enhances. It is further remarked that $f'(\eta)$ in the absence of M shows larger value compared to the velocity field in the presence of M. The similar trend was also reported by Hussain et al.45. Figure 2b shows that fluid velocity gradient tends to reduce due to rise in fluid parameter β. It holds physically because β varies inversely with momentum diffusivity, which causes a reduction in velocity gradient. Relatively, the β variation in presence of M shows lesser velocity than the absence of magnetic field. The influence of curvature parameter K over dimensionless velocity field in both cases (M=0, and M=1) is presented in Fig. 3a. Here it is revealed from the plot that both the velocity and thickness of the momentum layer rises for K in the absence of M. In fact K varies inversely with radius of cylinder. Thus larger estimation of K decays the cylinder radius and hence contact zone of the cylinder with fluid diminishes. Hence less resistive force occurs for the fluid and consequently velocity field improves. Behavior of velocity ratio parameter on the dimensionless fluid velocity in the presence/absence of M is sketched in Fig. 3b. Here, $f'(\eta)$ is higher against higher B values due to higher free stream velocity. Furthermore, when U_0 dominates over U_∞, then $f'(\eta)$ diminishes for larger B It is also noted from Fig. 2b that for B=1 there is no boundary layer as the free stream and stretching velocities are equivalent. On the other hand, fluid velocity in case of M=0 diminishes. Similarly, Fig. 4 is prepared to show the behavior of magnetic parameter M with and without fluid parameter β while retaining the remaining parameters fixed on the $f'(\eta)$ against η. It is revealed from Fig. 4 that an increase in the M values causes a rise in both the velocity and thickness of momentum layer. It holds physically that a rise in M causes an increase in Lorentz force, thus $f'(\eta)$ declines. Moreover, the flow field is more influenced with M when β=1.

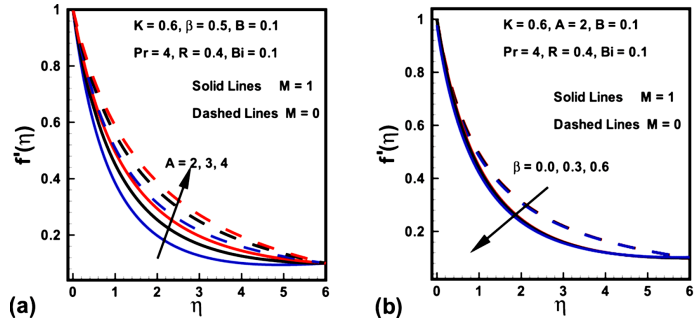


Figure 2: Variations in f′(η) (a) *A* for M=0 and M=1 (b) β for M=0 and M=1.

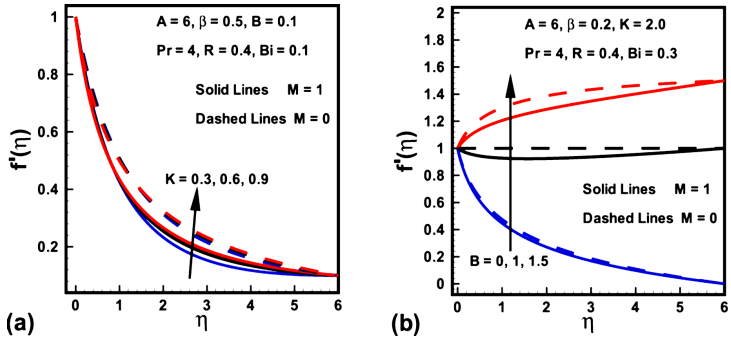


Figure 3: Variations in f′(η) (a) *K* for M=0 and M=1 (b) *B* for M=0 and M=1.

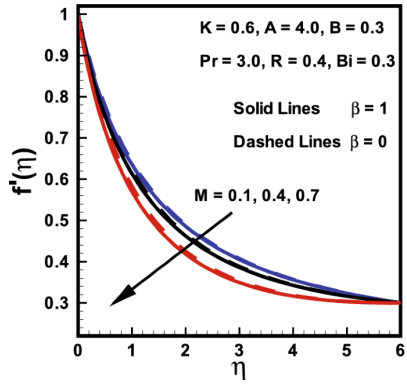


Figure 4: Impact of *M* and β on f′(η).

The effects of radiation parameter *R*, magnetic parameter *M*, Prandtl number *Pr*, curvature parameter *K* and Biot number *Bi*, over dimensionless temperature field are plotted in Figs. 5, 6, 7. Figure 5a is designed to show the behavior of Prandtl number *Pr* on the temperature against η with and without radiation parameter *R*. It is evident that temperature down with

improvement in *Pr*. Because by enhancing *Pr*, the fluid thermal diffusion declines, which accordingly drops the temperature and corresponding thermal layer. Additionally, the temperature field with *R* shows more heat transfer compared to the temperature field without radiation. The significance of Biot number *Bi* over the temperature for both cases (M=0 and M=1) is displayed in Fig. 5b. It is investigated from the plot that temperature and thickness of the related layer are enhancing functions of *M* and *Bi*. Higher values of *Bi* results in higher heat transfer coefficient which consequently boosts the temperature field. The influence of curvature parameter *K* in the presence/absence of magnetic parameter *M* over dimensionless temperature field is witnessed in Fig. 6a. It is clearly analyzed that for higher *K* near the surface thickness of thermal layer declines whereas it rises far away from the surface with *M*. It holds physically that rise in *K* causes an enhance in heat transfer due to which temperature distribution falls adjacent to the surface, on the other hand, it is the reason for rising the ambient temperature distribution. Figure 6b reveals that fluid temperature declines an increment in the ratio of velocities *B*. However, opposite behavior is found for magnetic parameter *M* on fluid temperature (see Fig. 7a). Because Lorentz force rises for higher *M* and consequently more heat is added which gives rise to temperature field. More improvement is observed when radiation parameter *R* is presented. Similarly, Fig. 7b highlight the behavior of fluid temperature against η for radiation parameter *R* in the presence/absence of *M*. It is witnessed from the graph that an increase in *R* causes a boost in the temperature distribution of the flow. This is because a rise in *R* generates the heat energy to the flow, as a result, the thermal layer thicknesses enhances. Also, fluid acquires high temperature in the presence of *M*.

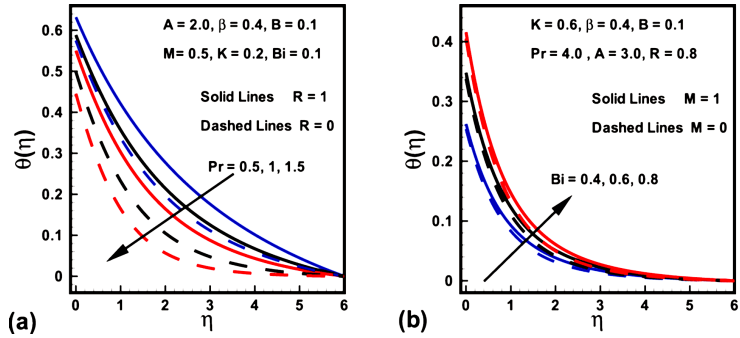


Figure 5: Variations in θ(η) (**a**) *Pr* for R=0 and R=1 (**b**) *Bi* for M=0 and M=1.

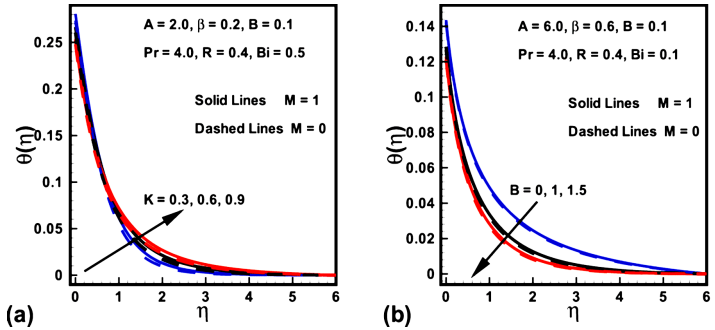


Figure 6: Variations in θ(η) (**a**) *K* for M=0 and M=1 (**b**) *B* for M=0 and M=1.

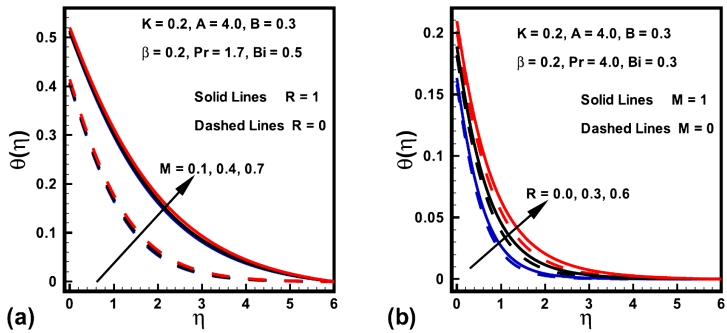


Figure 7: Variations in θ(η) (a) *M* for R=0 and R=1 (b) *R* for M=0 and M=1.

The skin friction coefficient $(Re^{\frac{1}{2}} C_{fx})$ and Nusselt number $(Re^{-\frac{1}{2}} Nu_x)$ variation due to the change in emerging parameters in the presence/absence of *M* are sketched in Figs. 8 to 9. It is perceived from Fig. 8 that the magnitude of the skin friction rises with magnetic parameter *M*. This is because *M* creates an opposing force which diminishes the fluid velocity and consequently, the skin friction rises for larger values of *M*. The results investigated in Fig. 8a shows that, the fluid parameters *A* and β have opposite behavior on the skin friction. Additionally, it is detected from Fig. 8b that as *K* boosts the $Re^{\frac{1}{2}} C_{fx}$ also boosts. Physically, velocity field at the surface of a cylinder is higher compared to that of a flat plate. On the other hand, the magnitude of the skin friction declines with rising values of *B*. Similarly, the behaviors of curvature parameter *K*, Prandtl number *Pr*, radiation parameter *R* and Biot number *Bi* in the presence/absence *M* on Nusselt number are witnessed in Fig. 9. It is revealed from Fig. 9 that the magnitude of heat transfer is higher

in absence of *M*. It is further explained in Fig. 9a that the magnitude of heat transfer is boosted for an increasing values in curvature parameter *K*. It is evidently analyzed that for higher *K* near the surface thickness of thermal boundary layer declines. From this Figure, it is investigated that with rise in *Pr* heat transfer rises. This is because *Pr* declines the fluid temperature which enhances the gap between fluid and surface temperature. Finally, it is revealed from Fig. 9b that the magnitude of heat transfer is higher for larger values of Biot number *Bi* and radiation parameter *R*.

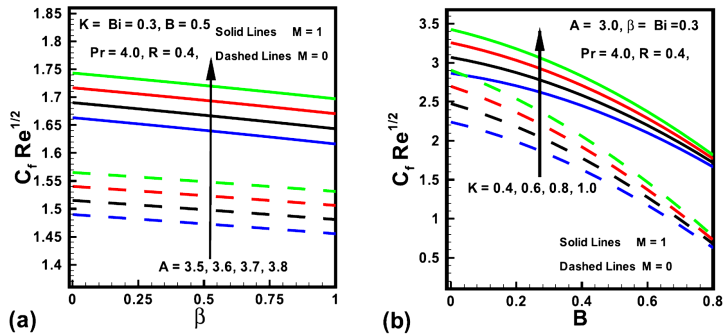


Figure 8: Variations in $Re^{\frac{1}{2}} C_{fx}$ (**a**) *A* against β for M=0 and M=1 (**b**) *K* against *B* for M=0 and M=1.

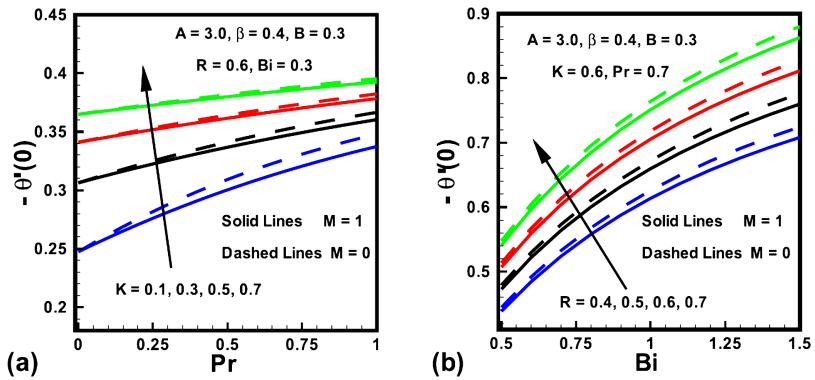


Figure 9: Variations in $Re^{-\frac{1}{2}} Nu_x$ (**a**) *K* against *Pr* for M=0 and M=1 (**b**) *R* against *Bi* for M=0 and M=1.

CONCLUSION

Here the numerical simulation of a 2D stagnation-point flow of MHD Prandtl–Eyring fluid over a stretching cylinder has been inspected. Further, convective boundary condition and radiation effect are also considered in this study. The computations of converted set of non-linear ODEs are performed successfully by Shooting method numerically using Mathematica software 11. The following are some of the significant findings from the present work:

- It is investigated that fluid velocity decays for higher values of magnetic parameter M while the fluid temperature enhances.
- Velocity field improves for fluid parameter A, curvature parameter K and ratio of velocities B; while decreasing function of fluid parameter B.
- Further, it is revealed that dimensionless fluid velocity and related layer thickness are enhancing functions of curvature parameter K, Biot number Bi and radiation parameter R; while decreasing functions of Prandtl number Pr and ratio of velocities B.
- It is concluded that the skin friction boosts by enhancing the fluid parameter A, curvature parameter K and magnetic parameter M.
- The heat transfer rate is boosted for Biot number Bi, radiation parameter R, Prandtl number Pr and curvature parameter K.
- Comparative study shows that current outcomes have better relevance with existing results.

REFERENCES

1. Hayat, T., Ullah, I., Alsaedi, A. & Asghar, S. Magnetohydrodynamics stagnation-point flow of sisko liquid with melting heat transfer and heat generation/absorption. *J. Therm. Sci. Eng. Appl.* **10**(5), 051015–051015 (2018).
2. Besthapu, P., Ul Haq, R., Bandari, S. & Al-Mdallal, Q. M. Thermal radiation and slip effects on MHD stagnation point flow of non-Newtonian nanofluid over a convective stretching surface. *Neural Comput. Appl.* **31**(1), 207–217 (2019).
3. Hiemenz, K. Die grenzschicht an einem in den gleichformigen flussigkeitsstrom eingetauchten geraden kreiszylinder. *Dinglers Polytech. J.* **326**, 321–324 (1911).
4. Ishak, A., Nazar, R. & Pop, I. Mixed convection boundary layers in the stagnation-point flow toward a stretching vertical sheet. *Meccanica* **41**(5), 509–518 (2006).
5. Farooq, M. *et al.* MHD stagnation point flow of viscoelastic nanofluid with non-linear radiation effects. *J. Mol. Liq.* **221**, 1097–1103 (2016).
6. Hayat, T., Khan, M. I., Tamoor, M., Waqas, M. & Alsaedi, A. Numerical simulation of heat transfer in MHD stagnation point flow of cross fluid model towards a stretched surface. *Results Phys.* **7**, 1824–1827 (2017).
7. Hayat, T., Khan, M. I., Waqas, M. & Alsaedi, A. Stagnation point flow of hyperbolic tangent fluid with Soret-Dufour effects. *Results Phys.* **7**, 2711–2717 (2017).
8. Vaidya, H., Prasad, K.V., Vajravelu, K., Wakif, A., Basha, N.Z., Manjunatha, G., & Vishwanatha, U.B. Effects of variable fluid properties on oblique stagnation point flow of a casson nanofluid with convective boundary conditions. In *Defect and Diffusion Forum*, Vol. 401, 183–196 (Trans Tech Publ, 2020).
9. Hayat, T., Ullah, I., Farooq, M. & Alsaedi, A. Analysis of non-linear radiative stagnation point flow of Carreau fluid with homogeneous-heterogeneous reactions. *Microsyst. Technol.* **25**(4), 1243–1250 (2019).
10. Aly, E. H. & Pop, I. MHD flow and heat transfer near stagnation point over a stretching/shrinking surface with partial slip and viscous dissipation: Hybrid nanofluid versus nanofluid. *Powder Technol.* **367**, 192–205 (2020).

11. Waini, I., Ishak, A. & Pop, I. Melting heat transfer of a hybrid nanofluid flow towards a stagnation point region with second-order slip. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, 0954408920961213 (2020).
12. Zhu, Q. Y., Zhuang, Y. J. & Yu, H. Z. Entropy generation due to three-dimensional double-diffusive convection of power-law fluids in heterogeneous porous media. *Int. J. Heat Mass Transf.* **106**, 61–82 (2017).
13. Khan, M. I. & Alzahrani, F. Nonlinear dissipative slip flow of Jeffrey nanomaterial towards a curved surface with entropy generation and activation energy. *Math. Comput. Simul.* **185**, 47–61 (2021).
14. Khan, M. I., Qayyum, S., Hayat, T., Alsaedi, A. & Khan, M. I. Investigation of Sisko fluid through entropy generation. *J. Mol. Liq.* **257**, 155–163 (2018).
15. Amanulla, C. H., Wakif, A., Boulahia, Z., Reddy, M. S. & Nagendra, N. Numerical investigations on magnetic field modeling for Carreau non-Newtonian fluid flow past an isothermal sphere. *J. Braz. Soc. Mech. Sci. Eng.* **40**(9), 1–15 (2018).
16. Khan, M. I. & Alzahrani, F. Binary chemical reaction with activation energy in dissipative flow of non-Newtonian nanomaterial. *J. Theor. Comput. Chem.* **19**(03), 2040006 (2020).
17. Khan, M. I., Qayyum, S., Hayat, T., Khan, M. I. & Alsaedi, A. Entropy optimization in flow of Williamson nanofluid in the presence of chemical reaction and joule heating. *Int. J. Heat Mass Transf.* **133**, 959–967 (2019).
18. Ullah, Z., Zaman, G. & Ishak, A. Magnetohydrodynamic tangent hyperbolic fluid flow past a stretching sheet. *Chin. J. Phys.* **66**, 258–268 (2020).
19. Hayat, T., Aslam, N., Khan, M. I., Khan, M. I. & Alsaedi, A. Physical significance of heat generation/absorption and Soret effects on peristalsis flow of pseudoplastic fluid in an inclined channel. *J. Mol. Liq.* **275**, 599–615 (2019).
20. Abdelsalam, S. I., Mekheimer, Kh. S. & Zaher, A. Z. Alterations in blood stream by electroosmotic forces of hybrid nanofluid through diseased artery: Aneurysmal/stenosed segment. *Chin. J. Phys.* **67**, 314–329 (2020).

21. Hayat, T., Ullah, I., Alsaedi, A. & Farooq, M. MHD flow of Powell-Eyring nanofluid over a non-linear stretching sheet with variable thickness. *Results Phys.* **7**, 189–196 (2017).
22. Eldesoky, I. M., Abdelsalam, S. I., El-Askary, W. A. & Ahmed, M. M. The integrated thermal effect in conjunction with slip conditions on peristaltically induced particle-fluid transport in a catheterized pipe. *J. Porous Media* **23**(7), 695–713 (2020).
23. Amanulla, C. H., Saleem, S., Wakif, A. & AlQarni, M. M. MHD Prandtl fluid flow past an isothermal permeable sphere with slip effects. *Case Stud. Therm. Eng.* **14**, 100447 (2019).
24. Abd Elmaboud, Y. & Abdelsalam, S. I. DC/AC magnetohydrodynamic-micropump of a generalized Burger's fluid in an annulus. *Physica Scripta* **94**(11), 115209 (2019).
25. Bhatti, M. M., Alamri, S. Z., Ellahi, R. & Abdelsalam, S. I. Intra-uterine particle-fluid motion through a compliant asymmetric tapered channel with heat transfer. *J. Therm. Anal. Calorim.* **144**(6), 2259–2267 (2021).
26. Ullah, Z. & Zaman, G. Lie group analysis of magnetohydrodynamic tangent hyperbolic fluid flow towards a stretching sheet with slip conditions. *Heliyon* **3**(11), e00443 (2017).
27. Bhatti, M. M. & Abdelsalam, S. I. Thermodynamic entropy of a magnetized Ree-Eyring particle-fluid motion with irreversibility process: a mathematical paradigm (2021).
28. Khan, M. I., Khan, S. A., Hayat, T., Khan, M. I. & Alsaedi, A. Nanomaterial based flow of Prandtl–Eyring (non-Newtonian) fluid using Brownian and thermophoretic diffusion with entropy generation. *Comput. Methods Programs Biomed.* **180**, 105017 (2019).
29. Akram, J., Akbar, N. S. & Maraj, E. Chemical reaction and heat source/sink effect on magnetonano Prandtl–Eyring fluid peristaltic propulsion in an inclined symmetric channel. *Chin. J. Phys.* **65**, 300–313 (2020).
30. Uddin, I., Ullah, I., Ali, R., Khan, I. & Nisar, K. S. Numerical analysis of nonlinear mixed convective mhd chemically reacting flow of Prandtl–Eyring nanofluids in the presence of activation energy and joule heating. *J. Therm. Anal. Calorim.* **145**(2), 495–505 (2020).
31. Ur Rehman, K., Malik, A. A., Malik, M. Y., Tahir, M. & Zehra, I. On new scaling group of transformation for Prandtl–Eyring fluid model with both heat and mass transfer. *Results Phys.* **8**, 552–558 (2018).

32. Abdelsalam, S. I., Velasco-Hernández, J. X. & Zaher, A. Z. Electromagnetically modulated self-propulsion of swimming sperms via cervical canal. *Biomech. Model. Mechanobiol.* **20**(3), 861–878 (2021).
33. Shankar, U. & Naduvinamani, N. B. Magnetized squeezed flow of time-dependent Prandtl–Eyring fluid past a sensor surface. *Heat Transf.-Asian Res.* **48**(6), 2237–2261 (2019).
34. Smith, J. W. Effect of gas radiation in the boundary layer on aerodynamic heat transfer. *J. Aeronaut. Sci.* **20**(8), 579–580 (1953).
35. Viskanta, R. & Grosh, R. J. Boundary layer in thermal radiation absorbing and emitting media. *Int. J. Heat Mass Transf.* **5**(9), 795–806 (1962).
36. Raza, J., Mebarek-Oudina, F. & Chamkha, A. J. Magnetohydrodynamic flow of molybdenum disulfide nanofluid in a channel with shape effects. *Multidiscip. Model. Mater. Struct.* **15**(4), 737–757 (2019).
37. Gireesha, B. J., Sowmya, G., Khan, M. I. & Öztop, H. F. Flow of hybrid nanofluid across a permeable longitudinal moving fin along with thermal radiation and natural convection. *Comput. Methods Programs Biomed.* **185**, 105166 (2020).
38. Wakif, A. A novel numerical procedure for simulating steady MHD convective flows of radiative Casson fluids over a horizontal stretching sheet with irregular geometry under the combined influence of temperature-dependent viscosity and thermal conductivity. *Math. Probl. Eng.* **2020**, 1675350 (2020).
39. Dogonchi, A. S. & Ganji, D. D. Effect of Cattaneo-Christov heat flux on buoyancy MHD nanofluid flow and heat transfer over a stretching sheet in the presence of joule heating and thermal radiation impacts. *Indian J. Phys.* **92**(6), 757–766 (2018).
40. Khan, M. I. & Alzahrani, F. Free convection and radiation effects in nanofluid (silicon dioxide and molybdenum disulfide) with second order velocity slip, entropy generation, Darcy-Forchheimer porous medium. *Int. J. Hydrogen Energy* **46**(1), 1362–1369 (2021).
41. Raza, R., Mabood, F., Naz, R. & Abdelsalam, S. I. Thermal transport of radiative Williamson fluid over stretchable curved surface. *Therm. Sci. Eng. Prog.* **23**, 100887 (2021).
42. Ullah, I., Hayat, T., Alsaedi, A. & Asghar, S. Dissipative flow of hybrid nanoliquid (H2O-aluminum alloy nanoparticles) with thermal radiation. *Physica Scripta* **94**(12), 125708 (2019).

43. Eldesoky, I. M., Abdelsalam, S. I., El-Askary, W. A., El-Refaey, A. M. & Ahmed, M. M. Joint effect of magnetic field and heat transfer on particulate fluid suspension in a catheterized wavy tube. *BioNanoScience* **9**(3), 723–739 (2019).
44. Abumandour, R. M., Eldesoky, I. M., Kamel, M. H., Ahmed, M. M. & Abdelsalam, S. I. Peristaltic thrusting of a thermal-viscosity nanofluid through a resilient vertical pipe. *Zeitschrift für Naturforschung A* **75**(8), 727–738 (2020).
45. Hussain, A., Malik, M. Y., Awais, M., Salahuddin, T. & Bilal, S. Computational and physical aspects of MHD Prandtl–Eyring fluid flow analysis over a stretching sheet. *Neural Comput. Appl.* **31**(1), 425–433 (2019).
46. Hussain, Z., Hayat, T., Alsaedi, A. & Ullah, I. On MHD convective flow of Williamson fluid with homogeneous-heterogeneous reactions: A comparative study of sheet and cylinder. *Int. Commun. Heat Mass Transf.* **120**, 105060 (2021).
47. Salahuddin, T. *et al.* Analysis of tangent hyperbolic nanofluid impinging on a stretching cylinder near the stagnation point. *Results Phys.* **7**, 426–434 (2017).
48. Hayat, T., Gull, N., Farooq, M. & Ahmad, B. Thermal radiation effect in MHD flow of Powell-Eyring nanofluid induced by a stretching cylinder. *J. Aerospace Eng.* **29**(1), 04015011 (2016).
49. Akbar, N. S., Ebaid, A. & Khan, Z. H. Numerical analysis of magnetic field effects on Eyring-Powell fluid flow towards a stretching sheet. *J. Magn. Magn. Mater.* **382**, 355–358 (2015).
50. Khan, I., Hussain, A., Malik, M. Y. & Mukhtar, S. On magnetohydrodynamics Prandtl fluid flow in the presence of stratification and heat generation. *Phys. Stat. Mech. Appl.* **540**, 123008 (2020).
51. Kumar, R. V. M. S. S. K., Kumar, G. V., Raju, C. S. K., Shehzad, S. A. & Varma, S. V. K. Analysis of Arrhenius activation energy in magnetohydrodynamic Carreau fluid flow through improved theory of heat diffusion and binary chemical reaction. *J. Phys. Commun.* **2**(3), 035004 (2018).
52. Khan, M. & Alshomrani, A. S. Mhd stagnation-point flow of a Carreau fluid and heat transfer in the presence of convective boundary conditions. *PLoS ONE* **11**(6), e0157180 (2016).

53. Aziz, A. A similarity solution for laminar thermal boundary layer over a flat plate with a convective surface boundary condition. *Commun. Nonlinear Sci. Numer. Simul.* **14**(4), 1064–1068 (2009).

54. Uddin, Md. J., Khan, W. A. & Ismail, A. IMd. MHD forced convective laminar boundary layer flow from a convectively heated moving vertical plate with radiation and transpiration effect. *PLoS ONE* **8**(5), e62664 (2013).

Chapter 7

A REVERSE LOGISTICS CHAIN MATHEMATICAL MODEL FOR A SUSTAINABLE PRODUCTION SYSTEM OF PERISHABLE GOODS BASED ON DEMAND OPTIMIZATION

Saeed Tavakkoli Moghaddam[1] , Mehrdad Javadi[2] , Seyyed Mohammad Hadji Molana[3]

[1] Young Researchers and Elites Club, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2] Department of Mechanical Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran
[3] Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

ABSTRACT

Sustainability in the supply chain means pushing the supply chain to focus on social, economic and environmental aspects, and addressing the existing

Citation: (APA): Tavakkoli Moghaddam, S., Javadi, M., & Hadji Molana, S. M. (2019). A reverse logistics chain mathematical model for a sustainable production system of perishable goods based on demand optimization. Journal of Industrial Engineering International, 15(4), 709-721. .(13 pages).

problems in the traditional supply chain. Considering the importance of evaluating supply chain networks, especially in the field of perishable commodities, this paper aimed to design a mathematical model for the reverse supply chain of perishable goods, taking into account the sustainable production system. In this research, four objective functions were considered to maximize profitability and the level of satisfaction with the use of technology, minimize costs and measure environmental impacts. The results of the implementation of the proposed model for a manufacturing company show that objective functions are sensitive to demand, so the change in demand changes the objective functions, in particular the profitability function.

INTRODUCTION

Reduction in raw materials, increase in pollutants and the extent of pollution caused by them have been important issues for organizations in recent decades. In addition, failure to observe ethical responsibilities will lead to increased costs and thus reduced profitability. Sustainable supply chain management is rooted in sustainability and includes an extensive approach to supply chain management. Sustainability in the supply chain means pushing the supply chain to focus on social, economic and environmental aspects, and addressing the existing problems in the traditional supply chain. Sustainable supply chain includes all logistics costs from an economic point of view, reducing the amount of contaminants released from an environmental point of view, and reviewing social responsibility from a social point of view.

The supply chain for perishable products includes products with a durable shelf life and limited production, the management of which requires making right decisions (Katsaliaki et al. 2014). Rapid food spoilage leads to a loss in the volume of many foods and more pressure on FSCs; it also reduces the quality, profitability and sustainability of food. Some of the food losses that occur after harvesting and in the supply chain transportation are inevitable. According to FAO reports, 20–60% of the total production in all countries and one-third of food products for human consumption in the world (about 1.3 billion tons per year) are lost after harvesting.

With an increase in food demand in the world, production is one of the ways to meet these needs; in addition, reducing waste in each stage of the food chain can be an option for productivity when increasing production. Many cases in manufacturing operations can be effective in causing waste, most of which, according to Lemma et al., are inefficiency in production,

storage and transportation. In addition, inappropriate planning and supply chain management practices are the main operational reasons for wastes in different countries (Lemma et al. 2014).

At the strategic level, the key issue is the design or reengineering of the supply chain network, which addresses the location and evaluation of facilities and the flow of materials through the network. In the meantime, supply chain management is seeking to achieve goals such as effective economic competitions, time and quality of service, specifically in the economic environment characterized by the globalization of transactions and acceleration of industrial cycles (Eskandarpour et al. 2015). Coordination, integration and management of business processes in the supply chain will determine the competitive success of food companies. Sustainable food supply chain management includes procurement of raw materials, production and distribution, and processes for collecting used or unused products, to ensure social, economic and environmental sustainability (Bloemhof and Soysal 2017).

According to Bloemhof and Soysal, about 40% of food waste is related to supply chain activities, such as transportation that requires specific conditions and storage, management and packaging of perishable products. So, supply chain sustainability means improving the mix of various and sometimes contradictory factors, and how to combine economic, social and environmental indicators (Bloemhof and Soysal 2017).

Due to competition, changes in customer demand and legal issues, corporate executives need to focus on aspects of the sustainability of value creation, including a new set of challenges in decision making. Companies are trying to develop products with a certain quality and minimal cost. Today, the environmental and social performance of products beyond the entire life cycle of the product should be taken into account. From an environmental point of view, product design should lead to products that are characterized by reducing the severity of harmful substances, less input of toxic materials, decomposition, durability, ease of recovery, and less energy consumption and life span. Stindt argued that supply chain design is a mutual planning issue that involves all processes of the value chain of the core company with interfaces with the supplier and consumer that shows the resources and flow of materials (Stindt 2017). Recently, more companies have turned to using sustainable proactive strategies and management operations of an evolving sustainable supply chain. In the meantime, researchers considered closed loop supply chain (CLSC) as one of the most important factors for achieving

sustainable operations. In the modern world of business, focus is not only on reducing costs and increasing profits, but also on achieving sustainability and creating a balance between social responsibility, environmental protection and economic prosperity; these factors result in sustainability (Sgarbossa and Russo 2017).

In the present paper, considering factors such as reverse chain trend, sustainability and environment, perishable goods logistics, the use of different vehicles with certain speeds, and determining the details of supplier and retailers, attention has been paid to the productivity increase throughout the chain from beginning to the end, reduction of environmental damages and intra-chain productivity by four different objective functions: (1) minimization of supply chain design cost; (2) measuring the overall environmental impact over the network; (3) maximization of the profitability of the chain according to the product's novelty; (4) maximization of the level of satisfaction of using technology. The research also created a potential for measuring the performance and predicting the process by creating the objective function of satisfaction of the use of technology.

LITERATURE REVIEW

The supply chain of products and services, especially when it comes to highly perishable products needing high level of services, is usually difficult to handle. In this case, simulation can offer a reliable approach toward studying and evaluating the processes and outcomes of such supply chains, and presenting suitable alternatives that can achieve optimal performance. Spoilage is a common phenomenon. Products may lose their value or quality suddenly or gradually. Fruits, vegetables, flowers, medicines, blood, dairy products, meat and food are prominent examples. Spoilage is the main concern of the supply chain, because the quality or value of most products is reduced over the life span. Spoilage is a nonlinear function that affects many factors, such as transportation types (Sazvar et al. 2016). Integrating objectives includes dimensions of sustainability, economic, environmental and social development, which are derived from the needs of stakeholders and customers (Galal and Moneim 2016).

Katsaliaki et al. have provided a game-based approach to facilitate decision-making on perishable products (Katsaliaki et al. 2014).

Researchers have used different methods to optimize the food supply chain and support the decision-making process, with some aiming to model food management and productivity, focusing on minimizing food waste

along the supply chain. Food supply chains (FSCs) can be considered as a component of variable supply chain due to continuous and significant changes in the quality of food products throughout the supply chain until the final consumption. Also, due to the product's perishable nature, high volatility in demand and price, and increasing consumer concerns for food safety, FSC is a more complex chain that is tied to environmental conditions, as compared to other supply chains. To reduce food waste, proper study and performance is needed to improve the entire supply chain. Many approaches have been taken by researchers and practitioners to reduce food wastage. However, some studies were made by two-echelon inventory system for perishable items in supply chain (AriaNezhad et al. 2013). At the governmental level, many countries have taken various approaches to reduce waste. For example, at the production stage, the government supports farmers to improve the availability of agricultural development services and to improve harvesting techniques. In addition, improvement of the availability for storage, process improvement and packaging techniques, consumer education campaigns, etc., are used in a variety of areas.

In most researches, LP methods have been used to improve the supply chain. In addition, some recent researches have used advanced optimization techniques, such as an evolutionary optimization approach. This suggests that advanced modeling methods are at maturity stage and require further studies on perishable food products.

Lemma have considered production, transportation and inventory as the main causes of waste generation, which have high impacts on this stage of the supply chain. Lots of wastes are generated throughout the supply chain; however, little attention is paid to minimizing food waste (Lemma et al. 2014).

When the market is disturbed, that is, expected demand or variance varies from one period to another according to a probability principle, there is typically less likelihood of sustaining long-term partnerships in a booming market or a market with low demand variations. Further information on future fluctuations may not help the supply chain to sustain long-term partnerships due to strategic considerations of the partners. With availability of the market signal, the overall supply chain profit will increase, but the profitability of the retailer may be even worse (Sun and Debo 2014).

Some of the challenges in sustainable supply chain management are more important to be analyzed. In the same vein, companies' survival not only needs to make the sustainability issues involved in the plan, but

also has to consider their strategic impact. Therefore, appropriate goals or functional indicators must be defined to achieve an appropriate decision-making process. Uncertainty is a factor that can deteriorate each model in supporting the decision making and reduce the importance of scheduled goals for such models. One of the origins of such uncertainty is the forecast mistakes that affect small and medium enterprises, especially in the food supply chain (Li et al. 2014).

To assess the sustainability of the Greek dairy chain and the performance of individuals, Bourlakis et al. did an analysis using key indicators related to the efficiency, flexibility, accountability and product quality. The importance of these indicators has been assessed based on the perceptions of the key members of this chain. They did a comparative analysis in terms of sustainable performance indicators on the members of the Greek dairy chain. This analysis depicts many of the major functions of the members of this chain. In particular, there are significant differences regarding the cost of raw materials, production, operations, storage costs, delivery and distribution, flexibility in delivery to an alternative point of sale, the time of product protection, and the quality of the product packaging (Bourlakis et al. 2014). In general, a good supply chain performance needs awareness of customer needs and information changes (Fedrigotti and Fischer 2015). On the other hand, selection of sustainable suppliers in supply chain has a great importance (Ghoushchi et al. 2018).

Supply chain network design (SCND) models and methods have been the subject of many recent studies. Eskandarpour et al. analyzed 87 articles in the design of sustainable supply chain network covering mathematical models that include economic factors, as well as social or environmental dimensions. Nagurney has designed sustainable supply chain design for sustainable cities. The supply chain provides the necessary infrastructure for the production and distribution of products and services in the network economy and serves as a channel for manufacturing, transporting and consuming a range of products from food, clothing, automotive and high-tech products, even to health care products. Cities, as mainstream population centers, serve as major demand points, distribution centers and large storage facilities, transport providers and even manufacturers. For sustainable supply chains, focusing on sustainable cities, we can use a precise mathematical modeling with computational framework (Nagurney 2015). In another place, Galal and Moneim addressed the development of sustainable supply chain in developing countries and used AHP and other indicators to arrive at the final solution. They believe that supply chain sustainability is achieved by

taking into account the economic, environmental and social aspects of the decision-making process. In developing countries where supply chains are often labor-intensive and environmental laws are still at their infancy, both environmental and social aspects must be taken into account. To achieve sustainability objectives, there is need for cooperation between supply chain managers. To maintain its position and role in the supply chain, each member must comply with environmental and social objectives. Competition must also be achieved through the fulfillment of customer requirements and economic aspects. It should be considered that the failure of one stage or player in the supply chain affects the performance of the entire supply chain and its competitions. The supply chain is considered as a system of individuals whose performance identifies the overall stability of the supply chain (Galal and Moneim 2016). There has been growing concern about the environmental and social effects of commercial operations in the last decade. The sustainability of the supply chain has attracted the attention of the academy and industry at the same time, taking into account the economic, environmental and social values. The issues of timely delivery and disposal of spoilt products are very worrying, especially for perishable and seasonal products such as the fresh crops. Sazvar introduced a multi-objective multi-supplier supply chain with perishable products in which a multi-objective linear mathematical method is used. Some variables, such as final consumer demand, the proportion of delayed orders and the rate of corruption are uncertain. The model of this paper simultaneously considers the economic and environmental objectives of perishable supply chains, emphasizing the details of the social aspects of the specific applications of the flower-picking industry. Integration of environmental and social aspects with economic considerations, which comprise the three dimensions of organizational sustainability, has gained importance in management decisions in supply chain management. As compared to the old SCM, typically, the focus is on financial and economic business performance, sustainable SCM with explicit integration of environmental or social objectives for the expansion of economic dimensions (Sazvar et al. 2016; Zhang et al. 2016).

Because the wastes in the emerging markets are high from harvesting to consumption in the supply chain of perishable food, such as fruit and vegetables, Balaji and Arshinder also analyzed the causes of waste generation in the supply chain of unsustainable foods. This study was conducted to identify the causes of food waste and their interdependencies and to analyze the interactions among them. This paper presents a fuzzy MICMAC and total interpretive structural modeling (TISM) (Balaji and

Arshinder 2016). Fresh fruits and vegetables (FFV) are among the most important components of the retail chain and serve as a strategic product in attracting customers. The demand for fresh fruits is growing year by year. There is also a higher potential for the future. Food products come from a farmer's land to the end customer through a long chain of intermediaries like farmers, cooperatives, wholesalers, retailers, commissioners, which can cause a lot of waste (Agarwal 2017).

In order to design a sustainable supply chain based on post-harvest losses and harvest timing equilibrium, a sustainable two-way optimization model was presented by An and Ouyang in which a food company maximizes its profit and minimizes the post-harvest waste by expanding process facilities and purchases, price determination, a group of non-cooperative early distributor farmers, harvesting time, transportation, storage and market decisions which have been considered as product uncertainty and market equilibrium (An and Ouyang 2016).

Given the evolution of the agricultural sector and the new challenges facing it, effective management of agricultural supply chains is an attractive topic for research. Therefore, uncertainty management in the supply chain for the agricultural crops is important in researches on the latest advances in operational research methods to manage the uncertainty that occurs in supply chain management issues (Borodin et al. 2016).

In another study, in order to achieve multi-objective optimization for the design of a sustainable supply chain network with respect to distribution channels, a new method for designing SCN with multiple distribution channels (MDCSCN) was presented. By providing direct products and services to customers by available facilities, as a substitution for the conventional products and services, this model benefits them. Sustainable objectives, such as reducing economic costs, increasing customer coverage, and mitigating environmental impacts, contribute to MDCSN design. A multi-objective artificial bee colony (MOABC) Algorithm for solving the MDCSCN model, which integrates the priority paradigm coding mechanism, Pareto optimization and the swarm intelligence of the bee colony, was provided. The concept of sustainable development would be taken into consideration when it can reduce economic costs for chain companies, increase the flexibility of customer orders and reduce environmental impacts (Zhang et al. 2016).

Because of competition, customer pressure and legal issues, corporate executives need to focus, during decision making, on aspects of sustainability

of value creation, including a new set of challenges. Companies are striving to develop products of a specific quality with minimal cost. Today, the environmental and social performance of products beyond its entire life cycle should be taken into account. From an environmental point of view, product design should result in products that are characterized by reduced material severity, lower input of toxic materials, biodegradability, durability, ease of recovery and lower energy consumption during the life cycle. Supply chain design is a mutual planning issue that involves all value chain processes of the core company with interfaces for supplier–consumer that illustrate the resources and flow of materials (Stindt 2017).

To optimize the fresh food logistics, an optimization model was proposed with three types of decision-making in gardening, which deal with the purchase, transportation and storage of fresh produce (Soto-Silva et al. 2017). The management of unsophisticated food in retail stores is very difficult due to the short life span of products and their spoilage. Many elements, such as price, shelf space allocation and quality that can affect the rate of consumption, should be considered when designing step for the retail chain perishable food. Xiao and Yang designed a retail chain for perishable foods and provided a mathematical model for a single-item retail chain, and determined the pricing strategy, shelf space allocation, and quantity assignment to maximize the overall profitability of the retailer with the use of tracer technologies (Xiao and Yang 2016).

In the contemporary business world, focus is not only on reducing costs and increasing profits, but also on achieving sustainability and balance between social responsibility, environmental protection and economic prosperity. These factors lead to sustainability; therefore, a preventive model in the food supply chain can be useful (Sgarbossa and Russo 2017).

In recent years, food safety incidents have occurred in many countries, and issues related to the quality of food and safety have become more socially appealing. Due to the concern about the quality sustainability of the food supply chain, many companies have developed a real-time data mining system to ensure the quality of the products in the supply chain. For food safety and quality issues, the food chain precautionary system helps in the analysis of the food safety risk and minimization of the production and distribution of poor quality or non-safe products. Precaution also helps in improving the quality of food due to ensuring the sustainability of the supply chain quality. Therefore, Wang and Yue introduced a data mining food safety precautionary system for a sustainable supply chain (Wang and

Yue 2017). Other aspects of deteriorating items have been studied by several researchers (Singh et al. 2017; Sundara Rajan and Uthayakumar 2017; Uthayakumar and Tharani 2017).

PROBLEM STATEMENT AND MATHEMATICAL FORMULATION

Considering the problem statement, the assumption considered in the design process of the mathematical model as well as the proposed model solution is as follows:

- The number of retailers is known.
- The demand for retailer l for the period p specified with d_{lp} is a specific variable, and retailers' demands are independent of each other.
- There are different vehicles with different capacities that should be considered.
- Every retailer/open top distribution center is visited at a maximum of once per period.
- Soft time windows are included.
- There is more than one vehicle for each route.
- If a retailer or open top distribution center needs service, there should be more than one vehicle for servicing.
- The time period is considered as 1 day.
- The capacity of manufacturers and distribution centers is limited.
- At all stages, vehicles are available from the morning and the maximum availability time for each vehicle is less than or equivalent to working time per day.
- Distribution centers meet retailers' demand, and manufacturers can meet the orders of distribution centers.
- Retailers and distribution centers can order more than they need (they also have the permission for storage).
- One type of product is considered.
- In retail and distribution centers, no return can be made.
- The time and cost of dispatching the vehicle are known.
- Travel cost and unit distance are specified.

- The cost of maintenance is known.
- The service time is specified for each retailer.
- The speed of the vehicle is known.
- Products should be ordered in such a way that none expires in the warehouse.
- The first round of work should start and end at the same open top production unit.
- The second round should start and end at the same center of the open top distribution center.
- Manufacturers cannot directly sell products to retailers.

Before dealing with the mathematical model, the sets, parameters and decision variables are described prior to the mathematical model.

Sets

- K: A set of various types of vehicles.
- M_1: Set of type 1 vehicles.
- M_k: Set of type *K* vehicles.
- Tech: Set of manufacturing technologies.
- M: Set of potential producers.
- D: Set of potential distribution centers.
- L: Set of retailers.
- P: Set of time intervals.
- N_1: Set of nodes including $\{M \cup D\}$
- N_2: Set of nodes including $\{D \cup L\}$

Parameters

- c_{ij}: The average cost of traveling from nodes *i* to *j*.
- OC_d: The cost of opening the distribution center *d*.
- OC_m: The cost of opening the manufacturing unit *m*.
- S_{cme}: The cost of technology deployment that must be built in the production center *m*.
- d_{lp}: Customer demand for retailer *l* over time interval *p*.
- VC_{dp}: Variable cost for maintaining a product at the distribution center *l* in the time interval *p*.

- VC_{mep}: The cost of producing each unit at the production center *d* with technology at the time interval *p*.
- FVF_k: The fixed cost of every vehicle launched in the first round for a *K*-type vehicle.
- FVS_k: The fixed cost of every vehicle launched in the second round for a *K*-type vehicle.
- EO_{me}: Environmental effects of the outdoor production unit *m* with the technology *e*.
- EO_d: Environmental impacts of open top distribution center *d*.
- VE_{dp}: Environmental impacts of preserving each unit in the open top distribution center *d* in the time interval *p*.
- VE_{mep}: Environmental impacts of manufacturing in each unit in the production unit *m* with the technology level *e* in the time interval *p*.
- ET_{ij}: Average transfer of environmental effects from node *i* to node *j*.
- D_{Max}: Maximum desirable number of distribution centers.
- M_{Max}: Maximum desirable number of producers.
- Q_k: Capacity of vehicle type *k*.
- $q_{ip}^{m_k k}$: Delivery time specified for vehicle type *k*.
- Cap_d: Storage capacity of distribution center *d*.
- Cap_{me}: Producer Capacity *m* for production with technology *e*.
- Id_p: The amount of product stored at the distribution center *d* as inventory at time interval *p*.
- I_{lp}: The amount of product stored in retail *l* as inventory at time interval *p*.
- ζ: A confidence coefficient that allows distribution centers to store a percentage of their previous period delivered to retailers.
- τ_{max}: Maximum continuous period to keep a perishable foodstuff.
- $D_{m;kp}$: Distance of movement of vehicle M_k type *K* in time interval *p*.
- WT: Working time per day.
- t: Time index.

- $RT_{ip}^{m_k k}$: A re-run time for a vehicle M_k type K for node i in the time interval p.
- dis_{ij}: The distance between the nodes i and j.
- $S_{m;k}$: Average speed of type K vehicle M_k.
- $ST_{ip}^{m_k k}$: The delivery distance assigned by a k type vehicle M_k for a node i in the time interval p.
- ud_{ip}: Time to enter distribution center i in time interval p.
- ed_{ip}: The earliest entry time for the time window for the distribution center i at the time interval p.
- ld_{ip}: The most delayed entry time for the time window for the distribution center i at the time interval p.
- pd_{ep}: Cost of waiting penalty or waiting time unit for Distribution Centers i at time interval p.
- pd_{lp}: Latency penalty fee or the delayed arrival time for distribution centers i at the time interval p.
- $pd_{ip}(ud_{ip})$: Time window deviation for distribution center i in time interval p.
- HC_{dp}: The earliest entry time for the time window for retail i in the period p.
- HC_{lp}: The most delayed entry time for the time window for retail i in the period p.
- pb_e: The cost of a waiting time penalty or time unit for retailer i at the time interval p.
- α_e: The latency penalty fee or the delayed arrival time for retailer i during the period p.
- q: Level of freshness.
- A_{ip}^{q}: The level of freshness of the products in the retailer i at the time interval p.
- B_{ip}: The quality of retail product i during the period p.
- pd_{dp}: The latency penalty fee or the delayed arrival time for the manufacturing unit i during the period p.
- ud_{dp}: Time to enter the manufacturing unit d in time interval p.

- pr_{lp}: The latency penalty fee or the delayed arrival time for the distribution center *i* during the period *p*.
- ur_{lp}: Time to enter the distribution center *l* in the time interval *p*.
- ur_{jp}: Time to enter the retail *l* in the time interval *p*.
- pd_{ed}: The latency penalty fee or the delayed arrival time for the manufacturing unit *i* with the technology *e*.
- pr_{ip}: The latency penalty fee or the delayed arrival time for retailers *i* during the period *p*.
- pr_{ep}: The latency penalty fee with the technology *e* in the time interval *p*.
- l': Undefined retailers.
- N': Prohibition of circulation subsets.
- $g_{jp}^{m_k k}$: The working time for a *K* type vehicle M_k for the node *j* in the time interval *p*.

Decision Variables

- $r_{dlp}^{m_k k}$: If the vehicle M_k type *K*, within the time interval *p*, travels the distance between manufacturers and distributors, otherwise 0.
- $r_{ijp}^{m_k k}$: If the vehicle M_k type *K*, within the time interval *p*, travels the distance $^{\text{arc}(i,j),\ N_2}$, otherwise 0.
- $r_{lip}^{m_k k}$: If the vehicle M_k type *K*, within the time interval *p*, travels the distance arc(*i*, *j*) from the retailer *1*, N_2, otherwise 0.
- $g_{lp}^{m_k k}$: If the vehicle M_k type *K*, within the time interval *p*, meets the retailer *1*. Otherwise 0.
- β_{dlp}: If the distribution center *d* services the retailer *l* within the time interval *p*. Otherwise 0.
- y_d: If the distribution center *d* is opened. Otherwise 0.
- z_{me}: If the manufacturing unit *m* with technology *e*, is opened. Otherwise 0.
- $x_{mdp}^{m_k k}$: If the vehicle M_k type *K*, within the time interval *p*, travels the distance between manufacturers and distributors in arc(*I*, *j*), otherwise 0.

- $x_{ijp}^{m_k k}$: If the vehicle M_k type K, within the time interval *p*, travels the distance arc(I, j) N_2, otherwise 0.
- $q_{dp}^{m_k k}$: If the vehicle M_k type *K*, within the time interval *p*, meets the distribution center *d*, otherwise 0.
- $\eta_{lp}^{m_k k}$: The amount of products delivered to the retailer *1* with the vehicle M_k type *K* within the time interval *p*.
- $\delta_{dp}^{m_k k}$: The amount of products delivered to the distribution center *d* with the vehicle M_k type *K* within the time interval *p*.
- h_{mep}: The amount of product produced in the manufacturing unit *m* with technology *t* within the time interval *p*.

Mathematical Modeling

In this section, the four objectives of the research problem were first discussed; then, the constraints were introduced suitable to the problem.

$$\begin{aligned}\text{Min } F_1 = & \sum_{m\in M} \text{OC}_m \sum_{m\in M} z_{me} + \sum_{m\in M}\sum_{e\in Tech} \text{SC}_{me} z_{me} + \sum_{d\in D} \text{OC}_d y_d \\ & + \sum_{p\in P}\left(\sum_{k\in K}\sum_{m_k\in M_K}\sum_{i,j\in N_1} c_{ij} x_{ijp}^{m_k k} + \sum_{k\in K}\sum_{m_k\in M_k}\sum_{i,j\in N_2} c_{ij} r_{ijp}^{m_K K}\right. \\ & + \sum_{m\in Me}\sum_{\in \text{Tech}} \text{VC}_{mep} h_{mep} + \sum_{d\in D} \text{VC}_{dp}\left(\sum_{l\in L}\beta_{dlp}\left(\sum_{k\in K}\sum_{m_k\in M_K}\eta_{lp}^{m_k k}\right)\right) \\ & + \sum_{m\in M}\sum_{d\in D}\sum_{k\in K}\sum_{m_k\in M_K} \text{FVF}_k x_{mdp}^{m_k k} + \sum_{d\in D}\sum_{l\in L}\sum_{k\in K}\sum_{m_k\in M_K} \text{FVS}_k r_{dlp}^{m_k k} \\ & \left. + \sum_{d\in D} \text{Pd}_{dp}(ud_{dp}) + \sum_{l\in L} \text{Pr}_{lp}(ur_{lp}) + \sum_{d\in D} \text{HC}_{dp} I_{dp} + \sum_{l\in L} \text{HC}_{lp} I_{LP}\right)\end{aligned} \tag{1}$$

The 1 objective function reduces the overall variable and fixed costs of supply chain design. The first part is the fixed cost of opening a production unit, and the second is the fixed cost of association with the consolidation and learning of technology. The third part is about the fixed cost of opening distribution centers. It is important to know that the above corrections are related to the first stage of a two-stage model that includes decisions that need to be made before identifying the demands and vehicle routes in different periods or the fixed costs of the opening. The remaining parts are related to the second stage. They show variable costs, and these decisions are made after demands have been periodically determined. Parts four and five are transportation costs for the first and second periods. The sixth and seventh

sections represent variable costs in manufacturing units and distribution centers. The next two parts are the fixed costs of each round of the first and second periods. The next two are the fine of distortion of the time window and the final two parts of the cost of inventory of distribution centers and retailers.

$$\begin{aligned}\text{Min } F_2 = & \sum_{m\in M}\sum_{e\in \text{Tech}} \text{EO}_{me} z_{me} + \sum_{d\in D} \text{EO}_d y_d \\ & + \sum_{p\in P}\left(\sum_{k\in K}\sum_{m_k\in M_k}\sum_{i,j\in N_1} \text{ET}_{ij} x_{ijp}^{m_k k} + \sum_{k\in K}\sum_{m_k\in M_K}\sum_{i,j\in N_2} \text{ET}_{ij} r_{ijp}^{m_K K}\right. \\ & \left. + \sum_{d\in D} \text{VE}_{dp}\left(\sum_{l\in L}\beta_{dlp}\left(\sum_{k\in K}\sum_{m_k\in M_K}\eta_{lp}^{m_k k}\right)\right) + \sum_{m\in M}\sum_{e\in \text{Tech}} \text{VE}_{mep} h_{mep}\right)\end{aligned} \tag{2}$$

The 2 objective function measures the overall environmental impact over the network. The first two parts are the environmental impacts related to the opening services of manufacturing units and distribution centers. The next two in the second phase are the environmental impacts associated with the marine transportation of products from production units to distribution centers in the first round and from distribution centers to retailers in the second. Finally, the two final sums are variable environmental impacts that arise from executive activities in production and distribution centers. All variables are described in constraints (35) to (44).

$$\begin{aligned}\text{Max } F_3 = & \sum d_{lp} \cdot A_{ip}^{q} \\ & - \sum\sum\sum\sum \left[\left(B_{ip} \cdot \text{VC}_{mep}\right) + \left(\delta_{dp}^{m_k} \cdot S_{cme}\right)\right]\end{aligned} \tag{3}$$

$$\begin{aligned}\text{Max } F_4 = & \sum_{s} pb_e \left[\alpha\left(\sum_{m}\sum_{e} \text{EO}_{me}\right)\right. \\ & \left. + (1-\alpha)\left(\sum_{c}\sum_{m}\sum_{e} S_{cme}\right)\right]\end{aligned} \tag{4}$$

The 3 objective function indicates the maximum profitability of the supply chain according to the freshness of the products. This function consists of two parts; the first part expresses the demand based on the products' freshness, and the second part expresses the cost of production (constant and variable) based on product quality. The 4 objective function also indicates the maximum level of satisfaction with the use of technology due to its use in the production process based on the amount of pollutants and the construction costs.

Constraints

$$\sum_{i\in N_2}\sum_{k\in K}\sum_{m_k\in M_K} r_{ilp}^{m_k k} \le 1 \quad \forall l \in L,\ p \in P \tag{5}$$

$$\sum_{i\in N_2} r_{ilp}^{m_k k} = \sum_{i\in N_2} r_{lip}^{m_k k} = g_{lp}^{m_k k} \quad \forall m_k \in M_K,\ k \in K,\ l \in L,\ p \in P \tag{6}$$

$$\sum_{d\in D}\sum_{l\in L} r_{dlp}^{m_k k} \le 1 \quad \forall m_k \in M_K,\ k \in K,\ p \in P \tag{7}$$

$$\sum_{l\in L}\sum_{i\in N_2} d_{lp} r_{dlp}^{m_k k} \le Q_k \quad \forall m_k \in M_K,\ k \in K,\ p \in P \tag{8}$$

$$\sum_{i\in L'}\sum_{j\in L'} r_{dlp}^{m_k k} \le |L'| - 1 \quad \forall m_k \in M_K,\ k \in K,\ L' \subseteq L,\ |L'| \ge 2,\ p \in P \tag{9}$$

$$\sum_{d\in D} \beta_{dlp} \le 1 \quad \forall l \in L,\ p \in P \tag{10}$$

$$\sum_{l\in L} d_{lp}\beta_{dlp} \le \mathrm{Cap}_d y_d \quad \forall d \in D,\ p \in P \tag{11}$$

$$\sum_{l\in L} r_{dlp}^{m_k k} + \sum_{i\in N_2} r_{lip}^{m_k k} - \beta_{dlp} \le 1 \quad \forall d \in D,\ l \in L,\ m_k \in M_K,\ k \in K,\ p \in P \tag{12}$$

$$\eta_{lp}^{m_k k} \le Q_k \times \sum_{i\in N_2} r_{dlp}^{m_k k} \quad \forall l \in L,\ m_k \in M_K,\ k \in K,\ p \in P \tag{13}$$

$$I_{lp-1} + \sum_{k\in K}\sum_{m_k\in M_K} \eta_{dlp}^{m_k k} = d_{lp} + I_{lp} \quad \forall l \in L,\ p \in P \tag{14}$$

$$\sum_{l\in L}\sum_{k\in K}\sum_{m_k\in M_K} r_{dlp}^{m_k k} = \sum_{i\in N_1}\sum_{k\in K}\sum_{m_k\in M_K} x_{idp}^{m_k k} = y_d \quad \forall d \in D,\ p \in P \tag{15}$$

$$\sum_{d\in D}\sum_{k\in K}\sum_{m_k\in M_K} x_{dlp}^{m_k k} \le \sum_{e\in \mathrm{Tech}} z_{me} \quad \forall m \in M,\ p \in P \tag{16}$$

$$\sum_{d\in D} y_d \le D_{\mathrm{Max}} \tag{17}$$

$$\sum_{m\in M}\sum_{e\in \text{Tech}} z_{me} \le M_{\text{Max}} \tag{18}$$

$$\sum_{j\in N_1} x_{ijp}^{m_k k} = \sum_{j\in N_1} x_{jlp}^{m_k k} = q_{ip}^{m_k k} \quad \forall i \in N_1,\ m_k \in M_K,\ k \in K,\ p \in P \tag{19}$$

$$\sum_{i\in N_1}\sum_{j\in N_1} x_{ijp}^{m_k k} \le |N'| - 1 \quad \forall m_k \in M_K,\ k \in K,\ \left|N_1'\right| \subseteq N_1' \ge 2, p \in P \tag{20}$$

$$\sum_{d\in D} \delta_{dp}^{m_k k} \le Q_k \quad \forall m_k \in M_K,\ k \in K,\ p \in P \tag{21}$$

$$\sum_{d\in D} \delta_{dp}^{m_k k} \le Q_k \times \sum_{d\in D}\sum_{i\in N_1} x_{idp}^{m_k k} \quad \forall m_k \in M_K,\ k \in K,\ p \in P \tag{22}$$

$$\sum_{d\in D} \delta_{dp}^{m_k k} \le \text{Cap}_d y_d \quad \forall m_k \in M_K,\ k \in K,\ p \in P \tag{23}$$

$$\sum_{e\in \text{Tech}} h_{mep} = \sum_{d\in D}\sum_{k\in K}\sum_{m_k\in M_K} x_{mdp}^{m_k k}\delta_{dp}^{m_k k} \quad \forall m \in M,\ p \in P \tag{24}$$

$$h_{mep} \le \text{Cap}_{me} z_{me} \quad \forall m \in M,\ e \in \text{Tech},\ p \in P \tag{25}$$

$$I_{dp-1} + \sum_{k\in K}\sum_{m_k\in M_K} \delta_{dlp}^{m_k k} = \sum_{l\in L}\sum_{k\in K}\sum_{m_k\in M_K} \eta_{lp}^{m_k k}\beta_{dlp} + I_{dp} \quad \forall d \in D, p \in P \tag{26}$$

$$I_{dp} \le \xi\left(\sum_{p-\tau_{\max}-1<\tau<p-1}\sum_{m_k\in M_K}\sum \eta_{lp}^{m_k k}\right) \quad \forall d \in D,\ p \in P \tag{27}$$

$$I_{lp} \le \sum_{P<\tau<P+\tau_{\text{MAX}}} d_{l\tau} \quad \forall l \in L,\ p \in P \tag{28}$$

$$D_{m_k k_p} = \sum_{i\in N_1} \text{dis}_{ij} x_{ijp}^{m_k k} + \sum_{i'\in N_2} \text{dis}_{i'j'} r_{ijp}^{m_k k} \quad \forall j \in N_1 | j' \in N_2,\ mk \in M_k,\ k \in K,\ p \in P \tag{29}$$

$$\sum_{i\in N_1} q_{ip}^{m_k k}\text{ST}_{ip}^{m_k k} + \sum_{j\in N_2} g_{jp}^{m_k k}\text{ST}_{jp}^{m_k k} + D_{m_k kp}/S_{m_k k} \le \text{WT} \quad m_k \in M_k,\ k \in K \tag{30}$$

$$ud_{jp} = ud_{ip} + \mathrm{ST}_{ip}^{m_k k} + \mathrm{dis}_{ij}/S_{m_k k}. \times x_{ijp}^{m_k k} \quad \forall i,j \in N_1,\ m_k \in M_k\ k \in K,\ p \in P \tag{31}$$

$$ur_{jp} = ur_{ip} + \mathrm{ST}_{ip}^{m_k k} + \mathrm{dis}_{ij}/S_{m_k k}. \times r_{ijp}^{m_k k} \quad \forall i,j \in N_2,\ m_k \in M_k\ k \in K,\ p \in P \tag{32}$$

$$pd_{ip}(ud_{ip}) = \left\{ \begin{array}{ll} pd_{ed}(ed_{ip} - ud_{ip}) & \text{if } ud_{ip} < ed_{ip} \\ 0 & \text{if } ed_{ip} \le ud_{ip} \le ld_{ip} \\ pd_{lp}(ud_{ip} - ld_{ip}) & \text{if } ld_{ip} < ud_{ip} \end{array} \right\}, \quad \forall i \in D,\ p \in P \tag{33}$$

$$pr_{ip}(ur_{ip}) = \left\{ \begin{array}{ll} pr_{ep}(er_{ip} - ur_{ip}) & \text{if } ur_{ip} < er_{ip} \\ 0 & \text{if } er_{ip} \le ur_{ip} \le lr_{ip} \\ pr_{lp}(ur_{ip} - lr_{ip}) & \text{if } lr_{ip} < ur_{ip} \end{array} \right\}; \quad \forall i \in L,\ p \in P \tag{34}$$

$$r_{dlp}^{m_k k} \in \{0,1\} \quad \forall m_k \in M_K,\ k \in K,\ d \in D,\ l \in L,\ p \in P \tag{35}$$

$$g_{lp}^{m_k k} \in \{0,1\} \quad \forall m_k \in M_K,\ k \in K,\ l \in L,\ p \in P \tag{36}$$

$$\beta_{dlp} \in \{0,1\} \quad \forall d \in D,\ l \in L,\ p \in P \tag{37}$$

$$y_d \in \{0,1\} \quad \forall d \in D \tag{38}$$

$$z_{me} \in \{0,1\} \quad \forall m \in M,\ e \in \mathrm{Tech} \tag{39}$$

$$x_{mdp}^{m_k k} \in \{0,1\} \quad \forall m_k \in M_K,\ k \in K,\ m \in M,\ d \in D,\ p \in P \tag{40}$$

$$q_{dp}^{m_k k} \in \{0,1\} \quad \forall m_k \in M_K,\ k \in K,\ d \in D,\ p \in P \tag{41}$$

$$\eta_{lp}^{m_k k} \ge 0 \quad \forall m_k \in M_K,\ k \in K,\ l \in L,\ p \in P \tag{42))}$$

$$\delta_{dp}^{m_k k} \ge 0 \quad \forall m_k \in M_K,\ k \in K,\ d \in D,\ p \in P \tag{43}$$

$$h_{mep} \ge 0 \quad \forall m \in M,\ e \in \mathrm{Tech},\ p \in P \tag{44}$$

Constraint (5) indicates that each customer has been visited only once. Constraint (6) shows the current visit to each retailer at any time interval and guarantees that the vehicle will return to the original distribution center. Constraint (7) shows that in the second level, each vehicle leaves

a maximum of one distribution center. Constraint (8) indicates that the capacity of each vehicle should be taken into account. Constraint (9) is the constraint of the elimination of circulation sets and ensures that each customer has been visited at any time interval. Constraint (10) states that each customer has entered a distribution center. The inequality (11) requires that in case a distribution center is closed, no retailer enters it. Otherwise, the overall demand of retailers by an open top distribution center could exceed its capacity. Constraint (12) states that the distribution center *d* serves the retailer if a vehicle M_k type *k* leaves *d* and reaches *l* and $l(\beta_{dlp})$ can also be equal to 1 if no vehicles go from *d* to *l*. Constraint (13) shows that if the vehicle M_k type *k* does not visit the retailer *l*, then the product amount transferred from the vehicle M_k type *k* to *l* should be zero. Constraint (14) shows the total balance in each retailer *l*. Constraint (15) shows that if a circulation enters the distribution center, the circulation must enter the retail; then, the distribution center is operationally balanced. Constraint (16) indicates that each circulation that leaves the manufacturing unit *m* should be defined. Constraints (17) and (18) limit the maximum distribution centers used and open top manufacturing units. Constraint (19) imposes ongoing observations in each distribution center at any time interval. Constraint (20) prohibits the circulation subsets.

Constraint (21) indicates the capacity of each vehicle. Constraint (22) shows that if the vehicle M_k type *k* does not enter the distribution center *d*, then the amount of product sent to the distribution center *d* by that vehicle must be zero and that the capacity of the vehicles should be taken into account. Inequality (23) states that the amount of product sent to the distribution center should be in line with its capacity. Constraint (24) requires that the product produced in the manufacturing unit *m* with technology e at the time interval *p* equals the amount of product to be delivered from that node. Constraint (25) indicates the capacity of the manufacturing unit *m*, and if no manufacturing unit *m* is used, then one cannot claim any product. Constraint (26) applies balance to each distribution center *d*. Constraint (27) ensures that the distribution center *d* has no inventory level higher than the total aggregate of products delivered by *d* in the previous continuous time period $(\tau_{\max} - 1)$. Constraint (28) ensures that retailer *l* has no inventory levels above the overall demand over the next continuous time period $(\tau_{\max} - 1)$. Constraint (29) calculates the delivery distance of each vehicle M_k type *k* in each time interval. Constraint (30) shows that the delivery time for a vehicle M_k type *k* at any time interval *p* should not exceed the working

time of each day. Constraints (31) and (32) show that at each interval, the arrival time for the *i* and *j* nodes is the same, plus the servicing time for node I with the vehicle in the node *i*, and the time to come from nodes *i* to *j* in first and second circulations. Finally, constraints (33) and (33) show that, for any time interval *p*, a penalty fee is incurred for the deviation from the time window because no arrival time exists for a node in the specified time interval in the first and second circulation rounds.

CASE STUDY

In this case study, the manufacturing group B.A was investigated. In this study, distribution of all types of ready-made foods of meat products to distribution centers was considered. This product should be consumed within 6 months from the time of production. Therefore, for this purpose, the importance of the subject was first introduced and then the details of the problem were addressed (Fig. 1).

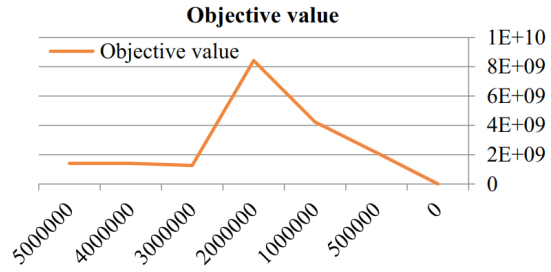


Figure 1: The amount of objective function with variation in demand.

As suggested in the mathematical modeling section (Sect. 3), the proposed model is a reverse logistical chain network model of the sustainable production system of perishable goods, which is used in this section. In this study, due to the sensitivity of meat products, it is considered to be a difficult type. Before proceeding to solve the model, the sustainable manufacturing system in the company under study was briefly described.

The Setareh Yakhi Asia Company delivers all types of Persian and Western ready-made food products with the most advanced and up-to-date methods of preparation, processing and packaging under the brand B.A. B.A production group, using today's modern technology and specialists, as the first and largest producer of ready-made and semi-ready Iranian and international halal foods, and according to the current and future needs of the stakeholders, aims to satisfy its customers and meet their various needs by

tireless endeavor through the implementation of participatory management, with the following policies:

- Compliance with national and international regulations and implementation of ISO9001-2008 quality management system requirements and ISO 22000-2005 Food Safety Management System.
- Client-oriented expansion at all levels of the organization and satisfying the consumers by implementing CRM system (Customer Relationship Management System).
- Continuous improvement of all processes in order to raise the level of quality and safety.
- Increasing the ability to "identify, assess, monitor and control" the risks to product safety and consumer health, and more efforts to maintain and improve GMP/GHPs (desirable conditions for construction and sanitation).
- Increasing the employees' participation and empathy in decision making.
- Raising the level of knowledge of employees through practical and strategic training.
- Increasing productivity in key processes and achieving organizational excellence.

Creating and promoting effective inter- and intra-organizational communications: In this regard, all managers and staff are required to work toward achieving the above-mentioned goals and try to increase the level of satisfaction of the stakeholders and control the risks to the consumers. Therefore, the management representative in the quality and food safety management system, with sufficient authority, is responsible for continuous monitoring, evaluation and ensuring the correct functioning of the above systems. For the first time in the country's food industry, designing products aimed at improving texture, taste, color, aroma and quality of food, as well as promotion of traditional and healthy Iranian foods that are unfortunately less common on the Iranian dining tables, such as vegetable omelet, potato omelet and cutlet, has been performed in the B.A. production group. For this purpose, in 1388, the formulation of more than 35 types of Iranian and foreign foods was performed under the supervision of reliable European and Iranian experienced experts in the food industry in cooperation with renowned Iranian chefs, and after the market test operations, they were gradually prepared for mass production and delivered to the consumer

markets. B.A., ready-made foods production group, has been embracing the latest technology in the production of fully cooked, frozen foods with traditional Iranian cuisine by the highly advanced machines and the expertise of managers in a land with an area of over 20,000 sq.m and a subtraction of 9000 sq.m in the large industrial complex of Shiraz with various venues including the following sections:

- A: The reception section for the red meat, consisting of below zero refrigerating room, as well as aging, bone removal, chopping, etc. (for warm meat).
- B: The reception section for the white meat consists of above and below zero refrigerating room, the initial washing and cleaning, in order to prevent microbial contamination (considering that in chicken slaughterhouses, chickens are not cleaned completely). Then, automatic transference to the chopping system which can split up to 6000 chicken carcasses into 2–14 pieces per hour automatically.
- C: Processing halls include two production halls: one for preparation of raw materials, the other for processing the product and packing it with the most advanced machinery and technology in the world. Therefore, the raw materials are received in compliance with all health conditions and after approval of the relevant systems, then stored in the best possible conditions and in the manufacturing halls, by using the standards of the USA and Japan, which are certainly the world's leading food producers. In order to prevent possible contamination, cold air generation and sterilizing equipment are used in the facilities to keep the temperature of the halls in all seasons at 12–15 °C. Also, positive pressure systems combined with microbial filtration help the company comply with all conditions mentioned in HACCP and minimize the risk by the highest and most advanced technology and novel preparation, processing, packaging and marketing methods.

Research Data

Since the company produces a diverse range of products like cooked foods, including chicken nugget, chicken and mushroom nugget, potato croquette, Krakow, ; semi-cooked foods, including hamburger, chicken burger, vegetable omelet, little omelets, ; and raw foods, including chicken kebab,

Lari kebab, in various volumes, in this research, the supply chain of fried foods (chicken nuggets) was examined.

- Storage conditions: 18° below zero
- Warehousing conditions: 18° below zero, inside the carton and plastic pallet
- Package weight: 250 g
- Bulk packaging weight: 2 kg
- Number in the package: 9 pcs

The full list of additives and packaging materials together with the amount of consumption per ton of nuggets is shown in Table 1.

Table 1: Additive consumption and inventory of the first period

Type of material	Unit	Consumption/ton
Active carbon	kg	0.12
Anti oxidants	kg	0.08
Acid citric	kg	0.44
Phosphoric acid	kg	0.6
Beta carotene	kg	0.025
Propylene glycol	kg	0.63
Liquid soda	kg	5.5
Catalyst nickel	kg	1.6
Aromatics	kg	0.75
Monodiglyceride	kg	3
Lecithin	kg	1.5
Potassium sorbate	kg	1
NaCl	kg	3
Cartons	–	100
Nylon	kg	2.7
Adhesives	–	0.18

Company planning is usually announced to all of the manufacturing departments at the beginning of the year as a forecast for the whole year by the planning unit and with the cooperation of the trading department with regard to the capacity of the manufacturing equipment. During the year, the planning director, production manager and the commercial manager accurately determine the amount of the monthly production. In the meat industry, production of oil drop is allowed to be between 2 and 5%, and it is

3% for this company. The company produces about 4500 to 5000 nuggets per month. The demand for nuggets is in an average of 200–250 *t*/*m*, which is about 5.5% of the total factory production.

Changes in demand are an effective factor in maximizing target functions. Table 2 shows how much change in demand affects the target functions.

Table 2: Sensitivity analysis of demand changes

***D* (demand)**	**Objective function**
0	−121,378
500,000	2,143,146,000
1,000,000	4,234,710,000
2,000,000	8,417,837,000
3,000,000	1,260,096,000
4,000,000	1,409,690,000
5,000,000	1,409,690,000

As shown in Table 2, demand changes lead to changes in the objective function, that is, with increase in the amount of demand, the company's profit also increases, and it is clear that when the demand does not exist, the amount of the objective function is negated. Therefore, the amount of optimized demand is created when the factory production is the same as the sales.

Table 3 describes the values of the four objective functions introduced in this study. As shown in the table, the two first objectives are minimized and the two following objectives are maximized; therefore, in sensitivity analysis for the first two functions that are minimized, the minimum and maximum values are displaced. Also, the model responses are ensured for (the feasibility of) all constraints, meaning that the optimal values obtained for all constraints are true.

Table 3: Results of the studied case calculations (together with sensitivity analysis)

Objective function	**Minimum**	**Optimized value of the objective function**	**Maximum**
Objective 1	2,500,756,000	3,750,009,781	1,570,000,000
Objective 2	3,700,840,000	3,205,685,000	1,764,375,000
Objective 3	2,746,874,000	3,746,870,000	4,005,874,000
Objective 4	89%	95%	95%

CONCLUSIONS

Reducing costs and increasing the level of service (satisfaction) are the most important factors in today's market competition. In this regard, in the framework of a comprehensive systematic approach, supply chain management considers the coordination between the members in order to reduce costs and increase the level of service in providing a product or service to the customers. In this system, the total costs of the facility are a special priority. Efforts have been made to model and optimize the supply chain design. But there are a few research projects that target the design of supply chain networks comprehensively (taking into account both strategic and tactical issues simultaneously). Many of these attempts use definitive methods, while in the real world, definitive assumption is unreasonable. Therefore, it is necessary to consider uncertainty in investigations and decisions. On the other hand, taking into account the reduction of environmental impacts, considering the importance of the environment and pollution prevention is of great significance. Environmental damage is one of the most intangible costs that the entire community is its beneficiary.

In the sustainable supply chain, the effects of a chain on the environment are also addressed, and this, together with the inclusion of uncertainty and the study of the supply chain for perishable goods, forms an efficient collection that is addressed in this study. In this paper, introducing two objective functions to minimize the cost of supply chain design and environmental impacts and two functions to maximize profitability and satisfaction with the use of technology, all aspects of a supply chain for perishable goods are considered. The proposed model of this research has been implemented for the B.A. food production company. This unit uses up-to-date equipment for production. The results of the study indicate the effect of demand on the objective functions. By analyzing the sensitivity to demand, it was found that a change in demand would lead to a change in the level of profitability, and the optimal demand would be reached when the production amounts of the factory are the same as the sales. In this paper, proper objective functions for each of the four objectives were introduced with appropriate constraints that consider all aspects. Further research can be done to investigate other issues. For example, customer demand maximization functions can be added to target functions by using strategic planning and maximizing customer satisfaction. For example, customer demand maximization functions by strategic planning and maximizing customer satisfaction can be added to objective functions. Fuzzy numbers can also be used instead of crisp numbers.

REFERENCES

1. Agarwal S (2017) Issues in supply chain planning of fruits and vegetables in agri-food supply chain: A review of certain aspects. IMS Business School Presents Doctoral Colloquium, Kolkata
2. An K, Ouyang Y (2016) Robust grain supply chain design considering post-harvest loss and harvest timing equilibrium. Transp Res Part E Logist Transp Rev 88:110–128
3. AriaNezhad MG, Makuie A, Khayatmoghadam S (2013) Developing and solving two-echelon inventory system for perishable items in a supply chain: case study (Mashhad Behrouz Company). J Ind Eng Int 9:1–10
4. Balaji M, Arshinder K (2016) Modeling the causes of food wastage in Indian perishable food supply chain. Resour Conserv Recycl 114:153–167
5. Bloemhof JM, Soysal M (2017) Sustainable food supply chain design. In: Bouchery Y, Corbett CJ, Fransoo JC, Tan T (eds) Sustainable supply chains, Springer, pp 395–412
6. Borodin V, Bourtembourg J, Hnaien F, Labadie N (2016) Handling uncertainty in agricultural supply chain management: a state of the art. Eur J Oper Res 254(2):348–359
7. Bourlakis M, Maglaras G, Gallear D, Fotopoulos C (2014) Examining sustainability performance in the supply chain: the case of the Greek dairy sector. Ind Market Manag 43(1):56–66
8. Eskandarpour M, Dejax P, Miemczyk J, Péton O (2015) Sustainable supply chain network design: an optimization-oriented review. Omega 54:11–32
9. Fedrigotti VB, Fischer C (2015) Sustainable development options for the chestnut supply chain in South Tyrol, Italy. Agric Agric Sci Procedia 5:96–106
10. Galal NM, Moneim AFA (2016) Developing sustainable supply chains in developing countries. Procedia Cirp 48:419–424
11. Ghoushchi SJ, Milan MD, Rezaee MJ (2018) Evaluation and selection of sustainable suppliers in supply chain using new GP-DEA model with imprecise data. J Ind Eng Int 14:613–625
12. Katsaliaki K, Mustafee N, Kumar S (2014) A game-based approach towards facilitating decision making for perishable products: an example of blood supply chain. Expert Syst Appl 41(9):4043–4059

13. Lemma Y, Kitaw D, Gatew G (2014) Loss in perishable food supply chain: an optimization approach literature review. Int J Sci Eng Res 5(5):302–311
14. Li D, Wang X, Chan HK, Manzini R (2014) Sustainable food supply chain management. Int J Prod Econ 152:1–8
15. Nagurney A (2015) Design of sustainable supply chains for sustainable cities. Environ Plan B Plan Des 42(1):40–57
16. Sazvar Z, Sepehri M, Baboli A (2016) A multi-objective multi-supplier sustainable supply chain with deteriorating products, case of cut flowers. IFAC PapersOnLine 49(12):1638–1643
17. Sgarbossa F, Russo I (2017) A proactive model in sustainable food supply chain: insight from a case study. Int J Prod Econ 183:596–606
18. Singh T, Mishra PJ, Pattanayak H (2017) An optimal policy for deteriorating items with time-proportional deterioration rate and constant and time-dependent linear demand rate. J Ind Eng Int 13:455–463
19. Soto-Silva WE, González-Araya MC, Oliva-Fernández MA, Plà-Aragonés LM (2017) Optimizing fresh food logistics for processing: application for a large Chilean apple supply chain. Comput Electron Agric 136:42–57
20. Stindt D (2017) A generic planning approach for sustainable supply chain management-How to integrate concepts and methods to address the issues of sustainability? J Clean Prod 153:146–163
21. Sun J, Debo L (2014) Sustaining long-term supply chain partnerships using price-only contracts. Eur J Oper Res 233(3):557–565
22. Sundara Rajan R, Uthayakumar R (2017) Optimal pricing and replenishment policies for instantaneous deteriorating items with backlogging and trade credit under inflation. J Ind Eng Int 13:427–443
23. Uthayakumar R, Tharani S (2017) An economic production model for deteriorating items and time dependent demand with rework and multiple production setups. J Ind Eng Int 13:499–512
24. Wang J, Yue H (2017) Food safety pre-warning system based on data mining for a sustainable food supply chain. Food Control 73:223–229
25. Xiao Y, Yang S (2016) The retail chain design for perishable food: the case of price strategy and shelf space allocation. Sustainability 9(1):12
26. Zhang S, Lee CKM, Wu K, Choy KL (2016) Multi-objective optimization for sustainable supply chain network design considering multiple distribution channels. Expert Syst Appl 65:87–99

Chapter

8

NEW MATHEMATICAL MODELING FOR A LOCATION–ROUTING–INVENTORY PROBLEM IN A MULTI-PERIOD CLOSED-LOOP SUPPLY CHAIN IN A CAR INDUSTRY

F. Forouzanfar[1] , R. Tavakkoli-Moghaddam[2,3] , M. Bashiri[4] , A. Baboli[5] , S. M. Hadji Molana[1]

[1] Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2] School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
[3] LCFC, Arts et Me´tier Paris Tech, Metz, France
[4] Department of Industrial Engineering, Faculty of Engineering, Shahed University, Tehran, Iran
[5] DISP Laboratory, INSA-Lyon, University of Lyon, Villeurbanne, France

Citation: (APA): Forouzanfar, F., Tavakkoli-Moghaddam, R., Bashiri, M., Baboli, A., & Hadji Molana, S. M. (2018). New mathematical modeling for a location–routing–inventory problem in a multi-period closed-loop supply chain in a car industry. Journal of Industrial Engineering International, 14(3), 537-553. .(17 pages).

ABSTRACT

This paper studies a location–routing–inventory problem in a multi-period closed-loop supply chain with multiple suppliers, producers, distribution centers, customers, collection centers, recovery, and recycling centers. In this supply chain, centers are multiple levels, a price increase factor is considered for operational costs at centers, inventory and shortage (including lost sales and backlog) are allowed at production centers, arrival time of vehicles of each plant to its dedicated distribution centers and also departure from them are considered, in such a way that the sum of system costs and the sum of maximum time at each level should be minimized. The aforementioned problem is formulated in the form of a bi-objective nonlinear integer programming model. Due to the NP-hard nature of the problem, two meta-heuristics, namely, non-dominated sorting genetic algorithm (NSGA-II) and multi-objective particle swarm optimization (MOPSO), are used in large sizes. In addition, a Taguchi method is used to set the parameters of these algorithms to enhance their performance. To evaluate the efficiency of the proposed algorithms, the results for small-sized problems are compared with the results of the ε-constraint method. Finally, four measuring metrics, namely, the number of Pareto solutions, mean ideal distance, spacing metric, and quality metric, are used to compare NSGA-II and MOPSO.

INTRODUCTION

In the 90s, with the improvement of production processes and spreading of reengineering patterns, managers of many industries were not satisfied only with the improvement of internal processes and flexibility in corporate capabilities. They found out that the suppliers of parts and materials have to also produce materials with the best quality and cost and distributors of the products must have a close relationship with the market development policies of the producers. With such an attitude, logistic approaches, supply chain, and its management have emerged. Conditions of global competition and environmental sensitivity have made corporations responsible for collecting the rejected products to recover, recycle, or devastation them to maintain the environment and gain the profit of rejected products that have been abandoned. Collecting products after the consumption by customers and returning them to supply chain or devastating them bring up the closed-loop supply chain problem. The concept of the closed-loop supply chain has gained attention as a result of identification forward and reverse supply chains that are managed seamlessly. In the last 3 decades, with the highlighting of

the importance of supply chain management to the industrialists, the role of coordination and integration of different components of the supply chain has become stronger in creating competitive advantages and the concept of integration has become one of the most important aspects of the supply chain management system. This concept addresses the dependence between the location of facilities, allocation of suppliers and customers to the facilities, the structure of transport system and routing them, and inventory control system.

With respect to the increasing attention to supply chain management subjects, the combinations of location, routing and inventory subjects have become of great importance in industry. Actually, the interest and attention that exist to the subjects of the supply chain have made more critical the importance of having an optimized supply chain system. Appropriate configuration of the supply chain network is considered as a continuous and noticeable competitive advantage and help the corporation against the other future problems and difficulties. Applying an integrated location–routing–inventory approach to optimize the closed-loop supply chain problem can be beneficial. Actually, the above decisions are highly dependent and only identification of the optimum of these variables in an interactive manner can result in finding an optimized supply chain system with the minimum possible costs.

In this paper, we present a new mathematical model for the location–routing–inventory problem in a closed-loop supply chain network that consists of multiple suppliers, producers, distribution centers, collecting centers, and recovery and recycling centers. Furthermore, we consider multiple periods, price increase factor for the operational costs at centers through the periods, existence the inventory or shortage (lost sales or backlog) at production centers, multiple levels of capacity for centers, arrival time to distribution centers and departure from them in the routing, and cost and time of transportation. To solve this problem, the ε-constraint method and NSGA-II and MOPSO algorithms are applied. To enhance the efficiency of these algorithms, the Taguchi method is used to tune their parameters. The remainder of this paper is as follows. In “Literature review”, the literature is reviewed. In “Problem definition”, the proposed problem is discussed and a mathematical model is presented. “Proposed solution methods” uses the ε-constraint method, NSGA-II and MOPSO algorithms to solve the model. Different comparing factors are expressed in “Comparing factors of multi-objective evolutionary algorithms”. In “Computational results”,

the computational results are presented. Finally, "Conclusion" provides the conclusion.

LITERATURE REVIEW

Some of the recent studies in the context of the location–routing–inventory problem, closed-loop supply chain, and their synthesis are presented to show the necessity of this research.

In the earliest publications about the location–routing–inventory problem, a paper titled integrating routing and inventory costs in strategic location problems was presented (Shen and Qi 2007). The intended paper considers the supply chain design problem in which the decision maker needs to decide on the number and location of distribution centers. The demand of the customers is random and each distribution center holds a certain amount of the warehouse storage with the purpose of achieving the appropriate service level to their dedicated customers. The intended problem has been formulated as a nonlinear integer programming model. For the first time, Ahmadi Javid and Azad (2010) presented a new model in a non-deterministic supply chain that simultaneously optimizes the location, allocation, capacity, inventory, and routing decisions. It has been assumed that the demand of customers is non-deterministic and follows a normal distribution. In addition, each distribution center holds a certain amount of warehouse storage.

Hiassat and Diabat (2011) presented a location–routing–inventory problem for deteriorating products. In the mentioned paper, one producer distributes one deteriorating product with a given imperishability to multiple retailers through a number of warehouses. It has been assumed that at each period, each vehicle travels at most in one route and all of the customers are served. The fleet is homogeneous and all vehicles are identical in capacity. Ahmadi Javid and Seddighi (2012) proposed the new location–routing–inventory model with deterministic demand for multi-resource distribution network. The objective of this problem is to minimize the total cost of location, routing, and inventory and was formulated in the form of mixed integer programming.

Xia (2013) presented a three-level multi-product model, in which the capacity and routing decisions, assignment decisions, transportation decisions, and routing and inventory decisions were considered. The objective of this model was the selection of locations for distribution centers, identification of transportation assignment, setup of inventory

policy according to the servicing needs, and scheduling the routes of the vehicle to meet the customer demands. In the above model, the demand of each retailer at each period follows a normal distribution. Nekooghadirli et al. (2014) in a two-level supply chain including of distribution centers and customers formulated a bi-objective multi-product multi-period location–routing–inventory problem. In their model, the demand of customers is unknown and follows a normal distribution. Each distribution center holds a certain amount of safety stock and shortage is not allowed. The objective of mentioned model is to minimize total cost and maximize the expected time of delivery goods to customers. Four algorithms, including MOICA, MOPSA, NSGA-II, and PAES (Pareto archived evolution strategy), were applied to solve the model. In the same area, Guerrero et al. (2013) presented a synthetic heuristic for a location–routing–inventory problem. They considered a multi-depot multi-retailer system with capacitated storage during a discrete planning horizon.

Zhang et al. (2014) presented a mathematical model for a two-level supply chain network that includes multiple capacitated potential depot and a set of customers. This model simultaneously optimizes the decisions of location, allocation, inventory, and routing and minimizes the system costs. The amount of delivery to customers at each period and their repletion under the state of vender-managed inventory (VMI) by the homogenous fleet of capacitated vehicles are identified by the model. In the above model, the demand during the planning time horizon is deterministic and dynamic and inventory are held in the customers' zones. Chen et al. (2017) published a paper for optimization of a multi-stage closed-loop supply chain for solar cell industry. The model formulated as a multi-objective mixed integer linear programming. Multi-objective particle swarm optimization algorithm with non-dominated sorting approach based on crowding distance was developed to search the near-optimal solution.

Zhalechian et al. (2016) presented a reliable closed-loop location–routing–inventory supply chain network under the synthetic uncertainty in the form of a new multi-objective linear programming. It has been assumed that each retailer has unknown demand which follows a normal distribution. Furthermore, different types of products have been considered in the closed-loop supply chain. Tang et al. (2016) presented a reliable location–routing–inventory model. They considered the customer environmental behavior. Aydin et al. (2016) published a paper on the coordination of closed-loop supply change for designing of the production line with considering of reproduced products. The NSGA-II algorithm was applied for identification

of Pareto optimal solution of multi-objective problems. This paper of Kadambala et al. (2017) measured the effective responsiveness of closed-loop supply chain in the terms of time and energy productivity. This model was formulated as a multi-objective mixed integer linear programming and multi-objective particle swarm optimization approach and NSGA-II were applied to solve it.

The main contributions of this paper, which differentiate our effort from related studies, are as follows:

- designing a new multi-objective mathematical model for a location–routing–inventory problem in a multi-period closed-loop supply chain in a car industry;
- minimizing the sum of the maximum time at each level in a closed-loop supply chain;
- considering the arrival/departure time of vehicles of each plant to/from its dedicated distribution centers;
- determining the percentage of lost sale and backlog of total shortage at each period according to the specified policy.

PROBLEM DEFINITION

Our problem is a multi-stage closed-loop supply chain including multiple suppliers, producers, distribution centers, customers, collecting centers, and recovery and recycling centers. The proposed model of the problem minimizes the inventory and shortage costs, production costs, fixed cost of the transportation vehicles of each plant at each period, costs of locating centers with a certain level of capacity, operational costs at centers at each period, and the sum of maximum time at each period.

The objective of this model is to identify the number of opened centers, their locations, and capacities, how to allocate centers at subsequent stages, amount of inventory or shortage at each opened plant at each period, value of production of each opened plant at each period, routes of vehicles with starting from an opened plant to serve its opened allocated distribution centers, and come back to it. Furthermore, this model calculates the vehicle arrival and departure time of each plant to/from distribution centers at each period, amount of transferred goods between opened centers at each stage during each period, so that the sum of costs and transportation time should be minimized. The corresponding decision variables are presented in the proposed mathematical model.

For better understanding of this problem, we consider a closed-loop supply chain with four and three levels in the forward and reverse supply chains, respectively, as depicted in Fig. 1. Forward supply chain levels include multiple suppliers, producers, distribution centers, and customers. Reverse supply chain levels consist of some collecting, recovery and recycling centers. The proposed model has been designed according to the real case study in the car industry in Iran. Some suppliers of spare parts for Iran Khodro Company are Ezam, Mehrkam Pars, Crouse group and the like. The Iran Khodro Industrial Group sells its products through its authorized agents to customers. These agents exist in the most cities of Iran. Using end-of-life vehicles not only bears a very high expense in the economic aspects as well as fuel consumption, but also leads to the extraordinarily heavy costs in environmental aspects for Iran. For this reason, Iranian government has paid special attention to collect end-of-life vehicles. Parts and components of thesis vehicles are disassembled. Recoverable parts are sent to recovery centers and recyclable parts are sent to recycle centers. After recycling recyclable parts, they are sold to raw materials customers and then delivered to suppliers of spare parts.

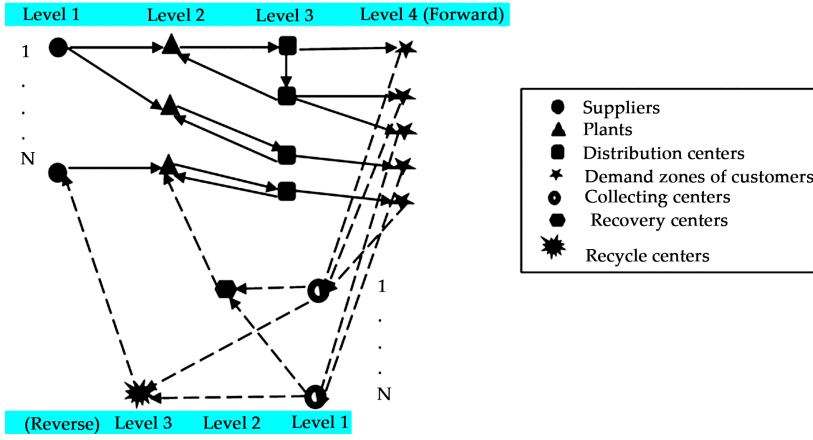


Figure 1: Closed-loop supply chain network.

Assumptions

Some of the assumptions considered for this model are as follows:

- The intended problem is single product.
- At each period, the factories can have inventory or shortage or none of them (inventory = shortage = 0) (Gorji et al. 2014).

- The shortage includes lost sale and backlog (Mousavi et al. 2015).
- The demand of the distribution centers and customers at each period are deterministic.
- The cost of centers opening with certain capacity is specified.
- All opened centers must be served.
- Different levels are considered for centers and eventually one capacity level is selected for each center.
- Time horizon planning is multi-period.
- The backlog of each distribution center at each period must be supplied at next period by its dedicated factor.
- The lost sale of the plant at each period is not compensable.
- For operational costs at centers, price increase factor is considered.
- It is assumed that production is done at the beginning of the period and its sum with the net inventory at the end of the previous period is always positive.
- Routing is considered from plant to distribution centers (Ahmadizar et al. 2015).

Sets

I :Set of plants

S :Set of distribution centers

K :Set of demand zones of customers

L :Set of collecting centers

M :Set of recovery centers

N :Set of recycle centers

λ ***:***Set of suppliers

T :Set of time period

V :Set of vehicles

$\boldsymbol{b}_j$***:***Set of capacity levels available to center j ($j \in i, s, l, m, n, \lambda$)

Parameters

$G_{1i}^{b_i}$ **:** Fixed cost of opening and operating plant i with capacity level b_i

$G_{2s}^{b_s}$ **:** Fixed cost of opening and operating distribution center s with capacity level b_s

$G_{3l}^{b_l}$ **:** Fixed cost of opening and operating collecting center *l* with capacity level b_l

$G_{4M}^{b_M}$ **:** Fixed cost of opening and operating recovery center *M* with capacity level b_M

$G_{5N}^{b_N}$ **:** Fixed cost of opening and operating recycle center *N* with capacity level b_N

$G_{6\lambda}^{b_\lambda}$ **:** Fixed cost of opening and operating supplier λ with capacity level b_λ

e_s **:** Price increase factor for processing each product unit at each distribution center *s*

e_L **:** Price increase factor for processing each product unit at each collecting center *L*

e_M **:** Price increase factor for recovery each product unit at each recovery center *M*

e_N **:** Price increase factor for recycle each product unit at each recycle center *N*

e_λ **:** Price increase factor for production each part at each supplier λ

e_i **:** Price increase factor for production each product unit at each plant *i*

P_s **:** Processing cost of each product unit at each distribution center *s* at the beginning of time horizon planning

P_L **:** Processing cost of each product unit at each collecting center *L* at the beginning of time horizon planning

P_M **:** Processing cost of each product unit at each recovery center *M* at the beginning of time horizon planning

P_N **:** Processing cost of each product unit at each recycle center *N* at the beginning of time horizon planning

P_λ **:** Expected value of production cost of one part at each supplier λ at the beginning of time horizon planning

P_i **:** Production cost of each product unit at each plant *i* at the beginning of time horizon planning

T_{KS} **:** Time between customer *K* and distribution center *S* (time is considered as a distance function)

T_{KL} **:** Time between customer *K* and collecting center *L* (time is considered as a distance function)

T_{LM} **:** Time between collecting center *L* and recovery center *M* (time is considered as a distance function)

T_{LN} **:** Time between collecting center *L* and recycle center *N* (time is considered as a distance function)

$T_{\lambda i}$: Time between supplier λ and plant *i* (time is considered as a distance function)
$T_{N\lambda}$: Time between recycle center *N* and supplier λ (time is considered as a distance function)
T_{Mi} : Time between recovery center *M* and plant *i* (time is considered as a distance function)
t'_{ivt} : Departure time of vehicle ν from plant *i* at period *t*
O''_{is} : Distance traveled between plant *i* and distribution center *s*
V'_{iv} : Average speed of vehicle ν of plant *i*
$O'''_{s's}$: Distance traveled between distribution center *s* and *s′*
L'_{ivs} : Required time for unloading each product unit from vehicle ν of plant *i* at distribution center *s*
γ_{ist} : Percent of the shortage at distribution center *s* to the total shortage at plant *i* at period *t*
ω'_{iv} : Capacity of vehicle ν of plant *i*
α_{it} : Percent of backlog to the total shortage at plant *i* at period *t*
T : The time of each period
h_t : Holding cost of each unit of inventory at period *t*
T'_{it} : A party of the period *t* that inventory exist
T''_{it} : A party of the period *t* that shortage exist
h'_t : Cost of one unit of backlog at period *t*
h''_t : Cost of one unit of lost sale at period *t*
c'_{ivt} : Unit transportation cost of vehicle ν of plant *i* at period *t*
f'_{iv} : Fixed cost of vehicle ν of plant *i*
u_{it} : Available budget for plant *i* at period *t*
d_{kt} : Amount of customer's demand of zone *k* in period *t*
ε : Number of used parts in one product
δ : Percent of recyclable parts of each product
ω_l : Capacity of collecting center *l*
ω_M : Capacity of recovery center *M*
ω_N : Capacity of recycle center *N*
ω_λ : Capacity of supplier λ
ω_i : Capacity of plant *i*
SP_1 : Volume of each product
SP_2 : Average volume of each part
F_{it} : Maximum number of producible products at plant *i* at each period with regard to available resources at that period

N : A large unbounded positive number

K′: A fixed number determined by plants as a bonus given to distribution centers

Decision Variables

$A_{1I}^{b_i}$ **:** 1, if plant *i* with capacity level b_i is opened; 0, otherwise

$A_{2s}^{b_s}$ **:** 1, if distribution center *s* with capacity level b_s is opened; 0, otherwise

$A_{3L}^{b_L}$ **:** 1, if collecting center *L* with capacity level b_L is opened; 0, otherwise

$A_{4M}^{b_M}$ **:** 1, if recovery center *M* with capacity level b_M is opened; 0, otherwise

$A_{5N}^{b_N}$ **:** 1, if recycle center *N* with capacity level b_N is opened; 0, otherwise

$A_{6\lambda}^{b_\lambda}$ **:** 1, if supplier λ with capacity level b_λ is opened; 0, otherwise

X_{KST} **:** 1, if customer *K* assigned to distribution center *S* at period *t*; 0, otherwise

X_{KLT} **:** 1, if collecting center *L* assigned to customer *K* at period *t*; 0, otherwise

X_{LMT} **:** 1, if collecting center *L* assigned to recovery center *M* at period *t*; 0, otherwise

X_{LNT} **:** 1, if collecting center *L* assigned to recycle center *N* at period *t*; 0, otherwise

$X_{\lambda it}$ **:** 1, if supplier λ assigned to plant *i* at period *t*; 0, otherwise

$X_{N\lambda t}$ **:** 1, if recycle center *N* assigned to supplier λ at period *t*; 0, otherwise

X_{Mit} **:** 1, if recovery center *M* assigned to plant *i* at period *t*; 0, otherwise

X_{ivst} **:** 1, if distribution center *s* be the first one met by vehicle ν of plant *i* at period *t*; 0, otherwise

X''_{ivst} **:** 1, if distribution center *s* be the last one met by vehicle ν of plant *i* at period *t*; 0, otherwise

X'''_{ivst} **:** 1, if distribution center *s* is served by vehicle ν of plant *i* at period *t*; 0, otherwise

$X'_{ivs'st}$ **:** 1, if distribution center *s* is met immediately after distribution center *s′* with vehicle ν of plant *i* at period *t*; 0, otherwise

Y_{it} **:** 1, if plant *i* have inventory at the end of period *t*; 0, otherwise

NS_{it} **:** Net value of inventory of plant *i* at the end of period *t*

Q_{it} **:** Amount of production in plant *i* at period *t*

d_{st} **:** Demand of distribution center *s* at period *t*

O_{skt} **:** Amount of transferred product from distribution center *s* to customer *K* at period *t*

O_{kLt} **:** Amount of transferred product from customer *K* to collecting center *L* at period *t*

O_{LMt} **:** Amount of transferred product from collecting center *L* to recovery center *M* at period *t*

O_{LNt} **:** Amount of transferred product from collecting center *L* to recycle center *N* at period *t*

η'_{ivst} **:** Arrival time of vehicle ν of plant *i* to distribution center *s* at period *t*

θ'_{ivst} **:** Departure time of vehicle ν of plant *i* from distribution center *s* at period *t*

Z'_{it} **:** Total cost of inventory, shortage (lost sale and backlog), production, fixed cost of vehicle, and transportation of plant *i* at period *t:*

ω_{it}: $\begin{cases} 1 & NS_{it} = 0 \\ 0 & NS_{it} < 0 \end{cases}$

M_{ist}: $\begin{cases} 1 & d_{st} > K' \\ 0 & d_{st} \le K' \end{cases}$.

Mathematical Model

$$\text{MIN } Z_1 = \sum_t (T_{1t} + T_{2t} + T_{3t} + T_{4t} + T_{5t} + T_{6t} + T_{7t} + T_{8t})$$

$$\begin{aligned}\text{MIN } Z_2 = {} & \sum_{b_i}\sum_{I} G_{1I}^{b_i} \cdot A_{1I}^{b_i} + \sum_{b_s}\sum_{S} G_{2s}^{b_s} \cdot A_{2s}^{b_s} + \sum_{b_L}\sum_{L} G_{3L}^{b_L} \cdot A_{3L}^{b_L} \\ & + \sum_{b_M}\sum_{M} G_{4M}^{b_M} \cdot A_{4M}^{b_M} + \sum_{b_N}\sum_{N} G_{5N}^{b_N} \cdot A_{5N}^{b_N} \\ & + \sum_{b_\lambda}\sum_{\lambda} G_{6\lambda}^{b_\lambda} \cdot A_{6\lambda}^{b_\lambda} + \sum_{t}\sum_{I} Z'_{it} \\ & + \sum_{S}\sum_{b_S}\sum_{K}\sum_{t} A_{2s}^{b_s} \cdot P_s \cdot X_{kst} \cdot O_{kst} \cdot 1 + e_s)^t \\ & + \sum_{L}\sum_{b_L}\sum_{k}\sum_{t} A_{3L}^{b_L} \cdot P_L \cdot O_{klt} \cdot X_{klt} \cdot (1 + e_L)^t \\ & + \sum_{M}\sum_{b_M}\sum_{L}\sum_{t} A_{3L}^{b_L} \cdot A_{4M}^{b_M} \cdot P_M \cdot O_{LMt} \cdot X_{LMt} \cdot (1 + e_M)^t \\ & + \sum_{N}\sum_{b_N}\sum_{L}\sum_{t} A_{5N}^{b_N} \cdot A_{3L}^{b_L} \cdot P_N \cdot O_{LNt} \cdot X_{LNt} \cdot (1 + e_N)^t \\ & + \sum_{N}\sum_{b_N}\sum_{\lambda}\sum_{t} A_{6\lambda}^{b_\lambda} \cdot A_{5N}^{b_N} \cdot P_\lambda \cdot O_{N\lambda t} \cdot X_{N\lambda t} \cdot (1 + e_\lambda)^t\end{aligned}$$

s.t.

$$T_{1t} = \max_i \left[\left(\max_s \sum_{b_s}\sum_{v} A^{b_s} 2s \cdot X'''_{ivst} \cdot \eta'_{ivst} \right) - t'_{ivt} \right] \quad \forall t \tag{1}$$

$$T_{2t} = \max_{k}\left[\sum_{S} T_{ks} \cdot X_{kst}\right] \quad \forall t \tag{2}$$

$$T_{3t} = \max_{k}\left[\sum_{L} T_{kL} \cdot X_{kLt}\right] \quad \forall t \tag{3}$$

$$T_{4t} = \max_{L}\left[\sum_{M} X_{LMt} \cdot T_{LM}\right] \quad \forall t \tag{4}$$

$$T_{5t} = \max_{L}\left[\sum_{N} X_{LNt} \cdot T_{LN}\right] \quad \forall t \tag{5}$$

$$T_{6t} = \max_{i}\left[\sum_{\lambda} X_{\lambda it} \cdot T_{\lambda i}\right] \quad \forall t \tag{6}$$

$$T_{7t} = \max_{\lambda}\left[\sum_{N} X_{N\lambda t} \cdot T_{N\lambda}\right] \quad \forall t \tag{7}$$

$$T_{8t} = \max_{i}\left[\sum_{M} X_{Mit} \cdot T_{Mi}\right] \quad \forall t \tag{8}$$

$$\eta'_{ivst} = X_{ivst}\left(t'_{ivt} + \frac{O''_{is}}{V'_{iv}}\right) + \sum_{st=1}^{s} X'_{ivs'st}\left(\theta'_{ivs't} + \frac{O'''_{s's}}{V'_{iv}}\right) \quad \forall i, v, s, t \tag{9}$$

$$\theta'_{ivst} = \eta'_{ivst} + L'_{ivs}[Y_{it} \cdot d_{st} + (1 - Y_{it})\omega_{it} \cdot d_{st} + (1 - Y_{it})(1 - \omega_{it})d_{st} \cdot (1 - \gamma_{st})] \quad \forall i, v, s, t \tag{10}$$

$$(Y_{it} - 1)(1 - \omega_{it}) \cdot NS_{it} = \sum_{S}\sum_{V} X'''_{ivst} \cdot d_{st} \cdot \gamma_{st} \quad \forall i, t \tag{11}$$

$$\left[X_{ivst} + \sum_{s'=1}^{s} X'_{ivs'st} = \sum_{s'=1}^{s} X'_{ivss't} + X''_{ivst}\right] \quad \forall i, v, s, t \tag{12}$$

$$\sum_{V}\sum_{S} X_{ivst} = \sum_{b_i} A_{1I}^{b_i} \quad \forall i, t \tag{13}$$

$$(SP_1) \cdot \sum_{S}\sum_{V} X'''_{ivst}[Y_{it} \cdot d_{st} + (1 - Y_{it})\omega_{it} \cdot d_{st} + (1 - Y_{it})(1 - \omega_{it})d_{st} \cdot (1 - \gamma_{st})] \le \omega'_{iv} \quad \forall i, t \tag{14}$$

$$Z_{it}^{'}=\begin{bmatrix}\left(Y_{it}\left(\frac{NS_{i(t-1)}+\left(Y_{i(t-1)}-1\right)\left(1-\omega_{i(t-1)}\right)NS_{i(t-1)}\left(1-\alpha_{i(t-1)}\right)+Q_{it}+NS_{it}}{2}\right)T\cdot h_t\right)\\ +\left((1-Y_{it})\omega_{it}\left(\frac{NS_{i(t-1)}+\left(Y_{i(t-1)}-1\right)\left(1-\omega_{i(t-1)}\right)NS_{i(t-1)}\left(1-\alpha_{i(t-1)}\right)+Q_{it}}{2}\right)T\cdot h_t\right)\\ +\left((1-Y_{it})(1-\omega_{it})\left(\frac{NS_{i(t-1)}+\left(Y_{i(t-1)}-1\right)\left(1-\omega_{i(t-1)}\right)NS_{i(t-1)}\left(1-\alpha_{i(t-1)}\right)+Q_{it}}{2}\right)T_{it}^{'}\cdot h_t\right)\\ +(Y_{it}-1)(1-\omega_{it})\left(\frac{\alpha_{it}\cdot NS_{it}}{2}\cdot T_{it}^{''}\cdot h_t^{'}\right)+(Y_{it}-1)(1-\omega_{it})\left(\frac{(1-\alpha_{it})\cdot NS_{it}}{2}\cdot T_{it}^{''}\cdot h_t^{''}\right)\\ +Q_{it}\cdot P_t\cdot(1+e_t)^t+\sum_V\sum_S\left[\left(X_{ivst}+X_{ivst}^{''}\right)O_{is}^{''}\cdot C_{ivt}^{'}+X_{ivst}\cdot f_{iv}^{'}\right]+\sum_S\sum_V\sum_{S'}X_{ivss't}^{'}\cdot O_{ss'}^{'''}\cdot C_{ivt}^{'}\end{bmatrix}\quad\forall i,t \tag{15}$$

$$Z_{it}^{'}\le u_{it}\quad\forall i,t \tag{16}$$

$$\begin{aligned}NS_{it}=&NS_{i(t-1)}-\left(1-\alpha_{i(t-1)}\right)NS_{i(t-1)}\\&\cdot\left(1-Y_{i(t-1)}\right)\left(1-\omega_{i(t-1)}\right)+Q_{it}-\sum_S\left(M_{ist}+d_{st}\right)\\&\cdot X_{ivst}^{'''}\\&\forall i,t\end{aligned} \tag{17}$$

$$NS_{it}\le N\cdot Y_{it}\quad\forall i,t \tag{18}$$

$$NS_{it}\succ\left(Y_{it}-1\right)\cdot N\quad\forall i,t \tag{19}$$

$$d_{st}\le\left(\sum_I\sum_V N\cdot X_{ivst}^{'''}\cdot M_{ist}\right)+K^{'}\quad\forall s,t \tag{20}$$

$$d_{st}\succ\left(\sum_I\sum_V N\cdot X_{ivst}^{'''}\cdot\left(M_{ist}-1\right)\right)+K^{'}\quad\forall s,t \tag{21}$$

$$NS_{it}\le\omega_{it}-1\quad\forall i,t \tag{22}$$

$$NS_{it}\ge(1-\omega_{it})(-N)\quad\forall i,t \tag{23}$$

$$\left(Y_{it}-1\right)(1-\omega_{it})NS_{it}=d_{it}\cdot T_{it}^{''}\quad\forall i,t \tag{24}$$

$$d_{it}=\sum_S\sum_V\sum_{b_s}A_{2s}^{b_s}\cdot X_{ivst}^{'''}\cdot d_{st}\quad\forall i,t \tag{25}$$

$$d_{st}=\sum_k X_{skt}\cdot d_{kt}\quad\forall s,t \tag{26}$$

$$T=T_{it}^{'}+T_{it}^{''}\quad\forall i,t \tag{27}$$

$$\sum_s X_{skt}=1\quad\forall k,t \tag{28}$$

$$\sum_{L} X_{kLt} = 1 \quad \forall k, t \tag{29}$$

$$\sum_{M} X_{LMt} = \sum_{b_L} A_{3L}^{b_L} \quad \forall L, t \tag{30}$$

$$\sum_{N} X_{LNt} = \sum_{b_L} A_{3L}^{b_L} \quad \forall L, t \tag{31}$$

$$\sum_{M} X_{Mit} = \sum_{b_i} A_{1I}^{b_i} \quad \forall i, t \tag{32}$$

$$\sum_{N} X_{N\lambda t} = \sum_{b_\lambda} A_{6\lambda}^{b_\lambda} \quad \forall \lambda, t \tag{33}$$

$$\sum_{\lambda} X_{\lambda it} = \sum_{b_i} A_{1I}^{b_i} \quad \forall i, t \tag{34}$$

$$\sum_{b_i} A_{1I}^{b_i} \leq 1 \quad \forall i \tag{35}$$

$$\sum_{b_s} A_{2s}^{b_s} \leq 1 \quad \forall s \tag{36}$$

$$\sum_{b_L} A_{3L}^{b_L} \leq 1 \quad \forall L \tag{37}$$

$$\sum_{b_M} A_{4M}^{b_M} \leq 1 \quad \forall M \tag{38}$$

$$\sum_{b_N} A_{5N}^{b_N} \leq 1 \quad \forall N \tag{39}$$

$$\sum_{b_\lambda} A_{6\lambda}^{b_\lambda} \leq 1 \quad \forall \lambda \tag{40}$$

$$\sum_{I} \sum_{V} \sum_{s'} X_{ivst}''' \cdot X_{ivs't} = \sum_{s'} A_{2s}^{b_s} \quad \forall s, t \tag{41}$$

$$\sum_{S}\sum_{v} X_{ivst} = \sum_{S}\sum_{v} X''_{ivst} \quad \forall i,t \tag{42}$$

$$X'''_{ivst} \le X_{ivst} + \sum_{s'} X'_{ivss't} + X''_{ivst} \quad \forall i,v,s,t \tag{43}$$

$$SP_2\left(\sum_{M} O_{Mit} + \sum_{\lambda} O_{\lambda it}\right) + SP_1(NS_{it} \cdot Y_{it}) \le \omega_i \quad \forall i,t \tag{44}$$

$$\sum_{S} O_{kst} = \sum_{L} O_{kLt} \quad \forall k,t \tag{45}$$

$$\sum_{k} O_{kLt} \cdot \delta \cdot \varepsilon = \sum_{N} O_{LNt} \quad \forall L,t \tag{46}$$

$$\sum_{k} O_{kLt} \cdot (1-\delta) \cdot \varepsilon = \sum_{M} O_{LMt} \quad \forall L,t \tag{47}$$

$$r \cdot \sum_{L} O_{LNt} = \sum_{\lambda} O_{N\lambda t} \quad \forall N,t \tag{48}$$

$$r' \cdot \sum_{N} O_{N\lambda t} = \sum_{I} O_{\lambda it} \quad \forall \lambda,t \tag{49}$$

$$\sum_{L} O_{LMt} = \sum_{i} O_{Mit} \quad \forall M,t \tag{50}$$

$$\sum_{K} O_{kLt} \le \omega_L \quad \forall L,t \tag{51}$$

$$\sum_{L} O_{LMt} \le \omega_M \quad \forall M,t \tag{52}$$

$$\sum_{L} O_{LNt} \le \omega_N \quad \forall N,t \tag{53}$$

$$\sum_{N} O_{N\lambda t} \le \omega_\lambda \quad \forall \lambda,t \tag{54}$$

$$Q_{it} \le F_{it} \quad \forall i,t \tag{55}$$

$$NS_{i(t-1)} - \left(1 - \alpha_{i(t-1)}\right) \cdot NS_{i(t-1)} \cdot \left(1 - Y_{i(t-1)}\right) \cdot \left(1 - \omega_{i(t-1)}\right) + Q_{it} \succ 0 \quad \forall i, t \tag{56}$$

$$O_{KLt} \leq X_{KLt} \cdot N \quad \forall K, L, t \tag{57}$$

$$O_{LMt} \leq X_{LMt} \cdot N \quad \forall L, M, t \tag{58}$$

$$O_{LNt} \leq X_{LNt} \cdot N \quad \forall L, N, t \tag{59}$$

$$O_{\lambda it} \leq X_{\lambda it} \cdot N \quad \forall \lambda, i, t \tag{60}$$

$$O_{N\lambda t} \leq X_{N\lambda t} \cdot N \quad \forall N, \lambda, t \tag{61}$$

$$O_{Mit} \leq X_{Mit} \cdot N \quad \forall M, i, t \tag{62}$$

$$O_{Kst} \leq X_{Kst} \cdot N \quad \forall K, s, t \tag{63}$$

$$A_{1I}^{b_i}, A_{2s}^{b_s}, A_{3L}^{b_L}, A_{4M}^{b_M}, A_{5N}^{b_N}, A_{6\lambda}^{b_\lambda}, X_{KST}, X_{KLT}, X_{LMT}, X_{LNT}, X_{\lambda iT}, X_{N\lambda t}, X_{Mit}, X_{ivst}, X''_{ivst}, X'''_{ivst}, X'_{ivs'st}, Y_{it}, \omega_{it}, M_{ist} \in [0, 1] \quad \forall i, s, v, k, L, M, N, t, \lambda \tag{64}$$

$$NS_{it}, Q_{it}, d_{st}, \eta'_{ivst}, \theta'_{ivst}, O_{sKt}, O_{kLt}, O_{LMt}, O_{LNt}, Z'_{it} \geq 0 \quad \forall i, s, v, k, L, M, N, t. \tag{65}$$

The first objective function minimizes the sum of maximum time between the centers of two subsequent stage dedicated together. The second objective function of the model minimizes the cost of centers construction with a certain capacity, inventory costs, shortage (lost sale and backlog), production, fixed cost of transportation vehicle of plant *i,* and operational costs of centers. Constraint (1) identifies the maximum time between factories and their last allocated distribution center on the route of the vehicle of that plant at levels between the factories and distribution centers. Constraint (2) identifies the maximum time between the distribution centers and their allocated customers at levels between the distribution centers and

customers. Constraint (3) identifies the maximum time between customers and their allocated collecting centers at levels between the customers and collecting centers. Constraint (4) identifies the maximum time between collecting centers and their allocated recovery centers at levels between the collecting centers and recovery centers. Constraint (5) identifies the maximum time between collecting centers and their allocated recycling centers at levels between the collecting centers and recycling centers. Constraint (6) represents the maximum time between plant i and its allocated supplier λ at levels between factories and suppliers.

Constraint (7) represents the maximum time between supplier λ and its allocated recycling centers at levels between suppliers and recycling centers. Constraint (8) represents the maximum time between plant i and its allocated recovery center M at levels between factories and recovery centers. Constraint (9) identifies the arrival time of vehicle ν of plant i to distribution center S at period t. Constraint (10) identifies the departure time of vehicle ν of plant i from distribution center S at period t. Constraint (11) means that the sum of shortages of dedicated distribution centers of plant i at period t is equal to the shortage at plant i at period t. Constraint (12) ensures the path continuity. If vehicle ν arrives a node, it must exit from it. Constraint (13) implies that vehicle of each plant at the start of its route only has one first visited distribution center. Constraint (14) implies that the amount of product transferred by vehicle ν of plant i must be at most equal to its capacity. Constraint (15) implies the costs of plant i, including inventory holding costs, shortage (lost sales and backlog), production, fixed cost of transportation vehicle of plant i, and transportation costs of plant i. Constraint (16) represents the maximum budget that is available for plant i at period t. Constraint (17) identifies the net inventory of plant i at the end of period t. Constraints (18) and (19) relate the net inventory of plant i at the end of period t with the binary variable Y_{it}. Constraints (20) and (21) represent the relationship of distribution center S at period t with constant k'.

Constraints (22) and (23) represent the relationship between the net inventory of plant i at the end of period t and binary variable w_{it}. Constraint (25) implies that the demand of plant i is equal to the total demands of distribution centers dedicated to it. Constraint (26) means that the demand of distribution center s is equal to the total demands of its dedicated customers. Constraint (27) ensures that the total time under the shortage and inventory state at each period is T. Constraint (28) ensures that each customer is served only by one distribution centers. Constraint (29) ensures that each customer

must be assigned to one collecting center. Constraint (30) means that if collecting center L is opened with capacity b_L, then it must be assigned definitely to a recovery center. Constraint (31) implies that if collecting center L is opened with capacity b_L, then it must be assigned certainly to a recycling center. Constraint (32) implies that if plant i is opened with capacity b_i, then it must be assigned to a recovery center. Constraint (33) implies that if supplier λ is opened with capacity b_λ, then it must be supplied with a recycling center. Constraint (34) implies that if plant i is opened with capacity b_i, then it must be assigned certainly to a supplier λ. Constraint (35) implies that if plant i is opened, then it certainly assigned a capacity. Constraint (36) implies that if distribution center s is opened, then it certainly assigned a capacity. Constraint (37) implies that if collecting center s is opened, then it certainly assigned a capacity.

Constraint (38) identifies that if recovery center M is opened, then it certainly assigned a capacity. Constraint (39) identifies that if recycling center N is opened, then it certainly assigned a capacity. Constraint (40) identifies that if supplier λ is opened, then it certainly assigned a capacity. Constraint (41) implies that if distribution center s is opened, then it is supplied with a plant and that plant has certainly a first met distribution center in its vehicle route. Constraint (42) identifies that if plant i visits the first distribution center in its route to distribution centers, then it certainly visits the last one. Constraint (43) means that if distribution center s is supplied with plant i, then that center is certainly on the vehicle route of that plant (the first distribution center on the route, or the last distribution center on the route, or the first and the last distribution centers on the route, or between the first and the last distribution centers on the route). Constraint (44) is about the maximum warehouse space of plant i. Constraint (45) implies that the number of products that delivered to customer k is delivered to its assigned collecting center. Constraints (46) and (47) imply that a part of collected products at collecting center L is sent to the recycling center and the other part is sent to a recovery center.

Constraint (48) ensures the balance between entrance and outputs at recycling centers using a transformation factor. Constraint (49) ensures the balance between entrance and outputs at supplier λ using a transformation factor. Constraint (50) ensures the entrance and output balance at recovery centers using a transformation factor. Constraint (51) shows that collected products at collecting center L are at most equal to its capacity. Constraints (52), (53), and (54) show the capacity constraint of recovery, recycling, and supplier λ. Constraint (55) is about the maximum production quantity of

plant i at period t. Constraint (56) means that the sum of production of plant i at each period with the net inventory of its previous period is positive. Constraints (57)–(63) show that if two centers at two consecutive levels are assigned together, then products, materials, or parts are transferred between them. Ultimately, constraints (64) and (65) represent the type of the variables.

PROPOSED SOLUTION METHODS

To show the applicability and validity of the presented model, we have solved a number of small-sized test problems by the ε-constraint method through a branch-and-bound module in the GAMS (General Algebraic Modeling System) software. Because the mentioned model is NP-hard, NSGA-II and MOPSO algorithms have been used to solve large-scale problems. To show the efficiency of the proposed algorithms, their results have been compared with the results obtained by the ε-constraint method in small-sized test problems. Notably, all the computations have been performed on a laptop with 2.09 GHz CPU and 1.92 GB RAM. Furthermore, the mentioned algorithms are coded in MATLAB R2009a.

Non-dominated Sorting Genetic Algorithm (NSGA-II)

The initial population consists of a number of solutions generated randomly. Matrices are used to represent solutions. Each solution contains several matrices designed in accordance with model outputs. For example, for variable Xijl1tXijl1t, a four-dimensional matrix $I \times J \times l_{1} \times T$ is defined. In the same way, matrices are defined for other outputs. In addition, after the generation of each solution, the constraints are checked and the solution is accepted if all constraints are satisfied, and otherwise, it is rejected. In the proposed algorithm, the objective value is used for fitness function to evaluation of solutions.

The selection strategy of parent population is done by the use of crowded tournament selection operator. In crowded tournament selection operator, solution i dominates solution j in the tournament if and only if one of the following conditions are met:

- Solution i has a better rank.
- Solutions i and j be of the same rank that solution i has a better crowd distance to solution j.

The crossover operator used in this algorithm is selected by a guideline matrix. This guideline matrix includes binary elements, and for each part, there is a separate chromosome with an equal dimension to that part. Thus, for each element of each part of chromosome, there is a corresponding element in guideline matrix. To produce a new offspring, if the corresponding element in guideline matrix is 1, then the related values of that element are replaced in two parents, otherwise, that element will be left unchanged. In proposed algorithm for mutation operator, a number of chromosome elements are randomly selected and their values are generated randomly.

The mechanism of Elitism operator is from lower fronts toward higher ones and among the solutions of one front is from larger crowd distance to smaller one. In the corrective procedure, the produced off springs are compared to the solutions in the last non-dominated front. If the produced offspring is dominated with none of the solutions of the last front, it is allowed to enter the new generation. Achievement to a certain number of repetitions has been considered as the stopping criterion.

Multi-objective Particle Swarm Optimization (MOPSO)

The representation of the solutions is the same as the structure presented in the NSGA-II section. To evaluate existing solutions in the population and integrate the objective functions, the general procedure of the algorithm is as follows:

The new position and velocity of the particles are calculated by Eqs. (66) and (67), and with regard to the objective function, a competence value is assigned. This process continues to reach the stopping criterion, and ultimately, the best position found by the particles is presented as the solution:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{66}$$

$$V_i^{t+1} = \omega V_i^t + \varphi_1 r_1 \left(Pbest_i^t - X_i^t\right) + \varphi_2 r_2 \left(Gbest^t - X_i^t\right). \tag{67}$$

Achievement to a certain number of repetitions has been considered as the stopping criterion.

Comparing Factors of Multi-objective Evolutionary Algorithms

- *Number of Pareto solutions (NPS)* The number of non-dominated solutions shows the number of alternatives that can be reported to the decision maker.
- *Mean ideal distance (MID)* This measure is a measurement of Pareto solutions closeness to the ideal solution (f_1-Best and f_2-Best).
- *Spacing metric (SM)* This measure identifies the uniformity of the width of non-dominated solutions.
- *Quality metric (QM)* The percent (rate) of the dominance of Pareto solutions of each algorithm indicate the solution quality of algorithms together.

COMPUTATIONAL RESULTS

According to the given assumptions and parameters in the proposed mathematical model and the size of existing problems in the literature, some small-, medium-, and large-scale problems have been randomly defined. To solve the problem, meta-heuristic algorithms NSGA-II and MOPSO have been applied and to validate results of the proposed algorithms in small-size problems, they are compared to the results of ε-constraint method obtained by GAMS software. To increase the efficiency of these algorithms, a Taguchi method is used to tune the parameters.

Setting the parameters of the proposed meta-heuristics

To increase the efficiency of the proposed meta-heuristic algorithm, we set some of their input parameters by use of the Taguchi method. The levels of factors of NSGA-II and MOPSO algorithms are demonstrated in Table 1.

Table 1: Factors of NSGA-II and MOPSO algorithms with their levels

	NSGA-II				MOPSO			
	Pop-size	Iteration	Crossover RATE	Mutation rate	Pop-size	Iteration	V-max	Inertia weight
Level 1	80	200	0.75	0.05	80	170	0.2	0.6
Level 2	100	160	0.85	0.1	100	140	0.3	0.8
Level 3	125	130	0.95	0.2	120	115	0.5	0.99

The required degrees of freedom for the algorithms corresponding to these four factors is $4 \times 2 + 1 = 9$. The most appropriate element for both NSGA-II and MOPSO algorithms is accordance with Table 2 that have necessary conditions for setting up algorithms parameters.

Table 2: Orthogonal array *L*

Trial	*A*	*B*	*C*	*D*
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

As it can be seen from Fig. 2, deviation reduction for this algorithm is when the parameters are set as follows: the population size is on level 3, the number of generation is on level 2, the crossover rate is on level 3, the and mutation rate is on level 1.

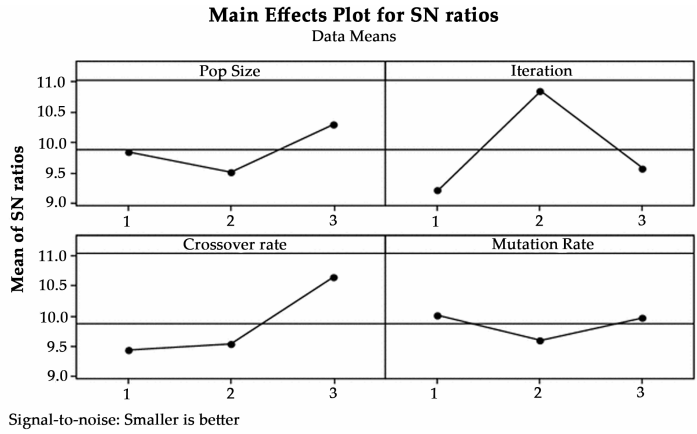


Figure 2: Chart of S/N rate of objective functions at different levels of NSGA-II algorithm factors.

Values of S/N rate for different levels of MOPSO parameters are presented in Fig. 3. As obvious in the figure, deviation reduction in this algorithm is when its parameters are set as follows: the number of initial particles on level 1, the number of repetitions on level 2, the maximum velocity of particles on level 1, and the inertia weight on level 3.

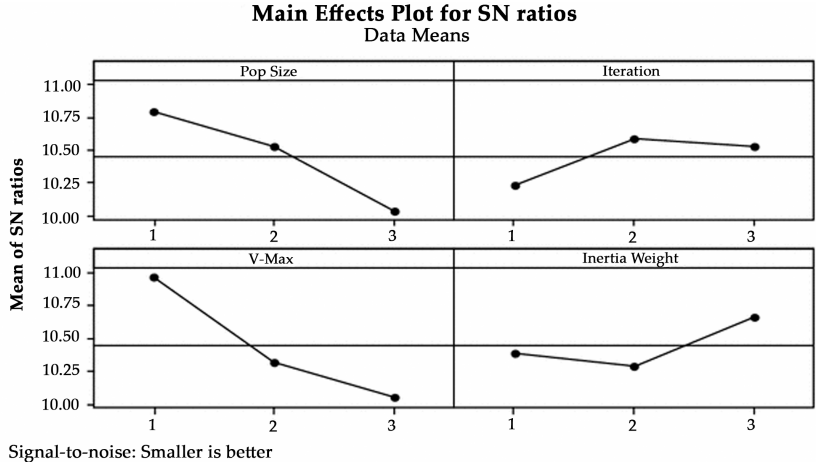


Figure 3: Chart of S/N rate of objective functions at different levels of MOPSO algorithm factors.

$\phi 1\phi 1$ and $\phi 2\phi 2$ are constants named cognitive and social parameters, respectively. In this study, the two parameters have been considered equal to 2.

Comparison of the Results for Small-sized Problem

In this section, a problem has been created with two suppliers, two distributers, three customers, two collecting centers, two recycling centers, two recovery centers, and three capacity levels for each of the centers, three types of vehicles, and three period time. We consider the first objective function as a baseline and the epsilon value equal to 28.9. For validation of presented NSGA-II and MOPSO algorithms, the problem has been solved by ε–constraints method and two proposed algorithms and the obtained points in their final Pareto front have been compared. The result of this comparison is demonstrated in Fig. 4. As it can be seen, two algorithms have conformity with ε–constraints method at some points and for other points are not dominated that indicates the validity of the proposed algorithms.

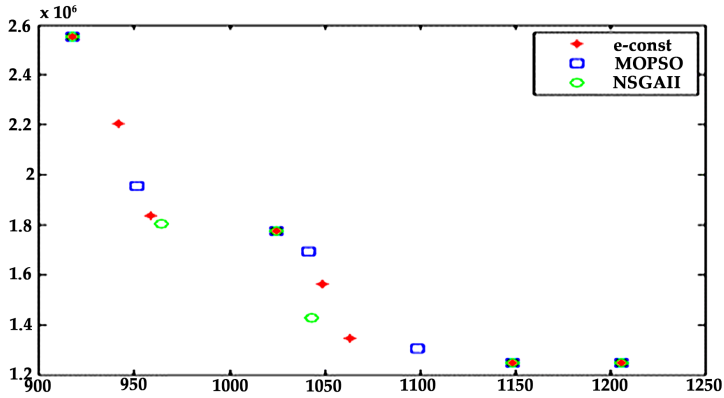


Figure 4: Obtained Pareto front of ε-constraints, NSGA-II and MOPSO methods.

Comparison of the Results for the Proposed Algorithms

In total, 27 problems have been solved and their results, as presented in Tables 3, 4, and 5. The considered range of the parameters is as follows:

Table 3: Results of NSGA-II and MOPSO algorithms for small-sized problems

Instance	No. of plants	No. of DCs	No. of customers' centers	No. of collecting centers	No. of recovery centers	No. of recycle centers	No. of period	QM		SM		MID		NPS	
								NSGA-II	MOSPO	NSGA-II	MOSPO	NSGA-II	MOSPO	NSGA-II	MOSPO
1	2	4	5	3	2	3	1	4.8	4	0.703617	0.734881	0.548263	0.387083	0.166667	0.833333
2	2	4	5	3	2	3	2	6.2	4	0.657624	0.947509	0.684212	0.62821	0.272222	0.727778
3	2	4	5	3	2	3	1	7.2	5.2	0.848426	0.880136	0.748178	1.10146	0.249048	0.750952
4	2	4	5	3	2	3	2	6.6	6.4	0.676526	0.884192	0.749638	0.828011	0.414394	0.585606
5	2	4	5	3	2	3	1	5.4	6	0.759956	0.807461	0.637741	0.758701	0.1	0.9
6	2	4	5	4	2	3	2	5.4	9.8	0.712331	0.804647	0.702026	0.748817	0.357143	0.642857
7	2	4	7	4	2	3	1	5.2	5.8	0.871974	0.754717	0.707911	0.606264	0.051515	0.948485
8	2	4	7	4	2	3	2	4.8	4.8	0.822937	0.919584	0.886044	0.663851	0.228571	0.771429
9	2	4	7	4	2	3	3	5.2	6.6	0.736214	0.97694	0.724271	0.684676	0.355556	0.644444

Table 4: Results of NSGA-II and MOPSO algorithms for medium-sized problems

Instance	No. of plants	No. of DCs	No. of customers' centers	No. of collecting centers	No. of recovery centers	No. of recycle centers	No. of period	QM		SM		MID		NPS	
								NSGA-II	MOSPO	NSGA-II	MOSPO	NSGA-II	MOSPO	NSGA-II	MOSPO
1	3	7	7	7	5	4	1	5.2	4.4	0.675866	0.795232	0.434928	0.898548	0.165714	0.834286
2	3	7	7	7	5	4	2	5.2	6.8	0.621076	0.838057	0.682714	0.809564	0.203571	0.796429
3	3	7	7	7	5	4	3	5.6	7.8	0.710011	0.806778	0.78044	0.756451	0.07	0.93
4	6	8	10	8	6	5	1	6	7.6	0.862634	0.935091	0.686637	1.061099	0.177662	0.822338
5	6	8	10	8	6	5	2	5.6	6.2	0.822928	0.824759	0.656155	0.677683	0.272143	0.727857
6	6	8	10	8	6	5	3	6.2	8.6	0.695284	0.868963	0.592814	0.705304	0.251991	0.748009
7	7	9	15	9	7	7	1	6.6	5.8	0.608399	0.70193	0.494604	0.519271	0.398095	0.601905
8	7	9	15	9	7	7	2	6.4	4	0.777689	0.922406	0.563588	0.410235	0.230952	0.769048
9	7	9	15	9	7	7	3	5.8	7.6	0.689365	1.008697	0.729985	0.678227	0.332308	0.667692

Table 5: Results of NSGA-II and MOPSO algorithms for large-sized problems

Instance	No. of plants	No. of DCs	No. of customers' centers	No. of collecting centers	No. of recovery centers	No. of recycle centers	No. of period	QM		SM		MID		NPS	
								NSGA-II	MOSPO	NSGA-II	MOSPO	NSGA-II	MOSPO	NSGA-II	MOSPO
1	10	20	30	16	7	6	1	5.8	5.8	0.689784	0.917126	0.631174	0.777097	0.303889	0.696111
2	10	20	30	16	7	6	2	6.4	8.6	0.589552	0.81157	0.64253	0.643982	0.261062	0.738938
3	10	20	30	16	7	6	3	6.4	5.6	0.74747	0.916762	0.698582	0.750484	0.385	0.615
4	15	40	70	35	12	10	1	6.6	6.6	0.557903	0.674509	0.582228	0.687046	0.388889	0.611111
5	15	40	70	35	12	10	2	5.2	6.8	0.634788	0.930261	0.551176	0.747531	0.188571	0.811429
6	15	40	70	35	12	10	3	5.4	6.2	0.593561	0.772562	0.672484	0.621205	0.415152	0.584848
7	15	45	90	40	15	13	1	1.2	4.2	0.886493	0.809355	0.625958	0.535442	0.083333	0.916667
8	15	45	90	40	15	13	2	3.6	5.4	0.53883	0.814527	0.509658	0.687479	0.238333	0.761667
9	15	45	90	40	15	13	3	3.4	4.8	0.58224	0.846821	0.604821	0.804037	0.404444	0.395556

- Transportation time of products among all of the centers has been considered in the uniform range [1…100].
- Required time for unloading has been considered in the uniform range [5…15].
- Annual costs of opening centers for all centers have been considered in the uniform range [1…40].
- Vehicles' cost has been considered in the uniform range [3000…6000].
- Centers' operational costs have been considered in the uniform range [1…20].
- Inventory holding cost has been considered in the uniform range [20…50].
- Backlog cost has been considered in the uniform range [30…50].
- Lost sale cost has been considered in the uniform range [50…100].
- Customers' demand has been considered in the uniform range [20…100].
- The speed of vehicles has been considered in the uniform range [20…50].
- Number of existed parts in a product has been considered 3.
- Total centers' capacity at all levels has been considered in the uniform range [1…1000].
- Price increase factor for the operations at each center has been considered in the uniform range [10…30%].
- The percent of existed parts in a product that is recyclable has been considered 0.5.

Results of the proposed algorithms according to NPS, MID, SM, and QM criteria are shown in Figs. 5, 6, 7, 8, 9, 10, 11, and 12. By comparing two algorithms according to the NPS criterion, it can be understood for medium- and large-size dimensions that in most cases, MOPSO algorithm has better performance than NSGA-II algorithm, especially for the growing scale of the problem. The average distance from the ideal point has a smaller value in NSGA-II algorithm that identifies its better performance to MOPSO algorithm. In comparison according to the SM criterion, the performance of NSGA-II algorithm is completely better for small size and is better for more than 60% for large-size problems to the MOPSO algorithm. From the comparison of the algorithms in the terms of QM criterion, it can be seen that MOPSO provides better solutions to in all problems, without exception.

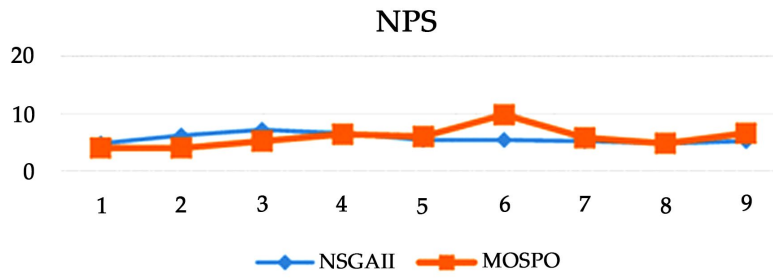


Figure 5: Comparison of the algorithms according to the NPS evaluation criterion for small-sized problems.

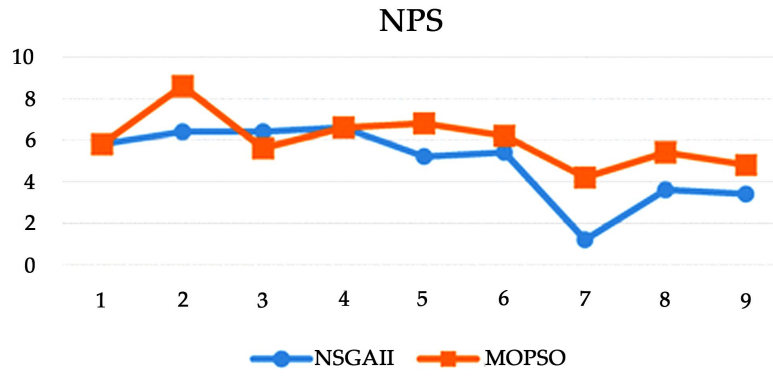


Figure 6: Comparison of the algorithms according to the NPS evaluation criterion for large-sized problems.

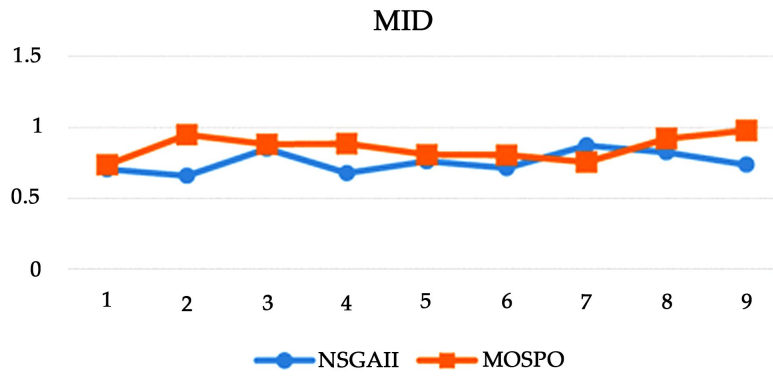


Figure 7: Comparison of the algorithms according to the MID evaluation criterion for small-sized problems.

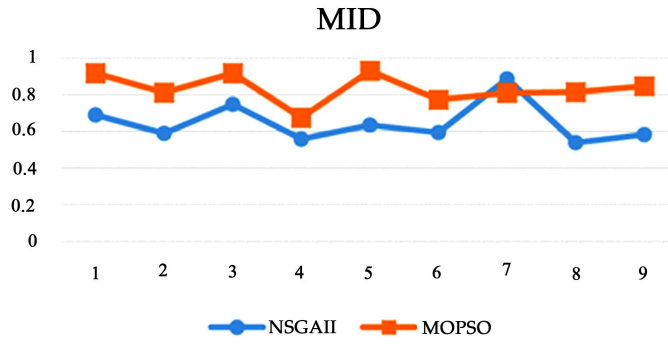


Figure 8: Comparison of the algorithms according to the MID evaluation criterion for large-sized problems.

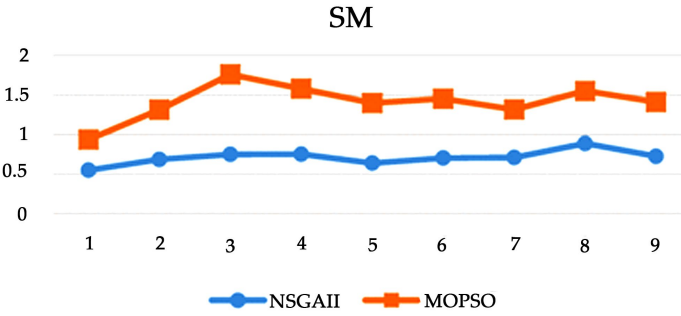


Figure 9: Comparison of the algorithms according to the SM evaluation criterion for small-sized problems.

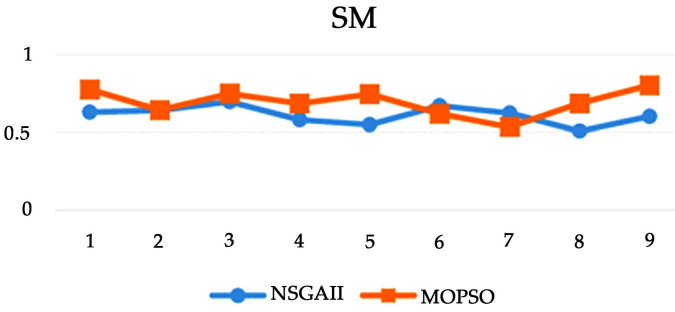


Figure 10: Comparison of the algorithms according to the SM evaluation criterion for large-sized problems.

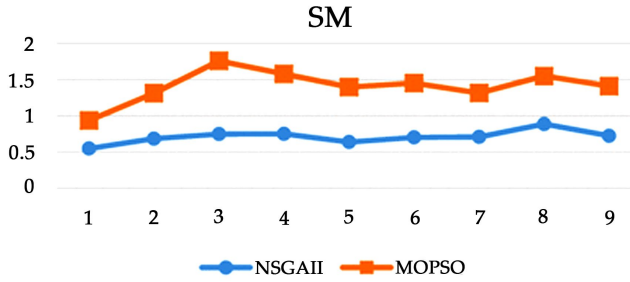


Figure 11: Comparison of the algorithms according to the QM evaluation criterion for small-sized problems.

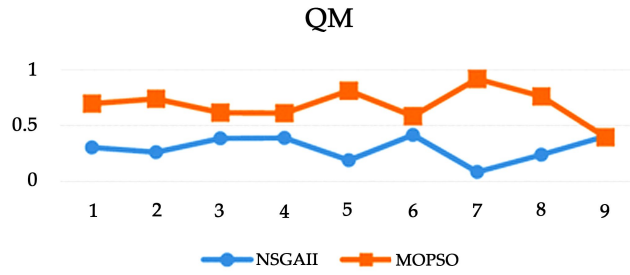


Figure 12: Comparison of the algorithms according to the QM evaluation criterion for large-sized problems.

For more results analysis, the expected value chart along with LSD distances has been presented for the two proposed algorithms in Figs. 13, 14, and 15 according to the NPS, MID, and SM criteria. The obtained results of MID criterion identify the existence of significant differences between them and statistical superiority of the MOPSO algorithm.

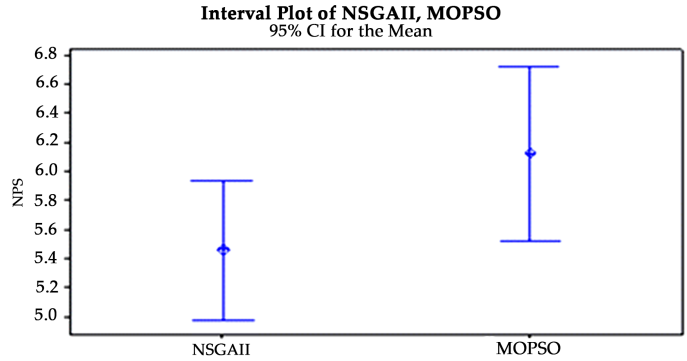


Figure 13: Expected value and LSD distances chart for proposed algorithms according to the NPS criterion.

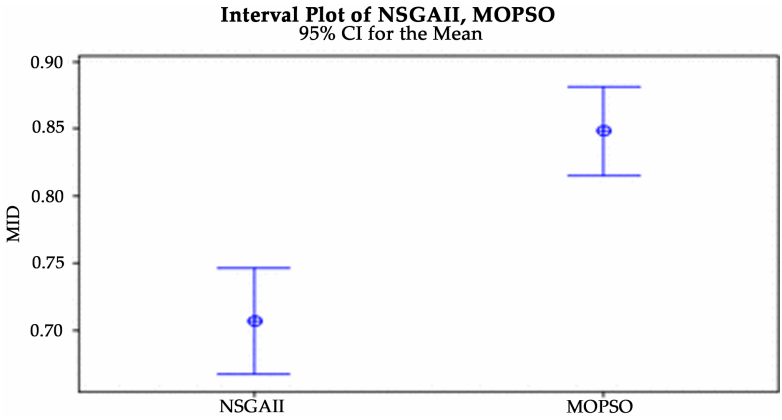


Figure 14: Expected value and LSD distances chart for proposed algorithms according to the MID criterion.

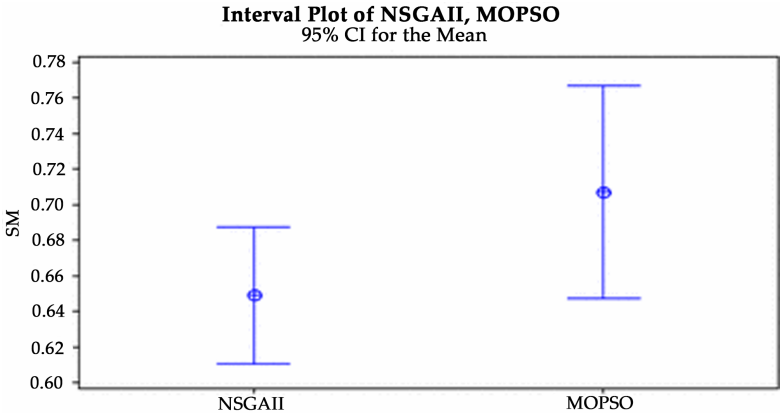


Figure 15: Expected value and LSD distances chart for proposed algorithms according to the SM criterion.

For a closer look at the results of two algorithms, we test the following hypothesis according to two SM and NPS criteria that their difference is not substantial. The *t* test is applied for this purpose.

Test 1: (in terms of NPS)

H_0 **:** The average NPS of MOPSO = the average NPS of NSGA-II.

H_1 **:** The average NPS of MOPSO < the average NPS of NSGA-II.

The statistic method of this test is as follows:

$$t_\circ = \frac{\overline{\mathrm{NPS}}_{\mathrm{MOPSO}} - \overline{\mathrm{NPS}}_{\mathrm{NSGA-II}}}{S_p \cdot \sqrt{\frac{1}{nX_{\mathrm{MOPSO}}} + \frac{1}{nY_{\mathrm{NSGA-II}}}}} = 2.02134. \tag{68}$$

The acceptance limits are as follows:

$$\left[-t_{\alpha, nX_{\mathrm{MOPSO}} + nY_{\mathrm{NSGA-II}} - 2}, +\infty\right) = \left[-1.6762 + \infty\right).$$

Because the statistic is in the acceptance region and we accept the null hypothesis at confidence level 95%. This emphasizes that the NPS average for MOPSO is statistically higher than NSGA-II. In other words, the number of non-dominated solutions of MOPSO method is more.

Test 1: (in terms of SM)

$\boldsymbol{H_0}$**:** The average SM of MOPSO = the average SM of NSGA-II.

$\boldsymbol{H_1}$**:** The average SM of MOPSO < the average SM of NSGA-II.

The statistic method of this test is as follows:

$$t_\circ = \frac{\overline{\mathrm{SM}}_{\mathrm{MOPSO}} - \overline{\mathrm{SM}}_{\mathrm{NSGA-II}}}{S_p \cdot \sqrt{\frac{1}{nX_{\mathrm{MOPSO}}} + \frac{1}{nY_{\mathrm{NSGA-II}}}}} = 0.756. \tag{69}$$

The acceptance limits are as follows:

$$\left[-t_{\alpha, nX_{\mathrm{MOPSO}} + nY_{\mathrm{NSGA-II}} - 2}, +\infty\right) = \left[-1.6762 + \infty\right).$$

Because the statistic is in the acceptance region and we accept the null hypothesis at confidence level 95%. This emphasizes that the SM average for MOPSO is statistically higher than NSGA-II.

The charts and statistical analysis of two algorithms MOPSO and NSGA-II identify that in the considered problem, MOPSO is better than NSGA-II in the terms of NPS and QM criterions. On the other hand, NSGA-II is superior in the terms of MID and SM criterions.

Sensitivity Analysis

To investigate the sensitivity analysis, the following two small- and large-sized test instances are considered to show the impact of parameter V'_{iv} on the objective functions of the proposed model.

Instance	No. of suppliers	No. of plants	No. of DCs	No. of customers' centers	No. of collecting centers	No. of recovery centers	No. of recycle centers	No. of period
1	5	6	8	10	8	6	5	2
2	10	15	40	70	35	12	10	2

Figures 16, 17 illustrate the sensitivity of the first and second objective functions on parameter V'_{iv} (i.e., average speed of vehicle ν of plant i) for above two small- and large-sized instances, respectively. As shown in these figures, when V'_{iv} (i.e., average speed of vehicle ν of plant i) is increased, the value of the first objective function (i.e., time) is decreased. By increasing V'_{iv}, the value of the second objective function (i.e., cost) is increased.

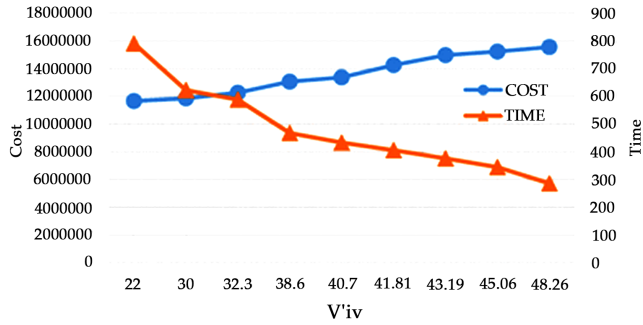


Figure 16: Sensitivity of the time and cost on parameter V'_{iv} for small-sized instances.

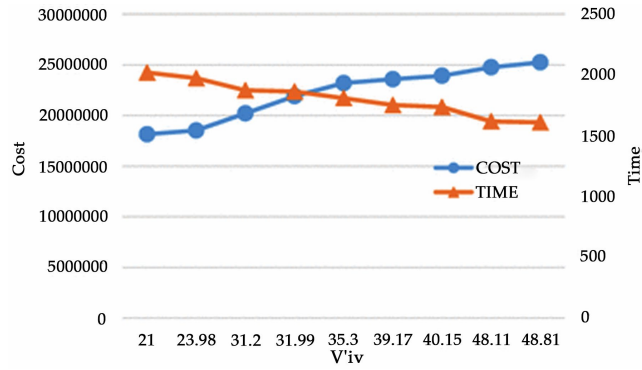


Figure 17: Sensitivity of the time and cost on parameter V'_{iv} for large-sized instances.

CONCLUSION

The proposed model represents a location–routing–inventory problem in a multi-period closed-loop supply chain with the consideration of shortage, price increase factor, arrival time to distribution centers, and departure time of them, so that the cost and maximum transportation time of the chain are minimized. The proposed model includes multiple producers, distribution centers, customers, collecting centers, recovery centers, and recycling centers. The percent of backlog and lost sales of the total shortage at each period is identified according to the predefined policies. Due to the NP-hard nature of the problem, NSGA-II and MOPSO algorithms have been applied and Taguchi approach has been used to increase the efficiency of these algorithms. A number of small-, medium-, and large-scale problems have been generated randomly. To evaluate the performance of the proposed algorithms, the results of produced small-size test problems with the results of ε-constraint method solved by the GAMS software. Finally, to identify the performance of the proposed algorithms, their performances were compared. The charts and statistical analysis of two algorithms (i.e., MOPSO and NSGA-II) identify that in the considered problem, MOPSO is better than NSGA-II in the terms of NPS and QM criterions. On the other hand, NSGA-II is superior in the terms of MID and SM criterions. Considering flexible time window, probabilistic nature for the input parameters, developing a model by considering all-unit and incremental discount policies, and applying and developing other meta-heuristic algorithms for large-scale multi-objective problems are suggested for future research.

REFERENCES

1. Ahmadi Javid A, Azad N (2010) Incorporating location, routing and inventory decisions in supply chain network design. Transp Res Part E Logist Transp Rev 46:582–597
2. Ahmadi Javid A, Seddighi A (2012) A location–routing–inventory model for designing multisource distribution networks. Eng Optim 44(6):637–656
3. Ahmadizar F, Zeynivand M, Arkat J (2015) Two-level vehicle routing with cross-docking in a three-echelon supply chain: a genetic algorithm approach. Appl Math Model 39(22):7065–7081
4. Aydin R, Kwong CK, Ji P (2016) Coordination of the closed-loop supply chain for product line design with consideration of remanufactured products. J Clean Prod 114:286–298
5. Chen YW, Ch Wang L, Wang A, Chen TL (2017) A particle swarm approach for optimizing a multi-stage closed loop supply chain for the solar cell industry. Robot Comput Integr Manuf 43:111–123
6. Gorji MH, Setak M, Karimi H (2014) Optimizing inventory decisions in a two-level supply chain with order quantity constraints. Appl Math Model 38:814–827
7. Guerrero WJ, Prodhon C, Velasco N, Amaya CA (2013) Hybrid heuristic for the inventory location–routing problem with deterministic demand. Int J Prod Econ 146(1):359–370
8. Hiassat A, Diabat A (2011) A location–inventory–routing problem with perishable products. In: Proceedings of the 41st international conference on computers and industrial engineering, pp 386–391
9. Kadambala DK, Subramanian N, Tiwari MK, Abdulrahman M, Liu Ch (2017) Closed loop supply chain networks: designs for energy and time value efficiency. Int J Prod Econ 183:382–393
10. Mousavi SM, Alikar N, Akhavan Niaki ST, Bahreininejad A (2015) Optimizing a location allocation–inventory problem in a two-echelon supply chain network: a modified Fruit Fly optimization algorithm. Comput Ind Eng 87:543–560
11. Nekooghadirli N, Tavakkoli-Moghaddam R, Ghezavati VR, Javanmard Sh (2014) Solving a new bi-objective location–routing–inventory problem in a distribution network by meta-heuristics. Comput Ind Eng 76:204–221

12. Shen ZJM, Qi L (2007) Incorporating inventory and routing costs in strategic location models. Eur J Oper Res 179(2):372–389
13. Tang J, Ji Sh, Jiang L (2016) The design of a sustainable location–routing–inventory model considering consumer environmental behavior. J Sustain 8(3):211
14. Xia M (2013) Integrated supply chain network design: location, transportation, routing and inventory decisions. PhD Dissertation, Department of Industrial Engineering, Arizona State University
15. Zhalechian M, Tavakkoli-Moghaddam R, Zahiri B, Mohammadi M (2016) Sustainable design of a closed-loop location–routing–inventory supply chain network under mixed uncertainty. Transp Res Part E Logist Transp Rev 89:182–214
16. Zhang Y, Qi M, Miao L, Liu E (2014) Hybrid meta-heuristic solutions to inventory location routing problem. Transp Res Part E Logist Transp Rev 70:305–323

Chapter

9

INFORMATION SHARING SYSTEMS AND TEAMWORK BETWEEN SUB-TEAMS: A MATHEMATICAL MODELING PERSPECTIVE

Hamid Tohidi[1] , Alireza Namdari[2] , Thomas K. Keyser[2] , Julie Drzymalski[3]

[1] College of Industrial Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran
[2] College of Engineering, Industrial Engineering Department, Western New England University, Springfield, MA, USA
[3] Drexel University, Philadelphia, PA 19104, USA

ABSTRACT

Teamwork contributes to a considerable improvement in quality and quantity of the ultimate outcome. Collaboration and alliance between team members bring a substantial progress for any business. However, it is imperative to acquire an appropriate team since many factors must be considered in this

Citation: (APA): Rabbani, M., Ramezankhani, F., Giahi, R., & Farshbaf-Geranmayeh, A. (2016). Biofuel supply chain considering depreciation cost of installed plants. Journal of Industrial Engineering International, 12(2), 221-235. (8 pages).

regard. Team size may represent the effectiveness of a team and it is of paramount importance to determine what the ideal team size exactly should be. In addition, information technology increasingly plays a differentiating role in productivity and adopting appropriate information sharing systems may contribute to improvement in efficiency especially in competitive markets when there are numerous producers that compete with each other. The significance of transmitting information to individuals is inevitable to assure an improvement in team performance. In this paper, a model of teamwork and its organizational structure are presented. Furthermore, a mathematical model is proposed in order to characterize a group of sub-teams according to two criteria: team size and information technology. The effect of information technology on performance of team and sub-teams as well as optimum size of those team and sub-teams from a productivity perspective are studied. Moreover, a quantitative sensitivity analysis is presented in order to analyze the interaction between these two factors through a sharing system.

INTRODUCTION

Productivity is a proof of total efficiency of production process and also a subject of maximization. It is determined by comparing the quantity of output and input and is also considered to be a significant measure of any economy, industry, and company's development. However, it requires an appropriate identification of real inputs and outputs within a business. Productivity is one of the considerable concerns of engineering management, so that it has been causing companies to follow procedures of collecting and analyzing data in order to evaluate their performance. Productivity improvement stems from a certain degree of complex interaction among factors. Teamwork and IT are two decisive factors which may cause immediate effect on the way productivity can be improved. Investing in ICT capital increases firm productivity by increasing the productivity of labor (Kılıçaslan et al. 2017).

Historically, teamwork has been defined as a process of working collaboratively within a group of individuals when team members pursue an identified goal. Teamwork is an integral part of progress in today's world. It has increasingly become prevalent among enterprises to benefit from teamwork. However, team members play a prominent role in consequences of teamwork. Every team member has particular responsibilities in order to accomplish tasks.

Information technology (IT), typically, refers to a set of applications to transmit, save, recover, and report data in the context of a business. However, it is mistakenly used in reference to personal or home computers. It actually involves all facets of managing information, data manipulation, and data storage architectures and methodologies. IT may contribute to improving organizational performance and productivity by assuming different variables. IT is a broad subject concerned with a range of attributes from personal computing and networking to information sharing system (ISS) within an organization. ISS consists of all layers of system from hardware to database and data management techniques. It is shown that the impact of IT capital on productivity is larger by about 25–50% than that of conventional capital. This contribution of IT capital is higher than that of non-ICT capital for small sized and low-tech firms (Kılıçaslan et al. 2017).

Team size is an effective parameter in teamwork. Studies in this field have shown that as team size increases the outcome will improve. However, team size contributes to productivity, but after a certain point the law of diminishing returns occurs which means adding to the team members will not result in a better team performance and improvement in productivity because of the irrational additional team size. Value-added analysis may be the solution in this regard.

In this research, a mathematical model is proposed to explain how teamwork may affect the productivity while considering information technology and optimized size of team and sub-teams as two effective factors. Teamwork may be processed within either a team or a group of sub-teams. In this study, a team with a group of sub-teams is presented. Furthermore, the size of team and sub-teams are investigated.

The related work is categorized according to two primary themes: productivity affected by teamwork and productivity affected by information technology. The following subsections address the mentioned research interests.

Productivity affected by Teamwork

Literatures on efficiency and productivity mainly focus on relations between teamwork and productivity and the importance of team size is not addressed. Stewart and Barrick (2000) examined data from many different teams including individuals and supervisors to resolve the appropriate structure. They studied the relationships between all characteristics and performances

for both conceptual and behavioral tasks and how the nature of the tasks may affect the consequences. Salas et al. (2008) reviewed the developments in team performance in five recent decades. They studied the shared cognition, team training, and task environments mainly from a human factor perspective. Moses and Stahelski (1999) studied the relation between productivity in an aluminum plant and problem-solving teams. Five productivity measures and four time periods in 1980s and 1990s were analyzed and significant and non-significant changes between the time periods were evaluated. The results were compared with three factors, technology improvements, changes in the price of finished aluminum, and changes in the number of employees. It was concluded that the study was not affected by those factors. Hatcher and Ross (1991) used different methodologies to analyze the changes in a transition from individual piecework plan to a gain-sharing plan at a company. The data observed in 4 years of operating presented a decrease in grievances and increase in final quality. Galegher and Kraut (1994) studied contingency theory to prove the difficulties of computer-based communication in order to reach complex collaborative work. A group of 67 MBA students was considered to do two writing projects in three different conditions; Computer, Computer plus Phone, and Face-to-Face. That study presented the difficulty of tasks which involve ambiguous goals, multiple perspectives, and multi-interpretation information using contingency hypothesis. Powell (2000) modeled a production process including variable processing times for different tasks in order to determine the optimal size of teams. In this research, the conditions under which assigning small tasks to individuals in comparison with assigning complex tasks to large teams were addressed. It was found that depending on the parameters different structures may be preferred.

Productivity affected by IT

IT has received increasing academic attention in the last two decades. Explained ahead, improving IT may cause improvement in efficiency. Bharadwaj (2000) studied IT capabilities and firm performance based on experiments by using a matched-sample methodology and ratings. IT resources in the area of firm were categorized into IT infrastructure, human IT resources, and IT-enabled intangibles. It was demonstrated that firms with high IT capability may contribute to cost-based performance measures. Whelan (2002) examined the importance of IT in general and computer in particular in productivity, calculated the computer-usage effect in US economic growth, and developed a theoretical framework to study the technological obsolescence. Bartel et

al. (2005) presented new empirical indications regarding the investments in new computer-based IT and productivity. In this research, a set of data was reviewed to examine the effects of new IT on production innovation, process improvements, employee skills, and work practices. The authors showed how new IT adoption may be defined more than new equipment installation. Furthermore, IT was studied as a factor which alters business strategies, improves the efficiency of production process, and increases the skill requirements of members. Badescu and Garcés-Ayerbe (2009) collected data from 341 medium size and large firms to evaluate the effects of investment in IT on productivity by using a Cobb–Douglas function. In this research, the effect of IT was categorized into firm-specific and period-specific and a significant improvement in productivity derived from IT was not observed within the defined time periods. Dehning and Richardson (2002) developed a model to assess investment in IT based on the data gathered from firm accounting and market performance. However, the relation between IT and business process on one hand, and business process and firm performance on the other hand were examined. Furthermore, the effects of contextual factors on performance of IT and IT management on performance of firm were reviewed. Wu et al. (2014) focused on two main concerns, information sharing and collaborative effort but in a supply chain context and identified the rudiments of implementing them in terms of issues related to partner exchanges including trust, commitment, reciprocity, and power. Finally, a positive relation between set-based variables, information sharing and collaboration, and supply chain performance was concluded. Martínez-Lorente et al. (2004) presented a survey-based research on the significant relationships between information technology (IT) and total quality management (TQM). However, the survey was conducted within the largest industrial companies in Spain and the results showed that intensive IT users observe the effect of IT on their TQM dimensions more significantly. Shao and Lin (2016) evaluated the performance of IT service industries of Organization of Economic Cooperation and Development (OECD) countries by using Malmquist Productivity Index (MPI) as a metric and Stochastic Production Frontier (SPF) as an approach and an annual rate of 7.4% growth in productivity in IT service industries was observed. The reported growth in productivity was mainly caused by technological advance process of IT services. Jones et al. (2011) studied the impact of implementing an Enterprise Resource Planning (ERP) system in a retail chain and firm and employee effects of an appropriate information system. It was found that employees

need to be informed of implementation of such an information system and the negative outcomes associated with them.

These literatures on productivity only deal with the approaches and models considering either IT or teamwork, and did not present the effect of both issues on production. Explained ahead, there are few researches considering both IT and teamwork at the same time. Tohidi and Tarokh (2006) studied the effect of changing IT on team output. They described the best coordination to increase team output and provided a good example of a team including two assembly lines and a supervisor. In addition, they categorized the factors which impact on coordination to hardware and software. In their research, they addressed the appropriate combination of those two factors from the output perspective. They proposed and analyzed a mathematical model in which productivity is driven by teamwork and information technology. They presented a sensitivity analysis to examine how IT and team size may increase the ultimate output. To the best of our knowledge this work is one of few studies about productivity considering both IT and teamwork. However, a structured model for a team with sub-teams has been lacking. The rest of the paper is organized as follows: first of all a model of teamwork is proposed. Then a mathematical model is produced. After that, interaction between IT and sub-teams is presented. To show the results of the paper as well as possible a sensitivity analysis is prepared in the next step. Finally, conclusion remarks are presented.

THE PROPOSED MODEL AND PROBLEM STATEMENT

Each team member affects output by collaborating with other team members to pursue team objectives. Coordination between team members is expected to lead to a considerable output. The issue of concern is how we can provide the best coordinated teamwork to improve output. With additional effort, according to the law of diminishing returns it will result in a decrease in output. This should be studied in order to determine the optimum size of each sub-team, and value of information related to each individual to benefit from a better collaboration.

As depicted in Fig. 1, in the presented model we assume a team size of n which contains m subgroups with n_s individuals in each of them producing product X.

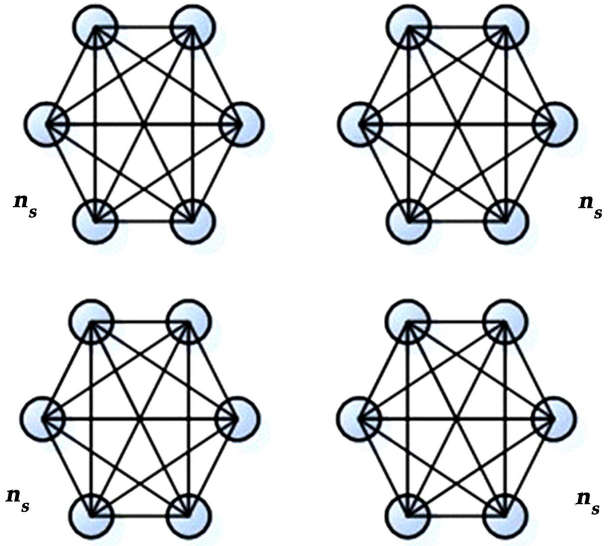


Figure 1: A team including sub-teams to produce product X

If several options were available in order to improve IT, the most cost-effective scenario would be the one with a combination of improving IT and increasing team size.

If the cost of adding new members to the team is more than the cost of improving IT, focusing on IT will be the best decision.

If the product demand is constant, organizations may achieve efficiency by investment in IT, and reducing the team size.

The above discussion has highlighted the importance of investment in IT. By doing so, the coordination and collaboration of activities among team members or sub-teams are facilitated.

The model assumptions are as follows:

- Sizes of sub-teams are the same.
- Each member spends their time on either production or information processing.
- IT as a parameter affects individuals and sub-teams in order to develop the output.
- There is exceptional value for the most effective coordination between individuals and sub-teams.
- Changes in IT contribute to changes in output.

- There is a one-to-one relation between each unit of product, IT, and team size.
- Individuals and sub-teams process all information received from other individuals and sub-teams, respectively.
- One unit of information is processed within one or less than a time unit.

The question that needs to be addressed is: how will you be able to predict the effect of IT on output and appropriate size of each sub-team by a mathematical model?

Mathematical Model

In this study, a mathematical model to evaluate the performance of a team associated with IT and optimized size of sub-teams is presented. Consider a team member who splits his/her time between information processing and production. Suppose that if one unit is exclusively dedicated to production, exactly one unit of output is produced. There is also exactly a unit of information generated, per each unit of output (Tohidi 2006).

It is assumed that a unit of information takes less than one unit of time to be produced. The time required to produce a particular piece of information by individuals is longer than the time consumed by team members if they work as a team to generate the same piece of information.

The model parameters are as follows:

- n: Team size.
- n_s: Size of each sub-team.
- m: Number of sub-teams.
- α: The rate of processing information created by members of a sub-team regarding the production.
- β: The rate of processing information created by another sub-team.
- t_1: The period of time required to create a report regardless of its size.
- t_2: The period of time required to generate a report.
- $\Omega(n_s)$: A fraction of the time that each member may spend on production after processing the information received from the other members of a sub-team (Tohidi 2008).

- p(n): The quantity of production of a team during one time period (Tohidi 2008).

It is assumed that the value of α is greater than the value of ββ and both variables are positive and less than 1. On the other hand, the coordination between internal sub-team requires more work than sub-teams coordination. Team size n is always more than sub-team size n_s.

$$0 < \beta < \alpha < 1 \tag{1}$$

$$1 \leq n_s \leq n \tag{2}$$

As was discussed earlier, each individual spends his/her time on either information sharing or production. Individuals spend a fraction of their time on processing information received from others and spend the rest of their time on production which is defined by $\Omega(n_s)$ and calculated by the following equation:

$$\Omega(n_s) = 1 - \alpha \cdot (n_s - 1) \cdot \Omega(n_s) - \beta \cdot (n - n_s) \cdot \Omega(n_s) - {}^{t_1}\!/_{t_2} \cdot \left({}^{n}\!/_{n_s} - 1\right) \tag{3}$$

Equation (3) is simplified to Eq. (4).

$$\Omega(n_s) = \frac{1 - t_1/t_2 \cdot \left(n/n_s - 1\right)}{1 + \alpha \cdot (n_s - 1) + \beta \cdot (n - n_s)} \tag{4}$$

The fraction of the time that each member may spend on production $\Omega(n_s)$ is between 0 and 1.

$$0 \leq \Omega(n_s) \leq 1 \tag{5}$$

The optimum size of a sub-team is determined by the following equation, which is derived from Eq. (4).

$$n_s^* = \frac{t_1 \cdot n + t_1 \cdot \sqrt{\frac{n \cdot (1-\alpha) \cdot \left(1 + t_2/t_1\right) + n^2 \cdot \left(\alpha + \beta \cdot t_2/t_1\right)}{\alpha - \beta}}}{t_1 + t_2} \tag{6}$$

In Fig. 2, it can be seen how sub-team size, the rate of processing information created by other sub-teams, and team size are interrelated and the following observations can be expressed.

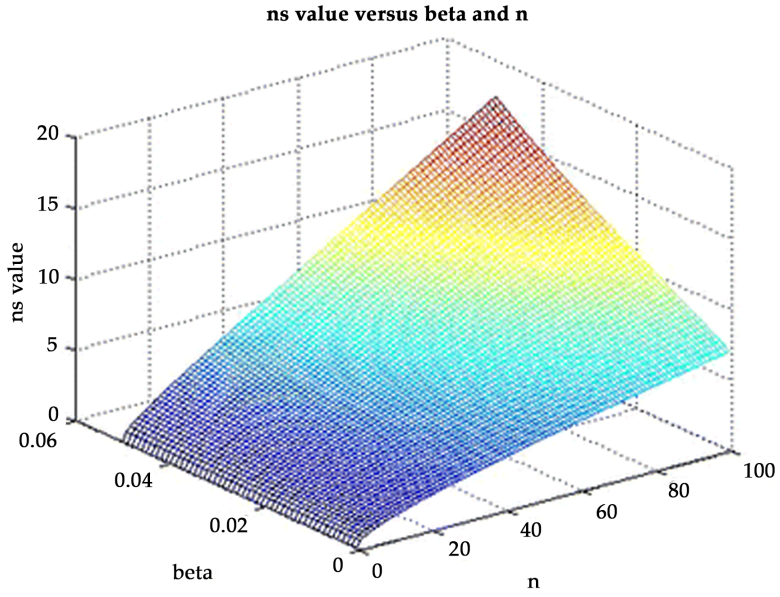


Figure 2: Size of sub-teams for different beta factors and team sizes.

- When α approaches β, it means the time it takes to process information created by members of a sub-team approaches the time to process information created by another sub-team, the optimum sub-team size goes to n, pointing that team dividing does not provide any benefits.
- As the constant time to process a report approaches 0, the optimum sub-team size approaches 1 That is, each team member becomes a sub-team of size one, pointing to perfect specialization on part of the individuals.
- When size of a team increases, an efficient sub-team size is the result. So there will be a trade-off between team members coordination and sub-teams coordination. By adding sub-teams to an organization, the coordination endeavor will increase since they will enhance the volume of information that needs to be processed.

The optimum quantity of production is a function of team size and $\Omega\left(n_s^*\right)$. It is concluded that by adding to the team members, productivity increases.

$$P_{n_s^*}(n) = n \cdot \Omega\left(n_s^*\right) \tag{7}$$

Theorem 1

$P_{n_s^*}(n)$ *is a monotonically and increasing function in n for all values of* $0 < \beta < \alpha < 1$.

Proof

$$\frac{dP_{n_s^*}(n)}{dn} = \frac{(1-\alpha)\cdot(t_2+t_1)^3\cdot\sqrt{t_1\cdot n\cdot(\alpha-\beta)\cdot[(t_1+t_2)\cdot(1-\alpha)+n\cdot(\alpha\cdot t_1+\beta\cdot t_2)}}{R\cdot S^2}, \tag{8}$$

where R and S are calculated by Eqs. (9) and (10).

$$R = t_1\cdot n\cdot t_2\cdot(\alpha-\beta)+t_2 \cdot\sqrt{n\cdot t_1\cdot(\alpha-\beta)\cdot[(t_1+t_2)\cdot(1-\alpha)+n\cdot(\alpha\cdot t_1+\beta\cdot t_2)} \tag{9}$$

$$S = t_1\cdot t_2\cdot(n-1)+t_2\cdot(n\cdot\beta+1-\alpha)+t_1 + \sqrt{n\cdot t_1\cdot(\alpha-\beta)\cdot[(t_1+t_2)\cdot(1-\alpha)+n\cdot(\alpha\cdot t_1+\beta\cdot t_2)} \tag{10}$$

$$\frac{dP_{n_s^*}(n)}{dn} > 0 \tag{11}$$

Hence, $P_{n_s^*}(n)$ is monotonically increasing function in *n*.

Theorem 1 indicates that team output can be increased by adding members to the team. However, the marginal product of team members is decreasing due to the increased coordination effort required, so that, for each added team member, there is a smaller and smaller increase in output. Beyond some value of *n*, the marginal cost of an additional team member exceeds the marginal value of the team's production (Tohidi 2006).

Theorem 2

For all values of $0 < \beta < \alpha < 1$, $P_{n_s^*}(n)$ *is a bounded function.*

Proof

From Theorem 1, $P_{n_s^*}(n)$ is a concave and monotonically increasing function of n. Also, $P_{n_s^*}(0) = 0$.

$$\lim_{n\to\infty} P_{n_s^*}(n) = \frac{2 + t_2/t_1 - t_1/t_2 \cdot \sqrt{\frac{(\alpha+\beta\cdot t_2/t_1)}{\alpha-\beta}}}{\alpha + \beta \cdot {}^{t_2}/_{t_1} + \sqrt{(\alpha-\beta)\left(\alpha + \beta \cdot {}^{t_2}/_{t_1}\right)}} \tag{12}$$

Hence, $P_{n_s^*}(n)$ is a bounded function.

The practical implication of Theorem 2 is that the maximum total production of a team during one time period depends on the speed at which the team members can coordinate their activities with their peers. To increase the team's maximum production capacity, it is necessary to change the communication and processing technology (i.e., decrease the value of α and β) or the work has to be re-organized so that each team member does not process all of the information provided by the other members (Tohidi 2008).

Theorem 3

The marginal product of team size is asymptotically zero.

Proof

$$\lim_{n\to\infty} \frac{\mathrm{d}P_{n_s^*}(n)}{\mathrm{d}n} = 0 \tag{13}$$

According to Eq. (13), if taken to a certain extent, adding to the team members may not result in productivity. Therefore, in order to increase total production units information sharing will need to be improved.

It can be understood from Eqs. 8 and 13 that management can grow the organization output by adding to the team members.

Figure 3 illustrates how $\frac{\mathrm{d}P_{n_s^*}(n)}{\mathrm{d}n}$ performs in different team size and the rate of processing information created by other sub-teams.

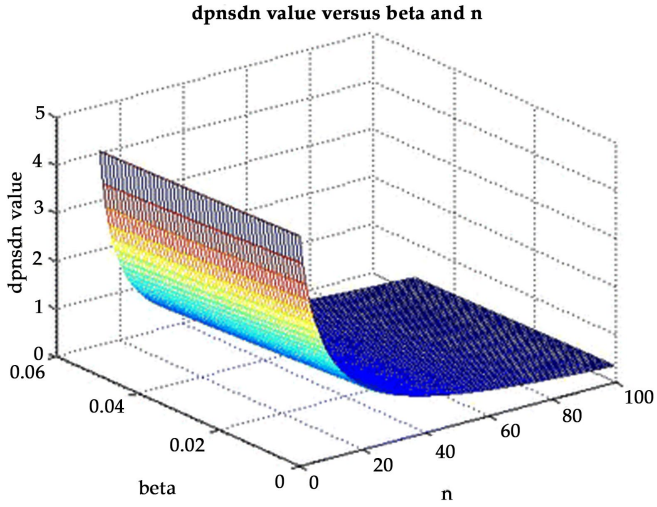


Figure 3: $\frac{dP_{n_s^*}(n)}{dn}$ for beta factors and team sizes.

Interaction between IT and Sub-teams

IT system may change team members' interactions through changing in one or some of the three information parameters α, β and t_1.

The derivative of sub-team size with respect to IT parameter α is calculated in Eq. (14).

$$\frac{dn_s^*}{d\alpha} = \frac{t_1 \cdot n \cdot [\beta \cdot (1-n) - 1] \cdot (1 + t_2/t_1)}{2.(t_1 + t_2) \cdot (\alpha - \beta)^2 \cdot \sqrt{\frac{n \cdot (1-\alpha) \cdot (1 + t_2/t_1) + n^2 \cdot (\alpha + \beta \cdot t_2/t_1)}{\alpha - \beta}}} \tag{14}$$

The value of dn_s^* by dα is negative.

$$\frac{dn_s^*}{d\alpha} < 0 \tag{15}$$

Equation 15 indicates, when sub-team's communication capabilities develop, the size of sub-team increases.

Figure 4 shows how $\frac{dn_s^*}{d\alpha}$, the rate of processing information created by other sub-teams, and team size affect each other.

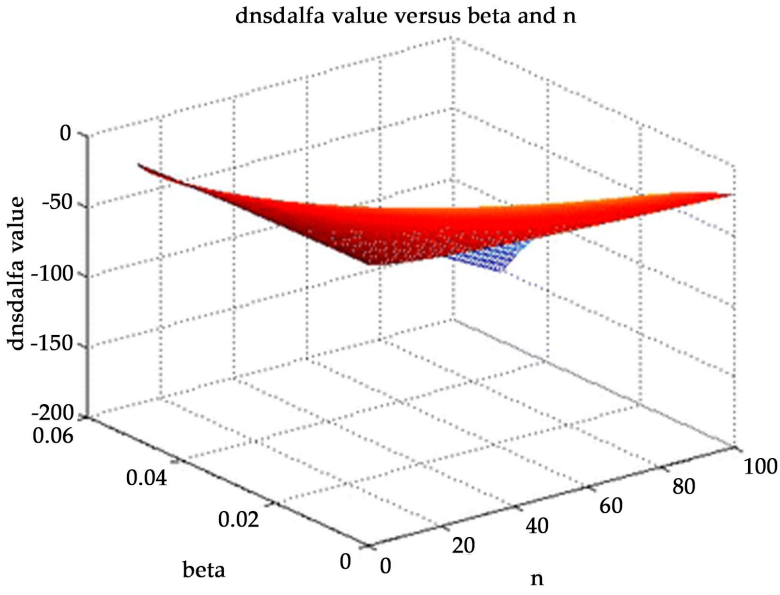


Figure 4: $\frac{dn_s^*}{d\alpha}$ for different beta factors and team sizes.

The derivative of sub-team size with respect to IT parameter β is calculated in Eq. (16).

$$\frac{dn_s^*}{d\beta} = \frac{t_1 \cdot n \cdot [(\alpha-\beta)\cdot t_2/t_1 + n\cdot(1-\alpha)\cdot(1+t_2/t_1) + n^2\cdot(\alpha+\beta\cdot t_2/t_1)}{2\cdot(t_1+t_2)\cdot(\alpha-\beta)^2\cdot\sqrt{\frac{n\cdot(1-\alpha)\cdot(1+t_2/t_1)+n^2\cdot(\alpha+\beta\cdot t_2/t_1)}{\alpha-\beta}}} \quad (16)$$

The value of dn_s^* by dβ is positive.

$$\frac{dn_s^*}{d\beta} > 0 \quad (17)$$

Equation 17 indicates, as the inter-sub-team coordination is simplified by using the new technology, the optimum sub-team size decreases.

Figure 5 shows how $\frac{dn_s^*}{d\beta}$, the rate of processing information created by other sub-teams, and team size affect each other.

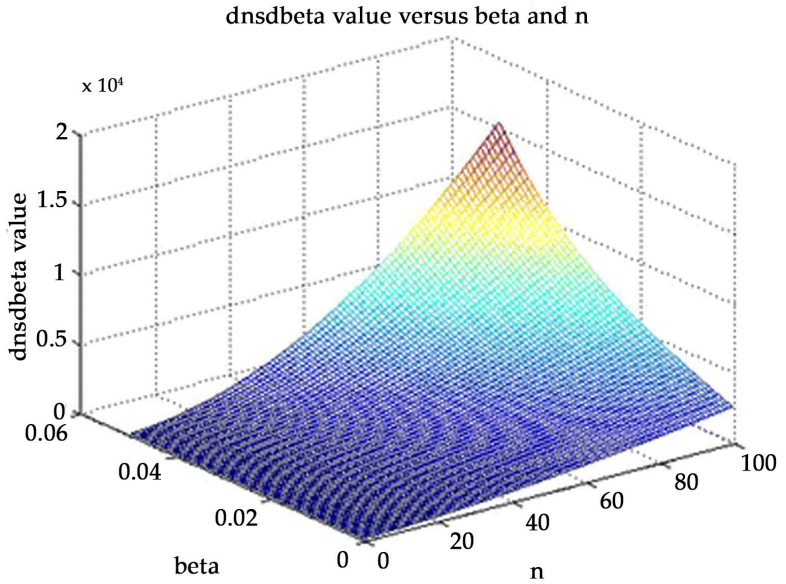


Figure 5: $\frac{dn_s^*}{d\beta}$ for different beta factors and team sizes.

The derivative of sub-team size with respect to the period of time required to create a report regardless of its size is calculated in Eq. (18).

$$\frac{dn_s^*}{dt_1} = \frac{n \cdot t_2}{(t_1 + t_2)^2} + \frac{n \cdot t_2 \cdot \frac{(1-\alpha)}{2} + n^2 \cdot t_2 \cdot \left(\alpha - \frac{\beta}{2}\right) + t_2 \cdot \left[n \cdot t_1 \cdot t_2 \cdot (1-\alpha) + \frac{\beta \cdot t_2 \cdot n^2}{t_1}\right.}{(\alpha - \beta) \cdot (t_1 + t_2)^2 \cdot \sqrt{\frac{n \cdot (1-\alpha) \cdot \left(1 + t_2/t_1\right) + n^2 \cdot \left(\alpha + \beta \cdot t_2/t_1\right)}{\alpha - \beta}}} \tag{18}$$

The value of dn_s^* by dt_1 is positive.

$$\frac{dn_s^*}{dt_1} > 0 \tag{19}$$

Equation 19 indicates as the time spent to process the information, the coordination time of tasks decreases, the size of organizational units will change. This change depends on changing the three information parameters. Of course, in all cases the coordination time decreases and the time spent on production increases.

Figure 6 illustrates how $\frac{dn_s^*}{dt_1}$ performs with variation in β and n.

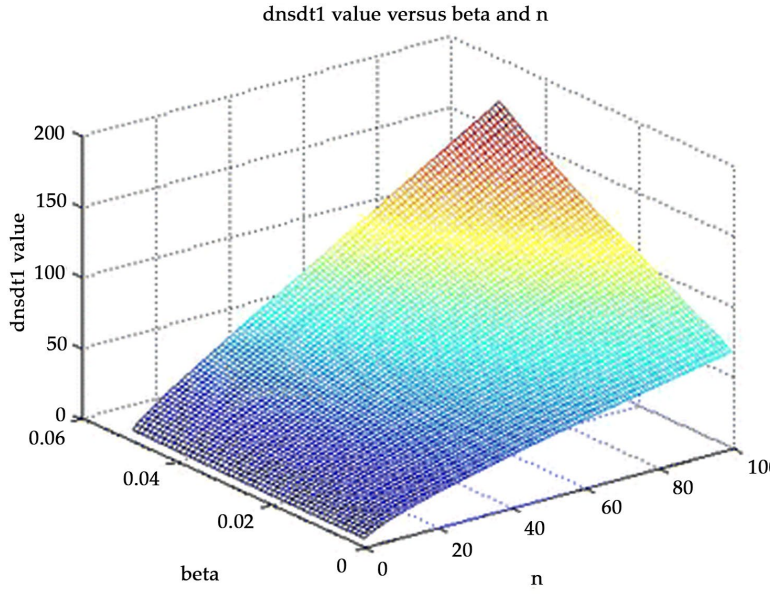


Figure 6: $\frac{dn_s^*}{dt_1}$ for different beta factors and team sizes.

Sensitivity Analysis

Equations 15, 17 and 19 emphasize the significance of investments in IT. By investing in IT that simplifies activities coordination among team members, the organization's production can be increased by management. The IT investment that adds intra-sub-team coordination, improves inter-sub-team coordination, or both. The suitable combination of investments depends on the labor cost, the task, and the price of the product at which the organization can sell.

Once the parameters, variables, and equations are defined and the results are obtained, a sensitivity analysis is performed to validate the presented mathematical model. The sensitivity analysis is developed in order to identify the variable which has the highest impact on the outcome of the model.

Three trials are reviewed and their numerical results are analyzed. The rate of processing information created by other sub-teams (β) and team

size (n) are varied in turn while the other variables remained the same. The results of the 2nd and 3rd trials are compared with outcomes of the 1st trial to determine how team size and information sharing system among sub-team members may affect the consequences, respectively.

As depicted in Table 1, in the original trial a team size of 21 is studied when the rate of processing information created by other sub-teams equals 0.03. According to the model, this team contributes to 9 units of product X. In the next trial, in order to increase P(n) from 9 to 12 units, team size needs to be changed to 54 while keeping the other variables unchanged. In the last trial, 12 units of product were obtained by improving information sharing system among sub-team members from 0.03 to 0.006. In other words, the same level of production may be achieved by 80% improvement in IT instead of adding 33 members to the team which is almost 157% more than the original team size.

Table 1: Values of parameters for three different trials

Parameters	1st trial	2nd trial	3rd trial
n	21	54	21
α	0.2	0.2	0.2
β	0.03	0.03	0.006
t_1	0.2	0.2	0.2
t_2	7	7	7
n_s^*	3	6	2
$\Omega(n_s)$	0.427	0.224	0.554
$P_{n_s^*}(n)$	9	12	12
m	7	9	11

CONCLUSION

In this study, a mathematical model has been proposed through which team performance was overviewed. The model is aimed at saving costs and improving productivity by collaboration and coordination of individuals within sub-teams and sub-teams within the whole team. Such a team is difficult to build and maintain, and it requires determining of optimum team size and sub-team size and the role that IT may possibly play. It has been

found that productivity increases with the increment of team size. However, increasing team size is not always cost-effective; beyond a certain point the cost of adding to the team members exceeds the value added to productivity. Investment in IT may also result in improvement in productivity. Hence, there should be a balance between increasing team size and improving IT in order to improve productivity. It is concluded that same numbers of product units may be attained by improving IT and increasing team size. Therefore, IT is an alternative for increasing team size. In summary, if improving information sharing system is more cost efficient then adding to the members of sub-teams and team is not the best scenario.

It is also concluded that if we separate a team into sub-teams and invest in IT, the efficiency and capacity of organization will be increased. Those interested in further studies in this research may investigate the methodologies and estimation approaches and measurement of IT parameters. Another future work in this research would be searching and providing an appropriate model which could be applied to a team with structured sub-teams, under the specific circumstances. There might be many uncertainties in more progressive cases in practice. Hence, the experiment may be further extended to test the improvement of productivity by increasing team size and IT using fuzzy logic.

REFERENCES

1. Badescu M, Garcés-Ayerbe C (2009) The impact of information technologies on firm productivity: empirical evidence from Spain. Technovation 29(2):122–129
2. Bartel AP, Ichniowski C, Shaw KL (2005) How does information technology really affect productivity? Plant-level comparisons of product innovation, process improvement and worker skills. NBER Working Paper No. 11773, November 2005
3. Bharadwaj AS (2000) A resource-based perspective on information technology capability and firm performance: an empirical investigation. MIS Q 24:169–196
4. Dehning B, Richardson VJ (2002) Returns on investments in information technology: a research synthesis. J Inf Syst 16(1):7–30
5. Galegher J, Kraut RE (1994) Computer-mediated communication for intellectual teamwork: an experiment in group writing. Inf Syst Res 5(2):110–138
6. Hatcher L, Ross TL (1991) From individual incentives to an organization-wide gainsharing plan: effects on teamwork and product quality. J Organ Behav 12(3):169–183
7. Jones DC, Kalmi P, Kauhanen A (2011) Firm and employee effects of an enterprise information system: micro-econometric evidence. Int J Prod Econ 130(2):159–168
8. Kılıçaslan Y, Sickles RC, Kayış AA et al (2017) Impact of ICT on the productivity of the firm: evidence from Turkish manufacturing. J Prod Anal. doi:10.1007/s11123-017-0497-3
9. Martínez-Lorente AR, Sánchez-Rodríguez C, Dewhurst FW (2004) The effect of information technologies on TQM: an initial analysis. Int J Prod Econ 89(1):77–93
10. Moses TP, Stahelski AJ (1999) A productivity evaluation of teamwork at an aluminum manufacturing plant. Group Organ Manag 24(3):391–412
11. Powell SG (2000) Specialization, teamwork, and production efficiency. Int J Prod Econ 67(3):205–218
12. Salas E, Cooke NJ, Rosen MA (2008) On teams, teamwork, and team performance: discoveries and developments. Hum Factors J Hum Factors Ergon Soc 50(3):540–547

13. Shao BBM, Lin WT (2016) Assessing output performance of information technology service industries: productivity, innovation and catch-up. Int J Prod Econ 172:43–53
14. Stewart GL, Barrick MR (2000) Team structure and performance: assessing the mediating role of intrateam process and the moderating role of task type. Acad Manag J 43(2):135–148
15. Tohidi H (2006) The team-based, information technology-enabled organizations. In: Proceedings of the second international conference on information management and business (IMB2006), pp 145–154
16. Tohidi H (2008) The relationship between teamwork effectiveness and information technology. J Appl Econ Sci 3(4(6)_Winter2008):464
17. Tohidi H, Tarokh MJ (2006) Productivity outcomes of teamwork as an effect of information technology and team size. Int J Prod Econ 103(2):610–615
18. Whelan K (2002) Computers, obsolescence, and productivity. Rev Econ Stat 84(3):445–461
19. Wu L, Chuang C-H, Hsu C-H (2014) Information sharing and collaborative behaviors in enabling supply chain performance: a social exchange perspective. Int J Prod Econ 148:122–132

Chapter

TOPOLOGY OPTIMISATION UNDER UNCERTAINTIES WITH NEURAL NETWORKS

10

Martin Eigel [1], Marvin Haase [2], and Johannes Neumann [3]

[1]Weierstrass Institute for Applied Analysis and Stochastics, 10117 Berlin, Germany
[2]Department of Mathematics, Technical University Berlin, 10623 Berlin, Germany
[3]Rafinex Ltd., Great Haseley OX44 7JQ, UK

ABSTRACT

Topology optimisation is a mathematical approach relevant to different engineering problems where the distribution of material in a defined domain is distributed in some optimal way, subject to a predefined cost function representing desired (e.g., mechanical) properties and constraints. The computation of such an optimal distribution depends on the numerical solution of some physical model (in our case linear elasticity) and robustness

Citation: (APA): Eigel, M., Haase, M. & Neumann, J. (2022). Topology Optimisation under Uncertainties with Neural Networks. Algorithms 15(7):241. (34 pages).

is achieved by introducing uncertainties into the model data, namely the forces acting on the structure and variations of the material stiffness, rendering the task high-dimensional and computationally expensive. To alleviate this computational burden, we develop two neural network architectures (NN) that are capable of predicting the gradient step of the optimisation procedure. Since state-of-the-art methods use adaptive mesh refinement, the neural networks are designed to use a sufficiently fine reference mesh such that only one training phase of the neural network suffices. As a first architecture, a convolutional neural network is adapted to the task. To include sequential information of the optimisation process, a recurrent neural network is constructed as a second architecture. A common 2D bridge benchmark is used to illustrate the performance of the proposed architectures. It is observed that the NN prediction of the gradient step clearly outperforms the classical optimisation method, in particular since larger iteration steps become viable.

Keywords: topology optimisation; deep neural networks; model uncertainties; random fields; convolutional neural networks; recurrent neural networks

INTRODUCTION

Structural topology optimisation is the (engineering-oriented) process of designing a construction part using optimisation algorithms under certain constraints. The resulting designs usually have a large influence on the subsequent production costs. The starting point of the process is a design domain that represents the maximum space available for the optimised component to be developed. The outcome presents information about which parts of the design space are occupied by material and which are void. Often, the task is motivated by mechanical requirements, e.g., sufficient stiffness of the constructed part with respect to assumed forces acting on it while certain predetermined points or surfaces should connect to other parts. A typical physical model for this comes from linear elasticity, describing the displacement field given material properties and forces. For the mathematical optimisation, it is repeatedly necessary to compute the stress distribution determined by the physical model in the design domain (more precisely, in the parts of the domain with material). This potentially complex computational task usually relies on the finite element method (FEM), which is based on a discretisation of the domain into elements.

Most commonly, the domain is represented as mesh consisting of disjoint simplices, i.e., triangles in 2D and tetrahedra in 3D.

Since the optimisation process easily requires several hundred evaluations of the state equation to evolve the material distribution, it is of significant interest to develop techniques that reduce this computational burden. This becomes much more pronounced when uncertainties of the model data should be considered in the computations. The treatment of uncertainties has been developed extensively from a theoretical and practical point of view over the last decade in the area of Uncertainty Quantification (UQ). A common way to describe uncertainties is by means of random fields, whose (Karhunen-Loève) expansions depend on a possibly very large number of random variables. The parameter space spanned by these random variables leads to very high-dimensional state problems for which the derived optimisation problems are very difficult to solve.

This paper investigates the application of a trend in scientific computing for current topology optimisation methods, namely the use of modern machine learning techniques. More precisely, our objective is to improve the efficiency of the structural topology optimisation problem by predicting gradient steps based on generated training data. This efficiency gain directly transfers to our ability to compute much more involved risk-averse stochastic topology optimisation problems with random data. In this case, the topology is optimised with an adjusted cost functional including the CVaR (conditional value at risk), by which unlikely events can be taken into account in contrast to simply optimising with the mean value of possible load scenarios. In addition to random loads, we also include random material properties which, e.g., can enter the model in terms of material errors or impurities. We emphasise that risk-averse optimisation based on some risk measure is a timely topic, which plays a role in many application areas. Despite its relevance, this type of problem has not been covered widely in the literature yet. In fact, the authors are not aware of any other machine-learning-assisted approach for risk-averse topology optimisation. This might be due to the more involved mathematical framework and substantially higher computational complexity. To achieve performance benefits with topology optimisation in this paper, we adapt concepts from the field of deep learning to approximate multiple iterations of the optimisation process and render the overall optimisation more efficient.

The goal of topology optimisation is to satisfy the technical requirements of a component (for instance, stiffness with respect to certain loading

scenarios) with minimal use of material. There are different approaches to describe the topology in a flexible way such that substantial changes are possible. We follow our previous work in [1] and use a phase field model which describes the density of material with a value in [0,1]. The starting point is the definition of a physical design space available for the component under consideration. This space is completely filled with a material in the sense of an initial solution. Furthermore, all points at which loads act on the component, as well as the type of the respective load, are prescribed. The optimisation aims to achieve a homogeneous, minimum possible deformation at all optimised points of the component under the imposed (possibly continuous and thus infinitely many) loading scenarios. Here, a minimum compliance corresponds to a maximum stiffness. In general, even solving the underlying partial differential equation (PDE) of this problem for deterministic coefficients of the PDE presents a complex task. Furthermore, PDE coefficients which describe material and the loads have a strong influence on the resulting topology, i.e., even small changes in these coefficients can lead to large differences in the resulting topology. This results in considerable computational effort in the stochastic settings, since the solution has to be calculated for a sufficient number of data realisations to become reliable. Hence, the modelling of these stochastic settings for example (with the most obvious approach) by a Monte Carlo (MC) simulation increases the required iteration steps linearly in the number of simulations.

A method to numerically tackle topology optimisation uncertainty was presented in [2]. In this paper, we extend the previous work by introducing Deep Neuronal Networks (DNN) that are designed to provide a prediction of the next gradient step. Since topologies discretised with finite elements can be represented as images, Convolutional Neural Networks (CNN) seem natural candidate architectures for this task and there has already been some research on this approach for the deterministic setting. An introduction is presented in [3] where the conventional topology optimisation algorithms are replicated in a computationally inexpensive way. Furthermore, a CNN is used in [4] to approximate the last iteration steps of a gradient method of a topology optimisation after a fixed number of steps to refine a “fuzzy” solution. A CNN architecture is also used in [5] to solve a topology optimisation problem and trained with large amounts of data. The resulting NNs were able to solve problems with boundary conditions different to their training data. In [6], the problem is stated as an image segmentation task and a deep NN with encoder–decoder architecture is leveraged for pixel-

wise labeling of the predicted topology. Another encoding–decoding U-Net CNN architecture is presented in [7], providing up- and down-sampling operators based on training with large datasets. In [8] a multilevel topology optimisation is considered where the macroscale elastic structure is optimised subject to spatially varying microscale metamaterials. Instead of density, the parameters of the micromaterial are optimised in the iteration, using a single-layer feedforward Gaussian basis function network as surrogate model for the elastic response of the microscale material.

A discussion on solving PDEs with the help of Neural Networks (NN) for instance of the Poisson equation and the steady Navier–Stokes equations is provided in [9]. In a relatively new approach, through a combination of Deep Learning and conventional topology optimisation, the Solid Isotropic Material with Penalisation (SIMP) was presented in [10], which could reduce the computational time compared to the classical approach. The authors use a similar method as [4] except that the underlying optimisation algorithm performs a mesh refinement after a fixed number of iterations. To improve this step, separately trained NNs are applied to the respective mesh in order to approximate the last steps of the optimisation on the corresponding mesh. The result is then projected to the next finer mesh and the procedure is repeated a fixed number of times. A SIMP density field topology optimisation is directly performed in [11]. The problem can be represented in terms of the NN activation function. Different beam problems comparable to our experiments are depicted. Fully connected DNNs are used in [12] to represent implicit level set function describing the topology. For optimisation, a system of parameterised ODEs is used. A two-stage NN architecture which by construction reduces the problem of structural disconnections is developed in [13]. Deep generative models for topology optimisation problems with varying design constraints and data scenarios are explored in [14]. In [15], direct predictions without any iteration scheme and also the nonlinear elastic setting are considered. Examples are only shown for a coarse mesh discretisation of the design domain. In [16], an NN-assisted design method for topology optimisation is devised, which does not require any optimised data. A predictor NN provides the designs on the basis of boundary conditions and degree of filling as input data for which no optimisation training data are required.

The main goal of this paper is to devise new NN architectures that lower the computational burden of structural topology optimisation based on a continuous phase-field description of the density in the design domain.

In particular, the approach should be able to cope with adaptive mesh refinements during the optimisation process, which has shown to significantly improve the performance of the optimisation. Moreover, as a consequence of an efficient computation in a deterministic setting, a goal is to transfer the developed techniques to the stochastic setting for the risk-averse topology optimisation task. The general strategy is to combine conventional topology optimisation methods and NNs in order to reduce the number of required iteration steps within the optimisation procedure, increasing the overall performance.

The main achievements of this paper are two new NN architectures that are demonstrated to yield state-of-the-art numerical results with a much lower number of iterations than with a classical optimisation. Moreover, in contrast to several other works that are solely founded on the image level of topology, our architectures make use of a very versatile functional phase-field description of the material distribution, which we have not observed in the literature with regard to NNs. This also holds true for the stochastic risk-averse framework, which to our knowledge has not been considered with NN predictions yet. Another novelty is the mixture of a fine reference mesh and adaptive iteration meshes during optimisation.

Inspired by the work of [5,10], as a first new NN architecture we develop a new CNN approach and show that it can replicate the reference results of [1,2]. In contrast to [10], we only have to train one NN for the entire optimisation despite mesh refinement being carried out in the iterative procedure. We subsequently extend this approach to the stochastic setting with risk-averse optimisation from [2]. Based on the CNN, we further extend the optimisation with a Long Short-Term Memory (LSTM) architecture as a second novel method. It encodes the change of the topology over several iteration steps, thus resulting in a more accurate prediction of the next gradient step.

In the numerical experiments, it can be observed that the two presented architectures perform equally well as our reference implementation. However, fewer iteration steps are required (i.e., larger steps can be used) since the gradient step predictions seem to be better than when computed with a classical optimisation algorithm.

The structure of the paper is as follows. In Section 2, we introduce the underlying setting from [1,2] and discuss the algorithms used for phase-field-based topology optimisation. In this context, we introduce the linear elasticity model and derive a state equation, the adjoint equation and a

gradient equation, whose joint solution constitutes the optimisation problem under consideration. In Section 3, we present two different architectures of the NNs approximating multiple steps of the gradient equation. We start with a CNN that is well suited for processing the discretised solutions of the equations from Section 2. This is then extended to a long short-term memory NN, which is able to process a sequence of these solutions simultaneously and thus achieves a higher prediction quality. Since the data of the finite element simulation do not directly match the required structure of NNs, we provide a discussion of the data preparation for both architectures. Section 4 illustrates the practical performance of the developed NN architectures with a standard benchmark (a 2D bridge problem). The work ends with a summary and discussion of the results and some ideas for further research in Section 5. The appendix provides some background on the used problem, in particular the standard benchmark problem in Appendix A and the finite element discretisation in Appendix B. The implementation and codes for the generation of graphics and data to reproduce this work are publicly available (https://github.com/MarvinHaa/DeepNNforTopoOptisation accessed at 1 June 2022).

TOPOLOGY OPTIMISATION UNDER UNCERTAINTIES

We are concerned with the task of topology optimisation with respect to a state equation of linear elasticity. This problem becomes more involved when stochastic data are assumed. In our case, this concerns material properties and the forces acting on the designed structure. These translate into the engineering world as material imperfections or fluctuations and natural forces such as wind or ocean waves. Such random phenomena are modelled in terms of random fields that often are assumed to be Gaussian with certain mean and covariance.

It is instructive to first present the deterministic topology optimisation task, which we discuss in the following Section 2.1. Subsequently, in Section 2.2 we extend the model to exhibit random data, allowing an extension of the cost functional to also include the fluctuations of the data in terms of a risk measure. In our case, this is the so-called conditional value at risk (CVaR).

For the sake of a self-contained presentation, we provide all equations that lead to the actual optimisation problem, which is given in terms of a gradient that evolves a phase field. Thus, the entire problem formulation can be understood and the required extensions to obtain the risk-averse

formulation become clear. However, in case the reader is only interested in the proposed NN architectures, it might be sufficient to simply gloss over the most important parts of the problem definition, for which we provide a guideline as follows: The linear state equation is given in Equation (1), leading to the weak form in Equation (2) that is used for the computation of finite element solutions. These are required in the deterministic minimisation problem given in Equation (4), which is solved iteratively by computing the gradient step defined by Equation (6). A similar problem formulation, extended by an approximation of the CVaR risk measure, can be obtained in case of the risk-averse optimisation. This is given in Equation (9) and can be solved iteratively with gradient steps defined by Equation (11).

The presentation of this section is based on [1,2] where the optimisation problem computes the distribution of material in a given design domain described by a continuous phase field depending on the realisation of the random parameters. The optimum of this problem maximises stiffness while minimising the volume of material for the given data.

Deterministic Model Formulation

The goal is to determine an optimal distribution of a material (with density or probability) $m \in [0,1]$ in a compact design domain $D \subset \mathbb{R}^d,\ d \in \mathbb{N}$. We further assume that D satisfies sufficient regularity assumptions such that the PDE state equation exhibits a unique solution. The desired optimality of the task means that the resulting topology is as resilient (or stiff) as possible with respect to the deformation caused by the expected forces acting on it, which are described by a differentiable vector field u: $D \to \mathbb{R}^d$.

Definition 1.

The distribution of a material $m \in [0,1]$ *in* $D \to \mathbb{R}^d$ *is represented by a phase field* $\varphi: D \to \mathbb{R}$ *with* $0 \le \varphi(x) \le 1$ *for all* $x \in D$*, where* $\varphi(x)=1$ *if there is material at position x and* $\varphi(x)=0$ *if there is no material at position x. At the phase transitions we allow* $0<\varphi(x)<1$ *to ensure sufficient smoothness for phase shift. We call the evaluation of* φ *topology.*

Note that the actual topology is reconstructed in a post-processing step by choosing some threshold in (0,1) to fix the interface between material and void phase of the phase field.

Linear Elasticity Model

The state equation corresponding to the above problem is described by the standard linear elasticity model [17]. To define the material tensor, we first define the *strain displacement* (or strain tensor) $\mathcal{E} : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ by

$$\mathcal{E}(x) := \frac{1}{2}(\nabla u(x) + \nabla u^T(x)),$$

which specifies the displacement of the medium in the vicinity of position x. Moreover, a so-called blend function $\omega : \mathbb{R} \to \mathbb{R}$ is given by

$$\omega(x) := \max\{x^3, 0\},$$

ensuring a smooth transition between the phases. According to Hooke's law and by using the *Lamé coefficients* $\mu_{mat}>0$ and $\lambda_{mat}>0$, the *isotropic material tensor* σ_{mat}: $\mathbb{R}^d \to \mathbb{R}^{d\times d}$ for the solid phase is given by

$$\sigma_{mat}(x) := 2\mu_{mat}\mathcal{E}(x) + \lambda_{mat}\mathrm{Tr}(\mathcal{E}(x))I.$$

This material tensor describes the acting forces between adjacent positions in the connected material, where λ_{mat} and μ_{mat} are two material parameters characterising the strain–stress relationship. For the void phase, to ensure solvability of the state equation in entire domain D, we define the tensor as a fraction of the material phases. More precisely, we set $\sigma_{\text{void}}(x) := \varepsilon^2\sigma_{\text{mat}}(x)$ with some small $\varepsilon>0$. Hence, the *material tensor* (or stress tensor) $\sigma : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is given by

$$\sigma(\varphi(x), u(x)) := \sigma_{\text{mat}}(u(x))\omega(\varphi(x)) + \sigma_{\text{void}}(u(x))\omega(1-\varphi(x))$$

Using the material tensor σ, a force with load $g \in \mathbb{R}^d$ (a pressure field) and the phase field φ, the displacement vector field u is described by the state equation of the standard linear elasticity model given by

$$\begin{aligned}
-\mathrm{div}[\sigma(\varphi(x), u(x))] &= 0 && \text{for all } x \in D,\\
u(x) &= 0 && \text{for all } x \in \Gamma_D,\\
\sigma(\varphi(x), u(x)) &= g && \text{for all } x \in \Gamma_g,\\
u(x)\cdot n(x) &= 0 && \text{for all } x \in \Gamma_s,\\
\sigma(\varphi(x), u(x))\cdot n(x) &= 0 && \text{for all } x \in \Gamma_0 = \partial D \setminus (\Gamma_D \cup \Gamma_g \cup \Gamma_s).
\end{aligned} \tag{1}$$

This implies that on boundary subspace $\Gamma_D \subset D$ the material is fixed while on $\Gamma_g \subset D$ the force g acts on the material. On the boundary $\Gamma_s \subset D$ the material is barred from movement in normal direction n. In the following, equality is to be generally understood in a pointwise manner.

State Equation

The weak formulation of the state Equation (1) can be formulated as: find $u \in H^1_{\Gamma_g}(D)$ such that

$$\int_D \sigma(\varphi, u)\mathcal{E}(v_u)\,\mathrm{d}\mu = \int_{\Gamma_g} g \cdot v_u\,\mathrm{d}\mu \qquad \forall\, v_u \in H^1_0(D), \tag{2}$$

where $H^1(D)$ is the usual Sobolev space and dμ the Lebesgue measure and

$$H^1_{\Gamma_g}(D) := \{u \in H^1(D) \mid \sigma(\varphi, u) = g \text{ on } \Gamma_g\}$$

and

$$H^1_0(D) := \{v_u \in H^1(D) \mid v_u = 0 \text{ on } \Gamma_0\}.$$

These definitions are in particular used for the finite element discretisation described in Appendix B.

Adjoint Equation

To define the optimisation problem, we introduce the *Ginsburg–Landau functional* $E^\varepsilon : \mathbb{R} \to \mathbb{R}$, which serves as a penalty term for undesired variations and is defined by

$$E^\varepsilon(\varphi) = \int_D \frac{\varepsilon}{2}|\nabla\varphi|^2 + \frac{1}{2\varepsilon}\psi_0(\varphi)\,\mathrm{d}\mu,$$

where $|\cdot|$ is the Euclidean norm. This ensures that the solution to the optimisation problem can be interpreted as an actual smooth shape. The *double well functional* $\psi_0 : \mathbb{R}^d \to \mathbb{R}$ with $\psi_0(x) = (\varphi(x) - \varphi(x)^2)^2$ penalises values of φ that differ from 0 or 1 and the leading term limits the changes of φ. This results in the cost functional $J^\varepsilon : \mathbb{R}^d \to \mathbb{R}$ to be minimised,

$$J^\varepsilon(\varphi, u) = \int_{\Gamma_g} g \cdot u\,\mathrm{d}\mu + \gamma E^\varepsilon(\varphi), \quad \gamma > 0. \tag{3}$$

The adaptivity parameter γ controls the weight of the interface penalty and hence has a direct influence on the minimum respective to the characteristics of the resulting shape of φ. In fact, γ is chosen adaptively to avoid non-physical or highly porous topologies, see [2]. Additionally, we require the volume constraint $\int_D \varphi\,\mathrm{d}\mu = m|D|$ with $m \in [0,1]$ to limit the amount of overall material.

The (displacement) state u from Equation (**3**) is obtained by solving the state Equation (**2**), which is used in the optimisation problem

$$\text{minimize } J^{\varepsilon}(\varphi, u) \text{ over } \varphi \in H^1(D) \tag{4}$$

s.t.Equation (2)holds, $0 \le \varphi(x) \le 1 \text{ for all } x \in D \text{ and } \int_D \varphi(x)\,\mathrm{d}\mu = m|D|$.

The *Allen–Cahn gradient flow approach* is used to determine the solution φ for which the adjoint problem of Equation (4) is used to avoid the otherwise more costly calculation. It is shown in [2] that for J^{ε} the corresponding adjoint problem can be formulated as: find $p \in H^1(D)$ such that

$$\int_D \sigma(\varphi, p))\mathcal{E}(v_p)\,\mathrm{d}\mu = \int_{\Gamma_g} g \cdot v_p\,\mathrm{d}\mu \qquad \forall\, v_p \in H_0^1(D), \tag{5}$$

which is identical to the state equation. Hence, the respective adjoint solution p is equal to the solution u of Equation (**2**) and no additional system has to be solved.

Gradient Equation

With the solutions u respective to p one gradient step with adaptive step size τ can be characterised by the unique solution $(\varphi,\lambda) \in H^1(D) \times \mathbb{R}$ such that, for all $(v_\varphi, v_\lambda) \in H_0^1(D) \times \mathbb{R}$,

$$\begin{aligned}\frac{\varepsilon}{\tau}\int_D(\varphi^* - \varphi_n)v_\varphi\,\mathrm{d}\mu + \varepsilon\gamma\int_D \nabla\varphi^* \cdot \nabla v_\varphi\,\mathrm{d}\mu + \frac{\gamma}{\varepsilon}\int_D \frac{\partial}{\partial\varphi}\psi_0(\varphi_n)v_\varphi\,\mathrm{d}\mu \\ - \int_D \frac{\partial}{\partial\varphi}\sigma(p, \varphi_n)v_\varphi\mathcal{E}(u)\,\mathrm{d}\mu + \int_D \lambda v_\varphi\,\mathrm{d}\mu + \int_D(\varphi^* - m)v_\lambda\,\mathrm{d}\mu = 0.\end{aligned} \tag{6}$$

The restriction on $0 \le \varphi \le 1$ for all $x \in D$ is realised by $\varphi(x) := \min\{\max\{0, \varphi^*(x)\}, 1\}$ in every iteration step. For the calculation of the minimum of Equation (4), the state Equation (1), the adjoint Equation (5) and subsequently the gradient Equation (6) are solved iteratively until φ converges. We always assume that solutions u and φ exist, which in fact can be observed numerically. The proposed procedure is described by Algorithm A1 where the solution of the integral equations takes place on a discretisation of D. The algorithm solves the state, adjoint and gradient equations in a loop until the solutions of the gradient equations only change slightly. The discretisation mesh is subsequently refined and the iterative process is restarted on this adjusted discretisation.

Stochastic Model Formulation

In the stochastic setting, the Lamé coefficients (determining the material properties) $\lambda_{\text{mat}} : \Omega \to \mathbb{R}^+$ and $\mu_{\text{mat}} : \Omega \to \mathbb{R}^+$ and the load scenarios $g : \Omega \to \mathbb{R}^d$ are treated as random variables on some probability space (Ω,P). The randomness of the data is inherited by the solution of the state equation as well as the adjoint equation. As a result, the gradient step can be considered as a random distribution, see again [1,2]. The goal is to minimise the functional for the expected value of φ as well as for particularly unlikely events. For the formulation of an adequate risk-averse cost functional, we introduce the *conditional value at risk* (CVaR). The CVaR, a common quantity in financial mathematics, is defined for a random variable X by

$$\text{CVaR}_\beta[X] := \mathbb{E}[X \, \mathbb{1}_{\{X > \text{VaR}_\beta[X]\}}],$$

with $\text{VaR}_\beta[X]:=\inf\{t\in \mathbb{R} | P(X\leq t)\geq\beta\}$ and $1>\beta\geq 0$. It characterises the expectation of the β-tail quantile distribution of X, hence accounting for bad outliers that may occur with low probability. The *stochastic state equation* can be formulated analogously to Equation (5) in the deterministic setting.

Adjoint Equation

For the risk-aware version of Equation (3) with respect to the CVaR parameter β, we define the cost $J_\beta^\varepsilon(\varphi) : \mathbb{R}^d \to \mathbb{R}$ by

$$J_\beta^\varepsilon(\varphi) = \text{CVaR}_\beta\Big[\int_{\Gamma_g} g \cdot u \, \mathrm{d}\mu\Big] + \gamma E^\varepsilon(\varphi), \quad \gamma > 0. \tag{7}$$

In the special case β=0, the CVaR is nothing else than the mean, i.e.,

$$J_0^\varepsilon(\varphi) = \mathbb{E}\Big[\int_{\Gamma_g} g \cdot u \, \mathrm{d}\mu\Big] + \gamma E^\varepsilon(\varphi).$$

This results in the *stochastic minimisation problem* analogous to Equation (4) given by

$$\text{minimise } J_\beta^\varepsilon(\varphi) \text{ over } \varphi \in H^1(D) \tag{8}$$

s.t. Equation (2)holdsa.s., $0 \leq \varphi(x) \leq 1$ for all $x \in D$ and $\int_D \varphi \, \mathrm{d}\mu = m|D|$.

Following [2], the CVaR can be approximated in terms of the plus

function. The solution of Equation (8) can hence be rewritten as

$$\min_{\varphi\in H^1(D)} J_\beta^\varepsilon(\varphi) = \min_{\varphi\in H^1(D),t\geq 0} \left(t + \frac{1}{1-\beta}\mathbb{E}\left[\left(\int_{\Gamma_g} g\cdot u \,\mathrm{d}\mu\right)_+\right] + \gamma E^\varepsilon(\varphi)\right). \tag{9}$$

An obvious approach to solve this optimisation problem is a Monte Carlo simulation, i.e., for each iteration step $n\in\mathbb{N}$ with evaluation of u_n, state Equation (1) have to be solved for different parameter realisations. The associated adjoint problem to Equation (9) reads

$$\int_D \sigma(\varphi,p)\mathcal{E}(v_p)\,\mathrm{d}\mu = \begin{cases} 0, & \text{if } \int_{\Gamma_g} gu\,\mathrm{d}\mu - t \leq 0 \\ \int_{\Gamma_g}(1-\beta)^{-1} gu\,\mathrm{d}\mu, & \text{else} \end{cases} \quad \text{a.s.} \tag{10}$$

Consequently, the solution of Equation (10) is given by

$$p = \begin{cases} 0, & \text{if } \int_{\Gamma_g} gu\,\mathrm{d}\mu - t \leq 0 \\ (1-\beta)^{-1}u, & \text{else} \end{cases} \quad \text{a.s.}$$

Gradient Equation

Analogous to the deterministic approach, the gradient can be defined corresponding to Equation (9) by the solution $(\varphi,\lambda,t)\in H^1(D)\times\mathbb{R}\times\mathbb{R}$, such that for all $(v_\varphi,v_\lambda,v_t)\in H_0^1(D)\times\mathbb{R},\times\mathbb{R}$ the following equation holds,

$$\begin{aligned} 0 = {} & \frac{\varepsilon}{\tau_\varphi}\int_D(\varphi^*-\varphi_n)v_\varphi\,\mathrm{d}\mu + \frac{\varepsilon}{\tau_t}\int_D(t-t_n)v_t\,\mathrm{d}\mu + \int_D \lambda v_\varphi\,\mathrm{d}\mu + \int_D(\varphi^*-m)v_\lambda\,\mathrm{d}\mu \\ & + \varepsilon\gamma\int_D \nabla\varphi^*\cdot\nabla v_\varphi\,\mathrm{d}\mu + \frac{\gamma}{\varepsilon}\int_D\frac{\partial}{\partial\varphi}\psi_0(\varphi_n)v_\varphi\,\mathrm{d}\mu - \int_D\frac{\partial}{\partial\varphi}\sigma(p,\varphi_n)v_\varphi\mathcal{E}(u)\,\mathrm{d}\mu \\ & + \begin{cases} \int_D v_t\,\mathrm{d}\mu, & \text{if } \int_{\Gamma_g} gu\,\mathrm{d}\mu - t\leq 0 \\ \int_D(1-\frac{1}{1-\beta})v_t\,\mathrm{d}\mu, & \text{else} \end{cases}_{a.s.} \end{aligned} \tag{11}$$

The actual solution for one gradient step of the minimisation problem in Equation (8) with respect to (9) follows from

$$\varphi \approx \frac{1}{S}\sum_{i=1}^{S}\varphi_i^*, \tag{12}$$

where $S\in\mathbb{N}$ is the number of samples of the Monte Carlo simulation. The larger β chosen, the larger t becomes, and thus the number of evaluations of u for which $\int_{\Gamma_g} gu\,\mathrm{d}\mu - t\leq 0$ holds true increases. In order to ensure a valid simulation of Equation (12), N must be chosen sufficiently large so that an adequate number of evaluations of u in each gradient step fulfils condition

$\int_{\Gamma_g} gu \, d\mu - t > 0$. The described procedure is depicted in Algorithm A2 where the optimisation process is basically the same as in Algorithm A1. The central difference is that $N \in \mathbb{N}$ realisations of the optimisation problem have to be computed in each iteration. In practice, these are solved in parallel for N different $\omega \in \Omega$ and the results are then averaged. The large computational efforts caused by the slow Monte Carlo convergence are alleviated by the neural-network-based machine learning approaches presented in Section 3. In particular, gradient steps for arbitrary parameter realisations can be evaluated very efficiently and significantly fewer iterations (i.e., optimisation iterations) are required.

NEURAL NETWORK ARCHITECTURES

In modern scientific and engineering computing, machine learning techniques have become indispensable in recent years. The central goal of this work is to devise neural network architectures to facilitate an efficient computation of the risk-averse stochastic topology optimisation task. In this section, we develop two such architectures. The first one described in Section 3.1 is based on the popular convolutional neural networks (CNN) that were originally designed for the treatment of image data. In Section 3.2, a classical long short-term memory architecture (LSTM) is adapted to predict the gradient step.

The usage of deep neural networks with topology optimisation tasks have been previously examined in [4,10]. However, in contrast to other approaches, our architecture aims at a single NN that can be trained to handle arbitrarily fine meshes in terms of the requirements warranted during the topology optimisation process. More precisely, we seek an NN that predicts the gradient step $\varphi_{n+k} \in \mathbb{R}^{|V(\mathcal{T}_m)|}$ from Equation (6) discretised on an arbitrarily fine mesh $\mathcal{T}_m$ at an arbitrary iteration step $n \in \mathbb{N}$ with given $k \in \mathbb{N}$, for $\mathcal{N}^k : \mathbb{R}^{1+d \times |V(\mathcal{T}_m)|} \to \mathbb{R}^{|V(\mathcal{T}_m)|}$ such that

$$\mathcal{N}^k([\varphi_n, u_{n+1}]) = \varphi_{n+k}. \tag{13}$$

Thus, the total number of iterations required for the topology optimisation iteration should ideally be reduced, resulting in improved practical performance. For the sake of a convenient presentation, we consider all other coefficients of Equation (6) as constant in the following analyses. Alternatively, one would have to increase the complexity in the number of

degrees of freedom which are the weights describing the NN as well as the required training data. It can be assumed that with more information in the form of coefficients provided to the NN during training the accuracy of the resulting approximation of φ_n increases. Within the optimisation procedure, the actual calculation of the gradient step given in Equation (6) is performed on the basis of variable coefficients (e.g., τ and γ).

Since the discretisation $\varphi_n \in \mathbb{R}^{|V(\mathcal{T}_m)|}$ can be rewritten rather easily in tensor form, which represents the input of a CNN, this is the first architecture we consider in the next section.

Topology Convolutional Neural Networks (TCNN)

When using a visual representation of topologies as images (as they can be generated as output of a finite element simulation), the solution of Equation (6) can be easily transferred to the data structures used in CNNs. Consequently, predicting the gradient step with a CNN can be understood as a projection of the optimisation problem into a pixel-structured image classification problem. Here we assume that the calculation of the learned gradient step is encoded in the weights that characterise the NN.

In principle, the structure of a classical CNN consists of one or more convolutional layers followed by a pooling layer. This basic processing unit can repeat itself as often as desired. If there are at least three repetitions, we speak of a deep CNN and a deep learning architecture. In the convolutional layers a convolution matrix is applied to the input. The pooling layers are to be understood as a dimensional reduction of their input. Although common for image classification tasks, pooling layers are not used in the presented architecture.

TCNN Architecture

We follow the presentation of the pytorch documentation [18]. The input of a layer of the CNN architecture is a tensor $\mathcal{I} \in \mathbb{R}^{S \times C_{\text{in}} \times H \times W}$. Here, $S \in \mathbb{N}$ is the number of input samples, in our case the evaluations of φ and u as presented in Section 3.1.2. It is therefore possible to calculate the gradient step φ_n from Equation (6) for several different loads g simultaneously. This way, Monte Carlo estimates become very efficient. $C_{\text{in}} \in \mathbb{N}$ corresponds to the number of input channels and each channel represents one dimension of an input (φ or u). $H \in \mathbb{N}$ and $W \in \mathbb{N}$ provide information about the dimension

of the discretisation of the space D. The output of one CNN layer is specified by $\mathcal{O} \in \mathbb{R}^{S \times C_{out} \times H \times W}$. For fixed s≤S and i≤$C_{out}$∈$\mathbb{N}$ with C_{out} the number of output channels is given as

$$\mathcal{O}_{s,i}(\mathcal{I}_s) = b_i + \sum_{k=1}^{C_{in}} \mathcal{W}_{i,k} * \mathcal{I}_{s,k}. \tag{14}$$

Here, * denotes the cross-correlation operator, $b \in \mathbb{R}^{C_{out} \times H \times W}$ and $I_{s,k}$ with s≤S,k≤C_{in} is a cutout of I. The weight tensor $\mathcal{W} \in \mathbb{R}^{C_{out} \times C_{in} \times H_K \times W_K}$ determines the dimensions of the kernel (or convolution matrix) of the layers with $H_K, W_K \in \mathbb{N}$.

For simplicity, we henceforth assume S=1 unless otherwise specified. In particular, the entries of the weight tensor W are parameters that are optimised during the training of the CNN. Depending on the architecture of the CNN, an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ evaluated element-wise can additionally be applied to Equation (14).

Definition 2.

Let $b \in \mathbb{R}^{C_{out} \times H \times W}$ *and* $\mathcal{W} \in \mathbb{R}^{C_{in} \times C_{out} \times H_K \times W_K}$ *with* $L, H, W, H_K, W_K, C_{in}, C_{out} \in \mathbb{N}$ *be given by one parameter vector* $\theta \in \mathbb{R}^d$ *with*

$$d = C_{out} \cdot H \cdot W + C_{out} \cdot C_{in} \cdot H_K \cdot W_K.$$

Furthermore, let σ be a continuously differentiable activation function. We call a function

$$Conv(\,\cdot\,;\theta) : \mathbb{R}^{C_{in} \times H \times W} \to \mathbb{R}^{C_{out} \times H \times W} \tag{15}$$

a convolution layer with activation function σ if it satisfies

$$Conv(\mathcal{I};\theta)_i = \sigma\left(b_i + \sum_{k=1}^{C_{in}} \mathcal{W}_{i,k} * \mathcal{I}_k\right) \tag{16}$$

with i=1,…,Cout.

A sequential coupling of this layer structure provides the framework for the CNN. Specifically, for $i^L = 1, \ldots, C_{out}^L \in \mathbb{N}$,

$$\begin{aligned}\mathrm{Conv}(\,\cdot\,;\theta^{L-1})_{i^L}^{L} \circ \mathrm{Conv}(\,\cdot\,;\theta^{L-1})^{L-1} \circ \ldots \circ \mathrm{Conv}(\mathcal{I};\theta^{1})^{1} = \\ \sigma^{L}\Big(b_{c_{\mathrm{out}}}^{L} + \sum_{k_L=1}^{C_{\mathrm{out}}^{L-1}} \mathcal{W}_{c_{\mathrm{out}},k_L}^{L} * \sigma^{L-1}\Big(b_{c_{\mathrm{out}}}^{L-1} + \sum_{k_{L-1}=1}^{C_{\mathrm{out}}^{L-2}} \mathcal{W}_{c_{\mathrm{out}},k_{L-1}}^{L-1} * \ldots \\ \ldots * \sigma^{1}\Big(b_{c_{\mathrm{out}}}^{1} + \sum_{k_1=1}^{C_{\mathrm{in}}} \mathcal{W}_{c_{\mathrm{out}},k_1}^{1} * \mathcal{I}_{k_1}\Big)\ldots\Big)\Big)_{i^L}. \end{aligned} \tag{17}$$

Definition 3

(CNN architecture). *Let* $\mathcal{W}^1 \in \mathbb{R}^{C_{in}\times C_{out}^1\times H_K\times W_K}$, $\mathcal{W}^l \in \mathbb{R}^{C_{out}^{l-1}\times C_{out}^l\times H_K\times W_K}$ *for* $1 < l \leq L$ *and* $b^l \in \mathbb{R}^{C_{out}^l\times H\times W}$ *for* l≤L *with* L,H,W, $H_K, W_K, C_{in}, C_{out}^1, \ldots, C_{out}^L \in \mathbb{N}$ *be given by some parameter vector* $\theta \in \mathbb{R}^d$ *with*

$$d = \underbrace{C_{out}^1 * (C_{in}(H_K \cdot W_K) + (H \cdot W))}_{\text{Dimension of } \mathcal{W}^1 \text{ and } b^1} + \underbrace{\sum_{l=2}^{L} C_{out}^l * (C_{out}^{l-1}(H_K \cdot W_K) + (H \cdot W))}_{\text{Dimension of } \mathcal{W}^l \text{ and } b^l \text{ for } 2\leq l\leq L}.$$

Furthermore, let $\sigma^1,\ldots,\sigma^L$ *be given continuously differentiable activation functions. We call an NN of the form of Equation (**17**) an L-layer topology convolutional neural network (TCNN) and characterise it as the mapping* $\mathcal{N}_{CNN}(\,\cdot\,;\theta) : \mathbb{R}^{C_{in}\times H\times W} \to \mathbb{R}^{C_{out}\times H\times W}$.

The approximation of NCNN is hence determined by its parameter vector θ. For general CNNs, the dimension H×W does not have to be constant across the different layers. The same holds true for the dimensions $H_K \times W_K$ of the kernel matrices. In fact, before implementing the convolution, we embed each channel of our input in a $(H + \lfloor \frac{H_K}{2} \rfloor) \times (W + \lfloor \frac{W_K}{2} \rfloor)$ space to preserve the dimension in the output.

Example 1.

The following specific TCNN has proved to be the most suitable for integration into Algorithm A1 for the selections of hyperparameters we have investigated. The architecture is given as an L=6 *layer TCNN with* C_{in}=3 *input channels,* $C_{out}^l = 15$ *for* 1<l<5 *hidden channel,* $C_{out}^6 = 1$ *output channel and kernel size* H_K=W_K=3 *as well as trained weights described by* $\theta \in \mathbb{R}^{8806}$ *which determine the mapping by*

$$\mathcal{N}_{CNN}(\,\cdot\,;\theta) : \mathbb{R}^{3\times 201\times 101} \to \mathbb{R}^{1\times 201\times 101}, \tag{18}$$

with activation function $\sigma^6(x):=\min\{\max\{x,0\},1\}$. *In contrast to many standard architectures, only the activation function of the output layer is not the identity. We chose* $\mathbb{R}^{3\times 201\times 101}$ *as input space in anticipation of the setting in Example 2, reflecting our mesh choice to discretise domain* $D=[-1,1]\times[0,1]$ *with* 201×101 *nodes, which for first-order finite elements then is the dimension of the discrete functions u and φ. The architecture is depicted in Figure 1.*

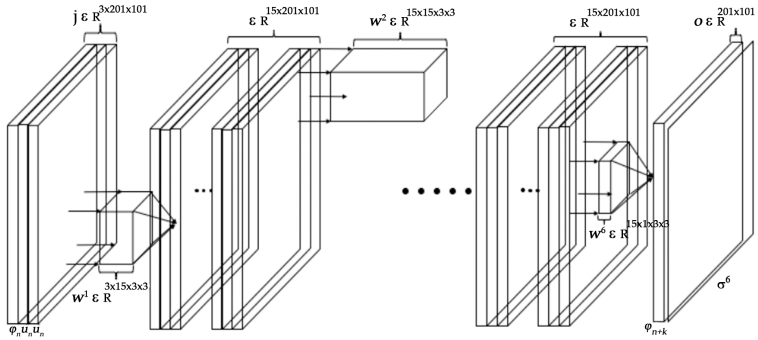


Figure 1: Visualisation of the TCNN from Example 1.

Data Preparation

On the algorithmic level, our goal is to replace the computationally costly lines 5 and 6 of all $c_m \in \mathbb{N}$ loop iterations of Algorithm A1 with a TCNN. This is not directly possible (at least for a TCNN) since the input space $\mathbb{R}^{C_{\text{in}}\times H\times W}$ of a TCNN does not match the mesh T_m on which the finite element discretisation and thus the optimisation of φ_n takes place in the current optimisation step *t*. Hence, it is necessary to project the evaluation of φ onto the format of a CNN. For this we define a transformation between $\mathbb{R}^{|V(\mathcal{T}_m)|\times C_{\text{in}}}$ and the input tensor $\mathbb{R}^{H\times W\times C_{in}}$ of $\mathcal{N}_{\text{CNN}}$. As described in Appendix B, we do not assume that the mesh T_m stays fixed in the optimisation algorithm and we instead generate a sequence of different meshes T_m by some adaptive mesh refinement, which has led to significant efficiency improvements in [1]. To obtain unique transformations between the discretisation finite element space and the input space of the NN, one can interpolate the current solutions of φ_n and u_{n+1} from T_m onto a constant reference mesh T_{const} by polynomial interpolations

$$p : \mathbb{R}^{|V(\mathcal{T}_m)|\times C_{\text{in}}} \to \mathbb{R}^{H\cdot W\times C_{\text{in}}},$$
$$q : \mathbb{R}^{H\cdot W\times C_{\text{out}}} \to \mathbb{R}^{|V(\mathcal{T}_m)|\times C_{\text{out}}}.$$

Hence, during the optimisation, the current solutions are interpolated via the operator p to the reference mesh, rendering the prediction independent from the actual adaptive mesh. After the N_{CNN} prediction of the gradient step on the reference mesh, it is mapped back to the actual computation mesh via q.

Consequently, we define the reference mesh $\mathcal{T}_{\text{const}} = (V(\mathcal{T}_{\text{const}}), E(\mathcal{T}_{\text{const}}))$ with vertices V and edges E as a graph such that $|V(\mathcal{T}_{\text{const}})| = H.W.$ Each node $v_i \in V(\mathcal{T}_{\text{const}})$ corresponds to the values of $\varphi_n^i = \varphi_n(v_i) \in \mathbb{R}$ and $u_n^i = u_n(v_i) \in \mathbb{R}^d$, $i \le H \cdot W = |V(\mathcal{T}_{\text{const}})|$ at node vi. The features of the nodes can hence be interpreted as rows of a feature matrix,

$$\widetilde{\mathcal{I}_n} = \begin{bmatrix} \varphi_1 & u_1 \\ \vdots & \vdots \\ \varphi n_{H\cdot W} & u n_{H\cdot W} \end{bmatrix} \in \mathbb{R}^{H\cdot W\times C_{in}}. \tag{19}$$

The structure of T_{const} is illustrated in Figure 2.

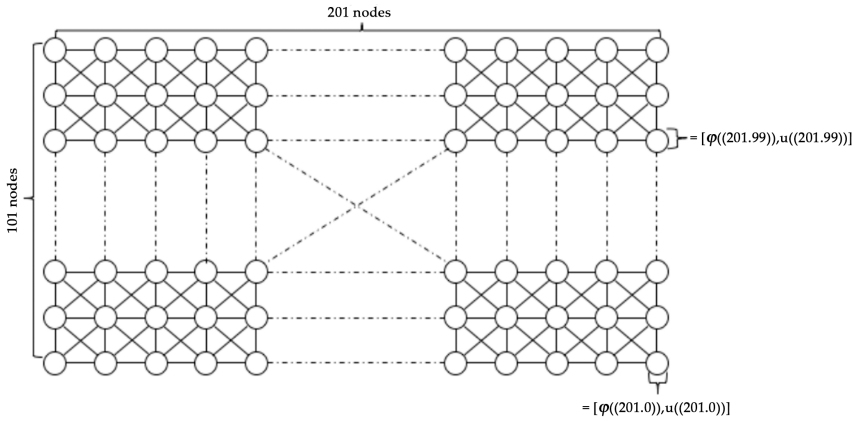


Figure 2: $\mathcal{T}_{\text{const}}$ for $\Phi_{C_{\text{in}}} : \mathbb{R}^{3\times 101\cdot 201} \to \mathbb{R}^{1\times 201\times 101}$ to transform N_{TCNN} from Example 1.

One can now define a transformation between $\mathbb{R}^{H\cdot W\times C_{in}}$ and $\mathbb{R}^{C_{in}\times H\times W}$ by

$$\begin{aligned}\Phi_{C_{in}} &: \mathbb{R}^{H\cdot W\times C_{\text{in}}} \to \mathbb{R}^{C_{\text{in}}\times H\times W},\\ \Phi_{C_{\text{out}}} &: \mathbb{R}^{C_{\text{out}}\times H\times W} \to \mathbb{R}^{H\cdot W\times C_{\text{out}}}.\end{aligned} \tag{20}$$

Hence, the approximation of the gradient step 6 of Algorithm A1 is basically a coupling of the mappings p, Φ and $\mathrm{N}_{\mathrm{CNN}}$, namely

$$\begin{aligned}\mathbb{R}^{|V(\mathcal{T}_m)|\times C_{\text{in}}} \xrightarrow{p} \mathbb{R}^{H\cdot W\times C_{\text{in}}} \xrightarrow{\Phi_{C_{\text{in}}}} \mathbb{R}^{C_{\text{in}}\times H\times W} \xrightarrow{\mathcal{N}_{\text{CNN}}}\\ \xrightarrow{\mathcal{N}_{\text{CNN}}} \mathbb{R}^{C_{\text{out}}\times H\times W} \xrightarrow{\Phi_{C_{\text{out}}}} \mathbb{R}^{H\cdot W\times C_{\text{out}}} \xrightarrow{q} \mathbb{R}^{|V(\mathcal{T}_m)|\times C_{\text{out}}}.\end{aligned} \tag{21}$$

Example 2

(Illustrating the TCNN). *The NN given by the coupling of functions in Equation (21) with* $\mathrm{N}_{\mathrm{CNN}}$ *given as in Example 1 can be described by*

$$\mathcal{N}^k_{CNN}(\,\cdot\,;\theta) : \mathbb{R}^{|V(\mathcal{T}_m)|\times 3} \to \mathbb{R}^{|V(\mathcal{T}_m)|}, \tag{22}$$

with

$$\begin{bmatrix}\varphi_n & u^x_{n+1} & u^y_{n+1}\end{bmatrix} \mapsto \begin{bmatrix}\varphi_{n+k}\end{bmatrix} \tag{23}$$

for $\varphi_n \in \mathbb{R}^{|V(\mathcal{T}_m)|}$ *and* $u_n = (u^x_n, u^y_n)$, $u^x_n, u^y_n \in \mathbb{R}^{|V(\mathcal{T}_m)|}$ *defined on* T_{m}. *Hence, this NN can be applied directly to the finite element discretisations* φ_{n} *and* un *used in Algorithms A1 and A2.*

With the TCNN from the example in Equation (23) we have extended Algorithm A1. More precisely, we have inserted an NN approximation $\mathcal{N}^k_{\mathrm{CNN}}(\varphi_n, u_{n+1}; \theta)$ in each of the $\mathrm{c_m}\in\mathbb{N}$ steps, which predicts k iteration steps by just one evaluation. For this, the sequence $\mathrm{c_m}$ has to be defined in advance. We leave it to future research to adaptively control the sequence cm dynamically within the optimisation algorithm. This extension of Algorithm A1 is described by Algorithm 1. In an analogous way, we also extend Algorithm A2 by the TCNN given in Equation (23). In particular, we are able to evaluate all samples $\mathrm{S}\in\mathbb{N}$ in parallel by adding additional sample dimensions to the input tensor of the TCNN given in Equation (14). This procedure is illustrated in Algorithm 2. Again, the parameter cm has to be chosen in advance.

Algorithm 1: Deterministic optimisation algorithm with TCNN approximated gradient step

Input: mesh $\mathcal{T}_0$, $\mathcal{T}_{const}$ and initial values φ_0 sequence $c_m \in \mathbb{N}$ and $j = 0 \in \mathbb{N}$
for $m = 0, 1, \ldots$ *until converged* **do**
  for $n = 0, 1, \ldots$ *until converged* **do**
    solve state equation on mesh $\mathcal{T}_m \Rightarrow u_{n+1}$
    if $j = c_m$ **then**
      interpolate φ_n onto $\mathcal{T}_{\text{const}}$ project u_{n+1}, φ_n to $\mathbb{R}^{d+1\times H\times W}$
      evaluate $\mathcal{N}^k_{\text{CNN}}(\varphi_n, u_{n+1}; \theta)$ as performed in Equation (21) $\Rightarrow \varphi_{n+k}$
      project φ_{n+k} to $\mathcal{T}_{\text{const}}$
      interpolate φ_{n+k} onto $\mathcal{T}_m$
      $j = 1$
    else
      solve adjoint equation on mesh $\mathcal{T}_m \Rightarrow p_{n+1}$
      solve gradient equation on mesh $\mathcal{T}_m \Rightarrow \varphi^*_{n+1}$
      project φ^*_{n+1} to $[0, 1] \Rightarrow \varphi_{n+1}$
      $j = j + 1$
    end
  end
  adapt mesh according to Appendix B $\Rightarrow \mathcal{T}_{m+1}$
end

Algorithm 2: Stochastic optimisation algorithm with TCNN approximated gradient step

Input: mesh $\mathcal{T}_0$, $\mathcal{T}_{\text{const}}$ and initial values φ_0, sequence $c_m \in \mathbb{N}$ and $j = 0 \in \mathbb{N}$
for $m = 0, 1, \ldots$ *until converged* **do**
  for $n = 0, 1, \ldots$ *until converged* **do**
    for $i = 1, \ldots, N$ **do**
      sample $g(\omega_i), \lambda_{mat}(\omega_i), \mu_{mat}(\omega_i)$
      solve state equation on mesh $\mathcal{T}_m \Rightarrow u_{n+1}(\omega_i)$
      solve adjoint equation on mesh $\mathcal{T}_m \Rightarrow p_{n+1}(\omega_i)$
      if $j = c_m$ **then**
        interpolate $\varphi_n(\omega_i)$ onto $\mathcal{T}_{\text{const}}$
        project $u_{n+1}(\omega_i), \varphi_n(\omega_i)$ to $\mathbb{R}^{d+1\times H\times W}$
        evaluate $\mathcal{N}^k_{\text{CNN}}(\varphi_n(\omega_i), u_{n+1}(\omega_i); \theta)$ as performed in Equation (21) $\Rightarrow \varphi_{n+k}(\omega_i)$
        project $\varphi_{n+k}(\omega_i)$ to $\mathcal{T}_{\text{const}}$
        interpolate $\varphi_{n+k}(\omega_i)$ onto $\mathcal{T}_m$
        $j = 1$
      else
        solve gradient equation on mesh $\mathcal{T}_m \Rightarrow \varphi^*_{n+1}(\omega_i)$
      end
    end
    compute the mean $\hat{\varphi}_{n+1} = \frac{1}{N}\sum_{i=1}^{N} \varphi^*_{n+1}(\omega_i)$
    project $\hat{\varphi}_{n+1}$ to $[0, 1] \Rightarrow \varphi_{n+1}$
    adapt mesh according to Appendix B $\Rightarrow \mathcal{T}_{m+1}$
    $j = j + 1$
  end
  adapt mesh according to Appendix B $\Rightarrow \mathcal{T}_{m+1}$
end

Topology Long Short-Term Memory Neural Networks (TLSTM)

One possible approach to improve the prediction of φ_n using an NN is to provide the classifier not just one tuple (φ_n, u_{n+1}) as an inpu, but to have it process a larger amount of information by a sequence of these tuples of the last $T \in \mathbb{N}$ iteration steps, i.e.,

$$\Big((\varphi_{n-T}, u_{n-T+1}), (\varphi_{n-T+1}, u_{n-T+2}), \ldots, (\varphi_n, u_{n+1})\Big).$$

Through this, the shift of the phase field or the change of the topology φ_n over time is also transferred as input to the NN. The sequence prediction problem considered in this case differs from the single step time prediction in the sense that the prediction target is now a sequence that contains both spatial and temporal information. Theoretically, this information can also be learned directly from the NN. However, in practice it is more effective to adapt the architecture to the information we have in advance (in our case with respect to the time dependency) to achieve better results. An NN that allows exactly this is a recurrent Neural Network (RNN). Unfortunately, standard RNNs often suffer from the vanishing gradient problem [19,20] which we try to prevent from the beginning. Therefore, we build on the special RNN concept of a Long Short-Term Memory (LSTM) in the context of our problem, which is more robust against the vanishing gradient issue and provides promising results, especially in the analysis of time series. For a background on time series analysis and a review of the different methods, we refer to the survey article [21]. In practice, time series are usually stored as one-dimensional sequences in vector format. Consequently, there is no out-of-the-box LSTM layer implementation for structures such as the input tensor we require in Equation (14). Nevertheless, we still do not wish to abandon the mechanism of convolution within the NN in order to keep the structural information of φ and u. An LSTM layer with a convolutional structure can be constructed by replacing the matrix vector multiplication within a standard LSTM layer by convolutional layers. The unique advantage of an LSTM according to [20] is its cell-gate architecture, which mitigates the vanishing gradient problem. More precisely, it consists of a "memory cell" $c_t : \mathbb{R}^{4\times d_{\text{in}}^t} \to \mathbb{R}^{d_{\text{out}}^t}$ that serves as an accumulator of the current state $t \le T, t \in \mathbb{N}$, in the processed sequence. The information capacity of the last status c_{t-1} within c_t is controlled by the activation of the so-called "forget gate" $f_t : \mathbb{R}^{3\times d_{\text{in}}^t} \to \mathbb{R}^{d_{\text{out}}^t}$. The information capacity of the input

state $x_t \in \mathbb{R}^{d_{\text{in}}^t}$ is controlled by the activation of the input gate it. Which information (or whether any at all) gets transferred from memory cell c_t to state $h_t : \mathbb{R}^{2\times d_{\text{in}}^t} \to \mathbb{R}^{d_{\text{out}}^t}$ is in turn controlled by the activation of the output gate o_t. From a technical point of view, the gates can be understood as learning forward layers.

TLSTM Architecture

An ordinary LSTM layer to generate complex sequences with long-range structure as presented in [22] corresponds to the described logic above and can be formulated numerically for a sequence of one-dimensional input state $x_t \in \mathbb{R}^{d_{\text{in}}^t}$ and output vector $h_t \in \mathbb{R}^{d_{\text{out}}^t}$ as an equation system

$$\begin{aligned} i_t(x_t, h_{t-1}, c_{t-1}) &= \sigma(\mathcal{W}_{xi}x_t + \mathcal{W}_{hi}h_{t-1} + \mathcal{W}_{ci} \odot c_{t-1} + b_i), \\ f_t(x_t, h_{t-1}, c_{t-1}) &= \sigma(\mathcal{W}_{xf}x_t + \mathcal{W}_{hf}h_{t-1} + \mathcal{W}_{cf} \odot c_{t-1} + b_f), \\ c_t(f_t, c_{t-1}, i_t, x_t, h_{t-1}) &= f_t \odot c_{t-1} + i_t \odot \tanh(\mathcal{W}_{xc}x_t + \mathcal{W}_{hc}h_{t-1} + b_c), \\ o_t(x_t, h_{t-1}, c_t) &= \sigma(\mathcal{W}_{xo}x_t + \mathcal{W}_{ho}h_{t-1} + \mathcal{W}_{co} \odot c_t + b_o), \\ h_t(o_t, c_t) &= o_t \odot \tanh(c_t), \end{aligned} \tag{24}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ with $\sigma(x) = \frac{1}{1+e^x}$ and tanh are evaluated element-wise. The operation $\odot$ denotes the Hadamard product and the subscripts of the weight matrices $\mathcal{W} \in \mathbb{R}^{d_{\text{out}}^t \times d_{\text{in}}^t}$ describe the affiliation to the gates. For example, W_{xi} is the weight matrix to input x_t of gate it. This illustrates how the weights of the LSTMs are transferred to the weights of convolution LSTMs in the following.

We want to reformulate Equation (24) by replacing all matrix-vector multiplications (i.e., the forward layer) with a convolution layer from Definition 2. This is inspired by [23], which has already provided the theoretic architecture of a convolutional LSTM layer with the approach on precipitation forecasting. Let Conv(·;θ): $\mathbb{R}^{C_{\text{in}}\times H\times W} \to \mathbb{R}^{C_{\text{out}}\times H\times W}$ be a convolutional layer and $\mathcal{I} \in \mathbb{R}^{T\times C_{\text{in}}\times H\times W}$ a sequence of inputs ordered by the discrete time dimension $T \in \mathbb{N}$. A convolutional LSTM layer to an input sequence $I_{t\leq T}$ and $\mathcal{H}_0 = 0 \in \mathbb{R}^{C_{\text{in}}\times H\times W}$, $\mathcal{C}_0 = 0 \in \mathbb{R}^{C_{\text{in}}\times H\times W}$ (since at t=1 we do not yet have any information about earlier steps in the sequence) is given by a system of equations,

$$\begin{aligned}
\hat{i}_t(\mathcal{I}_t, \mathcal{H}_{t-1}, \mathcal{C}_{t-1}) &= \sigma\Big(\mathrm{Conv}(\mathcal{I}_t; \theta_{xi}) + \mathrm{Conv}(\mathcal{H}_{t-1}; \theta_{hi}) + \mathcal{W}_{ci} \odot \mathcal{C}_{t-1}\Big),\\
\hat{f}_t(\mathcal{I}_t; \mathcal{H}_{t-1}, \mathcal{C}_{t-1}) &= \sigma\Big(\mathrm{Conv}(\mathcal{I}_t; \theta_{xf}) + \mathrm{Conv}(\mathcal{H}_{t-1}; \theta_{hf}) + \mathcal{W}_{cf} \odot \mathcal{C}_{t-1}\Big),\\
\mathcal{C}_t(\hat{f}_t, \mathcal{C}_{t-1}, \hat{i}_t, \mathcal{I}_t) &= \hat{f}_t \odot \mathcal{C}_{t-1} + \hat{i}_t \odot \tanh\Big(\mathrm{Conv}(\mathcal{I}_t; \theta_{xc}) + \mathrm{Conv}(\mathcal{H}_{t-1}; \theta_{hc})\Big),\\
\hat{o}_t(\mathcal{I}_t, \mathcal{H}_{t-1}, \mathcal{C}_t) &= \sigma\Big(\mathrm{Conv}(\mathcal{I}_t; \theta_{xo}) + \mathrm{Conv}(\mathcal{H}_{t-1}; \theta_{ho}) + \mathcal{W}_{co} \odot \mathcal{C}_t\Big),\\
\mathcal{H}_t(\hat{o}_t, \mathcal{C}_t) &= \hat{o}_t \odot \tanh(\mathcal{C}_t),
\end{aligned} \tag{25}$$

with $t \le T, t \in \mathbb{N}$ and $\mathcal{H} \in \mathbb{R}^{T \times C_{\mathrm{out}} \times H \times W}$ the output of the convolutional LSTM layer as well as $\mathcal{W}_{ci}, \mathcal{W}_{cf} \in \mathbb{R}^{C_{\mathrm{in}} \times H \times W}, \mathcal{W}_{co} \in \mathbb{R}^{C_{out} \times H \times W}$ in Equation (24). The subscript t indicates a cutout of the t-th element of sequence dimension T of the respective tensor.

Definition 4

(LSTM layer). *Let* $L, H, W, T, C_{\mathrm{in}}, C_{\mathrm{out}} \in \mathbb{N}$ *as well as* $\mathcal{W}_{ci}, \mathcal{W}_{cf} \in \mathbb{R}^{C_{in} \times H \times W}$, $\mathcal{W}_{co} \in \mathbb{R}^{C_{out} \times H \times W}$ *and parameter vectors, specifying the convolutional layer as in Definition 3 for* $L=1$ *from Equation (**25**),*

$$\theta_{xi} \in \mathbb{R}^{d_{xi}}, \theta_{hi} \in \mathbb{R}^{d_{hi}}, \theta_{xc} \in \mathbb{R}^{d_{xc}}, \theta_{hc} \in \mathbb{R}^{d_{hc}},$$

$$\theta_{xf} \in \mathbb{R}^{d_{xf}}, \theta_{hf} \in \mathbb{R}^{d_{hf}}, \theta_{xo} \in \mathbb{R}^{d_{xo}}, \theta_{ho} \in \mathbb{R}^{d_{ho}},$$

and described by the parameter vector $\theta \in \mathbb{R}^d$, *with*

$$d = d_{xi} + d_{hi} + d_{xc} + d_{hc} + d_{xf} + d_{hf} + d_{xo} + d_{ho} + 2 \cdot C_{in} \cdot H \cdot W + \cdot C_{out} \cdot H \cdot W.$$

Furthermore, let $\sigma : \mathbb{R} \to \mathbb{R}$ *with* $\sigma(x) = \frac{1}{1+e^x}$ *and tanh be evaluated element-wise. We call a function,*

$$LSTM(\,\cdot\,; \theta) : \mathbb{R}^{3 \times T \times C_{in} \times H \times W} \to \mathbb{R}^{2 \times T \times C_{out} \times H \times W},$$

an LSTM layer, if it satisfies the mapping rule given by the system of Equation (25).

For the forecasting of our gradient sequence, we use an encoder–decoder architecture (i.e., an "autoencoder") consisting of $2L, L \in \mathbb{N}$, LSTM layers,

$$\mathrm{LSTM}^l(\,\cdot\,; \theta^l) : \mathbb{R}^{3 \times T \times C^l_{\mathrm{in}} \times H \times W} \to \mathbb{R}^{2 \times T \times C^l_{\mathrm{out}} \times H \times W},$$

that satisfies the mapping rule given by the equation system (25) with $1 \le l \le 2L$. Therefore, the encoding and decoding blocks of the autoencoder have the same number of layers $L \in \mathbb{N}$. The autoencoder for an input

sequence $\mathcal{I} \in \mathbb{R}^{T \times C_{in} \times H \times W}$ can be described by the following system of equations of the encoder block,

$$\begin{aligned}
\mathrm{LSTM}^1(\mathcal{I}_t, \mathcal{C}_{t-1}^1, \mathcal{H}_{t-1}^1; \theta^1) &= [\mathcal{C}_t^1, \mathcal{H}_t^1], \\
\mathrm{LSTM}^2(\mathcal{H}_t^1, \mathcal{C}_{t-1}^2, \mathcal{H}_{t-1}^2; \theta^2) &= [\mathcal{C}_t^2, \mathcal{H}_t^2], \\
&\vdots \\
\mathrm{LSTM}^{L-1}(\mathcal{H}_t^{L-2}, \mathcal{C}_{t-1}^{L-1}, \mathcal{H}_{t-1}^{L-1}; \theta^{L-1}) &= [\mathcal{C}_t^{L-1}, \mathcal{H}_t^{L-1}], \\
\mathrm{LSTM}^{L}(\mathcal{H}_t^{L-1}, \mathcal{C}_t^{L}, \mathcal{H}_{t-1}^{L}; \theta^{L}) &= [\mathcal{C}_t^{L}, \mathcal{H}_t^{L}],
\end{aligned} \tag{26}$$

with $1 \le t \le T_{en}, t \in \mathbb{N}$. This is combined with the following decoder block by setting $\widetilde{\mathcal{O}}_0 = \mathcal{H}_{T_{en}}^L$ and $\mathcal{H}_0^{L+1} = \mathcal{H}_{T_{en}}^1, \ldots, \mathcal{H}_0^{2L} = \mathcal{H}_{T_{en}}^L$ and $\mathcal{C}_0^{L+1} = \mathcal{C}_{T_{en}}^1, \ldots, \mathcal{C}_0^{2L} = \mathcal{C}_{T_{en}}^L$,

$$\begin{aligned}
\mathrm{LSTM}^{L+1}(\widetilde{\mathcal{O}}_{t-1}, \mathcal{C}_{t-1}^{L+1}, \mathcal{H}_{t-1}^{L+1}; \theta^{L+1}) &= [\mathcal{C}_t^{L+1}, \mathcal{H}_t^{L+1}], \\
\mathrm{LSTM}^{L+2}(\mathcal{H}_t^{L+1}, \mathcal{C}_{t-1}^{L+2}, \mathcal{H}_{t-1}^{L+2}; \theta^{L+2}) &= [\mathcal{C}_t^{L+2}, \mathcal{H}_t^{L+2}], \\
&\vdots \\
\mathrm{LSTM}^{2L-1}(\mathcal{H}_t^{2L-2}, \mathcal{C}_{t-1}^{2L-1}, \mathcal{H}_{t-1}^{2L-1}; \theta^{2L-1}) &= [\mathcal{C}_t^{2L-1}, \mathcal{H}_t^{2L-1}], \\
\mathrm{LSTM}^{2L}(\mathcal{H}_t^{2L-1}, \mathcal{C}_t^{2L}, \mathcal{H}_{t-1}^{2L}; \theta^{2L}) &= [\mathcal{C}_t^{2L}, \widetilde{\mathcal{O}}_t],
\end{aligned} \tag{27}$$

with $1 \le t \le T_{dec}, t \in \mathbb{N}$.

It should be mentioned that the input and output sequences do not have to be of the same length (in fact, in general $T_{en} \neq T_{dec}$). Furthermore, $C_{out}^{l-1} = C_{in}^{l}$ holds for individual LSTM layers $2 \le l \le 2L$ defined in Equations (26) and (27). Especially, since the output sequence $\widetilde{\mathcal{O}}$ is also input of LSTM^{L+1}, it holds that $C_{out}^{2L} = C_{in}^{L+1}$. In order to be able to select the dimensions of the output tensor $\widetilde{\mathcal{O}}$ completely independently of the hidden channels, we additionally apply a convolutional layer $\mathrm{Conv}(\,\cdot\,; \theta^{\mathrm{final}}) : \mathbb{R}^{C_{out}^{2L} \times H \times W} \to \mathbb{R}^{C_{\mathrm{final}} \times H \times W}$ with activation function $\sigma_{\mathrm{final}} : \mathbb{R} \to \mathbb{R}$ to concatenate the hidden channel to an arbitrary number of output channels $C_{\mathrm{final}} \in \mathbb{N}$ given by

$$\mathrm{Conv}(\widetilde{\mathcal{O}}_t; \theta^{\mathrm{final}}) = \mathcal{O}_t \quad , 1 \le t \le T_{\mathrm{dec}}. \tag{28}$$

Definition 5

(TLSTM architecture). *Let* L,H,W, $T_{en}, T_{dec}, C_{final}, C_{in}^{1} \in \mathbb{N}$ *as well as* $C_{out}^{l} \in \mathbb{N}$, *with* 1≤l≤2L *be given. Hence, the respective LSTM layers from the encoder defined in Equation (26) and decoder in (27) block as well as the output layer in Equation (28) can be described by the parameter vectors of the CNN and LSTM layers (see Definition 3 for* L=1 *and Definition 4)* $\theta^{final} \in \mathbb{R}^{d^f}$, $\theta^{l} \in \mathbb{R}^{d^l}$ *with* $d^f \in \mathbb{N}, d^l \in \mathbb{N}$ *for* 1≤l≤2L *and an activation function* $\sigma_{final} : \mathbb{R} \to \mathbb{R}$ *of the output layer. These parameter vectors as well as the weight tensors* $\mathcal{W}_{ci}^{l}, \mathcal{W}_{cf}^{l} \in \mathbb{R}^{C_{in} \times H \times W}$ *and* $\mathcal{W}_{co}^{l} \in \mathbb{R}^{C_{out} \times H \times W}$ *for* l≤2L *can in turn be described collectively by the parameter vector* $\theta \in \mathbb{R}^{d}$ *, where*

$$d = d_f + \sum_{l=1}^{2L} d^l.$$

We call an NN as described in Equations (26)–(28) Convolutional Topology Long Short-Term Memory (TLSTM) and characterise it by

$$\mathcal{N}_{LSTM}(\,\cdot\,;\theta) : \mathbb{R}^{T_{en} \times C_{in} \times H \times W} \to \mathbb{R}^{T_{dec} \times C_{final} \times H \times W}.$$

The difference between a TLSTM and an LSTM is therefore the structure of the input tensor $\mathbb{R}^{T_{\text{en}} \times C_{in} \times H \times W}$ of a TLSTM instead of $\mathbb{R}^{T_{\text{en}} \times C_{\text{in}} \times H}$ and the internal calculation carried out with convolutional layers instead of standard multiplications.

Example 3.

For the experiments in Section 4.2, the underlying L=4 *layer TLSTM with* C_{in}=3 *input channels,* $C_{out}^{l} = 9$,l≤4 *hidden channels,* C_{final}=1 *output channel, kernel size of all included convolutional layers* H_K=W_K=3 *and sequence lengths* T_{en}=5, T_{dec}=10 *with trained weights described by* $\theta \in \mathbb{R}^{509266}$ *are given as*

$$\mathcal{N}_{LSTM}\,\cdot\,;\theta) : \mathbb{R}^{5\times3\times201\times101} \to \mathbb{R}^{10\times1\times201\times101}, \tag{29}$$

with activation function $\sigma^{final}(x) = \min\{\max\{x,0\}1\}$. *The autoencoder structure for an 8-layer LSTM described in Equations (26)–(28) for (29) is visualised in Figure 3.*

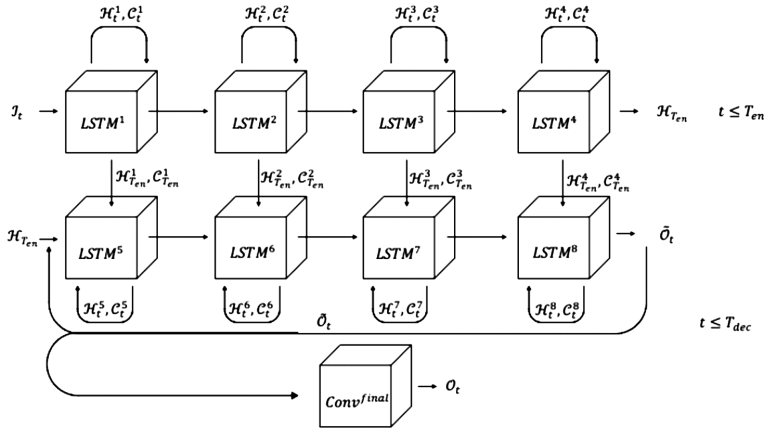


Figure 3: Autoencoder architecture N_{LSTM} of Example 3.

Data Preparation

As in the case of the integration of the NCNN proposed in Section 3.1, we want to replace lines 5 and 6 in Algorithm A1 with the approximation of the N_{LSTM} from Definition 5. In case of a TLSTM, the evaluations of φ and u have to be transformed from the graph structure of the finite element simulation into an appropriate tensor format. This in principle is analogous to the composition $\Phi_{C_{in}} \circ p$ in Equation (21). The only difference is that now this is performed on a sequence of evaluations φ_{m_n} and u_{m_n} of length $T_{en} \in \mathbb{N}$. As the subscripts suggest, such a sequence does not necessarily have to be evaluated on a fixed mesh T_m, it may extend over a sequence of meshes T_m. However, since we use polynomial interpolations

$$p_T : \mathbb{R}^{|V(\mathcal{T}_m)| \times C_{in} \times T} \to \mathbb{R}^{H \cdot W \times C_{in} \times T},$$

$$q_T : \mathbb{R}^{H \cdot W \times C_{out} \times T} \to \mathbb{R}^{|V(\mathcal{T}_m)| \times C_{out} \times T}$$

to transfer the sequence $\varphi_{n-T},\ldots,\varphi_n$ and $u_{n-T+1},\ldots,u_{n+1}$ onto some reference mesh $\mathcal{T}_{const} = (V(\mathcal{T}_{const}), E(\mathcal{T}_{const}))$.

We intend to process T_{en} feature matrices of the form of Equation (19). Hence, we define transformations

$$\Phi_{C_{\text{in}},T} : \mathbb{R}^{H\cdot W\times C_{\text{in}}\times T} \to \mathbb{R}^{T\times C_{\text{in}}\times H\times W},$$
$$\Phi_{C_{\text{out}},T} : \mathbb{R}^{T\times C_{\text{out}}\times H\times W} \to \mathbb{R}^{H\cdot W\times C_{\text{out}}\times T}. \tag{30}$$

Thus, the approximation of a gradient step in Algorithm A1 by a TLSTM can be understood as a concatenation of the form

$$\mathbb{R}^{|V(\mathcal{T}_m)|\times C_{\text{in}}\times T_{\text{en}}} \xrightarrow{p_T} \mathbb{R}^{H\cdot W\times C_{\text{in}}\times T_{\text{en}}} \xrightarrow{\Phi_{C_{\text{in}},T_{\text{en}}}} \mathbb{R}^{T\times C_{\text{in}}\times H\times W} \xrightarrow{\mathcal{N}_{\text{CNN}}}$$
$$\xrightarrow{\mathcal{N}_{\text{CNN}}} \mathbb{R}^{T\times C_{\text{out}}\times H\times W} \xrightarrow{\Phi_{C_{\text{out}},T_{\text{dec}}}} \mathbb{R}^{H\cdot W\times C_{\text{out}}\times T_{\text{dec}}} \xrightarrow{q_T} \mathbb{R}^{|V(\mathcal{T}_m)|\times C_{\text{out}}\times T_{\text{dec}}} \tag{31}$$

Example 4

(The TLSTM). *The NN given by the coupling of functions from Equation (**31**), where* $\mathrm{N}_{\mathrm{LSTM}}$ *as given in Example 3, can be described by*

$$\mathcal{N}_{LSTM}(\,\cdot\,;\theta) : \mathbb{R}^{|V(\mathcal{T}_m)|\times 3\times 5} \to \mathbb{R}^{|V(\mathcal{T}_m)|\times 10} \tag{32}$$

and

$$\begin{bmatrix} \varphi_{n-5} & u^x_{n-5+1} & u^y_{n-5+1} \\ & \vdots & \\ \varphi_n & u^x_{n+1} & u^y_{n+1} \end{bmatrix} \mapsto \begin{bmatrix} \varphi_{n+1} \\ \vdots \\ \varphi_{n+10} \end{bmatrix}$$

for $\varphi_n \in \mathbb{R}^{|V(\mathcal{T}_m)|}$ *and* $u_n = (u^x_n, u^y_n)$, $u^x_n, u^y_n \in \mathbb{R}^{|V(\mathcal{T}_m)|}$ *defined on mesh* T_{m}.

The TLSTM from Example 4 can directly be integrated into Algorithm A1 as with the previous integration with Algorithm 1. In fact, since in Algorithm A1 only the most recent gradient step is relevant, in practice we restrict the inverse mapping on the last element of the predicted sequences to save calculation time. The only difference is that the sequences $\varphi_{n-T},\ldots,\varphi_n$ and $u_{n-T+1},\ldots,u_{n+1}$ have to be stored in a list. We chose c_l=125 and c_m=50 for all $2\leq m\in\mathbb{N}$. This procedure is described in Algorithm 3. As in the case of the TCNN, we are able to include the extra sample dimension S to approximate gradient steps from multiple problems at once.

Algorithm 3: Deterministic optimisation algorithm with TLSTM approximated gradient step

Input: mesh $\mathcal{T}_0$, $\mathcal{T}_{\text{const}}$, initial values φ_0, sequence $c_m \in \mathbb{N}$ and $j = 1, T \in \mathbb{N}$ and list $L = [\varphi_0]$

for $m = 0, 1, \ldots$ *until converged* **do**
  for $n = 0, 1, \ldots$ *until converged* **do**
    solve state equation on mesh $\mathcal{T}_m \Rightarrow u_{n+1}$
    add u_{n+1} to L
    if $j = c_m$ **then**
      interpolate $\varphi_{n-T}, \ldots, \varphi_n$ and $u_{n-T1}, \ldots, u_{n+1}$ on to $\mathcal{T}_{\text{const}}$
      project $\varphi_{n-T}, \ldots, \varphi_n$ and $u_{n-T+1}, \ldots, u_{n+1}$ on to $\mathcal{T}_{\text{const}}$ to $\mathbb{R}^{T\times d+1\times H\times W}$
      evaluate $\mathcal{N}^T_{\text{LSTM}}(\varphi_{n-T}, \ldots, \varphi_n, u_{n-T+1}, \ldots, u_{n+1}; \theta)$ as performed in Equation (31) $\Rightarrow \varphi_n \ldots \varphi_{n+T}$
      project φ_{n+T} to $\mathcal{T}_{\text{const}}$
      interpolate φ_{n+T} onto $\mathcal{T}_m$
      declare list and add $\varphi_{n+T} \Rightarrow L = [\varphi_{n+T}]$
      $j = 1$
    else
      solve adjoint equation on mesh $\mathcal{T}_m \Rightarrow p_{n+1}$
      solve gradient equation on mesh $\mathcal{T}_m \Rightarrow \varphi^*_{n+1}$
      project φ^*_{n+1} to $[0, 1] \Rightarrow \varphi_{n+1}$
      add φ_{n+1} to L
      $j = j + 1$
    end
  end
  declare list L
  $j = 1$
  adapt mesh according to Appendix B $\Rightarrow \mathcal{T}_{m+1}$
end

RESULTS

This section is devoted to numerical results of the two previously described neural network architectures. The implementations were conducted with the open source packages PyTorch [18] for the NN part and FEniCS [24] for the FE simulations (see introduction for the link to the code repository). We first illustrate the performance of the TCNN in Section 4.1 with a deterministic bridge example compared to a classical optimisation. The important observation is that with the TCNN the optimisation can be carried out with far fewer optimisation steps while still leading to the reference topologies from [1]. Similar results can be observed for the risk-averse stochastic optimisation. In Section 4.2, numerical experiments of the TLSTM architecture are presented. The performance is revealed to be comparable to the TCNN architecture and the optimisation appears to be more robust with respect to the data realisations.

TCNN Examples

Before we can use the TCNN architecure for the optimisation in Algorithm 1, we have to train it on data that describe the system response of Equation (6). Note that it would not be useful to allow the N_{CNN} to learn the gradient steps of a fixed setting since different settings of the bridge problem from Appendix A.1 should efficiently be tackled. In considering the stochastic setting of the problem as defined in Section 2.2, the TCNN is trained to learn the gradient steps φ for a random $g : \Omega \to \mathbb{R}^d$.

Sampling the Data

To train the architecture, appropriate training data have to be generated. In order to achieve this, we chose the same setting for *g* as in Appendix A.2. Using the optimiser in Algorithm A1, we can generate $S \in \mathbb{N}$ different sample paths of gradient steps $\varphi_n(g)$ and solutions of the state equation $u_n(g)$ by generating *S* samples of *g*. In this procedure, we store every $k \in \mathbb{N}$ iteration step of φ_n and u_n in order to approximate *k* gradient steps at once. More precisely, we store $\left\lfloor \frac{N_{\max}}{k} \right\rfloor - 1$ tuples

$$\left(\left[\varphi_n \quad u^x_{n+1} \quad u^y_{n+1} \right], \left[\varphi_{n+k} \right] \right), \tag{33}$$

with $0 \le n \le N_{max} - k$, where $N_{max} \in \mathbb{N}$ is fixed in advance, representing the maximum number of iterations of an optimisation. For the training of the models in the following experiments, we have chosen $N_{max}=500$ since the topologies have mostly converged after this number of iterations. The overall number of $S\left(\left\lfloor \frac{N_{\max}}{k} \right\rfloor - 1\right)$ tuples are merged into an unsorted data set DCNN.

TCNN Predictions

The following experiment validates that the performance of Algorithm A1 can be replicated or (desirably) improved by including a CNN as described in Algorithm 1. As a first test, we illustrate that the proposed new architecture is indeed capable of predicting the gradients of the optimisation procedure.

Figure 4 shows the evaluations of the model Equation (22) after determining θ within the training of the NN on the data set D_{CNN}. Here, the prediction $\mathcal{N}^k_{CNN}(\varphi_n, u_{n+1}; \theta)$ and the actual gradient step φ_{n+k}

generated by Algorithm A1 are compared for different loads sampled from a truncated normally distributed g. Since the predictions of $\mathcal{N}^5_{\mathrm{CNN}}$ are hardly distinguishable from the reference fields, we have also trained Equation (22) to predict larger time steps (for 25 and 100 iteration steps at once). However, these NNs have proved to be less reliable in practice as prediction quality decreases. An illustrative selection of some predictions is provided in Figure A5 and Figure A6 in Appendix C.

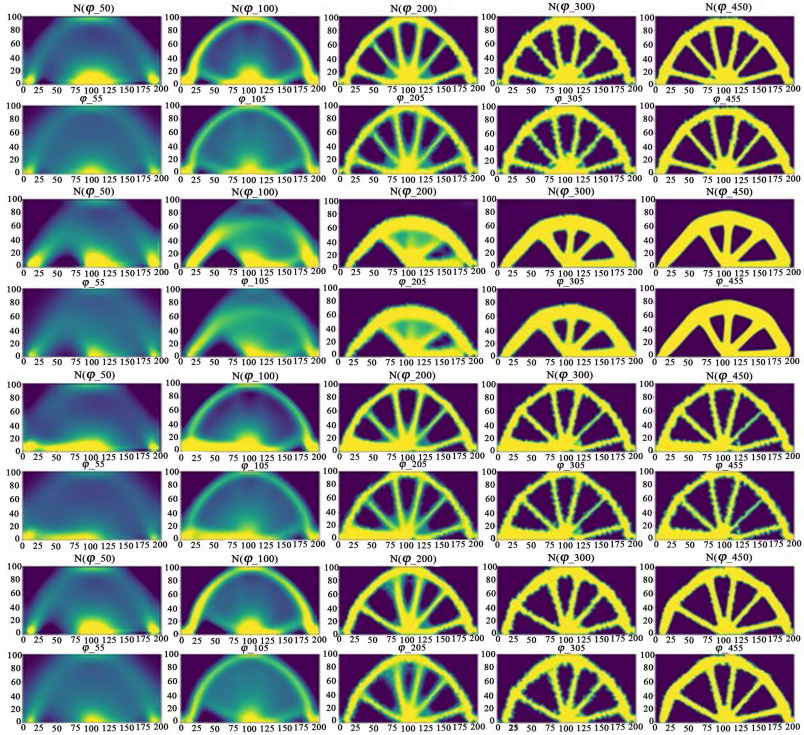


Figure 4: $\mathcal{N}^5_{CNN}(\varphi_n, u_{n+1}; \theta)$ (top row) in comparison with φ_{n+5} (bottom row).

Deterministic Bridge Optimisation

In these experiments we compare the performance of Algorithm 1 with that of Algorithm A1 in the setting of Appendix A.1. For the NN in Algorithm 1, we use Equation (22) from Example 2. For $c_m \in \mathbb{N}$ in Algorithm 1 we chose $c_m = 55$ for all $m \in \mathbb{N}$. In order to train Equation (22), data of the

form of Equation (33) from Section 4.1.1 are used. We expect that the best (reference) results in the setting from experiment A.1 can be obtained, since the distribution of the training data is a truncated normal distribution around this expected value and we thereby have an accumulation of training points around the load $g=(0,-5000)^T$. As a convergence criterion, we used the convergence criterion of the mesh refinement of Equation (A1) on a maximally fine mesh with $|V(T_m)|\leq 15{,}000$. A sub-sequence generated by Algorithm 1 is shown in Figure A7a in comparison to that of Appendix A.1 in Figure A7b in Appendix C. There, it can be observed that the classical and CNN-assisted optimisation results essentially appear identical with the faster CNN convergence.

The metrics for evaluating our algorithms have also improved through the application of the CNN as can be observed in Figure 5. For better comparability, we ran both algorithms 10 times and averaged all metrics. To be more precise, this is only the average of the calculation time, as the remaining metrics are deterministic and therefore always the same. It is easy to see how the metrics diverge with the first application of the CNN at iteration step 55, especially by the computation time required per iteration. The most significant indicator is the evaluation of $J^{\varepsilon}(\varphi_n)$ per iteration of Equation (3) at the top-right of Figure 5. The graph for Algorithm 1 reaches a constant lower level than that of Algorithm A1 after about 250 iterations and thus fulfils the convergence criterion earlier. Accordingly, the step size criterion for τ_n applies earlier by using the CNN, which further accelerates convergence. An interesting insight is provided by the calculation time, which shows that the actual time required per iteration step is more or less the same, except for the iteration steps in which Equation (**22**) is applied. This is indicated by the upward outliers in the computation time series. This additional computation cost can be explained by the application of the mesh projection of Equation (**21**), which represents an aspect requiring further improvements. Nevertheless, in total we achieve a shorter total run-time due to the faster convergence of Algorithm 1.

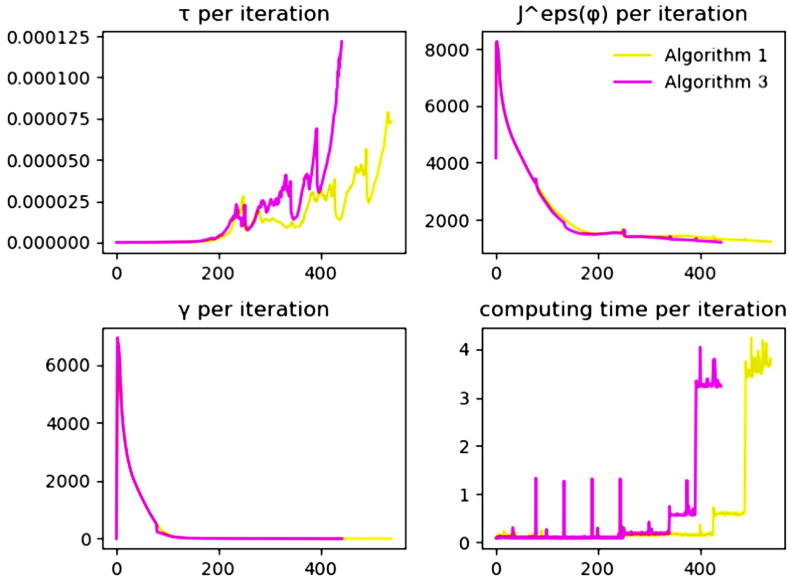


Figure 5. Comparison of metrics between Algorithms A1 and 1.

A detailed list of the run-times and target value metrics for the functional $J^{\varepsilon}(\varphi_n)$ is provided in Table 1. The evaluation of $J^{\varepsilon}(\varphi_{n_{\text{final}}})$ denotes the value of the functional for the topology $\varphi_{n_{\text{final}}}$ converged after $n_{\text{final}} \in \mathbb{N}$ iteration steps. The compliance is the value that is actually minimised in terms of the functional $J^{\varepsilon}(\varphi_{n_{\text{final}}})$. Algorithm 1 requires less computing time than the reference procedure after extending it with the CNN architecture from Equation (22).

Table 1: Comparison of metrics between Algorithms A1 and 1

Method	Applied Load g	Converges after n_{final}	Evaluation of $J^{\varepsilon}(\varphi_{n_{\text{final}}})$	Compliance $\int_{\Gamma_g} g \cdot u(\varphi)\, ds$
Algorithm A1	$(0,-5000)^T$	538	1475.39	1173.22
Algorithm 1	$(0,-5000)^T$	441	1206.49	1148.66

Stochastic Bridge Optimisation

Algorithm A2 can easily be extended by the TCNN of Equation (22) in order to improve the efficiency for topology optimisation under uncertainties. The corresponding procedure is shown in Algorithm 2 where we chose cm=55 for all $m \in \mathbb{N}$. Note that the predictions of the different realisations of $\varphi_n(\omega_i), \omega_i \in \Omega$ for $i=1,\ldots,S \in \mathbb{N}$ (where an evaluation $\varphi_n(\omega_i)$ are to be interpreted as transformation of an evaluation from $g(\omega_i)$) are actually not

executed within a loop but in parallel (lines 8–12 of Algorithm 2). This is possible because NNs are generally able to process batches of data in parallel. We have also implemented parallelisation for Algorithm A2, which is limited by the number of processor cores of the actual compute cluster. We want to compare the performance of Algorithm A2 and Algorithm 2 using the same setting as in Appendix A.2. To ensure comparable results despite stochastic parameters, we set the random seed to 42 before running both algorithms. The resulting sub-sequences of φ_n are compared side by side in Figure 6. Although the topology converges after fewer iterations with Algorithm A2, one can see that the topology resulting from Algorithm 2 has a more stable shape since the topology does not lose material to the unnecessary extra spoke. This is confirmed by the metrics in Table 2 where one can see that Algorithm 2 achieved a lower compliance after fewer iteration steps. Additionally, the optimisation of Algorithm 2 is stopped after 500 iterations to show that it achieves a better result in less time as shown in Figure 7. A notable observation is that the times of applying Equation (22) in Algorithm 2 can be identified by the spikes in the computation time of the iterations. It can be seen that despite the additional time required by the transformation in Equation (21), the calculation of a stochastic gradient step using Equation (22) is generally faster. This is due to the dynamic parallelisation that PyTorch provides when processing batches (in our case the approximation of multiple evaluations from $\varphi_n(\omega_i)$ with NNs). However, the amount of evaluations of $\varphi_n(\omega_i)$ that Algorithm A2 can process at once is limited by the number of available processors. Since the calculation time for the evaluation of an optimisation step $\varphi_{n+1}(\omega_i)$ increases with finer meshes, the evaluation of the approximation of all gradient steps $\varphi_{n+1}(\omega_1),\ldots,\varphi_{n+1}(\omega_S)$ at once results in a processing time advantage for the NN. It is to be expected that this time saving increases with the number of examples S.

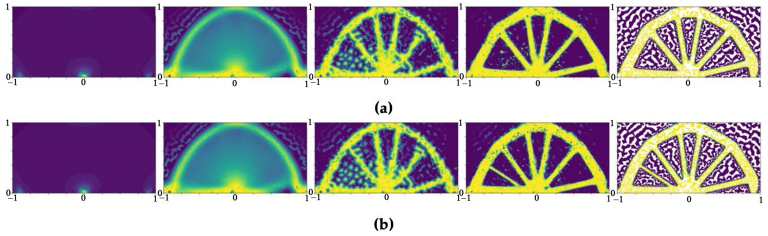


Figure 6: Classical risk-averse stochastic optimisation (top) and NCNN accelerated (bottom). (a) Sequence $\varphi_1, \varphi_{100}, \varphi_{200}, \varphi_{400}, \varphi_{517}$ from Algorithm 2. (b) Sequence $\varphi_1, \varphi_{100}, \varphi_{200}, \varphi_{400}, \varphi_{490}$ from Algorithm A2.

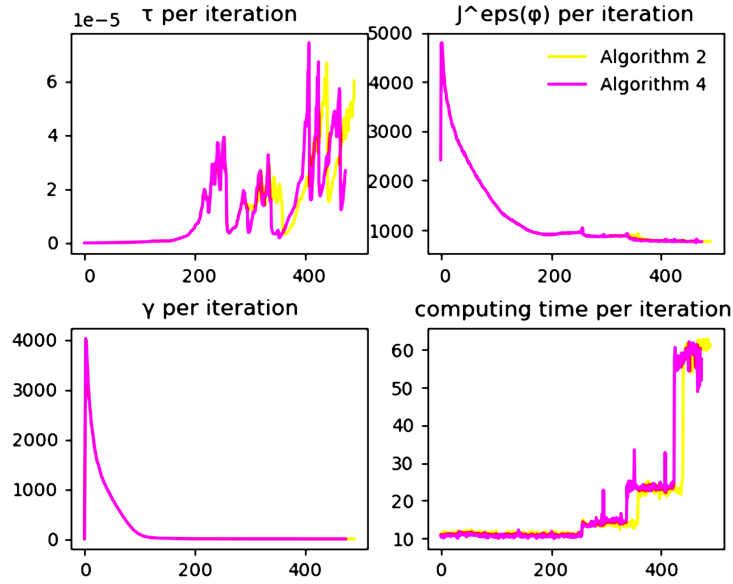


Figure 7: Comparison of metrics between Algorithms A2 and 2.

Table 2: Comparison of metrics between Algorithms A2 and 2

Method	Converges after n_{final}	Evaluation of $J_0^\varepsilon(\varphi_{n_{\text{final}}})$	Compliance $\mathbb{E}\left[\int_{\Gamma_g} g \cdot u(\varphi_{n_{\text{final}}})\,\mathrm{d}s\right]$	Samples
Algorithm A2	490	765.85	729.20	224
Algorithm 2	517	744.11	708.61	224
Algorithm 2	500	760.26	723.24	224

As mentioned at the beginning of the experiment, we expect to achieve good results close to the mean of $g=(0,-5000)^T$. In order to obtain a more general view of the quality of Algorithm 1, we have compiled a selection of extreme cases for the distribution of g (e.g., evaluations from g that deviate strongly from $(0,-5000)^T$) in Figure 8 and Table 3. The figure shows the sequence of φ_n in hundreds of steps as well as the final distribution of material $(\varphi_{100}, \varphi_{200}, \ldots, \varphi_{n_{\text{final}}})$ for the specific loads g. Table 3 indicates a noticeable saving in calculation time, but there is no guaranteed improvement in the results. In particular, when the topology "collapses" (i.e., the NN cannot generalise to the input data with strong deviations from the training data), the application of the CNN leads to worse results. Nevertheless, it can be seen that the NN extension gives the algorithm a greater robustness against porous fragments (see Figure 8b) in the optimisation of the topology and thus a higher stability against collapsing of the topology in the optimisation can be assumed. Finally, a critical aspect to be mentioned is the step size cm.

The time at which Equation (22) is applied and which is controlled by $c_m \in \mathbb{N}$ has a crucial impact on the viability or "compatibility" between the state of the optimisation procedure and the CNN. In some cases, a cm that is too small or too large can lead to the collapse of the topology, i.e., the topology deteriorates and does not recover. For the reliable use of Algorithm 1, a method for controlling cm would have to be devised.

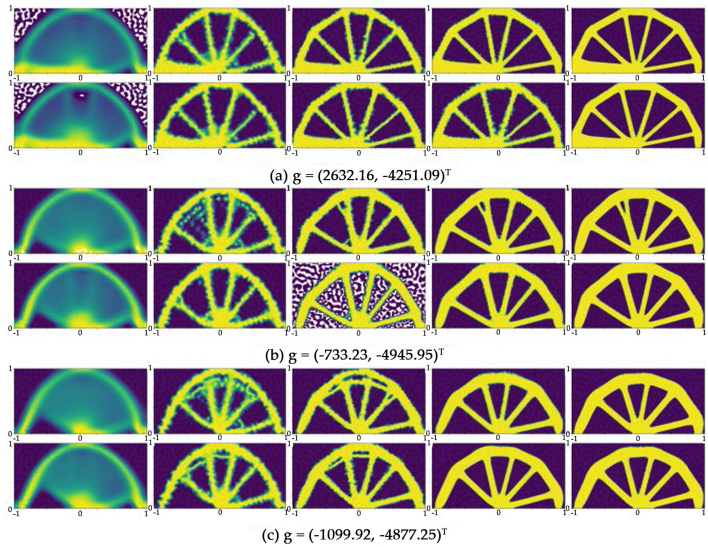


Figure 8: Comparison of metrics between Algorithms A1 (top row) and 1 (bottom row) for different loads g.

Table 3: Comparison of metrics between Algorithms A1 and 1

Method	Applied Load g	Converges after n_{final}	Evaluation of $J^{\varepsilon}(\varphi_{n_{\text{final}}})$	Compliance $\int_{\Gamma_g} g \cdot u(\varphi)\, ds$
Algorithm A1	$(0, -5000)^T$	538	1475.39	1173.22
Algorithm 1	$(0, -5000)^T$	441	1206.49	1148.66
Algorithm A1 (see Figure 8a)	$(2632.16, -4251.09)^T$	473	1487.39	1416.46
Algorithm 1 (see Figure 8a)	$(2632.16, -4251.09)^T$	517	1475.05	**1405.39**
Algorithm A1 (see Figure 8b)	$(-733.23, -4945.95)^T$	**457**	1115.66	1062.41
Algorithm 1 (see Figure 8b)	$(-733.23, -4945.95)^T$	497	1049.91	**1042.58**
Algorithm A1 (see Figure 8c)	$(-1099.92, -4877.25)^T$	470	1042.18	**992.28**
Algorithm 1 (see Figure 8c)	$(-1099.92, -4877.25)^T$	447	1048.44	998.34

TLSTM Examples

As in Section 4.1.3, randomly generated training data should be used in the following experiment with the derived LSTM transformation of Equation (31). The data tuples consist of input and output from N_{LSTM} according to Example 4.

Sampling the Data

Again, we assume an expected load $g=(0,-5000)^T$ and a random rotation characterised by a truncated normal distribution with bounds $[-\frac{\pi}{2},\frac{\pi}{2}]$, standard deviation 0.3 and mean 0. Using the optimiser in Algorithm A1, $S\in\mathbb{N}$ sample paths of gradient steps $\varphi(g)$ and solutions of the state equation $u(g)$ are generated by drawing S realisations of g. In contrast to Section 4.1.1, this time we do not only store every $T\in\mathbb{N}$ iteration step of φ_n and u_n but instead store all iteration steps of the optimisation of Algorithms A1. Afterwards, these are merged into disjoint subsets, each consisting of a sequence of T iteration steps. Thus, the feature or the sequence $\varphi_{n+1},\ldots,\varphi_{n+T}$ is the label for the assembled sequence from $\varphi_{n-T},\ldots,\varphi_n$ and $u_{n-T+1},\ldots,u_{n+1}$. More precisely, with

$$\left(\begin{bmatrix}\varphi_{n-T} & u^x_{n-T+1} & u^y_{n-T+1}\\ & \vdots & \\ \varphi_n & u^x_{n+1} & u^y_{n+1}\end{bmatrix},\begin{bmatrix}\varphi_{n+1}\\ \vdots\\ \varphi_{n+T}\end{bmatrix}\right),$$

for $0\le n\le N-(T-1)$, a total of $S(\lfloor\frac{N}{k}\rfloor-1)$ tuples are stored in an unsorted dataset DLSTM.

TLSTM Predictions

After training the TLSTM from Equation (29) with the data set D_{LSTM} generated in Section 4.2.1, we wish to investigate its predictive ability of $\mathcal{N}^T_{LSTM}(\varphi_n) := \mathcal{N}^T_{LSTM}(\varphi_{n-T},\ldots,\varphi_n,u_{n-T+1},\ldots,u_{n+1};\theta)$ compared to the real sequence $(_{\varphi n+T})_n$. For this purpose we have visualised both sub-sequences for T=10 in Figure 9 using the iteration sequence generated for a load $g=(0,-5000)$. The distorted topology in the first forecasts is striking. This can be attributed to the comparatively low weighting of training data in which the distribution of the material is constant $\varphi_n(x)=0{,}5$ for all $x\in D$ or almost constant. This forces us to choose a correspondingly high $c_m\in\mathbb{N}$ in Algorithm 3. Furthermore, it can be seen that especially in early phases of the partial sequence in which the change $\|\varphi_n-\varphi_{n+1}\|$ is very high, $\mathcal{N}^{10}_{LSTM}(\varphi_n)$ provides a better forecast from a visual perspective, i.e., the topology $\mathcal{N}^{10}_{LSTM}(\varphi_n)$ has already converged further than the target image φ_{n+10}. Since the topologies on the finer meshes no longer show any major

visual changes and therefore the differences in the predictions are no longer recognisable, we have decided not to present them at this point.

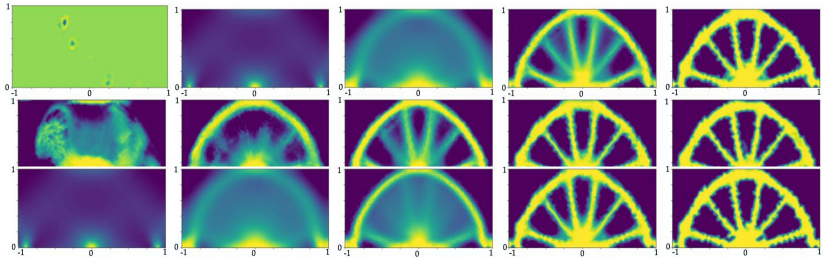


Figure 9: Input φ (top row), $\mathcal{N}^{10}_{\mathrm{LSTM}}(\varphi_n)$ (center row) in comparison with φ_{n+10} (bottom row).

The architecture of the TLSTM allows the length of the input sequence as well as the output sequence to be chosen independently of the training data. The expected consequence is a decrease in prediction quality. Despite this, Figure 10 depicts the prediction results of Equation (29) with unchanged input sequence (T_{en}=5) and output sequence of length 40. Since a shorter output sequence (T_{des}=10) is used to train Equation (29), the results of the longer output sequence indicate that N_{LSTM} has indeed learned to predict a gradient step for the given setting and that the training data from Section 4.2.1 describe the problem correctly.

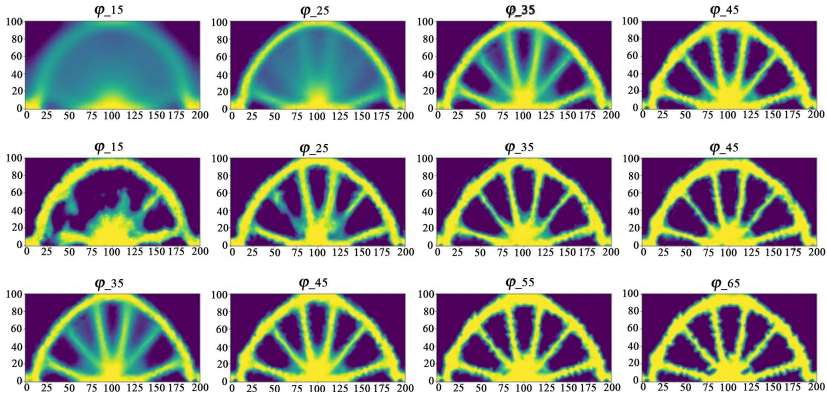


Figure 10: Input φ (top row), $\mathcal{N}^{20}_{\mathrm{LSTM}}(\varphi_n)$ (center row) in comparison with φ_{n+20} (bottom row).

Analogous to Section 3.1.1, the intention behind the construction of N_{LSTM} is to replace the gradient step in reference Algorithm A1 with Example 3. Algorithm 3 describes the integration of the LSTM prediction. When evaluating $\mathcal{N}_{LSTM}^{T}$, only the last iteration step φ_{n+T} of the predicted sequence is projected back to the current mesh Tm in order to save computational resources. We examine the performance of Algorithm 3 in the following deterministic experiment. As for the TCNN above, the beneficial performance of a single-gradient prediction transfers to the stochastic setting since it consists of a Monte Carlo estimator with $N \in \mathbb{N}$ samples in each step. It is hence not necessary to examine this in more detail.

Deterministic Bridge Optimisation

The motivation for the design of the TLSTM architecture was that the information contained in the time series of φ_n and u_n could ideally lead to an improvement in the forecast capabilities of the NN. This can be investigated as in Section 4.1.4 by calculating the optimal topology for different loading scenarios for *g* by Algorithm 3. Again, all results are based on the ten-fold averaged performance of the algorithms for each load *g*. Figure 11 shows the results in the setting similar to Appendix A.1 and compares the respective metrics. Analogous to Section 4.1.3 the sequence φ_n of the optimisation by Algorithm 3 is also more resilient to porous fragments in the structures than the reference optimisation procedure. In general, the pictures of φ_n hardly differ between Algorithms 1 and 3. Hence, apart from Figure 11, no further visualisations are presented. It should also be noted that the optimisation by Algorithm 3 is much less stable than the optimisation using N_{CNN} from Equation (23). This becomes apparent when the structure collapses which was the case in each of our test runs if the $c_m \in \mathbb{N}$ chosen was too small in Algorithm 3. Furthermore, it could be observed that the convergence criterion of Equation (A1) was not reached after applying N_{LSTM} because φ_n diverged too far from the actual minimum on T_m. In conclusion, the stability of Algorithm 3 is even more dependent on c_m than it is with Algorithm 1, which renders parameter calibration more difficult. However, the metrics in Figure 11 show that Algorithm 3 converges faster than Algorithm A1 and often achieves better results.

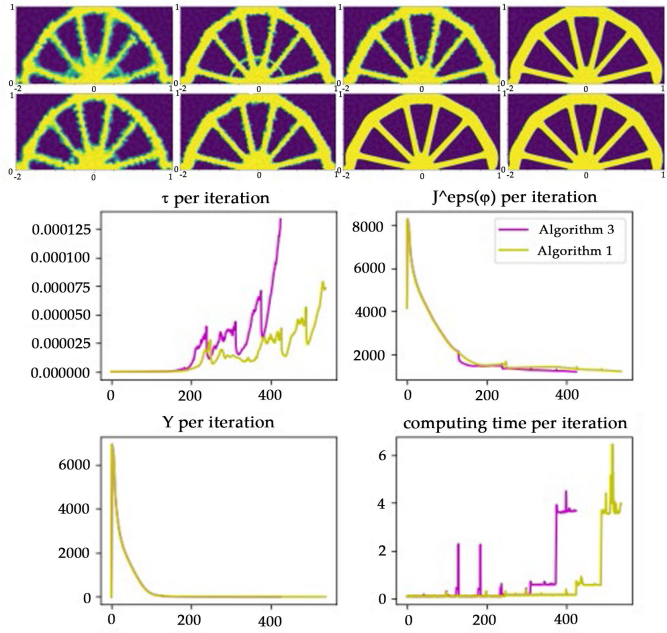


Figure 11: Comparison of $\varphi_{200}, \varphi_{300}, \varphi_{400}, \varphi_{n_{\text{final}}}$ between Algorithms A1 (top row) and 3 (bottom row).

One conspicuous feature is the high fluctuation in the calculation time per iteration in the applications of $N_{LSTM}(\cdot;\theta)$ during the optimisation when compared to Algorithm 1. On the one hand, this is due to the comparatively high complexity of the TLSTM. In this context, high complexity means a high dimension $d \in \mathbb{N}$ of the parameter vector $\theta \in \mathbb{R}^d$. On the other hand, the main driver of the higher calculation time is the transformation given by Equation (31) since on an algorithmic level the entire input sequence $\varphi_n, \varphi_{n+1}, \varphi_{n+2}, \varphi_{n+3}, \varphi_{n+4}$ has to be stored and transformed. In general, it becomes apparent at this point that the transformations in Equations (21) and (31) are the critical aspects that compromise the performance of Algorithms 1 and 3.

Table 4 compares the performance of the three presented algorithms. In overall terms, Algorithm 3 achieves better compliance, whereas Algorithm 1 stands out owing to its shorter calculation time.

Table 4: Comparison of metrics between Algorithms A1, 1 and 3

Method	Applied Load g	Computation Time	Converges after n_{final}	Evaluation of $J^{\varepsilon}(\varphi_{n_{\text{final}}})$	Compliance $\int_{\Gamma_g} g \cdot u(\varphi)\,\mathrm{d}s$
Algorithm A1	$(0, -5000)^T$	4 min 43 s	538	1475.39	1173.22
Algorithm 1	$(0, -5000)^T$	4 min 05 s	441	**1206.49**	**1148.66**
Algorithm 3	$(0, -5000)^T$	4 min 34 s	**425**	1209.24	1151.27
Algorithm A1	$(2632.16, -4251.09)^T$	4 min 31 s	473	1487.39	1416.46
Algorithm 1	$(2632.16, -4251.09)^T$	**4 min 14 s**	517	1475.05	1405.39
Algorithm 3	$(2632.16, -4251.09)^T$	4min 34 s	**436**	**1209.24**	**1151.27**
Algorithm A1	$(-733.23, -4945.95)^T$	4 min 50 s	**457**	1115.66	1062.41
Algorithm 1	$(-733.23, -4945.95)^T$	**4 min 21 s**	497	**1049.91**	1042.58
Algorithm 3	$(-733.23, -4945.95)^T$	4 min 41 s	484	1082.62	**1031.62**
Algorithm A1	$(-1099.92, -4877.25)^T$	5 min 11 s	470	1042.18	992.28
Algorithm 1	$(-1099.92, -4877.25)^T$	**4 min 42 s**	**447**	1048.44	998.34
Algorithm 3	$(-1099.92, -4877.25)^T$	4 min 43 s	542	**1041.28**	**992.16**
Algorithm A1	$(-3197.16, -3844.24)^T$	2 min 17 s	365	**845.93**	**805.94**
Algorithm 1	$(-3197.16, -3844.24)^T$	**2 min 13 s**	**346**	852.36	811.15
Algorithm 3	$(-3197.16, -3844.24)^T$	2 min 59 s	366	848.52	808.40

DISCUSSION AND CONCLUSIONS

The objective of this work is to devise neural network architectures that can be used for efficient topology optimisation problems. These tasks are computationally burdensome and typically are inevitably carried out with a large number of optimisation steps, each requiring (depending on the chosen method) the solution of state and adjoint equations to determine the gradient direction. Instead of learning a surrogate for state and adjoint equations, we present NN architectures that directly predict this gradient, leading to very efficient optimisation schemes. A noteworthy aspect of our investigation is the consideration of uncertainties of model data in a risk-averse optimisation formulation. This is a generalisation of the notion of "loading scenarios" that are commonly used in practice for a fixed set of parameter realisations. With our continuous presentation of uncertainties in the material and of the load acting on the considered structure, the robustness of the computed design with respect to these uncertainties can be controlled by the parameter of the CVaR used in the cost functional. Since computations with uncertainties require a substantial computational effort, our central goal is to extend the algorithms used in [1,2] by introducing appropriate NN predictions, reducing the iteration steps required. In contrast to other machine learning approaches, our aim is to achieve this even for adaptively adjusted finite element meshes since this has proven to be crucial for good performance in previous work. For this to function, an underlying sufficiently fine reference mesh is assumed for the training data and the prediction. Moreover, in contrast to other NN approaches for this problem, we consider the evolution of a continuous (functional) representation of a phase field determining the material distribution.

Ideally, the NN architectures should hasten the deterministic topology optimisation problem and consequently the risk-averse optimisation under uncertainties. This is achieved in Section 3.1 by embedding a CNN in the optimisation for both the deterministic and the stochastic setting.

The observed numerical results for a common 2D bridge benchmark are on par with the reference method presented in [1]. However, the gradient step predicted by the NN architectures allows for significantly larger iteration steps, rendering the optimisation procedure more efficient. This directly transfers to the Monte-Carlo-based risk-averse optimisation under uncertainties as defined by Algorithm 2 since the samples for the statistical estimation are obtained with minimal cost. In addition to the CNN, a second architecture is illustrated in terms of an LSTM. This generally leads to a better quality of the optimisation and is motivated by the idea that a memory of previous gradients may lead to a more accurate prediction of the next gradient step. However, it comes at the cost of longer computation times due to the transformation between the different adapted computation meshes (see Section 4.4). Hence, a substantial performance improvement could be achieved by reducing the complexity of the transformations of Equations (21) and (31).

There are several interesting research directions from the presented approach and observed numerical results. Regarding the chosen architectures, an interesting extension would be to consider graph neural networks (GNN) since there, the underlying mesh structure is mapped directly to the NN. Consequently, the costly transfer operators from current mesh to reference mesh of the design space could be alleviated, removing perhaps the largest computational burden of our approach. Moreover, transformer architectures have probably superseded LSTMs and it would be worth examining this modern architecture in the context of this work.

The loss function used in the training also leaves room for improvements. For example, instead of the simple mean squared error used here, one could approximate the objective functional of Equation (5) directly in the loss function. Regarding the training process, there are modern techniques to improve the efficiency and alleviate over-fitting such as early stopping, gradient clipping, adaptive learning rates and data augmentation as discussed in [25]. Moreover, transfer learning in a limited-data setting could substantially reduce the amount of training data required.

This work mainly serves as a proof of concept for treating the considered type of optimisation problems with modern NN architectures. An important

step towards practicability is the further generalisation of this model, e.g., to arbitrary problems (with parameterised boundary data and constraints), determined by descriptive parameters drawn from arbitrary distributions according to the problem at hand. Moreover, the models presented here can be used as a basis for theoretical proofs (e.g., regarding the complexity of the representation) and further systematic experiments.

AUTHOR CONTRIBUTIONS

Conceptualisation, M.E. and J.N.; methodology, M.E., M.H. and J.N.; software and numerical experiments, M.H.; investigation, M.H. and J.N.; writing—original draft, M.E. and M.H.; supervision, M.E.; project administration, M.E.; funding acquisition, M.E. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENTS

We thank the WIAS for providing the IT infrastructure to conduct the presented numerical experiments. We are also grateful to the anonymous reviewers and Anne-Sophie Lanier for remarks and suggestions that improved the original manuscript.

APPENDIX A. BRIDGE BENCHMARK PROBLEM

Appendix A.1. Deterministic Bridge Optimisation

For a comparison between Algorithm A1 from [**1**,**2**] and its extension to NNs, we make use of a bridge benchmark problem at selected locations. The name comes from the optimal shape resembling a bridge, which exhibits the best stiffness under the given constraints and forces acting on it. To be specific, the parameters are given as follows: Assume design domain $D=[-1,1]\times[0,1]$ with boundaries $\Gamma_D=[-1,-0.9]\times\{0\}$, $\Gamma_g=[-0.02,0.02]\times\{0\}$ on which the load $g=(0,-5000)^T$ is applied and the slip condition $\Gamma_S=[0.9,1.0]\times\{0\}$ is set. The Lamé coefficients are given by $\mu_{mat}=\lambda_{mat}=150$. Furthermore, the volume constraint is $m=0.4$ and $\varepsilon=116$. This is the same setting as in the deterministic experiment in [1,2].

The initial material distribution is given by $\varphi_0(x)=0.5\forall x\in D$. Several iteration results of φ_n from Algorithm A1 are depicted in Figure A1. As can be seen by the finer edges in the images, in the course of optimising the

topology (compare, e.g., φ_{200} and φ_{538}), an adapted mesh is used which is refined depending on φ_n in order resolve fine details of the topology and to save computational costs (see Appendix B).

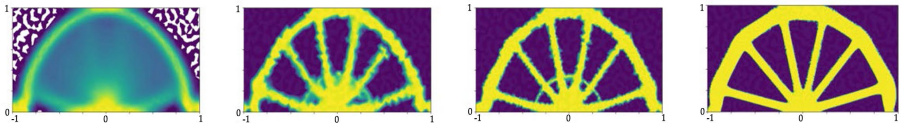


Figure A1: Iterations φ_{100},φ_{200},φ_{300},φ_{538} from Algorithm A1.

Algorithm A1 took 538 iterations to converge. Figure A2 illustrates other metrics in the optimisation. The convergence of $J^{\varepsilon}(\varphi)$ is clearly visible. Through the adaptation method of the step size τ_n (see [2]), it becomes increasingly larger when φ_n begins to converge towards the optimal mesh T_m. The small spikes in all time series are due to the refinement of the mesh T_m. In the lower-right corner, it can be seen that the calculation time increases with the fineness of the mesh T_m. The lower-left part of Figure A2 shows γ_n, which stabilises the form of φ_n, but plays no further role in our investigations.

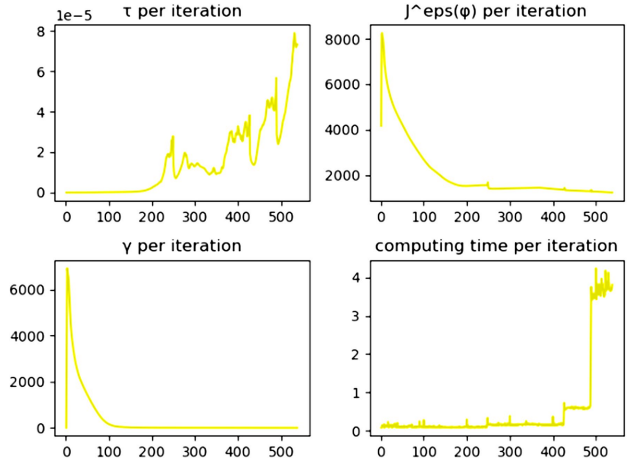


Figure A2: Metrics of the optimisation of Equation (**4**) using Algorithm A1.

Appendix A.2. Stochastic Bridge Problem

This modification of the experiment described in Appendix A.1 introduces uncertainties in the data, which render the problem much more involved. The Lamé–coefficients $\mu_{mat}=\lambda_{mat}$ are modelled as a truncated lognormal Karhunen–Loève expansion with 10 modes, a mean value of 150 and a

covariance length of 0.1 which is scaled by a factor of 100. The load g is assumed as a vector with mean (0,−5000) and a random rotation angle simulated through a truncated normal distribution with bounds $\left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$, standard deviation 0.3 and mean 0. In each iteration we use N=224 samples for the evaluation of the risk functional.

Some of the resulting iterations of the optimisation process are depicted in Figure A3. By calculating the expected value of the functional in Equation (7) (with parameter β=0), one can see a loss of symmetry in the resulting topology compared to the deterministic setting from Example 2 since the load is almost always not perpendicular to the load-bearing boundaries. The main difference is the strain on the Dirichlet boundary, which is introduced by the moved left-most spoke. In contrast to this, the right-hand side closely resembles the deterministic case since the slip boundary cannot absorb energy in the tangential direction. In this particular stochastic setting, an additional spoke is formed.

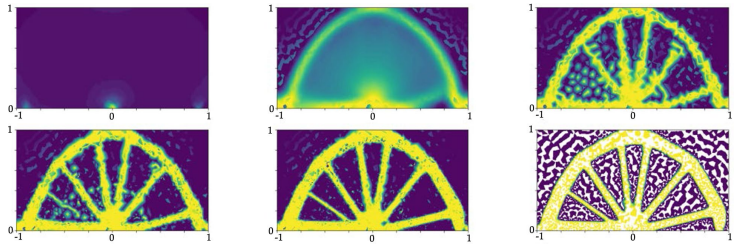


Figure A3: Iterations $\varphi_1, \varphi_{100}, \varphi_{200}, \varphi_{300}, \varphi_{400}, \varphi_{490}$ from Algorithm A2.

Algorithm A1: Deterministic optimisation algorithm from [2]

Input: mesh $\mathcal{T}_0$ and initial values φ_0
for $m = 0, 1, \ldots$ *until converged* **do**
  for $n = 0, 1, \ldots$ *until converged* **do**
    solve state equation on mesh $\mathcal{T}_m \Rightarrow u_{n+1}$
    solve adjoint equation on mesh $\mathcal{T}_m \Rightarrow p_{n+1}$
    solve gradient equation on mesh $\mathcal{T}_m \Rightarrow \varphi^*_{n+1}$
    project φ^*_{n+1} to $[0,1] \Rightarrow \varphi_{n+1}$
  end
  adapt mesh according to Appendix B $\Rightarrow \mathcal{T}_{m+1}$
end

Algorithm A2: Stochastic optimisation algorithm from [**2**]

Input: mesh $\mathcal{T}_0$ and initial values φ_0
for $m = 0, 1, \ldots$ *until converged* **do**
  for $n = 0, 1, \ldots$ *until converged* **do**
    for $i = 1, \ldots, N$ **do**
      sample $g(\omega_i), \lambda_{mat}(\omega_i), \mu_{mat}(\omega_i), \omega_i \in \Omega$
      solve state equation on mesh $\mathcal{T}_m \Rightarrow u_{n+1}(\omega_i)$
      solve adjoint equation on mesh $\mathcal{T}_m \Rightarrow p_{n+1}(\omega_i)$
      solve gradient equation on mesh $\mathcal{T}_m \Rightarrow \varphi^*_{n+1}(\omega_i)$
    end
    compute the mean $\hat{\varphi}_{n+1} = \frac{1}{N} \sum_{i=1}^{N} \varphi^*_{n+1}(\omega_i)$
    project $\hat{\varphi}_{n+1}$ to $[0, 1] \Rightarrow \varphi_{n+1}$
    adapt mesh according to Appendix B $\Rightarrow \mathcal{T}_{m+1}$
  end
end

APPENDIX B. FINITE ELEMENT DISCRETIZATION

The physical space $D \subset \mathbb{R}^2$ is discretised with first-order conforming finite elements with the FEniCS framework [24]. The reason for the favourable performance of the optimiser from [1,2] is the adaptive mesh refinement based on gradient information of the phase field. The idea is that the optimisation is started on a coarse mesh and refined over the course of the optimisation depending on the topology (more precisely on the phase transitions of φ). For this purpose it is assumed that the domain *D* is a convex polygon and is described with first-order conforming elements by the mesh $T_m=(V(T_m),E(T_m))$ at iteration step $m\in\mathbb{N}$, which also can be understood as a graph. The mesh consists of triangles $T\in T_m$ and an associated set of edges $E(T)\subset E(T_m)\subset D\times D$ and vertices $V(T)\subset V(T_m)\subset D$. Based on this mesh, the next mesh T_{m+1} is generated with a simple error indicator using the bulk criterion for the Dörfler marking [26] to determine which triangles *T* should be refined. Since φ moves within the domain *D*, the mesh refinement necessarily reflects this. So instead of simply refining the previous mesh, for every refinement we start with the initial mesh T_0, interpolate the current solution and displacement onto that mesh, refine according to the associated indicators and interpolate the current solutions φ_n and un onto the finer mesh. We repeat this process until a mesh T_{m+1} is obtained that is adequately finer than T_m. This refinement takes place whenever φ_n converges on the current mesh Tm. The refinement across an optimisation with Algorithm A1 is shown in Figure A4. It can be observed that the mesh is refined along the edges of φ. This dynamic characterisation (depending on φ and thus on the coefficients of the associated PDE) of the domain by the mesh Tm poses a

special challenge for the presented NN architectures when solving Equation (6) or (11), respectively. The relative change $e_n := \frac{\|\varphi_{n+1}-\varphi_n\|_{L^2(D)}}{\|\varphi_{n+1}\|_{L^2(D)}}$ in combination with the step size τ_n is used as convergence criterion, i.e.,

$$\frac{e_n}{\tau_n} < \varepsilon, \qquad \varepsilon > 0. \tag{A1}$$

This means that as soon as φ on a mesh Tm does not change significantly in relation to the step size τ_n, ϕ_n is considered converged. The refinement of the mesh is bounded by a chosen maximum of vertices $|V(T_m)| \leq b \in \mathbb{N}$.

We understand $\varphi_n \in \mathbb{R}^{|(\mathcal{T}_m)|}$ and $u_n \in \mathbb{R}^{d\times|(\mathcal{T}_m)|}$ as discretisation on T_m at iteration step $n \in \mathbb{N}$ in the sense that the value at every node $v_i \in V(T_m), i \leq |V(T_m)|$ is evaluated according to

$$u_n^i := u_n(v_i) \quad \text{or} \quad \varphi_n^i := \varphi_n(v_i)$$

in this node.

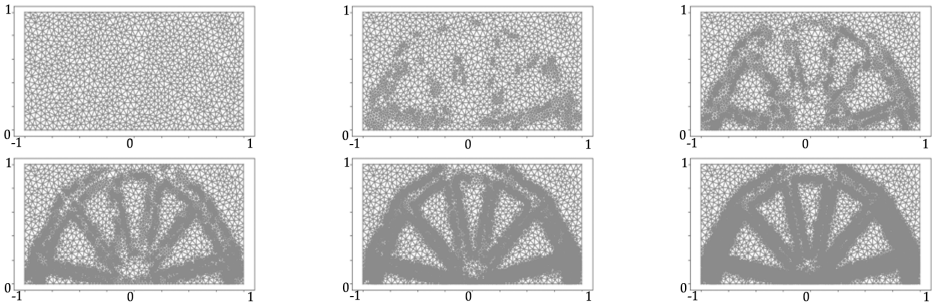


Figure A4: Iteration of adaptive mesh T_m from Appendix A.1.

APPENDIX C. ADDITIONAL EXPERIMENTS

Figure A5 and Figure A6 numerically illustrate less robust TCNN predictions for large gradient step sizes as mentioned in Section 4.1.2. In Figure A7, some iterations of a classical optimisation in comparison with the CNN assisted optimisation are depicted, showing basically identical results. However, it can also be observed that the distribution of the material converges faster when using the CNN. After 100 iterations, the first spokes for stabilising the arc can already be seen.

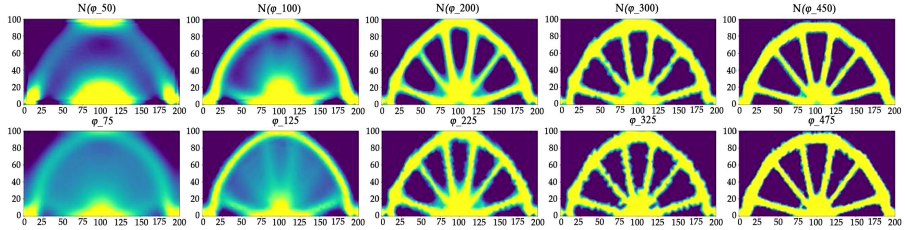


Figure A5: $\mathcal{N}_{\mathrm{CNN}}^{25}(\varphi_n, u_{n+1}; \theta)$ (top row) in comparison with φ_{n+25} (bottom row).

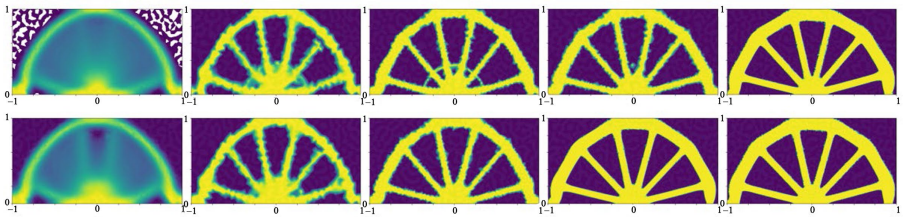


Figure A6: $\mathcal{N}_{\mathrm{CNN}}^{100}(\varphi_n, u_{n+1}; \theta)$ (top row) in comparison with φ_{n+100} (bottom row).

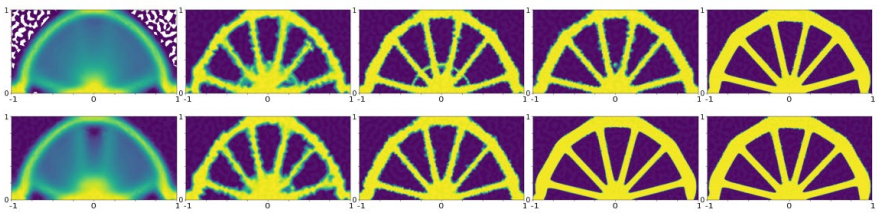


Figure A7: Optimisation without and with N_{CNN}. Sequence $\varphi_{100}, \varphi_{200}, \varphi_{300}, \varphi_{400}, \varphi_{538}$ from Algorithm A1 (top row). Sequence $\varphi_{100}, \varphi_{200}, \varphi_{300}, \varphi_{400}, \varphi_{441}$ from Algorithm 1 (bottom row).

REFERENCES

1. Eigel, M.; Neumann, J.; Schneider, R.; Wolf, S. Risk averse stochastic structural topology optimization. *Comput. Methods Appl. Mech. Eng.* 2018, *334*, 470–482.
2. Eigel, M.; Neumann, J.; Schneider, R.; Wolf, S. Stochastic topology optimisation with hierarchical tensor reconstruction. *WIAS* 2016, *2362*.
3. Rawat, S.; Shen, M.H. A Novel Topology Optimization Approach using Conditional Deep Learning. *arXiv* 2019, arXiv:1901.04859.
4. Cang, R.; Yao, H.; Ren, Y. One-Shot Optimal Topology Generation through Theory-Driven Machine Learning. *arXiv* 2018, arXiv:1807.10787.
5. Zhang, Y.; Chen, A.; Peng, B.; Zhou, X.; Wang, D. A deep Convolutional Neural Network for topology optimization with strong generalization ability. *arXiv* 2019, arXiv:1901.07761.
6. Sosnovik, I.; Oseledets, I. Neural networks for topology optimization. *Russ. J. Numer. Anal. Math. Model.* 2019, *34*, 215–223.
7. Wang, D.; Xiang, C.; Pan, Y.; Chen, A.; Zhou, X.; Zhang, Y. A deep convolutional neural network for topology optimization with perceptible generalization ability. *Eng. Optim.* 2022, *54*, 973–988.
8. White, D.A.; Arrighi, W.J.; Kudo, J.; Watts, S.E. Multiscale topology optimization using neural network surrogate models. *Comput. Methods Appl. Mech. Eng.* 2019, *346*, 1118–1135.
9. Dockhorn, T. A Discussion on Solving Partial Differential Equations using Neural Networks. *arXiv* 2019, arXiv:1904.07200.
10. Kallioras, N.A.; Lagaros, N.D. DL-Scale: Deep Learning for model upgrading in topology optimization. *Procedia Manuf.* 2020, *44*, 433–440.
11. Chandrasekhar, A.; Suresh, K. TOuNN: Topology optimization using neural networks. *Struct. Multidiscip. Optim.* 2021, *63*, 1135–1149.
12. Deng, H.; To, A.C. A parametric level set method for topology optimization based on deep neural network. *J. Mech. Des.* 2021, *143*, 091702.

13. Ates, G.C.; Gorguluarslan, R.M. Two-stage convolutional encoder-decoder network to improve the performance and reliability of deep learning models for topology optimization. *Struct. Multidiscip. Optim.* 2021, *63*, 1927–1950.
14. Malviya, M. A Systematic Study of Deep Generative Models for Rapid Topology Optimization. *engrXiv* 2020.
15. Abueidda, D.W.; Koric, S.; Sobh, N.A. Topology optimization of 2D structures with nonlinearities using deep learning. *Comput. Struct.* 2020, *237*, 106283.
16. Halle, A.; Campanile, L.F.; Hasse, A. An Artificial Intelligence–Assisted Design Method for Topology Optimization without Pre-Optimized Training Data. *Appl. Sci.* 2021, *11*, 9041.
17. Slaughter, W.; Petrolito, J. Linearized Theory of Elasticity. *Appl. Mech. Rev.* 2002, *55*, B90–B91.
18. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
19. Hochreiter, S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 1998, *6*, 107–116.
20. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* 1997, *9*, 1735–1780.
21. Ghaderpour, E.; Pagiatakis, S.D.; Hassan, Q.K. A survey on change detection and time series analysis with applications. *Appl. Sci.* 2021, *11*, 6141.
22. Graves, A. Generating Sequences With Recurrent Neural Networks. *arXiv* 2013, arXiv:1308.0850.
23. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; kin Wong, W.; chun Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Adv. Neural Inf. Process. Syst.* 2015, *28*, 802–810.

24. Alnæs, M.; Blechta, J.; Hake, J.; Johansson, A.; Kehlet, B.; Logg, A.; Richardson, C.; Ring, J.; Rognes, M.; Wells, G. The FEniCS Project Version 1.5. *Arch. Numer. Softw.* 2015, *3*, 9–23.
25. Naushad, R.; Kaur, T.; Ghaderpour, E. Deep transfer learning for land use and land cover classification: A comparative study. *Sensors* 2021, *21*, 8083.
26. Dörfler, W. A Convergent Adaptive Algorithm for Poisson's Equation. *SIAM J. Numer. Anal.* 1996, *33*, 1106–1124.

Chapter

11

A HYBRID ARITHMETIC OPTIMIZATION AND GOLDEN SINE ALGORITHM FOR SOLVING INDUSTRIAL ENGINEERING DESIGN PROBLEMS

Qingxin Liu [1],Ni Li [2,3],Heming Jia [4], Qi Qi [1], Laith Abualigah [5,6], and Yuxiang Liu [7]

[1]School of Computer Science and Technology, Hainan University, Haikou 570228, China
[2]School of Mathematics and Statistics, Hainan Normal University, Haikou 571158, China
[3] Key Laboratory of Data Science and Intelligence Education of Ministry of Education, Hainan Normal University, Haikou 571158, China
[4]School of Information Engineering, Sanming University, Sanming 365004, China
[5]Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan
[6]School of Computer Science, Universiti Sains Malaysia, Gelugor 11800, Malaysia
[7]College of Physics and Information Engineering, Fuzhou University, Fuzhou 350108, China

Citation: (APA): Eigel, M., Haase, M. & Neumann, J. (2022). Topology Optimisation under Uncertainties with Neural Networks. Algorithms 15(7):241. (30 pages).

ABSTRACT

Arithmetic Optimization Algorithm (AOA) is a physically inspired optimization algorithm that mimics arithmetic operators in mathematical calculation. Although the AOA has an acceptable exploration and exploitation ability, it also has some shortcomings such as low population diversity, premature convergence, and easy stagnation into local optimal solutions. The Golden Sine Algorithm (Gold-SA) has strong local searchability and fewer coefficients. To alleviate the above issues and improve the performance of AOA, in this paper, we present a hybrid AOA with Gold-SA called HAGSA for solving industrial engineering design problems. We divide the whole population into two subgroups and optimize them using AOA and Gold-SA during the searching process. By dividing these two subgroups, we can exchange and share profitable information and utilize their advantages to find a satisfactory global optimal solution. Furthermore, we used the Levy flight and proposed a new strategy called Brownian mutation to enhance the searchability of the hybrid algorithm. To evaluate the efficiency of the proposed work, HAGSA, we selected the CEC 2014 competition test suite as a benchmark function and compared HAGSA against other well-known algorithms. Moreover, five industrial engineering design problems were introduced to verify the ability of algorithms to solve real-world problems. The experimental results demonstrate that the proposed work HAGSA is significantly better than original AOA, Gold-SA, and other compared algorithms in terms of optimization accuracy and convergence speed.

Keywords: Meta-heuristics; arithmetic optimization algorithm; golden sine algorithm; hybrid optimization algorithm; industrial engineering design problem

INTRODUCTION

The main optimization process can be considered to obtain the best solution among all potential solutions according to the various NP-hard and engineering problems. Many real-world problems, such as image processing [1,2,3], engineering design [4,5,6,7,8], and job shop scheduling [9], can

be expressed as optimization problems and solved using optimization techniques. In the past two decades, the complexity of real-world optimization problems has increased sharply. However, the traditional (mathematical) methods cannot find the optimal solution or near-optimal solution in many cases [10]. Therefore, many researchers have turned their attention to meta-heuristic algorithms (MAs). Unlike traditional techniques, MAs are flexible and reliable in solving complex optimization problems [11].

Over the past few decades, various MAs had been proposed according to natural phenomena, physical principles, biological behaviors, etc. [12]. MAs can be separated into three main categories (as shown in Figure 1): (1) swarm intelligence-based methods, (2) physics-based methods, and (3) evolution-based methods. The first kind of method mimics the biological entities in nature that have collaboration behavior to finish hunting, migrating, etc. [13]. Developed algorithms in this category are Whale Optimization Algorithm (WOA) [14], Particle Swarm Optimization (PSO) [15], Grey Wolf Optimizer (GWO) [16], Salp Swarm Algorithm (SSA) [17], Ant Lion Optimization (ALO) [18], Moth Flame Optimization (MFO) [19], Slime Mould Algorithm (SMA) [20], Harris Hawks Optimization (HHO) [21], Reptile Search Algorithm (RSA) [22], and Aquila Optimizer (AO) [23]. The second type of method mainly simulates the physical phenomena of the universe and methods designed based on these laws are Multi-Verse Optimizer (MVO) [24], Sine Cosine Algorithm (SCA) [25], Arithmetic Optimization Algorithm (AOA) [26], Golden Sine Algorithm (Gold-SA) [27], Henry Gas Solubility Optimization (HGSO) [28], Gravity Search Algorithm (GSA) [29], Atom Search Optimization (ASO) [30], and Equilibrium Optimizer (EO) [31]. The evolution-based methods stem from the biological evolution process in nature. Some of the well-known algorithms developed by this behavior are Genetic Algorithm (GA) [32], Bio-geography-Based Optimizer (BBO) [33], Differential Evolution (DE) [34], and Evolution Strategy (ES) [35]. However, considering the No-Free-Lunch (NFL) theorem [36], no specific optimization algorithm can solve all real-world problems, which motivates us to design more efficient methods to solve them well.

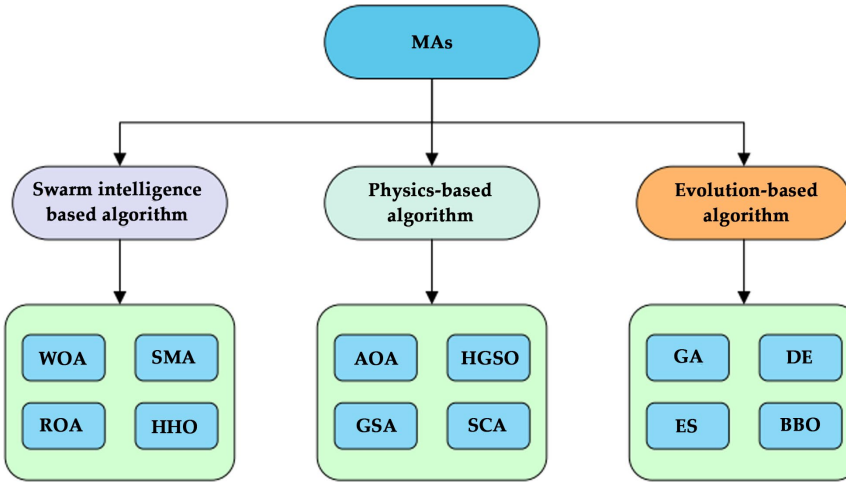


Figure 1. Classification of MAs.

The Arithmetic Optimization Algorithm (AOA) [26] is a physics-based and gradient-free method proposed by Abualigah et al. in 2021. It originated from the commonly used mathematical operators including Addition (+), Subtraction (−), Multiplication (×), and Division (÷). This approach integrates these four operators to realize different search mechanisms (exploration and exploitation) in the search space. Specifically, AOA uses the high distribution characteristics of (× and ÷) operators to realize the exploration approach. In the same way, the (+ and −) operators are used to obtain the high-dense results (exploitation approach). However, some researches denote that the original AOA has some defects, such as it easily suffering from a local optimal and slow convergence speed. Therefore, many variant versions of AOA were proposed to improve its searchability. For example, Azizi et al. [37] proposed an improved AOA based on Levy flight to determine the steel structure's optimal fuzzy controller parameters. Agushaka et al. [38] proposed an improved version of AOA called nAOA, which integrated the high-density values and beta distribution to enhance searchability. An Adaptive AOA, called APAOA, was proposed by Wang et al. [39]. In the APAOA, the parallel communication strategy was used to balance the exploration and exploitation ability of the original AOA. Another improved AOA that utilized a hybrid mechanism, named DAOA, was proposed by Abualigah et al. [40]. In DAOA, the differential evolution technique was integrated to enhance the local search ability of AOA, and to help it to jump out of the local optimal solution. Elkasem et al. [41] presented an eagle strategy AOA called ESAOA. In this work, the eagle strategy is

used to avoid premature convergence and increase the population's efficacy to obtain the optimal solution. Sharma et al. [42] introduced an opposition-based AOA namely OBAOA for identifying the parameters of PEMFCs. The opposition-based learning strategy is used to promote the algorithm to find the high-precision solution and improve the convergence rate. Abbassi et al. [43] developed an improved AOA to determine the solar cell parameters. In this work, the new operator called narrowed exploitation was used to narrow the search space and focus on the potential area to find the optimal or near-optimal solutions. Zhang et al. [44] proposed an improved AOA called IAO, which integrated the chaotic theory. The chaotic theory improves the algorithm to escape the optimal solution with a suitable convergence speed. Moreover, the IAO was used to optimize the weight of neural network.

Given the above discussion, some of the variants of AOA have strong searchability, but they cannot converge to the optimal solution at an appropriate time, i.e., they still easily fall into the local optimal solution. Furthermore, by considering the NFL theorem and increasingly complex real-world problems, the development of new and improved versions of MAs is ongoing. In general, a single optimizer also exposes some shortcomings; for example, it neglects to share useful information between populations, which may cause the algorithm to have insufficient search capability. Therefore, many researchers utilized the characteristic of two Mas, i.e., designing a hybrid algorithm to improve performance and applying it to solve complex real-world optimization problems. Unlike the single algorithm, the hybrid algorithm alleviates these shortcomings and increases diversity, and shares more helpful information within the population. Thus, the hybrid algorithm has more powerful searchability than the single algorithm. Gold-SA is a physics-based technique with a good exploitation ability to find the near-optimal solution. Furthermore, Gold-SA also has fewer parameters and is easy to program. Motivated by these considerations, in this paper, we propose an improved hybrid version of AOA called HAGSA that combines both AOA and Gold-SA. The proposed method uses Gold-SA to increase the population diversity and share more useful information between search agents. At the same time, Levy flight and a new strategy called Brownian mutation are used to enhance the exploration and exploitation capability of hybrid algorithms, respectively. To evaluate the effectiveness of the proposed method, we selected the CEC 2014 competition test suite as the benchmark function and compared the results with seven well-known methods, including AOA and Gold-SA. In addition, five classical engineering design problems,

including the car side crash design problem, pressure vessel design problem, tension spring design problem, speed reducer design problem, and cantilever beam design problem, were also used to evaluate HAGSA's ability to solve real-world problems. Experimental results demonstrate that the proposed work can provide complete results and achieve a faster convergence speed compared to other optimizers. The main contributions of this paper are as follows:

We propose a new hybrid algorithm based on the Arithmetic Optimization Algorithm and Golden Sine Algorithm (HAGSA).

- Levy flight and a new mechanism called Brownian mutation are carried out to enhance the exploration and exploitation ability of the hybrid algorithm.
- The performance of the proposed work is assessed on the CEC 2014 competition test suite and five classical engineering design problems.
- Several well-known MAs are compared with the proposed method.
- Experimental results indicate that HAGSA has more reliable performance than that of other well-known algorithms.

The remainder of this paper is structured as follows: Section 2 briefly illustrates the concepts of AOA and Gold-SA. Section 3 describes Levy flight, Brownian mutation, and the details of HAGSA. Section 4 presents and analyzes the experimental results of the proposed work. Finally, this paper's conclusion and potential research directions are discussed in Section 5.

PRELIMINARIES

This section introduces the inspiration and mathematical model of the original AOA and Gold-SA, in turn.

Arithmetic Optimization Algorithm (AOA)

The theory of AOA is described in this section. The main inspiration of AOA originates from the use of arithmetic operators such as Addition (A), Subtraction (S), Multiplication (M), and Division (D) to solve optimization problems [33]. In the following subsections, we discuss the different influences of these operators on optimization problems and the search method of AOA, as shown in Figure 2.

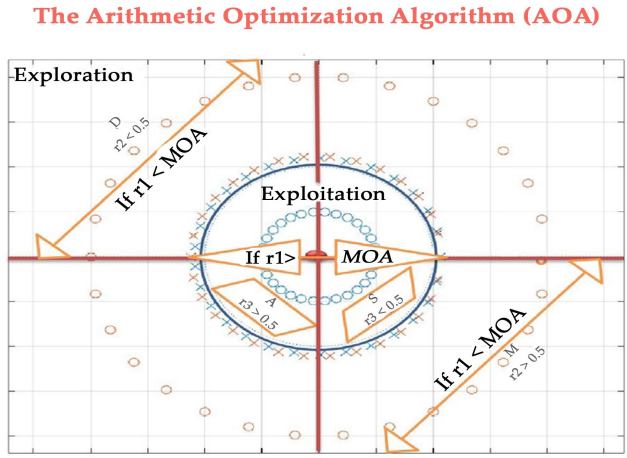


Figure 2. The different search phases of AOA.

Initialization Phase

Like other meta-heuristic optimization algorithms, AOA is based on population behavior. The set of a population X containing N search agents is illustrated in Equation (1). In the matrix, each row indicates a search agent [33].

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \tag{1}$$

After generating the population, the fitness of each search agent is computed, and the best one will be determined. Next, AOA decides to perform exploration or exploitation through the Math Optimizer Accelerated (MOA) value, which is defined as follows:

$$MOA(t) = Min + t \times \left(\frac{Max - Min}{T} \right) \tag{2}$$

where $MOA(t)$ indicates the value of MOA at the *t-th* iteration. Min and Max denote the minimum and maximum values of the accelerated function, respectively. t denotes the current iteration, and T denotes the maximum iteration. The search agent performs the

exploration phase when $r_1 > MOA$, otherwise the exploitation phase will be executed.

Exploration Phase

In this section, the exploration phase of AOA is described. According to the main inspiration, the Division (D) and Multiplication (M) operators are introduced to achieve high distributed values or decisions [33]. The Division and Multiplication operators can be mathematically described as follows:

$$X_{i,j}(t+1) = \begin{cases} X_{best,j}(t) \times MOP \times ((UB_j - LB_j) \times \mu + LB_j), & r_2 < 0.5 \\ X_{best,j}(t) \div (MOP + \varepsilon) \times ((UB_j - LB_j) \times \mu + LB_j), & \text{otherwise} \end{cases} \tag{3}$$

where $X_{i,j}(t+1)$ denotes the jth position of the ith solution in the next iteration. $X_{best,j}(t)$ denotes the best solution obtained so far in the jth position. LB_j and UB_j denote the lower and upper boundaries, respectively, of the search space at the jth dimension. ε is a small integer number, and r_2 denotes the random value between 0 and 1. $\mu = 0.5$, which represents the control function. Moreover, the Math Optimizer can be calculated as follows:

$$MOP(t) = 1 - \frac{t^{1/\alpha}}{T^{1/\alpha}} \tag{4}$$

where $\alpha = 0.5$ denotes the dynamic parameter, which determines the accuracy of the exploitation phase throughout iterations.

Exploitation Phase

In this section, we discuss the exploitation phase of AOA. In contrast to the D and M operator, AOA utilizes the Addition (A), and Subtraction (S) operators to derive high density solutions because (S and A) can easily approach the target region due to their low dispersion [33]. The mathematical formula can be described as follows:

$$X_{i,j}(t+1) = \begin{cases} X_{best,j}(t) - MOP \times ((UB_j - LB_j) \times \mu + LB_j), & r_3 < 0.5 \\ X_{best,j}(t) + MOP \times ((UB_j - LB_j) \times \mu + LB_j), & \text{otherwise} \end{cases} \tag{5}$$

where r_3 denotes a random value in the range 0 to 1.

The pseudo-code of AOA is illustrated in Algorithm 1.

Algorithm 1: pseudo-code of AOA [33]

Input: The parameter of AOA such as control function (μ), dynamic parameter (α), number of search agents (N), and maximum iteration (T)
Output: the best solution
Initialize the search agent randomly.
While ($t < T$) **do**
  Check if any search agent goes beyond the search space and amend it.
  Calculate fitness for the given search agent.
  Update the *MOA* and *MOP* using Equations (2) and (4), respectively.
  For $i = 1$ to N **do**
    For $j = 1$ to D **do**
      Update the random value r_1, r_2, r_3.
      If $r_1 > MOA$ **then**
        If $r_2 > 0.5$ **then**
          Update position by Division (÷) operator in Equation (3).
        Else
          Update position by Multiplication (×) operator in Equation (3).
        End if
      Else
        If $r_3 > 0.5$ **then**
          Update position by Addition (+) operator in Equation (5).
        Else
          Update position by Subtraction (−) operator in Equation (5).
        End if
      End if
    End for
  End for
  $t = t + 1$.
End while

Golden Sine Algorithm (Gold-SA)

This section introduces the basic theory of the Golden Sine Algorithm (Gold-SA). The inspiration of Gold-SA is a sine function in mathematics,

and the individuals explore the approximate optimal solution in the search space according to the golden ratio [27]. The range of the sine function is [−1, 1], with period 2π. When the value of x_1 changes, the corresponding variable y_1 also changes. Combining the sine function and golden ratio helps to continuously reduce the search space and search in regions where the optimal values are more likely to be generated, thereby improving the convergence speed [27]. The calculation formula is as follows:

$$X_{i,j}(t+1) = X_{i,j}(t) \times |\sin(p_1)| - p_2 \times \sin(p_1) \times \left| d_1 \times X_{best,j}(t) - d_2 \times X_{i,j}(t) \right| \quad (6)$$

where p_1 is the random value between [0, 2π], and p_2 is the random between [0, π], and d_1 and d_2 are the coefficient factors, which are obtained by the following equation:

$$d_1 = a \times \tau + b \times (1 - \tau) \quad (7)$$

$$d_2 = a \times (1 - \tau) + b \times \tau \quad (8)$$

where a and b are the initial values, which are −π and π. τ denotes the golden ratio, which is (5–√−1)/2. The pseudo-code of Gold-SA is shown in Algorithm 2.

Algorithm 2: pseudo-code of Gold-SA [27]

Input: The parameter of Gold-SA, such as the number of search agents (N), and maximum iteration (T).
Output: The best solution
Initialize the search agent randomly.
While ($t < T$) **do**
    Check if any search agent goes beyond the search space and amend it.
    Calculate fitness for the given search agent.
    For $i = 1$ to N **do**
        Update the random value p_1 and p_2, respectively.
        For $j = 1$ to D **do**
            Update position of search agent by the Equation (6).
        End for
    End for
    $t = t + 1$.
End while

THE PROPOSED ALGORITHM

In this section, we describe the proposed method. First, Levy flight is presented. Second, we propose a new strategy called Brownian mutation. Then, the details of the proposed work, HAGSA, are discussed and analyzed.

Levy Flight

Numerous studies reveal that the flight trajectories of many flying animals are consistent with characteristics typical of Levy flight. Levy flight is a class of non-Gaussian random walk that follows the Levy distribution [41,42]. It performs occasional long-distance walking with frequent short-distance steps, as shown in Figure 3. The mathematical formula for Levy flight is as follows:

$$Levy = 0.01 \times \frac{r_4 \times \sigma}{|r_5|^{\frac{1}{\beta}}} \tag{9}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{10}$$

where r_4 and r_5 are random values between [0, 1], and β is a constant equal to 1.5.

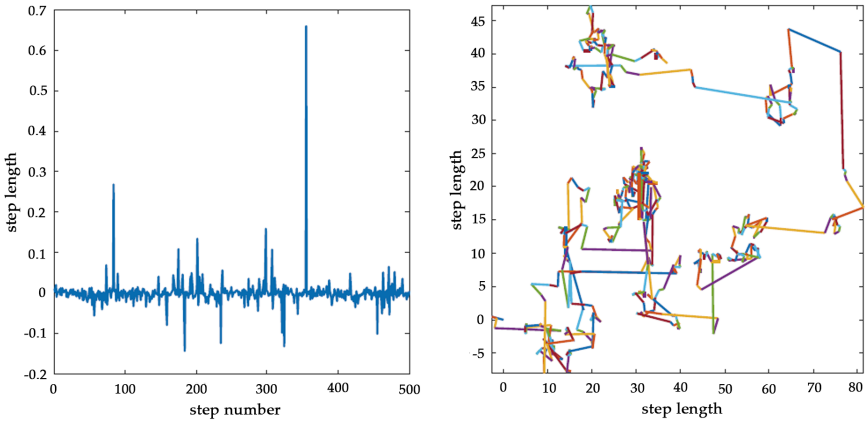


Figure 3: Levy distribution and 2D Levy trajectory.

Brownian Mutation

This paper proposes a Brownian mutation mechanism based on the mutation operator and Brownian motion. In 1995, differential evolution (DE) was proposed by Storn et al. [34], which was inspired by the mutation, crossover, and selection mechanisms in nature. Thus, DE obtains the optimal or near-optimal solution according to these operators. However, the crossover and mutation operators generate only one candidate solution in each iteration, limiting the population diversity and searchability of MAs [8]. Brownian motion (BM) is a stochastic process with a step size derived from a probability function defined by a normal distribution with $\mu = 0$ and $\sigma^2 = 1$ [43]. The formula of BM is listed as follows:

$$f_B(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \tag{11}$$

where x indicates a point following this motion, and the distribution and 2D trajectory of BM as shown in Figure 4. We can see that BM's trajectory can explore distant areas of the neighborhood, which shows more efficiency than a uniform random search in the search space. Therefore, considering the high performance of Brownian motion and the limitation of the mutation operator, we propose Brownian mutation, which generates two trail vectors with the Brownian motion strategy. This method generates two candidate solutions V_1 and V_2 of the i-th search agent in parallel through Equations (12) and (13), respectively.

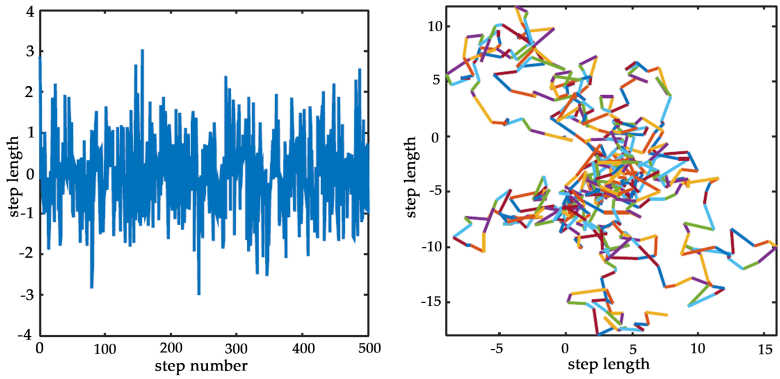


Figure 4: Brownian distribution and 2D Brownian trajectory.

The first mutation candidate solution V_1 is calculated as follows:

$$V_{1,j} = \begin{cases} X_{r6}(t) + Brownian \times (X_{r7}(t) - X_{r8}(t)), & \text{if } rand() < mr_1 \\ X_{i,j}, & \text{otherwise} \end{cases} \tag{12}$$

where r_6, r_7, and r_8 denote random values between 0 and 1. mr_1 is the mutation rate, and its value is 0.3. *Brownian* indicates the Brownian motion.

The second mutation candidate solution V_2 is calculated as follows:

$$V_{2,j} = \begin{cases} X_{best}(t) + Brownian \times (X_{r9}(t) - X_{r10}(t)), & \text{if } rand() < mr_2 \\ X_{i,j}, & \text{otherwise} \end{cases} \tag{13}$$

where r_9, r_{10}, and r_{11} denote random values between 0 and 1. mr_2 is the mutation rate equal to 0.5.

When two candidate solutions V_1 and V_2 are generated, they are first modified according to the lower and upper boundaries. Then, the best candidate solution V_{best} is selected using Equation (14) (lowest fitness as the criterion).

$$V_{best} = \begin{cases} V_1, \text{ if } f(V_1) < f(V_2) \\ V_2, \text{ otherwise} \end{cases} \tag{14}$$

Afterward, the best solution between V_{best} and X_i is selected as the ith search agent in the next iteration. The following equation describes this behavior:

$$X_i = \begin{cases} V_{best}, & \text{if } f(V_{best}) < f(X_i) \\ X_i, & \text{otherwise} \end{cases} \tag{15}$$

The Details of HAGSA

As mentioned above, single MAs have low diversity and cannot share useful information within the population. Moreover, the original AOA has shortcomings, such as easily stagnating into optimal local solutions and slow convergence speed. The Gold-SA has strong local searchability in the search space. Thus, to overcome the disadvantages of the original AOA and take full advantage of the benefits of Gold-SA, in this paper, we present a hybrid algorithm based on the AOA and Gold-SA, namely HAGSA. We divided the whole population into two subgroups, Group A and Group B, and optimized them using AOA and Gold-SA, respectively. Integrating both AOA and Gold-SA can increase population diversity and all the exchange

pf useful search information between search agents. This operation aims to enable search agents to find the valuable solution in the search space based on two MAs (AOA and Gold-SA) in less time and increase the diversity throughout the entire iterations. Furthermore, to enhance the searchability of the hybrid algorithm, it was integrated with Levy flight and Brownian mutation. Levy flight can improve the hybrid algorithm's exploration ability, allowing search agents to explore more potential regions in the search space. Thus, the improved exploration phase can be calculated by Equation (16). Furthermore, the Brownian mutation is used to strengthen the exploitation capability of the hybrid algorithm and help the individuals escape the local optimal solution.

$$X_{i,j}(t+1) = \begin{cases} X_{best,j}(t) \times Levy(j) \times MOP \times ((UB_j - LB_j) \times \mu + LB_j), & r_2 < 0.5 \\ X_{best,j}(t) \times Levy(j) \div (MOP + \varepsilon) \times ((UB_j - LB_j) \times \mu + LB_j), & \text{otherwise} \end{cases} \tag{16}$$

The pseudo-code of HAGSA is expressed in Algorithm 3, and the flowchart of the proposed work is shown in Figure 5.

Algorithm 3: pseudo-code of HAGSA

Input: The parameter such as control function (μ), dynamic parameter (α), number of search agent (N), and maximum iteration (T).
Output: best solution
Initialize the search agent randomly.
While ($t < T$) **do**
  Check if any search agent goes beyond the search space and amend it.
  Calculate fitness for the given search agent.
  Update the *MOA* and *MOP* using Equations (2) and (4), respectively.
  For $i = 1$ to N **do**
    For $j = 1$ to D **do**
      Update the random value r_1, r_2, r_3.
      If $i < N/2$ **then**
        If $r_1 > MOA$ **then**
          If $r_2 > 0.5$ **then**
            Update position by Division (÷) operator in Equation (16).
          Else
            Update position by Multiplication (×) operator in Equation (16).
          End if
        Else
          If $r_3 > 0.5$ **then**
            Update position by Addition (+) operator in Equation (5).
          Else
            Update position by Subtraction (−) operator in Equation (5).
          End if
        End if

Else
  Update position by Gold-SA operator in Equation (6).
End if
Generate candidate solution V_1 and V_2 by Equations (12) and (13).
Check if V_1 and V_2 goes beyond the search space and amend it.
Choose the best solution as V_{best} with the lower fitness from V_1 and V_2.
If $f(V_{best}) < f(X_i)$ **then**
  $X_i = V_i$.
  End if
End for
End for
$t = t + 1$.
End while

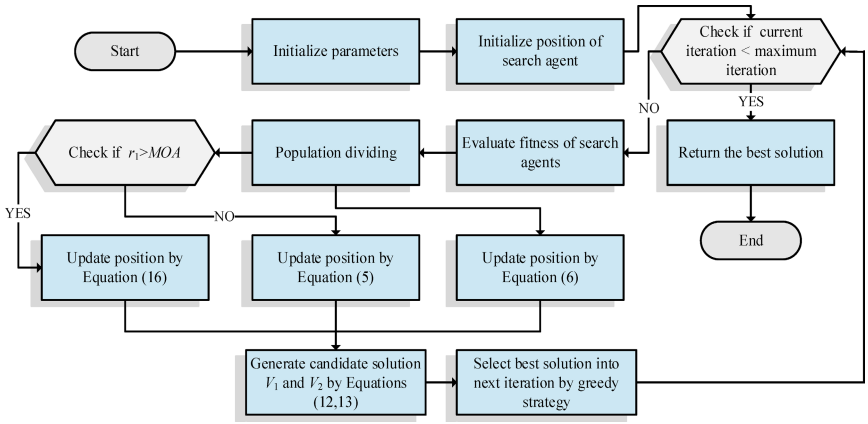


Figure 5: Flowchart of HAGSA.

Computational Complexity Analysis

In the initialization phase, HAGSA produces the search agents randomly in the search space, so the computational complexity of this phase is O($N \times D$), where N denotes the number of population and D denotes the dimension size. Afterward, HAGSA evaluates each individual's fitness during the whole iteration with the complexity O($T \times N \times D$), where T indicates the number

of iterations. Finally, we used AOA, Gold-SA, Levy flight, and Brownian mutation to obtain the best solution. Thus, the computational complexities of these phases are O($3 \times T \times N \times D$). In summary, the total computational complexity of HAGSA is O($T \times N \times D$).

EXPERIMENTAL RESULTS AND DISCUSSION

This section evaluates the effectiveness of the proposed HAGSA algorithm using the CEC 2014 competition test suite and five industrial engineering design problems. First, the benchmark functions and experimental setup are described. Next, the statistical results of the CEC 2014 benchmark functions are analyzed and discussed. Finally, the five industrial engineering design problems are used to prove the advantages of HAGSA.

Definition of CEC 2014 Benchmark Functions

To validate the searchability of the proposed HAGSA, we considered the CEC 2014 competition test suite as a benchmark function to evaluate the performance of HAGSA and its peers, which include 30 extremely complex functions [44]. The details of the benchmark functions are listed in Table 1, where f_{min} denotes the theoretical optimal fitness. According to their characteristics, the CEC 2014 test suite can be categorized into four classes. C01–C03 are unimodal functions with only one global optimum without any local optima, and are suitable for evaluating algorithms' exploitation capability. C04–C15 are multimodal functions with only one global optimal value with many local optimal values, and can evaluate algorithms' exploration and local minima avoidance ability. C16–C22 are hybrid functions, including both unimodal and multimodal functions, and can simultaneously examine the exploration and exploitation capability of algorithms. C23–C30 are composition functions that maintain continuity around the local and global optima. All these functions are rotated and shifted, so their complexity increases dramatically. Figure 6 provides a 2D visualization of some functions of the CEC 2014 test suite to understand its characteristics.

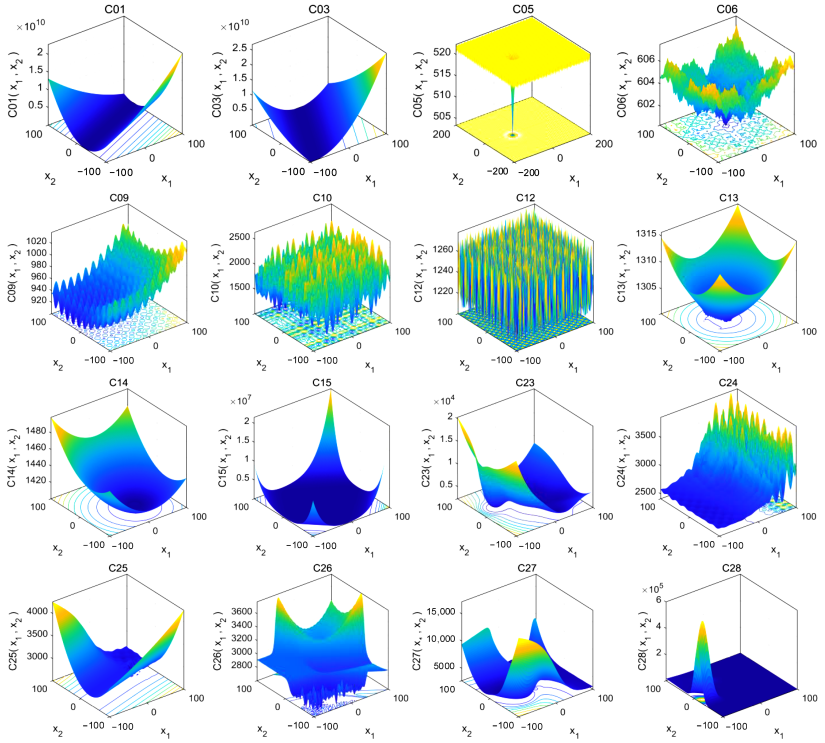


Figure 6: View of some CEC 2014 benchmark functions.

Table 1: CEC 2014 benchmark functions

Function Types	No.	Name of the Function	D	Range	f_{min}
Unimodal	C01	Rotated High Conditioned Elliptic Function	30	[−100, 100]	100
	C02	Rotated Bent Cigar Function	30	[−100, 100]	200
	C03	Rotated Discus Function	30	[−100, 100]	300
Multimodal	C04	Shifted and Rotated Rosenbrock Function	30	[−100, 100]	400
	C05	Shifted and Rotated Ackley Function	30	[−100, 100]	500
	C06	Shifted and Rotated Weierstrass Function	30	[−100, 100]	600
	C07	Shifted and Rotated Griewank Function	30	[−100, 100]	700
	C08	Shifted Rastrigin Function	30	[−100, 100]	800
	C09	Shifted and Rotated Rastrigin Function	30	[−100, 100]	900

	C10	Shifted Schwefel Function	30	[−100, 100]	1000
	C11	Shifted and Rotated Schwefel Function	30	[−100, 100]	1100
	C12	Shifted and Rotated Katsuura Function	30	[−100, 100]	1200
	C13	Shifted and Rotated HappyCat Function	30	[−100, 100]	1300
	C14	Shifted and Rotated HGBat Function	30	[−100, 100]	1400
	C15	Shifted and Rotated Expanded Griewank plus Rosenbrock Function	30	[−100, 100]	1500
Hybrid	C16	Shifted and Rotated Expanded Scaffer F6 Function	30	[−100, 100]	1600
	C17	Hybrid Function 1(N = 3)	30	[−100, 100]	1700
	C18	Hybrid Function 2(N = 3)	30	[−100, 100]	1800
	C19	Hybrid Function 3(N = 4)	30	[−100, 100]	1900
	C20	Hybrid Function 4(N = 4)	30	[−100, 100]	2000
	C21	Hybrid Function 5(N = 5)	30	[−100, 100]	2100
	C22	Hybrid Function 6(N = 5)	30	[−100, 100]	2200
Composition	C23	Composition Function 1(N = 5)	30	[−100, 100]	2300
	C24	Composition Function 2(N = 3)	30	[−100, 100]	2400
	C25	Composition Function 3(N = 3)	30	[−100, 100]	2500
	C26	Composition Function 4(N = 5)	30	[−100, 100]	2600
	C27	Composition Function 5(N = 5)	30	[−100, 100]	2700
	C28	Composition Function 6(N = 5)	30	[−100, 100]	2800
	C29	Composition Function 7(N = 3)	30	[−100, 100]	2900
	C30	Composition Function 8(N = 3)	30	[−100, 100]	3000

Experimental Setup

As stated above, the CEC 2014 test suite was utilized to evaluate HAGSA's optimization performance. To demonstrate the validity of the experimental results, the proposed algorithm HAGSA was compared with the basic AOA [26], Gold-SA [27], Remora Optimization Algorithm (ROA) [45], Aquila Optimizer (AO) [23], Sine Cosine Algorithm (SCA) [25], Whale Optimization Algorithm (WOA) [14], Flower Pollination Algorithm (FPA) [46], Differential Evolution (DE) [8], and Genetic Algorithm (GA) [47]. We set the maximum iteration $T = 500$, population size $N = 50$, dimension size $D = 30$, and 30 independent runs. The best results are highlighted in bold. All the experiments were conducted on a PC with an Intel (R) Core (TM) i5-11300H CPU @ 3.10 GHz, 16 GB RAM, Windows 10, and MATLAB

R2016b. Table 2 denotes the parameter setting of algorithms, and the details of the compared algorithms can be listed as follows:

- AOA: simulates four commonly used arithmetic operators as Division (÷), Multiplication (×), Subtraction (−), and Addition (+).
- Gold-SA: inspired by the sine function with the golden section search in mathematics compute.
- ROA: simulates remora's parasitism behavior on different hosts including whales and swordfish during the hunting process.
- AO: inspired by Aquila's four different hunting methods.
- SCA: simulates the distribution characteristics of sine and cosine functions.
- WOA: simulates the hunting behavior of humpback whales in oceans.
- FPA: simulates the pollination process of flowering plants in nature.
- DE: integrates the differential mutation, crossover, and selection mechanisms.
- GA: mimics the Darwinian evolution law and biological evolution of genetic mechanism in nature.

Table 2: Parameter setting of each algorithm

Algorithm	**Parameters**
AOA [**26**]	$\alpha = 5$; $\mu = 0.5$;
Gold-SA [**27**]	$c_1 = [1, 0]$; $c_2 \in [0, 1]$; $c_3 \in [0, 1]$
ROA [**45**]	$C = 0.1$
AO [**23**]	U = 0.00565; $r_1 = 10$; $\omega = 0.005$; $\alpha = 0.1$; $\delta = 0.1$;
SCA [**25**]	$a \in [2, 0]$
WOA [**14**]	$a_1 \in [2, 0]$; $a_2 \in [-1, -2]$; $b = 1$
FPA [**46**]	$p = 0.8$; $\beta = 1.5$
DE [**8**]	$F_{min} = 0.2$; $F_{max} = 0.8$; $CR = 0.1$
GA [**47**]	$Pc = 0.85$; $Pm = 0.01$

Statistical Results on CEC 2014 Benchmark Functions

Table 3 denotes the mean and standard deviation (std) values obtained by HAGSA and other competed algorithms for each CEC 2014 function with D

= 30. According to Table 3, the statistical results illustrate that the HAGSA provides better searchability than its peers. For unimodal functions, HAGSA better obtains the global optimal solution on C01 and C03 than others. For multimodal functions, HAGSA outperforms all other well-known algorithms on nine functions, except functions C07–08, C11, and C14; FPA, DE, ROA, and AO find the global optimal solution for these functions, respectively. For hybrid functions, HAGSA achieves the best results for C16, C19, C20, and C22 among all algorithms. Finally, HAGSA also outperforms the results for composition functions compared to the original AOA, Gold-SA, and other compared algorithms on C23–25 and C28–C30, but not on C26. Figure 7 shows HAGSA and competitor algorithms' ranking in various functions of the CEC 2014 test suite. In light of these results, HAGSA exhibits excellent performance by obtaining the best average over 21 functions.

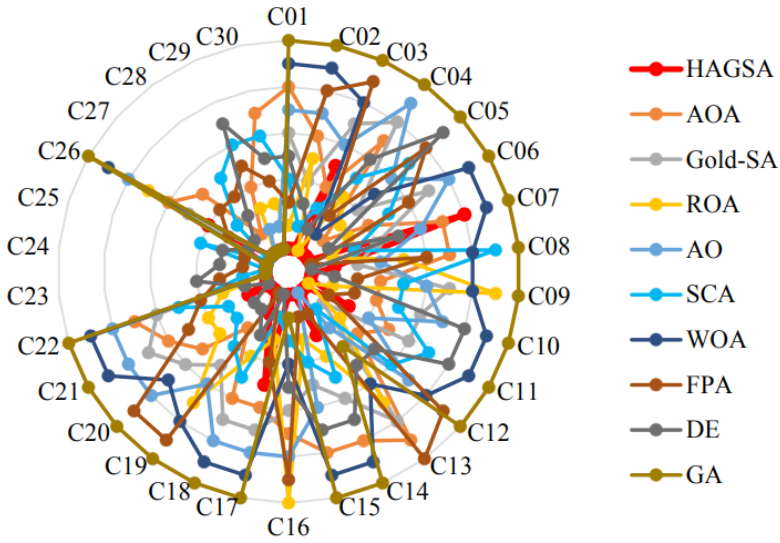


Figure 7: The radar graphs of algorithms on CEC 2014 benchmark functions.

Table 3: The mean fitness and std obtained with the different algorithms on the CEC 2014 test suite

Function		HAGSA	AOA	Gold-SA	ROA	AO	SCA	WOA	FPA	DE	GA
C01	Mean	**1.94×10^{8}**	1.08×10^{9}	6.73×10^{8}	3.59×10^{8}	7.85×10^{8}	5.11×10^{8}	1.97×10^{9}	4.63×10^{8}	5.30×10^{9}	2.77×10^{9}
	Std	7.50×10^{7}	3.49×10^{8}	2.21×10^{8}	1.63×10^{8}	3.92×10^{7}	1.26×10^{8}	3.25×10^{8}	1.97×10^{8}	2.23×10^{8}	1.07×10^{8}
C02	Mean	**2.40×10^{10}**	6.81×10^{10}	6.18×10^{10}	6.80×10^{10}	6.83×10^{10}	2.93×10^{10}	8.59×10^{10}	6.99×10^{10}	5.09×10^{10}	1.03×10^{11}
	Std	7.78×10^{9}	1.18×10^{10}	9.47×10^{9}	7.53×10^{9}	1.27×10^{9}	5.26×10^{9}	7.45×10^{9}	2.39×10^{9}	1.02×10^{10}	0.00
C03	Mean	8.55×10^{4}	8.19×10^{4}	8.73×10^{4}	**6.60×10^{4}**	8.72×10^{4}	7.58×10^{4}	9.20×10^{4}	1.26×10^{5}	7.01×10^{4}	1.42×10^{7}
	Std	2.10×10^{3}	6.52×10^{3}	2.50×10^{3}	7.55×10^{3}	7.66×10^{3}	1.61×10^{4}	1.22×10^{4}	6.25×10^{4}	1.56×10^{4}	1.25×10^{4}
C04	Mean	**1.45×10^{4}**	1.05×10^{4}	1.27×10^{4}	2.54×10^{3}	1.40×10^{4}	2.57×10^{3}	1.73×10^{4}	1.74×10^{3}	6.37×10^{3}	2.58×10^{4}
	Std	7.37×10^{2}	2.84×10^{3}	3.47×10^{3}	1.17×10^{3}	1.95×10^{2}	6.06×10^{2}	2.18×10^{3}	3.95×10^{2}	2.59×10^{3}	5.19×10^{2}
C05	Mean	**5.20×10^{2}**	5.21×10^{2}	5.21×10^{2}	5.21×10^{2}	5.21×10^{2}	5.21×10^{2}	5.21×10^{2}	5.21×10^{2}	5.21×10^{2}	5.21×10^{2}
	Std	8.39×10^{2}	8.06×10^{2}	7.03×10^{2}	1.07×10^{-1}	8.92×10^{-2}	7.53×10^{-2}	8.15×10^{-2}	8.17×10^{-2}	6.61×10^{-2}	8.05×10^{-2}
C06	Mean	**6.17×10^{2}**	6.38×10^{2}	6.42×10^{2}	6.35×10^{2}	6.42×10^{2}	6.39×10^{2}	6.45×10^{2}	6.39×10^{2}	6.34×10^{2}	6.50×10^{2}
	Std	3.56	2.45	2.42	3.07	2.76	1.97	1.43	2.82	2.63	2.05
C07	Mean	1.47×10^{3}	1.34×10^{3}	1.13×10^{3}	9.16×10^{2}	1.19×10^{3}	9.50×10^{2}	1.56×10^{3}	**7.41×10^{2}**	1.16×10^{3}	1.75×10^{3}
	Std	7.04×10	1.06×10^{2}	9.78×10	9.09×10	1.24×10	$3. \times 10$	6.79×10	1.56×10	1.12×10^{2}	7.10×10
C08	Mean	1.09×10^{3}	1.14×10^{3}	1.12×10^{3}	1.13×10^{3}	1.13×10^{3}	1.19×10^{3}	1.18×10^{3}	1.13×10^{3}	**1.08×10^{3}**	1.31×10^{3}
	Std	2.42×10	3.04×10	3.10×10	2.57×10	2.02×10	2.22×10	1.37×10	4.63×10	2.32×10	2.23×10
C09	Mean	**1.14×10^{3}**	1.22×10^{3}	1.26×10^{3}	1.37×10^{3}	1.26×10^{3}	1.22×10^{3}	1.29×10^{3}	1.20×10^{3}	1.20×10^{3}	1.38×10^{3}
	Std	1.65×10	2.17×10	2.71×10	2.23×10	1.89×10	2.49×10	1.67×10	5.07×10	2.70×10	2.31×10^{-13}
C10	Mean	**6.12×10^{3}**	7.26×10^{3}	8.04×10^{3}	6.34×10^{3}	8.16×10^{3}	7.97×10^{3}	9.45×10^{3}	6.57×10^{3}	8.93×10^{3}	1.07×10^{4}
	Std	6.25×10^{2}	3.79×10^{2}	5.47×10^{2}	7.11×10^{2}	5.84×10^{2}	4.49×10^{2}	3.61×10^{2}	7.50×10^{2}	2.87×10^{2}	5.36×10^{2}
C11	Mean	7.56×10^{3}	7.85×10^{3}	8.90×10^{3}	**7.28×10^{3}**	7.81×10^{3}	8.96×10^{3}	1.01×10^{4}	7.47×10^{3}	9.31×10^{3}	1.10×10^{4}
	Std	7.10×10^{2}	4.20×10^{2}	5.68×10^{2}	6.88×10^{2}	6.68×10^{2}	2.55×10^{2}	3.79×10^{2}	7.89×10^{2}	4.51×10^{2}	4.69×10^{2}
C12	Mean	**1.20×10^{3}**	1.20×10^{3}	1.20×10^{3}	1.20×10^{3}	1.20×10^{3}	1.20×10^{3}	1.20×10^{3}	1.20×10^{3}	1.20×10^{3}	1.21×10^{3}
	Std	5.52×10	5.78×10^{-1}	5.35×10^{-1}	5.62×10^{-1}	5.98×10^{-1}	5.89×10^{-1}	6.48×10^{-1}	6.75×10^{-1}	5.80×10^{-1}	9.19×10^{-1}
C13	Mean	**1.30×10^{3}**	1.31×10^{3}	1.31×10^{3}	1.31×10^{3}	1.31×10^{3}	1.31×10^{3}	1.31×10^{3}	1.31×10^{3}	1.31×10^{3}	1.31×10^{3}
	Std	8.34×10^{-1}	9.07×10^{-1}	8.99×10^{-1}	8.64×10^{-1}	4.09×10^{-1}	3.93×10^{-1}	8.37×10^{-1}	9.22×10^{-1}	7.67×10^{-1}	4.48×10^{-1}
C14	Mean	1.45×10^{3}	1.63×10^{3}	1.57×10^{3}	1.47×10^{3}	**1.41×10^{3}**	1.49×10^{3}	1.73×10^{3}	1.42×10^{3}	1.59×10^{3}	1.79×10^{3}
	Std	1.44×10	4.41×10	4.36×10	2.20×10	5.87×10	1.99×10	2.46×10	9.00	3.95×10	3.58×10
C15	Mean	**4.34×10^{3}**	2.50×10^{5}	4.92×10^{4}	9.04×10^{3}	8.92×10^{4}	2.54×10^{4}	5.38×10^{5}	4.74×10^{3}	1.03×10^{5}	9.16×10^{5}
	Std	2.25×10^{3}	1.31×10^{5}	3.55×10^{4}	8.15×10^{3}	3.54×10	1.62×10^{4}	1.55×10^{5}	2.24×10^{3}	1.35×10^{5}	4.74×10^{-10}
C16	Mean	**1.61×10^{3}**	1.61×10^{3}	1.61×10^{3}	1.61×10^{3}	1.61×10^{3}	1.61×10^{3}	1.61×10^{3}	1.61×10^{3}	1.61×10^{3}	1.61×10^{3}
	Std	3.71×10^{-1}	3.70×10^{-1}	3.21×10^{-1}	4.71×10^{-1}	4.08×10^{-1}	1.95×10^{-1}	2.08×10^{-1}	4.66×10^{-1}	2.37×10^{-1}	1.88×10^{-1}
C17	Mean	8.59×10^{7}	8.90×10^{7}	1.36×10^{8}	1.61×10^{7}	1.64×10^{8}	1.56×10^{7}	2.47×10^{8}	2.35×10^{7}	**9.17×10^{6}**	5.48×10^{8}
	Std	7.10×10^{6}	6.17×10^{7}	8.13×10^{7}	1.40×10^{7}	5.24×10^{6}	5.76×10^{6}	6.66×10^{7}	2.13×10^{7}	1.30×10^{7}	2.33×10^{8}
C18	Mean	1.36×10^{7}	2.44×10^{9}	2.78×10^{9}	2.90×10^{8}	3.80×10^{9}	4.77×10^{8}	7.52×10^{9}	**6.11×10^{6}**	2.65×10^{8}	1.20×10^{10}
	Std	2.06×10^{7}	2.04×10^{9}	1.67×10^{9}	6.85×10^{8}	2.05×10^{6}	2.52×10^{8}	2.30×10^{9}	7.19×10^{6}	3.73×10^{8}	3.82×10^{9}
C19	Mean	**2.01×10^{3}**	2.24×10^{3}	2.27×10^{3}	2.30×10^{3}	2.30×10^{3}	2.25×10^{3}	2.49×10^{3}	2.32×10^{3}	2.10×10^{3}	2.80×10^{3}
	Std	5.12×10	1.05×10^{2}	9.98×10^{1}	9.65×10	3.15×10	3.29×10	7.58×10	5.24×10	6.27×10	2.51×10
C20	Mean	**3.67×10^{4}**	1.86×10^{5}	2.45×10^{5}	9.06×10^{4}	4.34×10^{5}	5.90×10^{4}	3.43×10^{6}	4.97×10^{5}	2.75×10^{4}	1.07×10^{8}
	Std	3.96×10^{4}	9.23×10^{4}	1.27×10^{5}	6.04×10^{4}	5.11×10^{4}	2.94×10^{4}	4.51×10^{6}	7.78×10^{5}	2.08×10^{4}	2.87×10^{7}

C21	Mean	1.12×10^{6}	3.36×10^{7}	5.47×10^{7}	9.65×10^{6}	5.65×10^{7}	5.18×10^{6}	1.07×10^{8}	1.25×10^{7}	**5.17×10^{5}**	2.80×10^{8}
	Std	6.21×10^{5}	2.39×10^{7}	2.92×10^{7}	9.83×10^{6}	1.66×10^{6}	2.86×10^{6}	5.82×10^{7}	9.65×10^{6}	7.13×10^{5}	2.14×10^{8}
C22	Mean	**2.85×10^{3}**	4.93×10^{3}	4.69×10^{3}	3.28×10^{3}	6.49×10^{3}	3.37×10^{3}	3.08×10^{4}	3.32×10^{3}	3.08×10^{3}	1.68×10^{5}
	Std	2.12×10^{2}	2.12×10^{7}	1.78×10^{3}	7.41×10^{2}	2.78×10^{2}	1.72×10^{2}	3.05×10^{4}	2.89×10^{2}	2.52×10^{2}	7.01×10^{4}
C23	Mean	**2.50×10^{3}**	**2.50×10^{3}**	**2.50×10^{3}**	**2.50×10^{3}**	**2.50×10^{3}**	2.72×10^{3}	**2.50×10^{3}**	2.72×10^{3}	2.84×10^{3}	**2.50×10^{3}**
	Std	0.00	0.00	0.00	0.00	0.00	3.17×10	0.00	4.01×10	9.01×10	0.00
C24	Mean	**2.60×10^{3}**	**2.60×10^{3}**	**2.60×10^{3}**	**2.60×10^{3}**	**2.60×10^{3}**	**2.63×10^{3}**	**2.60×10^{3}**	2.61×10^{3}	2.69×10^{3}	**2.60×10^{3}**
	Std	0.00	8.87×10^{-2}	0.00	1.46×10^{-7}	2.34×10^{-5}	1.87×10	0.00	5.81	1.34×10	0.00
C25	Mean	**2.70×10^{3}**	**2.70×10^{3}**	**2.70×10^{3}**	**2.70×10^{3}**	**2.70×10^{3}**	2.75×10^{3}	**2.70×10^{3}**	2.72×10^{3}	2.73×10^{3}	**2.70×10^{3}**
	Std	0.00	0.00	0.00	0.00	0.00	1.15×10	0.00	1.83×10	8.74	0.00
C26	Mean	2.77×10^{3}	2.77×10^{3}	2.77×10^{3}	2.77×10^{3}	2.78×10^{3}	**2.70×10^{3}**	2.79×10^{3}	2.74×10^{3}	2.73×10^{3}	2.79×10^{3}
	Std	4.33×10	4.41×10	4.25×10	4.62×10	4.99×10	4.96×10^{-1}	2.35×10^{1}	8.00×10	4.33×10	2.39×10
C27	Mean	**2.90×10^{3}**	4.05×10^{3}	**2.90×10^{3}**	**2.90×10^{3}**	**2.90×10^{3}**	3.91×10^{3}	**2.90×10^{3}**	3.99×10	3.86×10^{3}	**2.90×10^{3}**
	Std	0.00	3.71×10^{2}	0.00	0.00	3.98	2.65×10^{2}	0.00	2.42×10^{2}	2.46×10^{2}	0.00
C28	Mean	**3.00×10^{3}**	5.34×10^{3}	**3.00×10^{3}**	**3.00×10^{3}**	**3.00×10^{3}**	5.95×10^{3}	**3.00×10^{3}**	5.40×10^{3}	5.36×10^{3}	**3.00×10^{3}**
	Std	0.00	2.75×10^{3}	0.00	0.00	0.00	6.11×10^{2}	0.00	8.95×10^{2}	4.64×10^{2}	0.00
C29	Mean	**3.10×10^{3}**	4.32×10^{8}	**3.10×10^{3}**	7.17×10^{6}	1.46×10^{4}	4.43×10^{7}	**3.10×10^{3}**	1.79×10^{7}	6.87×10^{7}	**3.10×10^{3}**
	Std	0.00	1.80×10^{8}	0.00	7.00×10^{6}	6.28×10^{4}	1.75×10^{7}	0.00	1.63×10^{7}	5.50×10^{7}	0.00
C30	Mean	**3.20×10^{3}**	4.16×10^{6}	**3.20×10^{3}**	3.29×10^{5}	1.66×10^{5}	6.97×10^{5}	**3.20×10^{3}**	4.02×10^{5}	4.17×10^{5}	**3.20×10^{3}**
	Std	0.00	2.65×10^{6}	0.00	2.80×10^{5}	1.44×10^{5}	2.87×10^{5}	0.00	2.76×10^{5}	2.34×10^{5}	0.00

Boxplot Behavior Analysis

The distribution characteristics of data can be displayed through boxplot analysis. The boxplot describes the data distribution as quartiles. The lowest and largest points of the edges of the boxplot are the minimum and maximum values obtained by the algorithm. The lower and upper quartiles are separated by the endpoints of the rectangle [**5**]. In this subsection, we use boxplot behavior to represent each algorithm's distribution of the obtained value. Each sample runs 30 times independently for each CEC 2014 benchmark function with D = 30. The boxplot behavior of each algorithm is shown in **Figure 8**. HAGSA has better stability for most benchmark functions and shows excellent performance compared to the others. In particular, for C01, C04, C05, C08, C09, C12, C13, and C15, the boxplot of the proposed HAGSA method is very narrow compared to others and shows lower values. For C06, C14, and C16, HAGSA achieves the lower values obtained than most algorithms. However, the performance is not obvious when solving C10, C17, C18, C19, C21, C23, C25, C27, and C30.

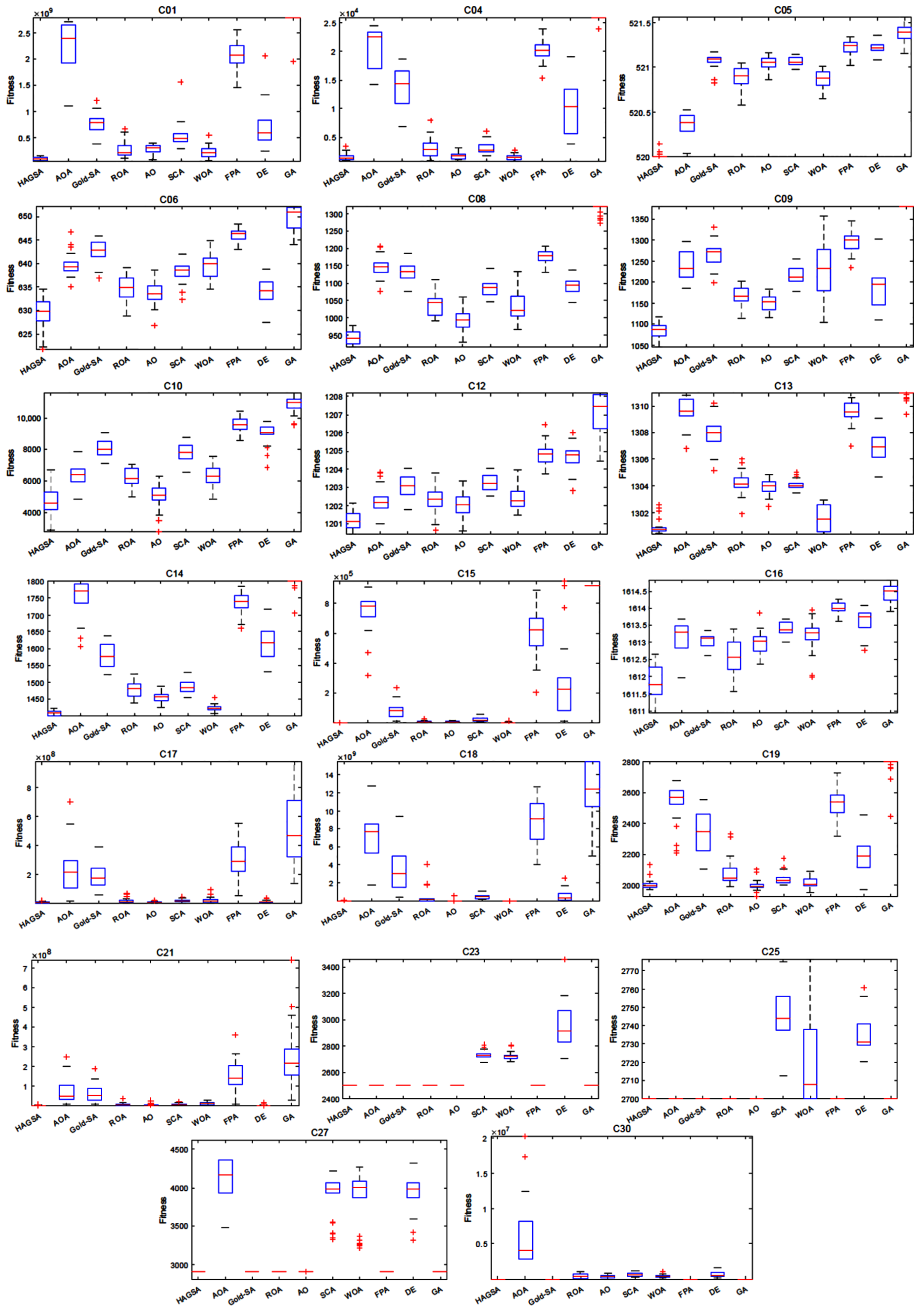
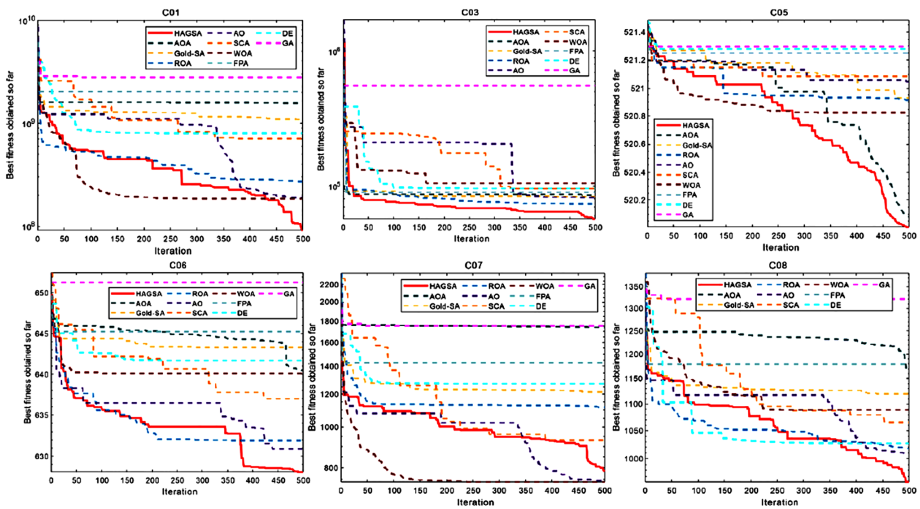


Figure 8: Boxplot behavior of algorithms on some functions.

Convergence Behavior Analysis

In this subsection, we analyze the convergence behavior of each algorithm used over some benchmark functions. Figure 9 shows the convergence behavior of HAGSA, AOA, Gold-SA, ROA, AO, SCA, WOA, FPA, DE, and GA for selected functions. As can be seen from this figure, HAGSA achieves excellent behavior for most functions, which suggests the convergence of the proposed method. For unimodal functions (C01 and C03), although the convergence speed is slower than WOA in the early iteration (for C01), the convergence accuracy is higher than WOA at the end of the iteration. For C03, HAGSA has the fastest convergence speed and highest convergence accuracy. On the multimodal functions, HAGSA still maintains the fastest convergence speed and highest accuracy on most functions. In particular, for C05 and C06, although the global optimal is not found, HAGSA still has excellent performance compared to the others. However, the optimal value of HAGSA is ranked third and the WOA and AO are ranked first and second, respectively, when solving C07. For C10 and C11, it can be seen that the convergence curve of HAGSA is accelerated in the later stage of iteration; this is due to the excellent ability to jump out of the local optimal as a result of Brownian mutation. On hybrid functions, the convergence accuracy is still good compared to the others. For C16, C20, C21, and C22, the proposed HAGSA algorithm demonstrates its better performance compared to the original AOA and Gold-SA. On composition functions, the improvement is not obvious compared to the original Gold-SA and other well-known algorithms such as GA and FPA.



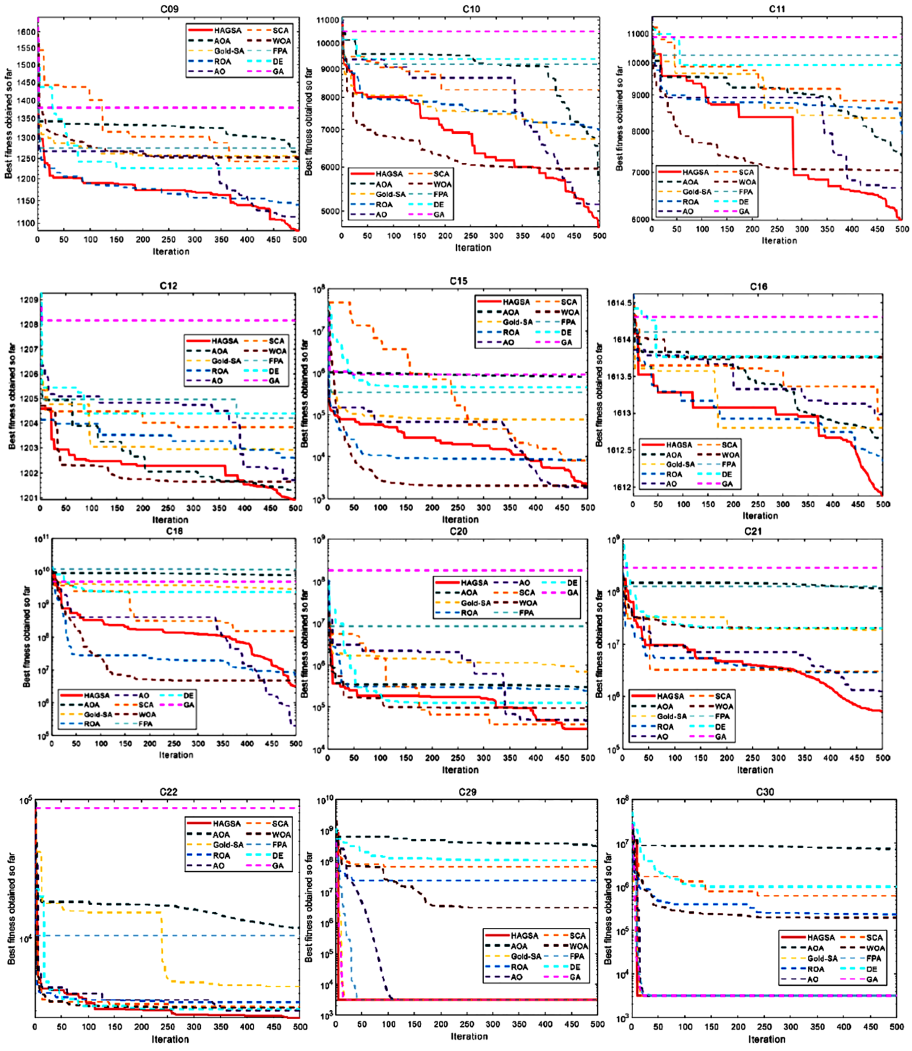


Figure 9: Convergence curve of algorithms on some functions.

Wilcoxon Rank-Sum Test

Because the results obtained by each algorithm are random, in this subsection, we utilize the Wilcoxon rank-sum test (WRS) to evaluate the statistical significance difference between two samples at a significance level of 5% [**2**]. Specifically, if the *p*-value is less than 0.05, it indicates the statistical difference is significant; otherwise, the difference is not obvious.

Furthermore, NaN denotes there is no difference between the two samples. The statistical results of the Wilcoxon rank-sum test are listed in Table 4; from this table, we can see that the proposed HAGSA algorithm shows better significant performance than the other algorithms on most benchmark functions.

Table 4: Statistical results of Wilcoxon rank-sum test obtained by each algorithm

Function	HAGSA vs.								
	AOA	Gold-SA	ROA	AO	SCA	WOA	FPA	DE	GA
C01	3.02×10^{-11}	3.02×10^{-11}	2.71×10^{-2}	2.13×10^{-4}	4.08×10^{-11}	2.64×10^{-1}	3.02×10^{-11}	1.29×10^{-9}	2.37×10^{-12}
C02	3.02×10^{-11}	3.02×10^{-11}	7.01×10^{-2}	3.02×10^{-11}	5.83×10^{-13}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	1.21×10^{-12}
C03	3.02×10^{-11}	3.02×10^{-11}	3.32×10^{-6}	2.49×10^{-6}	2.00×10^{-5}	6.52×10^{-9}	3.02×10^{-11}	3.82×10^{-10}	3.02×10^{-11}
C04	3.02×10^{-11}	3.02×10^{-11}	2.06×10^{-2}	2.61×10^{-10}	4.71×10^{-4}	1.31×10^{-8}	3.02×10^{-11}	3.69×10^{-11}	1.21×10^{-12}
C05	1.78×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
C06	4.62×10^{-10}	3.02×10^{-11}	2.75×10^{-3}	7.73×10^{-2}	2.23×10^{-9}	1.29×10^{-11}	3.02×10^{-11}	1.30×10^{-1}	2.95×10^{-11}
C07	3.02×10^{-11}	6.07×10^{-11}	1.45×10^{-1}	3.02×10^{-11}	2.13×10^{-5}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	3.16×10^{-12}
C08	3.02×10^{-11}	3.02×10^{-11}	1.73×10^{-6}	1.25×10^{-7}	3.02×10^{-11}	2.84×10^{-4}	3.02×10^{-11}	5.49×10^{-11}	9.40×10^{-12}
C09	3.02×10^{-11}	3.02×10^{-11}	1.44×10^{-2}	7.70×10^{-8}	3.34×10^{-11}	4.42×10^{-6}	3.02×10^{-11}	6.12×10^{-10}	1.21×10^{-12}
C10	2.23×10^{-9}	3.02×10^{-11}	5.09×10^{-8}	2.97×10^{-1}	3.02×10^{-11}	1.46×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
C11	3.50×10^{-9}	3.02×10^{-11}	1.37×10^{-3}	5.01×10^{-1}	3.34×10^{-11}	2.96×10^{-5}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
C12	6.91×10^{-4}	2.39×10^{-8}	1.76×10^{-2}	2.90×10^{-1}	1.69×10^{-9}	5.27×10^{-5}	4.50×10^{-11}	3.02×10^{-11}	2.80×10^{-11}
C13	3.02×10^{-11}	3.69×10^{-11}	1.38×10^{-2}	6.07×10^{-11}	1.68×10^{-3}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}	7.88×10^{-12}
C14	3.02×10^{-11}	3.02×10^{-11}	4.22×10^{-4}	1.33×10^{-10}	1.39×10^{-6}	1.09×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	1.72×10^{-12}
C15	3.02×10^{-11}	3.02×10^{-11}	5.49×10^{-1}	3.02×10^{-11}	1.69×10^{-9}	2.37×10^{-10}	3.02×10^{-11}	3.02×10^{-11}	1.21×10^{-12}
C16	1.56×10^{-8}	4.18×10^{-9}	1.64×10^{-5}	2.23×10^{-9}	4.50×10^{-11}	3.20×10^{-9}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
C17	3.02×10^{-11}	3.02×10^{-11}	6.97×10^{-3}	3.95×10^{-1}	7.22×10^{-6}	2.43×10^{-5}	3.02×10^{-11}	2.32×10^{-2}	3.00×10^{-11}
C18	3.02×10^{-11}	3.02×10^{-11}	4.86×10^{-3}	1.21×10^{-10}	6.70×10^{-11}	3.96×10^{-8}	3.02×10^{-11}	4.08×10^{-11}	2.63×10^{-11}
C19	3.02×10^{-11}	3.34×10^{-11}	2.39×10^{-4}	4.35×10^{-5}	5.61×10^{-5}	3.55×10^{-1}	3.02×10^{-11}	4.57×10^{-9}	1.72×10^{-12}
C20	3.02×10^{-11}	1.41×10^{-9}	1.00×10^{-3}	9.83×10^{-8}	1.91×10^{-2}	2.20×10^{-7}	3.69×10^{-11}	7.96×10^{-3}	3.02×10^{-11}
C21	3.02×10^{-11}	3.02×10^{-11}	3.18×10^{-4}	4.17×10^{-2}	2.28×10^{-5}	1.07×10^{-7}	3.02×10^{-11}	3.38×10^{-2}	3.02×10^{-11}
C22	5.49×10^{-11}	1.46×10^{-10}	5.32×10^{-3}	3.03×10^{-2}	7.70×10^{-8}	1.64×10^{-5}	3.02×10^{-11}	4.06×10^{-2}	3.02×10^{-11}
C23	1.21×10^{-12}	NaN	NaN	NaN	1.21×10^{-12}	1.21×10^{-12}	NaN	1.21×10^{-12}	NaN
C24	1.21×10^{-12}	NaN	1.61×10^{-1}	6.62×10^{-4}	1.21×10^{-12}	1.21×10^{-12}	NaN	1.21×10^{-12}	NaN
C25	1.21×10^{-12}	NaN	NaN	NaN	1.21×10^{-12}	1.93×10^{-9}	NaN	1.21×10^{-12}	NaN
C26	8.11×10^{-8}	3.55×10^{-1}	2.86×10^{-4}	4.56×10^{-2}	3.98×10^{-6}	9.59×10^{-9}	8.00×10^{-1}	7.40×10^{-3}	1.89E-02
C27	1.21×10^{-12}	NaN	NaN	4.19×10^{-2}	1.21×10^{-12}	1.21×10^{-12}	NaN	1.21×10^{-12}	NaN
C28	1.21×10^{-12}	NaN	NaN	NaN	1.21×10^{-12}	1.21×10^{-12}	NaN	1.21×10^{-12}	NaN
C29	1.21×10^{-12}	NaN	6.61×10^{-5}	1.61×10^{-1}	1.21×10^{-12}	1.21×10^{-12}	NaN	1.21×10^{-12}	NaN
C30	1.21×10^{-12}	NaN	6.25×10^{-10}	1.31×10^{-7}	1.21×10^{-12}	1.21×10^{-12}	NaN	1.21×10^{-12}	NaN

Computational Time Analysis

To show the computational cost of the proposed HAGSA, in this subsection, we record the computational time cost obtained by algorithms on the CEC 2014 test suite. The statistical results are listed in Table 5; although HAGSA has the same computational complexity as AOA and Gold-SA, the computational time cost of HAGSA is more than that of AOA and Gold-SA. This is because HAGSA uses Brownian mutation to generate two candidates' solutions to enhance the algorithm's searchability and Levy flight is used to improve the exploitation ability of the hybrid algorithm. In addition, considering the NFL theorem, it is acceptable to increase computational time to obtain reliable solutions.

Table 5: The computational time for HAGSA and its peers

Function	HAGSA	AOA	Gold-SA	ROA	AO	SCA	WOA	FPA	DE	GA
C01	0.5375	0.1722	**0.1260**	0.3587	0.3303	0.1756	0.1482	0.2102	0.2743	0.1516
C02	0.5998	0.1487	**0.0918**	0.2918	0.2854	0.1491	0.1332	0.1697	0.2010	0.1048
C03	0.5519	0.1659	0.1094	0.2395	0.2817	0.1588	0.1391	0.1558	0.2050	**0.1043**
C04	0.5085	0.1585	**0.0929**	0.2545	0.2499	0.1490	0.1803	0.1472	0.1971	0.1027
C05	0.5959	0.1564	0.1334	0.3615	0.3404	0.1521	0.1794	0.1705	0.2365	**0.1136**
C06	6.7234	1.1244	1.4928	5.5571	2.3889	1.5203	1.5837	1.5670	3.1240	**1.3135**
C07	0.6473	0.1605	0.1203	0.2872	0.3283	0.1850	0.1192	0.1661	0.2290	**0.1047**
C08	0.4786	0.1447	**0.1027**	0.2972	0.2493	0.1391	0.1212	0.1707	0.1799	0.1051
C09	0.6048	0.1847	0.1061	0.3045	0.2735	0.1681	0.1289	0.1791	0.2101	**0.1046**
C10	0.8256	0.1980	**0.1440**	0.4217	0.4360	0.2012	0.1474	0.2088	0.3306	0.1520
C11	0.8986	0.2100	**0.1596**	0.7439	0.4011	0.2126	0.1802	0.2147	0.3569	0.2055
C12	1.2724	0.3245	**0.2602**	1.1208	0.6199	0.3206	0.2946	0.3260	0.7370	0.3250
C13	0.5331	0.1451	**0.0933**	0.2555	0.2930	0.1541	0.1137	0.1541	0.2013	0.0944
C14	0.5130	0.1508	**0.1108**	0.3054	0.2816	0.1956	0.1180	0.1509	0.1855	0.1184
C15	0.4946	0.1610	0.1320	0.3372	0.3080	0.1914	0.1421	0.1771	0.2245	**0.1277**
C16	0.5078	0.1538	**0.0978**	0.3274	0.3163	0.1599	0.1164	0.1952	0.2397	0.1200
C17	0.6081	0.1684	0.1537	0.4210	0.3202	0.1730	0.1852	0.1790	0.2857	**0.1479**
C18	0.4803	0.1439	**0.1028**	0.3162	0.3332	0.2505	0.1138	0.2067	0.2115	0.1057
C19	1.6777	0.3450	0.3121	1.2646	0.7030	0.5136	0.3195	0.5490	0.8568	**0.2681**
C20	0.4975	0.1584	**0.0987**	0.3122	0.3295	0.1577	0.1371	0.1958	0.2195	0.1226

C21	0.5846	0.2016	**0.1269**	0.4338	0.3211	0.1794	0.1587	0.1744	0.2701	0.1310
C22	0.6756	0.1951	**0.1308**	0.5880	0.3688	0.1932	0.1534	0.2029	0.3152	0.1749
C23	1.8811	0.3695	**0.3146**	1.3598	0.7576	0.3692	0.4163	0.3930	0.9236	0.3200
C24	1.4512	0.2916	**0.2459**	1.1242	0.5935	0.4264	0.2725	0.4199	0.9495	0.2777
C25	1.6396	0.3334	**0.2878**	1.3109	0.7071	0.3465	0.3106	0.4518	1.0307	0.3435
C26	6.4800	**1.5258**	2.0984	5.0402	3.0212	2.0210	1.7759	1.7384	4.3710	1.6796
C27	6.3308	1.5334	**1.2872**	4.7350	2.8645	1.8617	1.8453	1.7505	4.3188	1.5384
C28	1.7570	0.4684	0.3955	1.1151	0.8401	0.4585	0.5362	0.6638	1.1272	**0.3811**
C29	2.0752	**0.4942**	0.6205	1.5271	0.9543	0.7252	0.6315	0.6343	1.4255	0.6466
C30	1.2367	**0.3321**	0.2759	0.8986	0.6684	0.3431	0.3148	0.3581	0.7941	0.4417

Industrial Engineering Design Problems

This subsection introduces five real-world industrial engineering design problems to evaluate the proposed algorithm's searchability, including the car side crash design problem, pressure vessel design problem, tension spring design problem, speed reducer design problem, and cantilever beam design problem. Unlike benchmark functions, these industrial engineering design problems have many inequality and equality constraints, which is a vital challenge to MAs. In addition, using these problems helps evaluate the potential of algorithms to solve real-world problems.

Car Side Crash Design Problem

This problem aims to maintain the side impact crash performance and minimize the vehicle weight [48]. It has 11 parameters that need to be optimized; also, ten constraints were integrated into this problem. The model of this problem can be established as follows:

Consider $x = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}]$

Minimize f(x)=Weight,

$$\text{Subject to} \begin{cases} g_1(x) = F_a(\text{load in abdomen}) \le 1\ \text{kN}, \\ g_2(x) = V \times Cu\ (\text{dummyupperchest}) \le 0.32\ \text{m/s}, \\ g_3(x) = V \times Cm\ (\text{dummymiddlechest}) \le 0.32\ \text{m/s}, \\ g_4(x) = V \times Cl\ (\text{dummylowerchest}) \le 0.32\ \text{m/s}, \\ g_5(x) = \Delta_{\text{ur}}\ (\text{upperribdeflection}) \le 32\ \text{mm}, \\ g_6(x) = \Delta_{\text{mr}}\ (\text{middleribdeflection}) \le 32\ \text{mm}, \\ g_7(x) = \Delta_{\text{lr}}\ (\text{lowerribdeflection}) \le 32\ \text{mm}, \\ g_8(x) = F\ (\text{Publicforce})_p \le 4\ \text{kN}, \\ g_9(x) = V_{\text{MBP}}(\text{Velocity of V} - \text{Pillarat middle point}) \le 9.9\ \text{mm/ms}, \\ g_{10}(x) = V_{\text{FD}}(\text{Velocity of front door at V} - \text{Pillar}) \le 15.7\ \text{mm/ms}, \end{cases}$$

$$\text{Variable range} \begin{cases} 0.5 \le x_1 - x_7 \le 1.5,\ 0.192 < x_8, x_9 < 0.345, \\ -30 \le x_{10}, x_{11} \le 30, \end{cases}$$

Table 6 shows the best results obtained by all algorithms. As shown in this table, the results of the proposed HAGSA are superior to those of other optimization techniques, and ROA and AO approaches are ranked second and third, respectively.

Table 6: Statistical results of car side crash design problem

Algorithm	Optimum Variables											Optimum Cost
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	
HAGSA	**0.5**	**1.253**	**0.5**	**1.109**	**0.5**	**0.5**	**0.501**	**0.344**	**0.192**	**3.904**	**6.381**	**22.9765**
AOA	0.5	1.262	0.5	1.156	0.5	0.772	0.5	0.310	0.192	0.365	1.162	23.2139
Gold-SA	0.5	1.278	0.612	1.102	0.544	1.323	0.5	0.345	0.345	0.170	0.294	23.9711
ROA	0.5	1.235	0.5	1.166	0.5	1.110	0.5	0.341	0.192	0.275	2.926	23.0801
AO	0.724	1.175	0.502	1.200	0.5	0.792	0.5	0.308	0.192	0.739	2.837	23.1694
SCA	0.567	1.334	0.540	1.167	0.5	1.109	0.5	0.233	0.263	0.301	2.393	24.3513
WOA	0.953	1.106	0.5	1.206	0.524	0.559	0.501	0.282	0.298	0.246	7.326	24.6495
FPA	0.532	1.322	0.515	1.143	0.616	0.516	0.534	0.197	0.197	0.710	1.892	24.1309
DE	0.505	1.446	0.521	1.182	0.5	1.466	0.5	0.312	0.192	1.008	13.266	24.7181
GA	1.073	1.0465	0.595	1.096	0.714	0.502	0.521	0.322	0.264	5.549	8.215	25.4504

Pressure Vessel Design Problem

The pressure vessel design problem is shown in Figure 10. The goal of this problem is to minimize the total cost [49]. It has four design parameters: shell thickness (T_s), ball thickness (T_h), shell radius (R), and shell length (L). The constraints and objective function can be expressed as follows:

Consider $x = [x_1\ x_2\ x_3\ x_4] = [T_s\ T_h\ R\ L]$

Minimize $f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$

$$\text{Subject to} \begin{cases} g_1(x) = -x_1 + 0.0193x_3 \le 0,\ g_2(x) = -x_3 + 0.00954x_3 \le 0, \\ g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1{,}296{,}000 \le 0,\ g_4(x) = x_4 - 240 \le 0 \end{cases}$$

$$\text{Variable range} \begin{cases} 0 \le x_1 \le 99,\ 0 \le x_2 \le 99, \\ 10 \le x_3 \le 200,\ 10 \le x_4 \le 200 \end{cases}$$

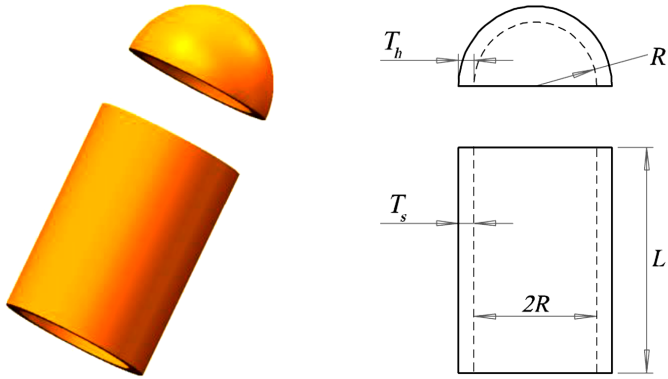


Figure 10: Pressure vessel design problem.

Table 7 shows the statistical results obtained by HAGSA and other comparison algorithms including AOA, Gold-SA, ROA, AO, SCA, WOA, FPA, DE, and GA. As can be seen from this table, HAGSA achieves competitive results in this design problem, and the results of ROA and AO are ranked second and third, respectively.

Table 7: Statistical results of the pressure vessel design problem

Algorithm	Optimum Variables				Optimum Cost
	T_s	T_h	R	L	
HAGSA	0.8304795	0.3770664	44.00935	154.9557	5982.8355
AOA	0.8395475	0.4113845	44.27936	156.8883	6068.3284
Gold-SA	0.7140179	0.4619435	40.49522	197.7362	6090.4062
ROA	0.8610026	0.3934984	44.96907	144.2921	6023.0145
AO	0.8030047	0.4524486	43.65139	158.3146	6024.2153
SCA	0.963087	0.476939	51.4412	87.3095	6246.7789
WOA	0.937726	0.473373	49.9436	98.8134	6195.7655
FPA	0.971843	0.478402	52.5479	81.3225	6393.2109
DE	1.009677	0.498834	54.0470	69.2270	6398.6641
GA	1.025422	0.484037	54.7458	64.6720	6439.9228

Tension Spring Design Problem

The main goal of this problem is to find the optimal parameters to minimize the production cost [50]. There are three parameters: wire diameter (d), mean diameter of the spring (D), and number of active coils (N), as shown in Figure 11. The mathematical model is expressed as follows:

$$\text{Consider } x = [x_1\ x_2\ x_3] = [d\ D\ N]$$

$$\text{Minimize } f(x) = (x_3 + 2)x_2x_1^2$$

$$\text{Subject to } \begin{cases} g_1(x) = 1 - \frac{x_2^3 x_3}{71{,}785x_1^4} \le 0,\ g_2(x) = \frac{4x_2^2 - x_1x_2}{12{,}566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \le 0, \\ g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0,\ g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0 \end{cases}$$

$$\text{Variable range } \begin{cases} 0.05 \le x_1 \le 2.00,\ 0.25 \le x_2 \le 1.30 \\ 2.00 \le x_3 \le 15.00 \end{cases}$$

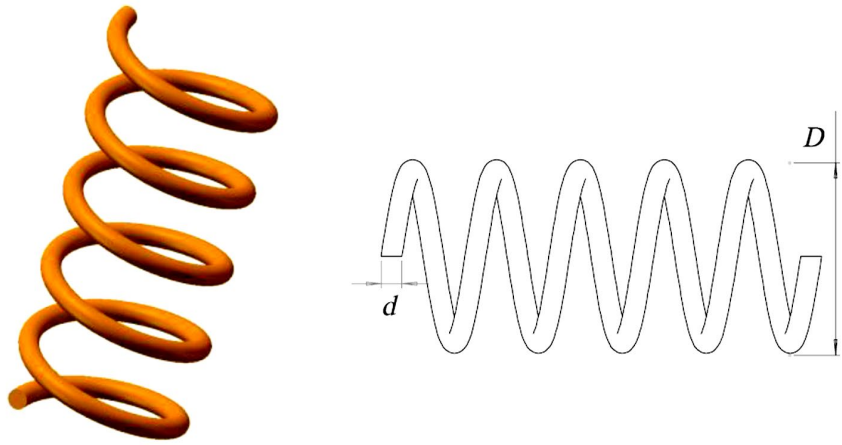


Figure 11: Tension spring design problem.

The statistical results of the tension spring design problem were obtained by HAGSA and other comparison algorithms as listed in Table 8. As can be seen from this table, the best cost of this design problem is 0.011196, and the three parameters are 0.050411, 0.37384, and 9.7854, respectively.

Table 8: Statistical results of the tension spring design problem

Algorithm	Optimum Variables			Optimum Cost
	d	*D*	*N*	
HAGSA	0.050411	0.37384	9.7854	0.011196
AOA	0.051791	0.388	9.5556	0.012026
Gold-SA	0.060683	0.67982	3.1063	0.012783
ROA	0.059221	0.6308	3.5188	0.012209
AO	0.05	0.337193	13.0905	0.012721
SCA	0.061365	0.70355	2.9232	0.013043
WOA	0.0502069	0.351224	12.336	0.012692
FPA	0.10187	1.093	9.5387	0.130890
DE	0.06766	0.907935	2.0871	0.016985
GA	0.05401	0.465113	9.6797	0.015848

Speed Reducer Design Problem

This problem aims to construct a speed reducer with a minimum weight under constraints [51]. There are seven parameters: face width, the module of teeth, number of teeth in the pinion, length of the first shaft between bearings, length of the second shaft between bearings, the diameter of the first shafts, and the diameter of second shafts. Figure 12 shows the design of this problem, and its mathematical formula is as follows:

$$\text{Consider } x = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7]$$

$$\text{Minimize } f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3),$$

$$\text{Subject to} \begin{cases} g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \le 0,\ g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0, \\ g_3(x) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0,\ g_4(x) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0, \\ g_5(x) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9\times10^6}}{110.0x_6^3} - 1 \le 0,\ g_6(x) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 157.5\times10^6}}{85.0x_6^3} - 1 \le 0, \\ g_7(x) = \frac{x_2x_3}{40} - 1 \le 0,\ g_8(x) = \frac{5x_2}{x_1} - 1 \le 0, \\ g_9(x) = \frac{x_1}{12x_2} - 1 \le 0,\ g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0, \\ g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0, \end{cases}$$

$$\text{Variable range} \begin{cases} 2.6 \le x_1 \le 3.6,\ 0.7 \le x_2 \le 0.8, \\ 17 \le x_3 \le 28,\ 7.3 \le x_4 \le 8.3, \\ 7.8 \le x_5 \le 8.3,\ 2.9 \le x_6 \le 3.9, \\ 5.0 \le x_7 \le 5.5 \end{cases}$$

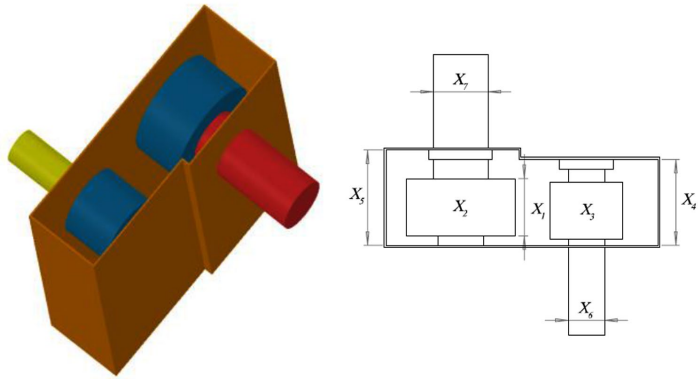


Figure 12: Speed reducer problem.

The proposed HAGSA is compared with AOA, Gold-SA, ROA, AO, SCA, WOA, FPA, DE, and GA. The statistical results are shown in Table 9. As can be seen, HAGSA is excellent for solving speed reducer design problems, and the results obtained by HAGSA are ranked first. The results of AOA and ROA are ranked second and third, respectively.

Table 9: Statistical results of the speed reducer design problem

Algorithm	Optimum Variables							Optimum Cost
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
HAGSA	**3.49767**	**0.7**	**17**	**7.3**	**7.8001**	**3.34982**	**5.28559**	**2995.4897**
AOA	3.50776	0.7	17	7.77685	7.96133	3.35075	5.28557	3007.0806
Gold-SA	3.49441	0.7	17	7.3	7.8	3.42383	5.2872	3016.2163
ROA	3.50776	0.7	17	7.77685	7.96133	3.35075	5.28557	3007.0806
AO	3.49748	0.7	17	8.07645	7.8	3.35162	5.28573	3002.8462
SCA	3.6	0.7	17	8.3	8.3	3.43032	5.30013	3085.2732
WOA	3.5247	0.7	17	8.14441	8.05897	3.35091	5.28568	3019.883
FPA	3.6	0.7	17	7.3	7.8	3.41261	5.28143	3056.8032
DE	3.5119	0.7	17	8.3	8.3	3.37356	5.38151	3088.6759
GA	3.4896	0.7	17	7.71388	7.8	3.65614	5.29218	3094.3185

Cantilever Beam Design

The design of the cantilever beam is shown in Figure 13, and the goal of this problem is to minimize the total weight [52]. There are five parameters that need to be optimized. The objective function and constraints of this problem are as follows:

Consider $x = [x_1\ x_2\ x_3\ x_4\ x_5]$

Minimize $f(x) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5)$

Subject to $g(x) = \frac{60}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$

Variable range $0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$

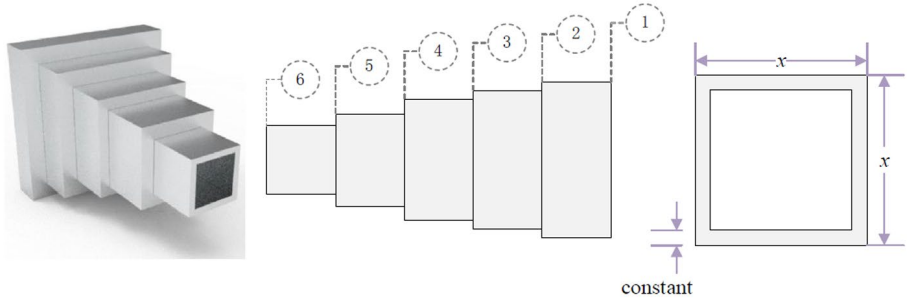


Figure 13: Cantilever beam structure.

The statistical results obtained by HAGSA, AOA, Gold-SA, ROA, AO, SCA, WOA, FPA, DE, and GA are shown in Table 10. From this table, HAGSA shows a lower cost than that of other optimization techniques, and the results of ROA and AO are ranked second and third, respectively.

Table 10: Statistical results of the cantilever beam design problem

Algorithm	Optimum Variables					Optimum Cost
	x_1	x_2	x_3	x_4	x_5	
HAGSA	**5.9271**	**5.3962**	**4.5081**	**3.476**	**2.1726**	**1.3404**
AOA	6.4746	5.515	4.1138	3.7827	1.8724	1.3577
Gold-SA	5.7908	5.0142	4.9397	3.4175	2.5713	1.3562
ROA	5.8567	5.4316	4.4342	3.6542	2.1263	1.3418
AO	5.8219	5.4572	4.4551	3.5517	2.2198	1.342
SCA	5.781	5.5669	4.9992	3.5049	2.5094	1.3954
WOA	6.6424	5.0184	4.8451	3.0428	2.287	1.3626
FPA	5.7763	6.4239	4.6938	3.6501	1.6685	1.3861
DE	7.1323	4.9612	4.2559	3.3748	2.5797	1.3918
GA	6.5195	4.1943	5.7643	4.1847	2.2862	1.4320

CONCLUSIONS AND FUTURE WORK

Considering the characteristic of AOA and Gold-SA, this paper proposes a hybrid optimization algorithm, namely HAGSA. First, Gold-SA is utilized to alleviate the shortcomings of AOA, such as low population diversity, premature convergence, and easy stagnation into local optimal solutions. Second, Levy flight and a new strategy called Brownian mutation are used to enhance the searchability of the hybrid algorithm.

We first used the CEC 2014 competition test suite to validate the optimization performance of HAGSA and its peers. The experimental results demonstrate that HAGSA outperforms other competitors in terms of optimization accuracy, convergence speed, robustness, and statistical difference. In addition, five industrial engineering design problems were carried out to test the ability of HAGSA to solve real-world problems. The experimental results also show that HAGSA is significantly better than its peers. Therefore, it is believed that HAGSA is a valuable method and can provide high-quality solutions to solve these kinds of problems. Although HAGSA has significant improvements over the original AOA and Gold-SA, its time consumption is a potential issue. This is because the BM strategy produces two candidate solutions and uses fitness evaluation to select the best solution. Thus, determining how to reduce the computational time under the premise of ensuring performance needs further research. In future works, we will: (1) improve the BM strategy to reduce the computational time without degrading HAGSA's performance; (2) seek to hybridize other MAs to improve AOA's optimization performance; and (3) apply HAGSA to solve combinatorial optimization problems (e.g., the traveling salesman problem, knapsack problem, and graph coloring problem). In addition, multilevel thresholding image segmentation would also be an interesting and meaningful research area.

AUTHOR CONTRIBUTIONS

Q.L., Conceptualization, methodology, software, formal analysis, investigation, data curation, visualization, writing-original draft preparation, writing—review and editing, funding acquisition, validation, resources, project administration. N.L., supervision, writing-review and editing, resources, validation, funding acquisition. H.J., project administration, validation, conceptualization, supervision, methodology, writing—review and editing, funding acquisition. Q.Q., project administration, resources, supervision, validation, conceptualization, methodology, writing—review

and editing, funding acquisition. L.A., writing—review and editing, supervision. Y.L., validation, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENTS

We acknowledge the anonymous reviewers for their constructive comments.

REFERENCES

1. Esparza, E.R.; Calzada, L.A.Z.; Oliva, D.; Heidari, A.A.; Zaldivar, D.; Cisneros, M.P.; Foong, L.K. An efficient harris hawks-inspired image segmentation method. *Expert Syst. Appl.* 2020, *155*, 113428.
2. Liu, Q.; Li, N.; Jia, H.; Qi, Q.; Abualigah, L. Modified remora optimization algorithm for global optimization and multilevel thresholding image segmentation. *Mathematics* 2022, *10*, 1014.
3. Ewees, A.A.; Abualigah, L.; Yousri, D.; Sahlol, A.T.; Al-qaness, A.A.; Alshathri, S.; Elaziz, M.A. Modified artificial ecosystem-based optimization for multilevel thresholding image segmentation. *Mathematics* 2021, *9*, 2363.
4. Wang, S.; Liu, Q.; Liu, Y.; Jia, H.; Liu, L.; Zheng, R.; Wu, D. A hybrid SSA and SMA with mutation opposition-based learning for constrained engineering problems. *Comput. Intell. Neurosci.* 2021, *2021*, 6379469.
5. Houssein, E.H.; Mahdy, M.A.; Blondin, M.J.; Shebl, D.; Mohamed, W.M. Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. *Expert Syst. Appl.* 2021, *174*, 114689.
6. Wang, S.; Jia, H.; Liu, Q.; Zheng, R. An improved hybrid aquila optimizer and harris hawks optimization for global optimization. *Math. Biosci. Eng.* 2021, *18*, 7076–7109.
7. Wu, D.; Wang, S.; Liu, Q.; Abualigah, L.; Jia, H. An Improved Teaching-Learning-Based Optimization Algorithm with Reinforcement Learning Strategy for Solving Optimization Problems. *Comput. Intell. Neurosci.* 2022, *2022*, 1535957.
8. Zhang, H.; Wang, Z.; Chen, W.; Heidari, A.A.; Wang, M.; Zhao, X.; Liang, G.; Chen, H.; Zhang, X. Ensemble mutation-driven salp swarm algorithm with restart mechanism: Framework and fundamental analysis. *Expert Syst. Appl.* 2021, *165*, 113897.
9. Giovanni, L.D.; Pezzella, F. An improved genetic algorithm for the distributed and flexible Job-shop scheduling problem. *Eur. J. Oper. Res.* 2010, *200*, 395–408.
10. Wu, B.; Zhou, J.; Ji, X.; Yin, Y.; Shen, X. An ameliorated teaching–learning-based optimization algorithm based study of image segmentation for multilevel thresholding using Kapur's entropy and Otsu's between class variance. *Inf. Sci.* 2020, *533*, 72–107.

11. Wang, S.; Jia, H.; Abualigah, L.; Liu, Q.; Zheng, R. An improved hybrid aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems. *Processes* 2021, *9*, 1551.
12. Lin, S.; Jia, H.; Abualigah, L.; Altalhi, M. Enhanced slime mould algorithm for multilevel thresholding image segmentation using entropy measures. *Entropy* 2021, *23*, 1700.
13. Su, H.; Zhao, D.; Yu, F.; Heidari, A.A.; Zhang, Y.; Chen, H.; Li, C.; Pan, J.; Quan, S. Horizontal and vertical search artificial bee colony for image segmentation of COVID-19 X-ray images. *Comput. Biol. Med.* 2021, *142*, 105181.
14. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* 2016, *95*, 51–67.
15. Khare, A.; Rangnekar, S. A review of particle swarm optimization and its applications in solar photovoltaic system. *Appl. Soft Comput.* 2013, *13*, 2997–3006.
16. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* 2014, *69*, 46–61. [Green Version]
17. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* 2017, *114*, 163–191.
18. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* 2015, *83*, 80–98.
19. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* 2015, *89*, 228–249.
20. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* 2020, *111*, 300–323.
21. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* 2019, *97*, 849–872.
22. Abualigah, L.; Elaziz, M.A.; Sumari, P.; Geem, Z.; Gandomi, A.H. Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* 2022, *191*, 116158.
23. Abualigah, L.; Yousri, D.; Abd, E.M.; Ewees, A.A. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* 2021, *157*, 107250.

24. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* 2015, *27*, 495–513.
25. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* 2016, *96*, 120–133.
26. Abualigah, L.; Diabat, A.; Mirjalili, S.; Elaziz, A.E.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* 2021, *376*, 113609.
27. Tanyildizi, E.; Demir, G. Golden sine algorithm: A novel math-inspired algorithm. Golden sine algorithm: A novel math-inspired algorithm. *Adv. Electr. Comput. Eng.* 2017, *17*, 71–78.
28. Neggaz, N.; Houssein, E.H.; Hussain, K. An efficient henry gas solubility optimization for feature selection. *Expert Syst. Appl.* 2020, *152*, 113364.
29. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* 2009, *179*, 2232–2248.
30. Sun, P.; Liu, H.; Zhang, Y.; Meng, Q.; Tu, L.; Zhao, J. An improved atom search optimization with dynamic opposite learning and heterogeneous comprehensive learning. *Appl. Soft Comput.* 2021, *103*, 107140.
31. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl.-Based Syst.* 2021, *191*, 105190.
32. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* 2021, *80*, 8091–8126.
33. Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* 2008, *12*, 702–713. [Green Version]
34. Slowik, A.; Kwasnicka, H. Evolutionary algorithms and their applications to engineering problems. *Neural Comput. Appl.* 2020, *32*, 12363–12379. [Green Version]
35. Hansen, N.; Ostermeier, A. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evol. Comput.* 2001, *9*, 159–195.
36. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* 1997, *1*, 67–82. [Green Version]
37. Azizi, M.; Talatahari, S. Improved arithmetic optimization algorithm for design optimization of fuzzy controllers in steel building structures with nonlinear behavior considering near fault ground motion effects. *Artif. Intell. Rev.* 2021.

38. Agushaka, J.O.; Ezugwu, A.E. Advanced arithmetic optimization algorithm for solving mechanical engineering design problems. *PLoS ONE* 2021, *16*, 0255703.
39. Wang, R.; Wang, W.; Xu, L.; Pan, J.; Chu, S. An adaptive parallel arithmetic optimization algorithm for robot path planning. *J. Adv. Transport.* 2021, *2021*, 3606895.
40. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A.H. A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of COVID-19 CT images. *Processes* 2021, *9*, 1155.
41. Liu, Y.; Cao, B. A novel ant colony optimization algorithm with Levy flight. *IEEE Access* 2020, *8*, 67205–67213.
42. Iacca, G.; Junior, V.C.S.; Melo, V.V. An improved jaya optimization algorithm with Levy flight. *Expert Syst. Appl.* 2021, *165*, 113902.
43. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* 2020, *152*, 113377.
44. Li, M.; Zhao, H.; Weng, X.; Han, T. A novel nature-inspired algorithm for optimization: Virus colony search. *Adv. Eng. Softw.* 2016, *92*, 65–88.
45. Jia, H.; Peng, X.; Lang, C. Remora optimization algorithm. *Expert Syst. Appl.* 2021, *185*, 115665.
46. Zhou, Y.; Wang, R.; Luo, Q. Elite opposition-based flower pollination algorithm. *Neurocomputing* 2016, *188*, 294–310.
47. Ewees, A.A.; Elaziz, M.A.; Houssein, E.H. Improved grasshopper optimization algorithm using opposition-based learning. *Expert Syst. Appl.* 2018, *112*, 156–172.
48. Yildiz, B.S.; Pholdee, N.; Bureerat, S.; Yildiz, A.R.; Sait, S.M. Enhanced grasshopper optimization algorithm using elite opposition-based learning for solving real-world engineering problems. *Eng. Comput.* 2021.
49. Houssein, E.H.; Helmy, B.E.; Rezk, H.; Nassef, A.M. An efficient orthogonal opposition-based learning slime mould algorithm for maximum power point tracking. *Neural Comput. Appl.* 2022, *34*, 3671–3695.

50. Taheri, A.; Rahimizadeh, K.; Rao, R.V. An efficient balanced teaching-learning-based optimization algorithm with individual restarting strategy for solving global optimization problems. *Inf. Sci.* 2021, *576*, 68–104.
51. Ahmadianfar, I.; Heidari, A.A.; Gandomi, A.H.; Chu, X.; Chen, H. RUN beyond the metaphor: An efficient optimization algorithm based on runge kutta method. *Expert Syst. Appl.* 2021, *181*, 115079.
52. Cheng, Z.; Song, H.; Wang, J.; Zhang, H.; Chang, T.; Zhang, M. Hybrid firefly algorithm with grouping attraction for constrained optimization problem. *Knowl. Based Syst.* 2021, *220*, 106937.

Chapter

12

MODELING AND OPTIMIZING THE SYSTEM RELIABILITY USING BOUNDED GEOMETRIC PROGRAMMING APPROACH

Shafiq Ahmad [1], Firoz Ahmad [2,3], Intekhab Alam [3], Abdelaty Edrees Sayed [1] and Mali Abdollahian [4]

[1]Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia

[2]Department of Management Studies, Indian Institute of Science, Bangalore 560012, India

[3]Department of Statistics and Operations Research, Aligarh Muslim University, Aligarh 202002, India

[4]School of Science, College of Sciences, Technology, Engineering, Mathematics, RMIT University, Melbourne, VIC 3001, Australia

ABSTRACT

The geometric programming problem (GPP) is a beneficial mathematical programming problem for modeling and optimizing nonlinear optimization

Citation: (APA): Ahmad, S., Ahmad, F., Alam, I., Sayed, A.E., & Abdollahian, M. (2022). Modeling and Optimizing the System Reliability Using Bounded Geometric Programming Approach. Mathematics 2022, 10, 2435. . (19 pages).

problems in various engineering fields. The structural configuration of the GPP is quite dynamic and flexible in modeling and fitting the reliability optimization problems efficiently. The work's motivation is to introduce a bounded solution approach for the GPP while considering the variation among the right-hand-side parameters. The bounded solution method uses the two-level mathematical programming problems and obtains the solution of the objective function in a specified interval. The benefit of the bounded solution approach can be realized in that there is no need for sensitivity analyses of the results output. The demonstration of the proposed approach is shown by applying it to the system reliability optimization problem. The specific interval is determined for the objective values and found to be lying in the optimal range. Based on the findings, the concluding remarks are presented.

Keywords: interval-based parameters; geometric programming problems; bounded optimization approach; system reliability

INTRODUCTION

Mathematical programming problems have different forms based on the nature of objective functions and constraints. The geometric programming problem, a typical form of mathematical optimization characterized by objective and constraint functions of a particular form, was introduced by [1]. Later, the advanced study in the domain of the GPP was performed by [2,3]. Several engineering applications [4] have investigated the effectiveness and importance of the GPP. The GP optimization approach inevitably outperforms other existing techniques due to the objective function's relative magnitudes instead of the decision variables. Initially, the GP technique's basic working principle is based on finding the optimal solution of the objective function and then proceeding further to determine the optimal values of the design variables. This characteristic feature of the GPP is essential and fruitful in circumstances where the decision-makers are interested in first finding the optimal values of the objective function. Thus, the polynomial structure of the objectives and constraints leads the GPP towards the simpler convex solution space [2,4]. GP optimization techniques can tackle this situation, and the computational activities are aborted to obtain the optimum design vectors. One of the GP techniques' most crucial advantages over others can be regarded as it mitigates the complex optimization problems into the different piecewise linear algebraic equations. On the other hand, the GP

approach deals only with the posynomial types of algebraic terms, meaning that it solely facilitates the objective function and the constraints with posynomial structures, which can be considered significant drawbacks.

Many engineering optimization problems deal with the manufacturing and production processes of some products, machinery parts, and the raw equipment used in the final usable machines and products. They care about the structure, dimensions, quality, and specifications of the raw material parts that are very important to be transformed and converted into usable products. For example, the cofferdam, shaft, journal bearings, etc., are the raw parts that are the building block raw parts of the various products and machines. Hence, the mathematical models with the specifications of these raw parts are built up and further used in manufacturing and producing the final products. Sometimes, the perfect specifications cannot be achieved due to some vagueness or technical errors in the functioning machine, for which the experts/managers allow some marginal variations among the specifications and dimensions of such raw parts. Afterward, it can be managed or adjusted to some extent. In system reliability modeling, various parameters can be taken as varying between some specified intervals. This means that the parameters can be taken as uncertain, and using some specified tools, they can be converted into crisp ones. In the literature, the concept of fuzzy and random parameters is available, which deals with the vagueness and randomness in the parameters. However, we have provided an opportunity to define the parameters under the continuous variations bounded by upper and lower limits. Instead of taking the vague and random parameters, one can assume the continuous variations in the parameters' values can be tackled with the two-level mathematical programming techniques discussed in this paper. Additionally, the sensitivity and post-optimality of the obtained solution results are waived off due to the working procedure of the proposed approach. Hence, the proposed bonded approach for the GPP can be easily implemented on various non-engineering problems while dealing with varying parameters.

The remaining part of the paper is summarized as follows: In Section 2, some relevant literature is discussed, while Section 3 presents the basic concepts and modeling of standard geometric programming problems along with the proposed bounded solution methods. The computational study is presented with a particular focus on system reliability optimization in Section 4. Analyses of the computational complexity are also performed with other existing approaches. Finally, conclusions and the future scope are discussed based on the present work in Section 5.

LITERATURE REVIEW

The GPP is a relatively new method of solving nonlinear programming problems. It is used to minimize functions in the form of posynomials subject to constraints of the same type. Practical algorithms have been developed for solving geometric programming problems [1,2,3]. Liu [5] proposed the posynomial GPP subject to fuzzy relation inequalities. In 2018, Lu and Liu [6] also studied a class of posynomial GPPs that considers the evaluation of a posynomial GPP subject to fuzzy relational equations with max–min composition. Ahmad and Adhami [7] also addressed the interval-based solution approach for solving transportation problems under varying input parameters. Chakraborty et al. [8] discussed the multiobjective GPP with the aid of fuzzy geometry. Garg et al. [9] presented the reliability optimization problem under an intuitionistic environment. Islam and Roy [10] investigated the modified GPP and applied it to many engineering problems. Islam and Roy [11] developed a new multiobjective GP model and used it to solve the transportation problem. Recently, a interesting study on GPP was presented by [12,13,14,15]. Khorsandi et al. [16] developed a new optimization technique for GPP. Mahapatra and Roy [17] also solved the reliability of a system using the GPP approach.

However, the geometric programming research approach in the field of reliability optimization is being performed in the context of mathematical modeling and real-life applications. Some recent work is also available on the system reliability, ensuring a significant contribution to the literature. Negi et al. [18] presented a hybrid optimizer model for system reliability. Roustaee and Kazemi [19] developed a stochastic model for multi-microgrid constrained reliability system and applied it to clean energy management. Zolfaghari and Mousavi [20] proposed an integrated system reliability model for the inbuilt component under uncertainty. Sedaghat and Ardakan [21] developed a novel computational strategy for redundant components in the system reliability optimization. Meng et al. [22] discussed the interval parameters by the sequential moving asymptote method for the system reliability based on the integrated co-efficient approach. Kugele et al. [23] presented a research work by integrating the second-degree difficulty in carbon ejection controlled, reliable, innovative production management and implemented it on a computational dataset. Son et al. [24] used the modeling texture of the GPP in the levelized cost of energy-oriented modular string inverter design and discussed it in the field of PV generation systems. Shen et al. [25] also introduced a novel method for energy-efficient ultrareliability using the outage probability bound and the GPP technique. Rajamony et

al. [26] designed multi-objective single-phase differential buck inverters by considering an active power decoupling and applied it to power generation. Singh and Singh [27] suggested the geometric programming approach for optimizing multi-VM migration by allocating transfer and compression.

All the studies are confined to either fuzzy- or stochastic-based approaches, but it may possible that input parameters may vary within some specified intervals bounded by upper and lower bounds. In this situation, the fuzzy and stochastic approaches may not be applied successfully. Thus, to overcome this issue, we developed a bounded solution method comprising the two-level GPP, and the values of the objective function are obtained directly. Hence, the present study lays down a new direction for obtaining the optimal solution under the varying parameters. The proposed method is applied to system reliability optimization problems and yields a result without affecting the system reliability under variations.

GEOMETRIC PROGRAMMING PROBLEM: BASIC CONCEPTS

In this sub-section, we discuss some important basic concepts related to geometric programming problems.

Basic Concepts

Definition 1

(Monomial)**.** *The word "monomial" is derived from the Latin word mono, meaning only one, and mial solely means term. Therefore, a monomial literally means "an expression in algebra having only one term".*

Thus, if $x_1, x_2, \cdots, x_n$ *represent the n non-negative variable, then a real-valued function F of x, in the following form*

$$F(x) = cx_1^{a_1} x_2^{a_2} \cdots x_n^{a_n},$$

where $c>0$ *and* $a_i \in R$*, is known as the monomial function.*

Illustrative Example 1: If a, b, and c are non-negative variables, then $6, 0.84, 9a^3b^{-9}, 17\sqrt{c/a}$ *are monomial, but* $5+a$*,* $6a-8c$*, and* $7(a+8a^6b^{-7})$ *are not monomials.*

Definition 2

(Polynomial)**.** *The word "polynomial" is also derived from the Latin word poly, meaning many, and mial solely means term. Therefore, a polynomial literally means "an expression in algebra having many terms", i.e., many monomials.*

Suppose $x_1,x_2,\cdots,x_n$ *represent the n non-negative variable, then the sum of one or more monomials in the following form of a real-valued function F of x:*

$$F(x) = \sum_{i=1}^{m} c_i x_1^{a_{1i}} x_2^{a_{2i}} \cdots x_n^{a_{ni}},$$

where $a_{ni} \in R$ *is known as a polynomial function or simply a polynomial.*

Illustrative Example 2: If a, b, and c are non-negative variables, then 6, 0.84, $9a^3b^{-9}$*,* $-5c/a$*,* $6a-8c$*, and* $7(a+8a^6b^{-7})$ *are polynomials.*

Definition 3

(Posynomial)**.** *If the coefficients* $c_i>0$ *in the polynomial, then it is called a posynomial. Therefore, the sum of one or more monomials in the following form of a real-valued function F of x:*

$$F(x) = \sum_{i=1}^{m} c_i x_1^{a_{1i}} x_2^{a_{2i}} \cdots x_n^{a_{ni}}, c_i > 0$$

where $c_i>0$ *and* $a_{ni} \in R$ *is called a posynomial function or simply posynomial. Illustrative Example 3: If a, b, and c are non-negative variables, then 6, 0.84,* $9a^3b^{-9}$*,* $17\sqrt{c/a} + a^7b^4$ *are posynomial, but* $5-a$*,* $6a-8c$*, and* $7(a+8a^6b^{-7})$ *are not posynomial.*

Note 1: The term posynomial is used to suggest a combination of positive and polynomial, that is **POSITIVE + POLYNOMIAL = POSYNOMIAL**.

Definition 4

(Degree of difficulty)**.** *This is defined as a quantity* $(N-n-1)$ *present in geometric programming called the degree of difficulty. In the case of a constrained geometric programming problem, N represents the total number of terms in all the posynomials and n represents the number of design variables.*

Note 2: The comparison and differences between monomial, polynomial, and posynomial are summarized in Table 1.

Table 1: Comparison between monomial, polynomial, and posynomial

Monomial	Polynomial	Posynomial
(1) Deals with a single term	Having one or more term	Having one or more term
(2) Sum of monomials is not a monomial	Sum of polynomials is a polynomial	Sum of posynomials is a posynomial
(3) Subtraction of monomials is not a monomial	Subtraction of polynomials is a polynomial	Subtraction of posynomials is not a posynomial
(4) Multiplication of monomials is a monomial	Multiplication of polynomials is a polynomial	Multiplication of posynomials is a posynomial
(5) Division of a monomial by other monomials is a monomial	Division of a polynomial by other monomials is a polynomial	Division of a posynomial by other monomials is a posynomial
(6) The mathematical expression of a monomial is $F(x) = cx_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$, $c > 0$	The mathematical expression of a polynomial is $F(x) = \sum_{i=1}^{m} c_i x_1^{a_{1i}} x_2^{a_{2i}} \cdots x_n^{a_{ni}}$	The mathematical expression of a posynomial is $F(x) = \sum_{i=1}^{m} c_i x_1^{a_{1i}} x_2^{a_{2i}} \cdots x_n^{a_{ni}}$, $c_i > 0$
(7) Example: $0.84, 9a^3b^{-9}, 17\sqrt{c/a}$	Example: $0.43, 9a^3b^{-9}, -5c/a, 6a - 8c$	Example: $0.59, 9a^3b^{-9}, 17\sqrt{c/a} + a^7b^4$

Geometric Programming Problems

Geometric programming problems fall under a class of nonlinear programming problems characterized by objective and constraint functions in a special form. The texture of GPP is quite different from other mathematical programming problems and depends on the characterization of decision variables in its product form. Thus, the modeling structure of different engineering problems inevitably adheres to the form of the GPP while optimizing the real-life problems. It is introduced for the solution of the algebraic nonlinear programming problems under the linear or nonlinear constraints, used to solve dynamic optimization problems. The useful impact in the area can be realized by its enormous application in integrated circuit design, manufacturing system design, and project management. Therefore, the standard form of GPP formulations can be represented as follows (1):

$$\begin{aligned} \mathrm{F} = \underset{x}{\text{Minimize}} \quad & \textstyle\sum_{k=1}^{l_0} c_{0k} \prod_{j=1}^{n} x_j^{\alpha_{0kj}} \\ \text{s. x.} \quad & \\ & \textstyle\sum_{k=1}^{l_i} c_{ik} \prod_{j=1}^{n} x_j^{\beta_{ikj}} \le 1, \quad i = 1, 2, \ldots, m, \\ & x_j \ge 0, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{1}$$

where l_0 is the number of terms present in the objective function, while the inequality constraints include l_i terms for i=1,2,⋯,m. Geometric programming problems have a strong duality theorem, and hence, geometric programming problems with enormously nonlinear constraints can be depicted correspondingly as one with only linear constraints. Moreover, if the primal problem is in the form of a posynomial, then a global solution of a minimization-type problem can be determined by solving its dual maximization-type problem. The dual problem contains the desirable

characteristics of being linearly constrained and with an objective function having wholesome features. This leads towards the development of the most promising solution methods for the geometric programming problems.

Assume that we replace the right-hand-side term (RHS) of the constraints in the GPP (1). Then, the modified GPP can be given as follows (2):

Primal

$$\begin{aligned} \mathrm{F} = \underset{x}{\text{Minimize}} \quad & \textstyle\sum_{k=1}^{l_0} c_{0k} \prod_{j=1}^{n} x_j^{\alpha_{0kj}} = F_0(x)(say) \\ \text{s. x.} \quad & \\ & \textstyle\sum_{k=1}^{l_i} c_{ik} \prod_{j=1}^{n} x_j^{\beta_{ikj}} \le B_i, \quad i = 1, 2, \ldots, m, \\ & x_j \ge 0, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{2}$$

where $B_i, \forall i=1,2,\cdots,m$ are non-negative numbers. If $B_i=1, \forall i$, then this modified geometric programming problem (2) is the standard geometric programming problem (1).

Consider that the geometric programming problem (2) is the primal problem, then its dual problem can be presented in the geometric programming problem (4). For this purpose, we formulate an auxiliary geometric programming problem (3) by dividing the constraint co-efficient by its RHS value B_i, which can be depicted as follows:

$$\begin{aligned} \mathrm{F} = \underset{x}{\text{Minimize}} \quad & \textstyle\sum_{k=1}^{l_0} c_{0k} \prod_{j=1}^{n} x_j^{\alpha_{0kj}} \\ \text{s. x.} \quad & \\ & \textstyle\sum_{k=1}^{l_i} \frac{c_{ik}}{B_i} \prod_{j=1}^{n} x_j^{\beta_{ikj}} \le 1, \quad i = 1, 2, \ldots, m, \\ & x_j \ge 0, \quad j = 1, 2, \ldots, n. \end{aligned} \tag{3}$$

The derivation for the dual formulation of the geometric programming problem (2) can be carried out using the concept of [1,2,3]. Furthermore, the potential complexity in obtaining and solving the dual geometric programming problem (4) can be realized by the research work in [6,16]. Thus, the dual formulation of the geometric programming problem (2) is presented in the geometric programming problem (4).

Dual

$$\begin{aligned} \mathrm{F} = \underset{y}{\text{Maximize}} \quad & \prod_{k=1}^{l_0} \left(\frac{c_{0k}}{y_{0k}} \right)^{y_{0k}} \prod_{i=1}^{m} \prod_{k=1}^{l_i} \left(\frac{c_{ik} y_{i0}}{B_i y_{ik}} \right)^{y_{ik}} = F(y)(say) \\ \text{s. x.} \quad & \\ & \textstyle\sum_{k=1}^{l_0} y_{0k} = 1, \quad i = 1, 2, \ldots, m, \\ & \textstyle\sum_{k=1}^{l_0} \alpha_{0kj} y_{0k} + \sum_{i=1}^{m} \sum_{k=1}^{l_i} \beta_{ikj} y_{ik} = 0, \quad j = 1, 2, \ldots, n, \\ & y_{ik} \ge 0, \quad \forall \ i, k. \end{aligned} \tag{4}$$

Theorem 1.

If δ is a feasible vector for the constraint posynomial geometric programming (2), then $F_0(x) \geq n \sqrt[n]{F(y)}$.

Proof.

The expression for $F_0(x)$ can be written as

$$F_0(x) = \sum_{i=1}^{n} \sum_{k=1}^{T_0} y_{ik} \left(\frac{c_{0ik} \prod_{j=1}^{m} x_{ij}^{\alpha_{0ikj}}}{y_{ik}} \right) \tag{5}$$

We can apply the weighted A.M.≥G.M. inequality to this new expression for F0(x) and obtain

$$\left(\frac{c_{0ik} \prod_{j=1}^{m} x_{ij}^{\alpha_{0ikj}}}{\sum_{i=1}^{n} \sum_{k=1}^{T_0} y_{ik}} \right)^{\sum_{i=1}^{n} \sum_{k=1}^{T_0} \delta_{ik}} \geq \prod_{i=1}^{n} \prod_{k=1}^{T_0} \left(\frac{c_{0ik} \prod_{j=1}^{m} x_{ij}^{\alpha_{0ikj}}}{y_{ik}} \right)^{\delta_{ik}}$$

or

$$\left(\frac{F_0(x)}{n} \right)^{n} \geq \prod_{i=1}^{n} \left(\frac{C_{0ik}}{y_{ik}} \right)^{y_{ik}} \prod_{j=1}^{m} x_{ij}^{\sum_{k=1}^{T_0} \alpha_{0ikj} y_{ik}}$$

using normality condition

$$= \prod_{i=1}^{n} \prod_{k=1}^{T_0} \left(\frac{C_0 ik}{y_{ik}} \right)^{\delta_{ik}} \prod_{j=1}^{m} x_{ij}^{\sum_{k=1}^{T_0} \alpha_{0ikj} y_{ik}} \tag{6}$$

Again, $F_r(x)$ can be written as

$$g_r(x) = \sum_{i=1}^{n} \sum_{T_{r-1}+1}^{T_r} y_{ik} \left(\frac{c_{rik} \prod_{j=1}^{m} x_{ij}^{\alpha_{rikj}}}{y_{ik}} \right) \tag{7}$$

Applying the weighted A.M.≥G.M. inequality in (7), we have

$$\left(\frac{F_r(x)}{\sum_{i=1}^{n} \delta_{ik}} \right)^{\sum_{i=1}^{n} y_{ik}} \geq \prod_{i=1}^{n} \prod_{k=T_{r-1}+1}^{T_r} \left(\frac{c_{rik}}{\prod_{j=1}^{m}} x_{ij}^{\alpha_{rikj}} y_{ik} \right)^{y_{ik}}$$

and

$$(F_r(x))^{\sum_{i=1}^{n} y_{ik}} \geq \prod_{i=1}^{n} \prod_{k=T_{r-1}+1}^{T_r} \left(\frac{C_{rik}}{y_{ik}}\right)^{y_{ik}} \prod_{j=1}^{m} x_{ij}^{\sum_{k=T_{r-1}+1}^{T_r} \alpha_{rikj} y_{ik}} \left(\sum_{s=1}^{n} y_{sk}\right)^{y_{ik}}$$

$(r = 1, 2, \ldots, l)$. Using $1 \geq (F_r(x))^{\sum_{i=1}^{n} y_{ik}} (r = 1, 2, \ldots, l)$ (since $F_r(x) \leq 1 (r = 1, 2, \ldots, l)$),

$$1 \geq \prod_{i=1}^{n} \prod_{k=T_{r-1}+1}^{T_r} \left(\frac{C_{rik}}{\delta_{ik}}\right)^{y_{ik}} \left(\sum_{s=1}^{n} y_{sk}\right)^{y_{ik}} \prod_{j=1}^{m} x_{ij}^{\sum_{k=T_{r-1}+1}^{T_r} \alpha_{rikj} y_{ik}} \tag{8}$$

Multiplying (6) and (8), we have

$$\left(\frac{F_0(x)}{n}\right)^n \geq \prod_{i=1}^{n} \prod_{k=T_{r-1}+1}^{T_r} \left(\frac{C_{0ik}}{y_{ik}}\right)^{y_{ik}} \left(\sum_{s=1}^{n} y_{sk}\right)^{y_{ik}} \prod_{j=1}^{m} x_{ij}^{\sum_{k=1}^{T_0} \alpha_{0ikj} \delta_{ik}} \prod_{j=1}^{m} x_{ij}^{\sum_{k=1}^{T_0} \alpha_{0ikj} y_{ik} + \sum_{r=1}^{l} \sum_{k=T_{r-1}+1}^{T_r} \alpha_{rikj} y_{ik}} \tag{9}$$

(r=0,1,2,⋯,l). Using orthogonality conditions, the inequality (9) becomes

$$\left(\frac{F_0(x)}{n}\right)^n \geq \prod_{i=1}^{n} \prod_{k=1}^{T_r} \left(\frac{C_{ik}}{y_{ik}}\right)^{\delta_{ik}} \left(\sum_{s=1}^{n} y_{sk}\right)^{y_{ik}}, (r = 0, 1, 2, \ldots, l)$$

i.e., $F_0(x) \geq n \sqrt[n]{F(y)}$. This completes the proof.

Theorem 2.

Suppose that the constraint PGP (2) is super-consistent and that x^* *is a solution for GP. Then, the corresponding DP (4) is consistent and has a solution* δ^* *that satisfies*

$$F_0(x^*) = n \sqrt[n]{F(y^*)}$$

and

$$y_{ik}^* = \begin{cases} \frac{u_{ik}(x^*)}{g_0(x^*)}, & (i = 1, 2, \ldots, n; k = 1, 2, \ldots, T_0) \\ \lambda_{ir}(y^*) u_{ik}(x^*) & (i = 1, 2, \ldots, n; k = T_{r-1} + 1, 2, \ldots, T_r; r = 1, 2, \ldots, l) \end{cases}$$

Proof.

Since GP is super-consistent, so is the associated CGP. Furthermore, since GP has a solution $x^* = (x_{i1}^*, x_{i2}^*, \ldots, x_{ij}^*)$, the associated GP has a solution $p^* = (p_{i1}^*, p_{i2}^*, \ldots, p_{ij}^*)$ given by $p_{ij}^* = \ln x_{ij}^*$.

According to the Karush–Kuhn–Tucker (K-K-T) conditions, there is a vector $\lambda^* = (\lambda_{i1}^*, \ldots, \lambda_{il}^*)$ such that

$$\lambda_{ir}^{*} \geq 0 \tag{10}$$

$$\lambda_{ir}^{*}(h_{ir}(p_{*}) - 1) = 0 \tag{11}$$

$$\frac{\partial h_{i0}(p^{*})}{\partial p_{ij}} + \sum_{r=1}^{l} \lambda_{ir}^{*} \frac{\partial h_{ir}(p^{*})}{\partial p_{ij}} = 0 \tag{12}$$

Because $x_{ij} = e^{p_{ij}}$ for i=1,2,⋯,n, j=1,2,⋯,m, it follows that r=1,2,⋯,l

$$\frac{\partial h_{ir}(p)}{\partial p_{ij}} = \frac{\partial h_{ir}(p)}{\partial x_{ij}} \frac{\partial x_{ij}}{\partial p_{ij}} = \frac{\partial g_{ir}(p)}{\partial x_{ij}} e^{p_{ij}}$$

Therefore, the condition (12) is equivalent to

$$\frac{\partial h_{i0}(p^{*})}{\partial x_{ij}} + \sum_{r=1}^{l} \lambda_{ir}^{*} \frac{\partial h_{ir}(p^{*})}{\partial x_{ij}} = 0 \tag{13}$$

since $e^{pij}>0$ and $x_{ij}>0$. Hence (13) is equivalent to

$$x_{ij}^{*} \frac{\partial h_{i0}(p^{*})}{\partial x_{ij}} + \sum_{r=1}^{l} \lambda_{ir}^{*} x_{ij}^{*} \frac{\partial h_{ir}(p^{*})}{\partial x_{ij}} = 0 \tag{14}$$

Now, the terms of $F_{ir}(p)$ are of the form

$$u_{ir}(p) = c_{rik} \prod_{i=1}^{n} x_{ij}^{\alpha_{rikj}}$$

It is clear that

$$x_{ij}^{*} \frac{\partial h_{i0}(p^{*})}{\partial x_{ij}} = \sum_{k=T_{r-1}+1}^{T_r} \alpha_{rikj}, (i = 1, 2, \ldots, n; j = 1, 2, \ldots, n; r = 1, 2, \ldots, l)$$

Therefore, (14) implies

$$\sum_{k=1}^{T_r} \alpha_{0ikj} u_{ik}(p^{*}) + \sum_{r=1}^{l} \sum_{k=T_{r-1}+1}^{T_r} \lambda_{ir}^{*} u_{ik}(p^{*}) = 0, (i = 1, 2, \ldots, n; j = 1, 2, \ldots, n) \tag{15}$$

If we divide the last equation by

$$F_{i0}(p^{*}) = \sum_{k=1}^{T_0} u_{ik} p^{*}$$

we obtain

$$\frac{\sum_{k=1}^{T_r} \alpha_{0ikj} u_{ik}(p^*)}{F_{i0}(p^*)} + \frac{\sum_{r=1}^{l} \sum_{k=T_{r-1}+1}^{T_r} \lambda_{ir}^* u_{ik}(x^*)}{F_{i0}(p^*)} = 0$$

Define the vector y_{ik}^* by

$$y_{ik}^* = \begin{cases} \frac{u_{ik}(p^*)}{F_{i0}(p^*)}, & (i = 1,2,\ldots,n; k = 1,2,\ldots,T_0) \\ \frac{\lambda_{ir} u_{ik}(p^*)}{F_{i0}(p^*)} & (i = 1,2,\ldots,n; k = T_{r-1}+1, 2,\ldots,T_r; r = 1,2,\ldots,l) \end{cases}$$

Note that $y_{ik}^*>0(i=1,2,\cdots,n;k=1,2,\cdots,T_0)$ and $r\geq 1$, either $y_{ik}^*>0$ for all k with $T_{r-1}+1\leq k\leq T_r$ or $y_{ik}^*=0$ for all k with $T_{r-1}+1\leq k\leq T_r$; according to the corresponding Karush–Kuhn–Tucker multipliers $y_{ir}^*,(i=1,2,\cdots,n;r=1,2,\cdots,l)$ is positive or zero.

Furthermore, observe that vector y^* satisfies all of the m exponent constraint equations in DP, as well as the constraint

$$\sum_{k=1}^{T_r} y_{ik}^* = \sum_{k=1}^{T_0} \frac{u_{ik}(p^*)}{F_{i0}(p^*)} = \frac{F_{i0}(p^*)}{F_{i0}(p^*)} = 1$$

Therefore, $y^* = (y_{i1},\ldots,y_{iT_0}^*)$ is a feasible vector for DP. Hence DP is constrained.

The Karush–Kuhn–Tucker multipliers λ_{ir}^* are related to the corresponding $\lambda_{ir}(y^*)$ DP as follows:

$$\lambda_{ir}(y_*) = \sum_{k=1}^{T_r} y_{ik}^* = \sum_{k=1}^{T_0} \lambda_{ir}^* \frac{u_{ik}(p^*)}{g_{i0}(p^*)} = \lambda_{ir}^* \frac{F_{i0}(p^*)}{F_{i0}(p^*)}, (i = 1,2,\ldots,n; r = 1,2,\ldots,l)$$

The Karush–Kuhn–Tucker condition (11) becomes

$$\lambda_{ir}^*(F_{ir}(p^*) - 1) = 0 \tag{16}$$

Therefore, we obtain

$$\lambda_{ir}^* F_{ir}(p^*) = \lambda_{ir}^*$$

Therefore, for $r=1,2,\cdots,l$ and $k=T_{r-1}+1,\cdots,T_r$, we see that

$$y_{ik}^* = \frac{\lambda_{ir}^* u_{ik}(p^*)}{F_{i0}(p^*)} = \frac{\lambda_{ir}^* F_{ir}(p^*) u_{ik}(p^*)}{F_{i0}(p^*)} = \lambda_{ir}(y^*) u_{ik}(p^*) \tag{17}$$

The fact that δ^* is a feasible for DP and x^* is a feasible for GP implies that

$$F_0(p^*) = n\sqrt[n]{F(y^*)}$$

because of the primal-dual inequality.

Moreover, the values of y_{ik}^* (i=1.2.⋯,n;r=1,2,⋯,l;k=1,2,⋯,T_{r-l}+1,⋯, T_r) are precisely those that force equality in the arithmetic-geometric mean inequalities that where used to obtain the duality inequality. Finally, Equation (17) shows that either $F_{ir}(p^*)=1$ or $y_{ir}^*=0$(i=1,2,⋯,n;r=1,2,⋯,l). This means that the value of y_{ik}^* actually forces equality in the primal-dual inequality. This completes the proof.

Geometric Programming Problem under Varying Parameters

In reality, optimization problems may contain uncertainty among the parameters that cannot be ignored. Due to the existence of uncertainty among parameters in the real world, many researchers have investigated the problem of decision-making in a fuzzy environment and management science. Different real-life problems inherently involve uncertainty in the parameters' values. In this case, the decision-makers are not able to provide fixed/exact values of the respective parameters. However, depending on some previous experience or knowledge, the decision-makers may furnish some estimated/most likely values of the parameters that lead to vagueness or ambiguousness. The inconsistent, inappropriate, inaccurate, indeterminate knowledge and lack of information result in vague and ambiguous situations. Thus, the parameters are not precise in such cases. Briefly, one can differentiate between stochastic and fuzzy techniques for tackling the uncertain parameters. Uncertainty arises due to randomness, which can be tackled by using stochastic techniques, whereas the fuzzy approaches can be applied when uncertainty arises due to vagueness.

Various interactive and effective algorithms are investigated for solving the GPP when the RHS in the constraint is known exactly. However, many applications of geometric programming are engineering design problems in which some of the deterministic parameters in the RHS are defined in an estimated interval of actual values. There are also many cases when the RHS may not be depicted in a precise manner. For example, in the machining economics model, the tool life may fluctuate due to different machining

operations and conditions. In this proposed GPP, uncertainty present in the data us varying between some specified intervals that differ from both types of the above-discussed uncertainties. The mathematical model of the GPP under varying parameters can be represented as follows (18):

Proposed Model

$$\begin{aligned} \mathrm{F} = \underset{x}{\text{Minimize}} \quad & \sum_{k=1}^{l_0} c_{0k} \prod_{j=1}^{n} x_j^{\alpha_{0kj}} \\ \text{s. x.} \quad & \\ & \sum_{k=1}^{l_i} c_{ik} \prod_{j=1}^{n} x_j^{\beta_{ikj}} \leq \tilde{B}_i, \quad \forall \ i = 1, 2, \ldots, m, \\ & x_j \geq 0, \qquad \forall \ j = 1, 2, \ldots, n. \end{aligned} \tag{18}$$

where $\tilde{B}_i \in [\underline{B_i}, \overline{B_i}], \forall\, i = 1, 2, \ldots, m$. The geometric programming problem (18) represents the proposed geometric programming model under varying parameters $(\tilde{B}_i)$ that are allowed to vary between some specified bounded intervals, i.e., lower $(\underline{B_i})$ and upper $(\overline{B_i})$ bounds, respectively.

Proposed Bounded Solution Method for Geometric Programming Problem

Intuitively, when the input values are varying within some specified intervals, then it is obvious to have the varying or fluctuating output as well while solving the problems. Hence, the value of the objective function can be determined in a specified interval according to the varying parameters. In this paper, we developed a bounded solution scheme to obtain the lower and upper bound of the geometric programming problems under varying parameters. The GPP (18) inherently involves variation among the RHS parameters. The following consideration is taken into account while proposing the bounded solution method.

Suppose that $S = \{\tilde{B}_i \,|\, \underline{B_i} \leq \tilde{B}_i \leq \overline{B_i}, \quad \forall \ i = 1, 2, \ldots, m\}$ is a set of varying parameters defined between the fixed intervals. Now, for each $\tilde{B}_i \in S$, we define $\tilde{F}(\tilde{B}_i)$ as the objective function value of geometric programming problem (18) under the set of given constraints. Assume that $\underline{F}$ *and* $\overline{F}$ is the minimum and maximum value of $\tilde{F}(\tilde{B}_i)$ defined on S, respectively. Therefore, mathematically, it can be expressed as follows:

$$\underline{F} = \text{Minimum}\,\{\tilde{F}(\tilde{B}_i) | \tilde{B}_i \in S\} \tag{19}$$

$$\overline{F} = \text{Maximum}\,\{\tilde{F}(\tilde{B}_i) | \tilde{B}_i \in S\} \tag{20}$$

With the aid of Equations (19) and (20), we can elicit the corresponding pair of two-level mathematical programming problems as follows:

$$\begin{aligned}
&\underline{F} = \underset{(\tilde{B}_i)\in S}{\text{Minimize}} \quad \underset{x}{\text{Minimize}} \quad \sum_{k=1}^{l_0} c_{0k} \prod_{j=1}^{n} x_j^{\alpha_{0kj}} \\
&\text{s. x.} \\
&\qquad \sum_{k=1}^{l_i} c_{ik} \prod_{j=1}^{n} x_j^{\beta_{ikj}} \le \tilde{B}_i, \ \forall\ i = 1,2,\ldots,m, \\
&\qquad x_j \ge 0, \qquad \forall\ j = 1,2,\ldots,n.
\end{aligned} \tag{21}$$

and

$$\begin{aligned}
&\overline{F} = \underset{(\tilde{B}_i)\in S}{\text{Maximize}} \quad \underset{x}{\text{Minimize}} \quad \sum_{k=1}^{l_0} c_{0k} \prod_{j=1}^{n} x_j^{\alpha_{0kj}} \\
&\text{s. x.} \\
&\qquad \sum_{k=1}^{l_i} c_{ik} \prod_{j=1}^{n} x_j^{\beta_{ikj}} \le \tilde{B}_i, \ \forall\ i = 1,2,\ldots,m, \\
&\qquad x_j \ge 0, \qquad \forall\ j = 1,2,\ldots,n.
\end{aligned} \tag{22}$$

The above problems (21) and (22) represent the two-level geometric programming problems under varying parameters. Since Problem (21) reveals the minimum of the best possible values on S, it would be justifiable to insert the constraints of the outer level into the inner level to simplify the two-level mathematical programming problems into the single-level mathematical programming problem, which can be presented as follows (23):

$$\begin{aligned}
&\underline{F} = \underset{x}{\text{Minimize}} \quad \sum_{k=1}^{l_0} c_{0k} \prod_{j=1}^{n} x_j^{\alpha_{0kj}} \\
&\text{s. x.} \\
&\qquad \sum_{k=1}^{l_i} \frac{c_{ik}}{\tilde{B}_i} \prod_{j=1}^{n} x_j^{\beta_{ikj}} \le 1, \ \forall\ i = 1,2,\ldots,m, \\
&\qquad x_j \ge 0, \qquad \forall\ j = 1,2,\ldots,n, \\
&\qquad \underline{B_i} \le \tilde{B}_i \le \overline{B_i}, \ \tilde{B}_i \in [\underline{B_i}, \overline{B_i}], \ \forall\ i = 1,2,\ldots,m.
\end{aligned} \tag{23}$$

However, in Problem (23), the value of x_j is not known. Thus, it is necessary to obtain the dual of Problem (23), which can be stated as follows (24):

Model A

$$\begin{aligned}
&\underline{F} = \underset{y}{\text{Maximize}} \quad \prod_{k=1}^{l_0} \left(\frac{c_{0k}}{y_{0k}} \right)^{y_{0k}} \prod_{i=1}^{m} \prod_{k=1}^{l_0} \left(\frac{c_{ik} y_{i0}}{\overline{B}_i y_{ik}} \right)^{y_{ik}} \\
&\text{s. x.} \\
&\qquad \sum_{k=1}^{l_i} y_{0k} = 1, \\
&\qquad \sum_{k=1}^{l_i} \alpha_{0kj} y_{0k} + \sum_{i=1}^{m} \sum_{k=1}^{l_i} \beta_{ikj} y_{ik} = 0, \\
&\qquad y_{ik} \ge 0, \qquad \forall\ k = 1,2,\ldots,l_0, \\
&\qquad \underline{B_i} \le \tilde{B}_i \le \overline{B_i}, \ \tilde{B}_i \in [\underline{B_i}, \overline{B_i}], \ \forall\ i = 1,2,\ldots,m.
\end{aligned} \tag{24}$$

Finally, Model A is a nonlinear programming problem and can be solved by using some optimizing software.

The problem (22) would give the maximum value among the best possible objective values over all decision variables. In order to find the upper bound of the geometric programming problem, the dual of the inner

problem of Problem (22) must be obtained with the fact that in the geometric programming problem, the primal problem and the dual problem have the same objective value. By using the strong duality theory of the geometric programming problem, the dual of inner problem (22) is transformed into a maximization-type problem to be similar to the maximization type of outer problem (22). Hence, the problem (22) can be re-expressed as follows:

$$\begin{aligned}
&\overline{F} = \underset{(\tilde{B}_i)\in S}{\text{Maximize}} \quad \underset{y}{\text{Maximize}} \quad \prod_{k=1}^{l_0}\left(\frac{c_{0k}}{y_{0k}}\right)^{y_{0k}} \prod_{i=1}^{m}\prod_{k=1}^{l_i}\left(\frac{c_{ik}y_{i0}}{\tilde{B}_i y_{ik}}\right)^{y_{ik}} \\
&\text{s. x.} \\
&\qquad \sum_{k=1}^{l_0} y_{0k} = 1, \quad i = 1,2,\ldots,m, \\
&\qquad \sum_{k=1}^{l_0} \alpha_{0kj}y_{0k} + \sum_{i=1}^{m}\sum_{k=1}^{l_i} \beta_{ikj}y_{ik} = 0, \quad j = 1,2,\ldots,n, \\
&\qquad y_{ik} \geq 0, \ \forall \ i,k, \\
&\qquad \tilde{B}_i \in [\underline{B_i}, \overline{B_i}], \ \forall \ i = 1,2,\ldots,m.
\end{aligned} \tag{25}$$

Since Problem (25) represents the maximum of the best possible values on S, so it would be justifiable to insert the constraints of the outer level into the inner level to simplify the two-level mathematical programming problems into the single-level mathematical programming problem (26), which can be stated as follows:

Model B

$$\begin{aligned}
&\overline{F} = \underset{y}{\text{Maximize}} \quad \prod_{k=1}^{l_0}\left(\frac{c_{0k}}{y_{0k}}\right)^{y_{0k}} \prod_{i=1}^{m}\prod_{k=1}^{l_i}\left(\frac{c_{ik}y_{i0}}{\underline{B}_i y_{ik}}\right)^{y_{ik}} \\
&\text{s. x.} \\
&\qquad \sum_{k=1}^{l_0} y_{0k} = 1, \quad i = 1,2,\ldots,m, \\
&\qquad \sum_{k=1}^{l_0} \alpha_{0kj}y_{0k} + \sum_{i=1}^{m}\sum_{k=1}^{l_i} \beta_{ikj}y_{ik} = 0, \quad j = 1,2,\ldots,n, \\
&\qquad y_{ik} \geq 0, \ \forall \ i,k, \\
&\qquad \underline{B_i} \leq \tilde{B}_i \leq \overline{B_i}, \ \tilde{B}_i \in [\underline{B_i}, \overline{B_i}], \ \forall \ i = 1,2,\ldots,m.
\end{aligned} \tag{26}$$

Model B is a nonlinear constrained programming problem and can be solved by using several efficient methods. Thus, Model A and Model B provide the lower and upper bound to the geometric programming problem under varying parameters and calculate the objective value directly without violating the optimal range of the objective values where they should lie. A comprehensive study about the relationship between the globally optimal cost and the optimal dual value can be found in [4].

COMPUTATIONAL STUDY

The proposed bounded solution method for the geometric programming problem under varying parameters was implemented in different real-life applications. The following two examples were adopted from engineering

problems. Furthermore, it was also applied to the system reliability optimization problem. All the numerical illustrations were coded in AMPL and solved using the optimizing solver CONOPT through the NEOS server version 5.0 on-line facility provided by Wisconsin Institutes for Discovery at the University of Wisconsin in Madison for solving optimization problems; see (Server [28]).

Example 1

([4]). *A cofferdam is an engineering design optimization problem. It is a prominent structure to attach a trivial submerged area to permit building a permanent structure on an allocated site. The cofferdam function is elicited in a random environment by transitions in surrounding water levels. The architecture designs a dam of height* x_1, *length* x_2 *breadth* x_3, *and total required perimeter* x_4 *and intends to estimate the most promising total cost for making decisions. The RHS parameters can be of any simplex dimensions such as area, volume, etc., which is not quite certain. These are no longer crisp or deterministic, but the allowable lower and upper bounds over each area/volume are determined in the closed interval. Thus, the use of varying parameters is quite worthwhile and the decision under such variation will be helpful in determining the range of optimal outcomes. Figure 1 depicts an illustrative example of a cofferdam.*

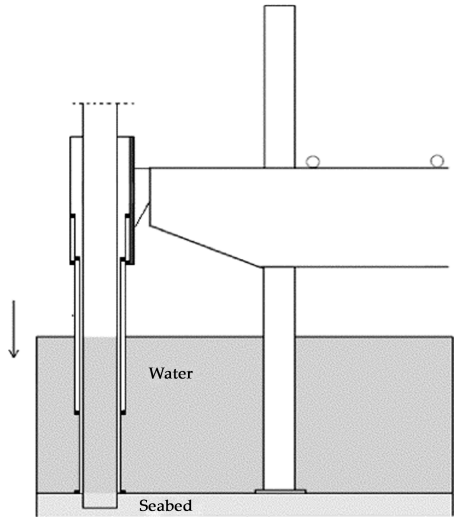


Figure 1. Illustrative figure of a cofferdam.

Thus, the equivalent mathematical programming problem with varying parameters is given as follows (27):

$$\begin{aligned} \mathrm{F} = \underset{x}{\text{Minimize}} \quad & 2x_1^{-0.9}x_2^{-1.5}x_3^{-1}x_4^{-1.6} + 4x_1^{-1}x_2^{-1}x_3^{-0.1}x_4^{-1} \\ \text{s. x.} \quad & \\ & 2x_1^{-2}x_2^{-1}x_3^{2}x_4 + 1.6x_1x_3x_4^{2} \le \tilde{B}_1 \\ & 1.9x_1^{2}x_2^{1.4}x_3x_4 + 3.1x_1^{2.2}x_4 \le \tilde{B}_2 \\ & x_1, x_2, x_3, x_4 \ge 0. \end{aligned} \tag{27}$$

where $\tilde{B}_1 \in (3, 3.2)$ and $\tilde{B}_2 \in (2, 2.4)$ are the varying parameters. Since all the parameters are crisp except the RHS, then Problems (23) and (26) can be utilized to obtain the upper and lower bounds of the objective values in Problem (27). According to Problems (23) and (26), the formulations of the upper and lower bounds for Problem (27) can be presented as follows:

Model A

$$\begin{aligned} \underline{F} = \underset{y}{\text{Minimize}} \quad & \left(\frac{2}{y_{01}}\right)^{y_{01}}\left(\frac{4}{y_{02}}\right)^{y_{02}}\left(\frac{2y_{10}}{\tilde{B}_1 y_{11}}\right)^{y_{11}}\left(\frac{1.6y_{10}}{\tilde{B}_1 y_{12}}\right)^{y_{12}}\left(\frac{1.9y_{20}}{\tilde{B}_2 y_{21}}\right)^{y_{21}}\left(\frac{3.1y_{20}}{\tilde{B}_2 y_{22}}\right)^{y_{22}} \\ \text{s. x.} \quad & \\ & y_{01} + y_{02} = 1, \\ & y_{01} + y_{02} - 2y_{11} + y_{12} + 2y_{21} + 2.2y_{22} = 0, \\ & y_{01} - y_{02} - y_{11} + 1.4y_{21} = 0, \\ & -y_{01} + y_{02} + 2y_{11} + y_{12} + y_{21} = 0, \\ & y_{01} - y_{02} + y_{11} + 2y_{12} + y_{21} + y_{22} = 0, \\ & y_{01}, y_{02}, y_{11}, y_{21}, y_{12}, y_{22} \ge 0 \\ & 3 \le \tilde{B}_1 \le 3.2, \quad 2 \le \tilde{B}_2 \le 2.4, \quad \forall \ i = 1, 2, \ldots, m. \end{aligned} \tag{28}$$

and

Model B

$$\begin{aligned} \overline{F} = \underset{y}{\text{Maximize}} \quad & \left(\frac{2}{y_{01}}\right)^{y_{01}}\left(\frac{4}{y_{02}}\right)^{y_{02}}\left(\frac{2y_{10}}{\tilde{B}_1 y_{11}}\right)^{y_{11}}\left(\frac{1.6y_{10}}{\tilde{B}_1 y_{12}}\right)^{y_{12}}\left(\frac{1.9y_{20}}{\tilde{B}_2 y_{21}}\right)^{y_{21}}\left(\frac{3.1y_{20}}{\tilde{B}_2 y_{22}}\right)^{y_{22}} \\ \text{s. x.} \quad & \\ & y_{01} + y_{02} = 1, \\ & y_{01} + y_{02} - 2y_{11} + y_{12} + 2y_{21} + 2.2y_{22} = 0, \\ & y_{01} - y_{02} - y_{11} + 1.4y_{21} = 0, \\ & -y_{01} + y_{02} + 2y_{11} + y_{12} + y_{21} = 0, \\ & y_{01} - y_{02} + y_{11} + 2y_{12} + y_{21} + y_{22} = 0, \\ & y_{01}, y_{02}, y_{11}, y_{21}, y_{12}, y_{22} \ge 0 \\ & 3 \le \tilde{B}_1 \le 3.2, \quad 2 \le \tilde{B}_2 \le 2.4, \quad \forall \ i = 1, 2, \ldots, m. \end{aligned} \tag{29}$$

Thus, Problems (28) and (29) are the required upper and lower bounds for the geometric programming problem (27). Upon solving the problem at zero degree of difficulty, the upper and lower bounds for the objective functions are obtained as $\overline{F}$ =8.5429 and $\underline{F}$ =4.9271, respectively. However,

the objective values at $\tilde{B}_1 = 3.2$ and $\tilde{B}_2 = 2.4$ are found to be $\tilde{F}$=5.6212. Therefore, the obtain objective function lies between the range of the upper and lower bounds, which shows that it is justified to reduce the objective values at the maximum RHS under variations.

Example 2

([4]). *This illustration belongs to a design problem of a journal bearing. The texture of the journal bearing is an inverse problem, where the eccentricity ratio and attitude angle are obtained for a defined load and speed. The engineers may not have experience in modeling the structure of this new type of journal bearing. The volume of steel, the thickness of the intermediate layer and nickel barrier, and the dimension of the plated overlay of the journal bearing are assumed to be unknown. Thus, the values of these parameters have been depicted between some specified closed intervals and taken in the form of lower and upper bounds, respectively. Hence, the varying solution outcomes will also come by ensuring the optimal objectives between corresponding intervals. Figure 2 represents the structure of the journal bearing used in this example. Hence, some parameters of the model are approximately known and are estimated by the engineers. Suppose that* x_1 *is the radial clearance,* x_2 *the fluid force,* x_3 *the diameter,* x_4 *the rotation speed, and* x_5 *the length-to-diameter ratio.*

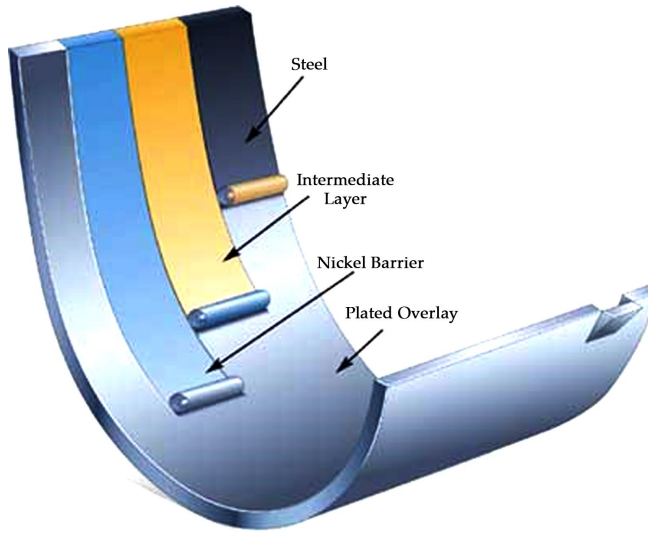


Figure 2: Illustrative figure of a journal bearing.

The following mathematical programming formulation can depict the design problem as a geometric programming problem (30):

$$\begin{aligned} \mathrm{F} = \underset{x}{\text{Minimize}} \quad & 0.5x_1^2x_2x_4x_5 + 1.1x_1^{-1}x_2^{-1}x_3^{-1} \\ \text{s. x.} \quad & \\ & 8.4x_1x_2^{-1}x_3^{-1}x_4^{-1}x_5 \le \tilde{B}_1 \\ & 0.5x_2x_3 + x_1x_4^{-1}x_5^{-1} + 1.6x_3x_4 \le \tilde{B}_2 \\ & x_1, x_2, x_3, x_4, x_5 \ge 0. \end{aligned} \tag{30}$$

where $\tilde{B}_1 \in (4, 4.2)$ and $\tilde{B}_2 \in (0, 1)$ are the varying parameters. Since all the parameters are crisp except the RHS, then Problems (23) and (26) can be utilized to obtain the upper and lower bound of the objective values in Problem (30). According to the problems (23) and (26), the formulations of the upper and lower bounds for Problem (30) can be presented as follows (31):

Model A

$$\begin{aligned} \underline{F} = \underset{y}{\text{Minimize}} \quad & \left(\frac{0.5}{y_{01}}\right)^{y_{01}}\left(\frac{1.1}{y_{02}}\right)^{y_{02}}\left(\frac{8.4}{\tilde{B}_1}\right)^{y_{11}}\left(\frac{8.4}{\tilde{B}_2}\right)^{y_{11}}\left(\frac{0.5y_{20}}{y_{21}}\right)^{y_{21}}\left(\frac{y_{20}}{y_{22}}\right)^{y_{22}}\left(\frac{1.6y_{20}}{y_{23}}\right)^{y_{23}} \\ \text{s. x.} \quad & \\ & y_{01} + y_{02} = 1, \\ & 2y_{01} - y_{02} + y_{11} + y_{22} = 0, \\ & y_{01} - y_{02} - y_{11} + y_{21} = 0, \\ & -y_{02} - y_{11} + y_{21} + y_{23} = 0, \\ & y_{01} - y_{11} - y_{22} + y_{23} = 0, \\ & y_{01} + y_{11} - y_{22} = 0, \\ & y_{01}, y_{02}, y_{11}, y_{21}, y_{22}, y_{23} \ge 0 \\ & 4 \le \tilde{B}_1 \le 4.2, \ \ 0 \le \tilde{B}_2 \le 1, \ \ \forall \ i = 1, 2, \ldots, m. \end{aligned} \tag{31}$$

where $y_{21}+y_{22}+y_{23}=y_{20}$ and the upper bound can be stated as follows (32):

Model B

$$\begin{aligned} \overline{F} = \underset{y}{\text{Maximize}} \quad & \left(\frac{0.5}{y_{01}}\right)^{y_{01}}\left(\frac{1.1}{y_{02}}\right)^{y_{02}}\left(\frac{8.4}{\tilde{B}_1}\right)^{y_{11}}\left(\frac{8.4}{\tilde{B}_2}\right)^{y_{11}}\left(\frac{0.5y_{20}}{y_{21}}\right)^{y_{21}}\left(\frac{y_{20}}{y_{22}}\right)^{y_{22}}\left(\frac{1.6y_{20}}{y_{23}}\right)^{y_{23}} \\ \text{s. x.} \quad & \\ & y_{01} + y_{02} = 1, \\ & 2y_{01} - y_{02} + y_{11} + y_{22} = 0, \\ & y_{01} - y_{02} - y_{11} + y_{21} = 0, \\ & -y_{02} - y_{11} + y_{21} + y_{23} = 0, \\ & y_{01} - y_{11} - y_{22} + y_{23} = 0, \\ & y_{01} + y_{11} - y_{22} = 0, \\ & y_{01}, y_{02}, y_{11}, y_{21}, y_{22}, y_{23} \ge 0 \\ & 4 \le \tilde{B}_1 \le 4.2, \ \ 0 \le \tilde{B}_2 \le 1, \ \ \forall \ i = 1, 2, \ldots, m. \end{aligned} \tag{32}$$

where $y_{21}+y_{22}+y_{23}=y_{20}$.

The above Problems (31) and (32) provide the required upper and lower bound for the problem (30). Both of these problems are concave programming problems with linear constraints. Upon solving the problem

at zero degree of difficulty, the upper and lower bounds for the objective functions are obtained as $\overline{F}$ =4.314 and $\underline{F}$=3.045, respectively. However, the objective values at $\tilde{B}_1 = 4.2$ and $\tilde{B}_2 = 1$ are found to be $\tilde{F}$=3.561. Therefore, the obtain objective function lies between the range of the upper and lower bounds, which shows that it is justified to reduce the objective values at the maximum RHS under variations.

Application to System Reliability Optimization

Assume system reliability having *n* components connected in series. Suppose r_i(i=1,2,⋯,n) represents the individual reliability of the *i*-th component of the system. Similarly, $R_s(r_1,r_2,\cdots,r_n)$ is the reliability of the whole series system. Consequently, $C_s(r_1,r_2,\cdots,r_n)$ depicts the total cost of *n* components associated with the system reliability. It seldom happens that the system reliability is maximized when the cost of the associated system is exactly known; however, some varying cost may make it easier to execute the smooth function of the framework. The obtained lower and upper bounds on the cost objective function will ensure the variation in total cost associated with the system and help with allocating the budget for maintenance or renovation, etc. In the same manner, minimizing the system cost under the varying reliability of the whole system would be quite a worthwhile task. The minimization of the system cost without affecting the system reliability is much needed to ensure the longer performance of the components. Thus, we considered that the system reliability is varying between some specified intervals and bounded by upper and lower bounds. This situation is quite common due to uncertainty in the failure of any components. In real-life scenarios, the minimization of the total system cost by maintaining the system reliability would be a more prominent modeling texture of the reliability optimization problems (see [17,29,30]). Therefore, the mathematical model for the minimization of system cost under varying system reliability takes the form of the geometric programming problem and can be represented as follows (33):

$$
\begin{aligned}
\mathrm{F} = \text{Minimize} \quad & C_s\,(r_1, r_2, \ldots, r_n) = +\textstyle\sum_{i=1}^{n} C_i r_i^{\alpha_i} \\
\text{s. x.} \quad & \\
& (r_1 \times r_2 \times \cdots \times r_n) = \textstyle\prod_{i=1}^{n} r_i \geq \tilde{R}_s \\
& \underline{R}_s \leq \tilde{R}_s \leq \overline{R}_s \\
& 0 \leq r_i \leq 1 \quad \forall\ i = 1, 2, \ldots, n.
\end{aligned}
\tag{33}
$$

where α_i is the acceptable tolerance linked with the i-th component. We considered the three components connected in series, and the relevant data are summarized in Table 2.

Table 2: Input data for the system reliability optimization problem

C_1	C_2	C_3	α_1	α_2	α_3	$\underline{R}_s \leq \tilde{R}_s \leq \overline{R}_s$
150	210	270	20	15	10	$0.6358 \leq \tilde{R}_s \leq 0.9776$

Since all the parameters are crisp except the system reliability, then Problems (23) and (26) can be utilized to obtain the upper and lower bounds of the objective values in Problem (33). According to the problems (23) and (26), the formulations of the upper and lower bounds for Problem (33) can be presented as follows (34):

Model A

$$\underline{F} = \underset{y}{\text{Minimize}} \quad \left(\frac{150}{y_{01}}\right)^{y_{01}} \left(\frac{210}{y_{02}}\right)^{y_{02}} \left(\frac{270}{y_{03}}\right)^{y_{03}} \left(\frac{1}{\tilde{R}_s y_{11}}\right)^{y_{11}} \left(\frac{1}{\tilde{R}_s y_{12}}\right)^{y_{12}} \left(\frac{1}{\tilde{R}_s y_{13}}\right)^{y_{13}}$$

s. x.

$$\begin{aligned} & y_{01} + y_{02} + y_{03} = 1, \\ & 20y_{01} + 15y_{02} + 10y_{03} + y_{11} + y_{12} + y_{13} = 0, \\ & y_{01} - y_{02} - y_{03} + y_{11} + y_{12} + y_{13} = 0, \\ & y_{01}, y_{02}, y_{03}, y_{11}, y_{12}, y_{13} \geq 0 \\ & 0.6358 \leq \tilde{R}_s \leq 0.9776, \quad \forall \ i = 1, 2, 3. \end{aligned} \tag{34}$$

whereas the upper bound can be stated as follows (35):

Model B

$$\overline{F} = \underset{y}{\text{Maximize}} \quad \left(\frac{150}{y_{01}}\right)^{y_{01}} \left(\frac{210}{y_{02}}\right)^{y_{02}} \left(\frac{270}{y_{03}}\right)^{y_{03}} \left(\frac{1}{\tilde{R}_s y_{11}}\right)^{y_{11}} \left(\frac{1}{\tilde{R}_s y_{12}}\right)^{y_{12}} \left(\frac{1}{\tilde{R}_s y_{13}}\right)^{y_{13}}$$

s. x.

$$\begin{aligned} & y_{01} + y_{02} + y_{03} = 1, \\ & 20y_{01} + 15y_{02} + 10y_{03} + y_{11} + y_{12} + y_{13} = 0, \\ & y_{01} - y_{02} - y_{03} + y_{11} + y_{12} + y_{13} = 0, \\ & y_{01}, y_{02}, y_{03}, y_{11}, y_{12}, y_{13} \geq 0 \\ & 0.6358 \leq \tilde{R}_s \leq 0.9776, \quad \forall \ i = 1, 2, 3. \end{aligned} \tag{35}$$

The above problems (34) and (35) provide the required upper and lower bound for the problem (33). Upon solving the problem at zero degree of difficulty, the upper and lower bounds for the system cost are obtained as $\overline{C_s}$=521.95 and $\underline{C_s}$=216.35, respectively. However, the system cost at $\tilde{R}_s$ =0.88 is found to be $\tilde{C}_s$=351.29. Therefore, the obtained system cost lies between the range of its upper and lower bounds, which shows that it is justified to reduce the system cost at maximum system reliability under variations.

Analyses of Computational Complexity and Discussions

This proposed bounded solution method captures the behavior of varying parameters and provides the interval-based solution of the objective function. Most often, uncertain parameters exist in any form, such as they may take the form of randomness, fuzziness, and any other aspects of uncertainty. The uncertainty among parameters arises due to vagueness being able to be dealt with by using fuzzy approaches, whereas the stochastic technique is applied when the uncertainty involves randomness among the parameters. More precisely, contrary to other uncertain optimization approaches, the developed approach adheres to comparatively less computational complexity in the sense of mathematical computation (e.g., some mathematical calculations are used to derive the crisp or deterministic version of fuzzy or random parameters), and there is no scope for obtaining the deterministic version of the problem for such varying geometric programming problems. The beauty of the proposed method can be highlighted by the fact that sensitivity analyses (post-optimality analysis) doe not need to be performed because the continuous variations among the parametric values directly produce the range of optimal objective functions from the interval parameters. Thus, the propounded solution approach can be the most prominent and efficient decision-maker while dealing with uncertain parameters other than the fuzzy or stochastic form.

The generalization of the conventional geometric programming problem of constant parameters is highlighted for interval parameters. The most prominent and extensive idea is to determine the lower and upper bounds of the range by applying the two-level mathematical programming technique to geometric programming problems. With the aid of a strong duality theorem, the two-level geometric programming problems are converted into a pair of one-level geometric programming problems to implement the computational study. When all the varying parameters degenerate to constant parameters, the two-level geometric programming problem turns into the conventional geometric programming problem. In general, in interval geometric programming problems, it may probably happen that the problem is infeasible for some specified range of varying parameters. Thus, our proposed methods are free from infeasibility and ignore those complexities due to infeasible values. The proposed method obtains the lower and upper bounds of the feasible solutions directly. In addition, the suggested method does not examine the range of values that results in

infeasibility. Furthermore, developing two-level geometric programming problems can determine the lower and upper bounds of the objective values. However, mathematical programming is nonlinear in the case of geometric programming problems, which may be very typical for solving large-scale problems. The comparative study is presented in Table 3.

Table 3: Comparison between the proposed method and traditional methods

Proposed Method	Traditional Methods
(1) Deals with the varying parameters	No scope for dealing with such parameters
(2) No need to consider the fuzzy or random parameters while dealing with uncertainty	It may require the fuzzy or random parameters while dealing with uncertainty
(3) No need to obtain the crisp or deterministic version of uncertain models	It requires the crisp or deterministic version of the uncertain models
(4) Sensitivity or post-optimal analysis is not required for the obtained solutions	Sensitivity or post-optimal analysis can be performed separately
(5) Less computational complexity in terms of mathematical calculations	Comparatively involves more computational complexity in terms of mathematical calculations

The presented work can be described as an empirical case research work by filling the various gaps [8,9,17,19,29] such as instant variation among parameters, two-level mathematical programming, duality theory in the GPP, and the automatic post-optimal analysis metric. In system reliability modeling, various parameters can be taken as varying between some specified intervals. For example, the cofferdam, shaft, journal bearings, etc., are the raw parts that are the building block materials of the various products and machines. This means that the parameters can be taken as uncertain, and using some specified tools, they can be converted into a crisp one. In the literature, the concept of fuzzy and random parameters is available, which deals with the vagueness and randomness in the parameters. However, we have provided an opportunity to define the parameters under the continuous variations bounded by upper and lower limits. Instead of taking the vague and random parameters, one can assume the continuous variations in the parameters' values can be tackled with the two-level mathematical programming techniques discussed in this paper. Additionally, the sensitivity and post-optimality of the obtained solution results are waived off due to the working procedure of the proposed approach.

In the future, a solution method that involves all the parameters under variation in the geometric programming formulation is much required to ensure solvability. The values near the lower and upper bounds have a significantly lower probability of occurrence. If the distributions of varying data are known in the stochastic environment, then the distribution of the

objective function would be obtained, which is more realistic, and the scenario is generated for consequent decision-making. Therefore, this lays down another direction for future research by deriving the distribution of the objective functions based on the distributions of the varying parameters.

CONCLUSIONS

The geometric programming problem is an integrated part of mathematical programming and has real-life applications in many engineering problems such as gravel-box design, bar–truss region texture, system reliability optimization, etc. The concept of varying parameters under the objective functions is discussed with the aim that uncertainty is critically involved and affects engineering problems' formulations directly. The propounded research work is developed and introduces an interval-based solution approach to finding the upper and lower limits on the objective function of the varying parameters. The outer- and inner-level geometric programming problem is transformed into a single-level mathematical programming problem. The outcomes are summarized in the numerical illustrations and observed in the precise interval where they should exist. The system reliability optimization problem also provides evidence of the discussed problem's successful implementation and dynamic solution results. The minimum system cost is obtained at the utmost system reliability, which also falls into the lower and upper bounds of the system costs.

The scope of usual sensitivity analysis is not further required due to the flexible nature of the proposed solution method. The propounded approach allows the abrupt fluctuations among the parameters between given intervals for which bounds over the objective functions are directly obtained. It also makes the computational algorithm easier than other methods by ignoring the uncertain parameters such as fuzzy, stochastic, and other uncertain forms that yield a solution procedure that is comparatively more complex. The developed approach may be extended for future research to stochastic programming, bi-level or multilevel programming, and various engineering problems with real-life applications.

AUTHOR CONTRIBUTIONS

Conceptualization, A.E.S., F.A. and S.A.; methodology, M.A. and S.A.; formal analysis, M.A. and F.A.; writing—original draft preparation, M.A., S.A. and A.E.S.; writing—review and editing, S.A., A.E.S. and F.A.; project ad-

ministration, S.A. and I.A.; funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENTS

The authors are very thankful to the Anonymous Reviewers and Editors for their insightful comments, which made the manuscript clearer and more readable. The authors extend their appreciation to King Saud University for funding this work through the Researchers Supporting Project (RSP-2021/387), King Saud University, Riyadh, Saudi Arabia.

REFERENCES

1. Duffin, R.; Peterson, E.L. Duality theory for geometric programming. *SIAM J. Appl. Math.* 1966, *14*, 1307–1349.
2. Duffin, R.J. Linearizing geometric programs. *SIAM Rev.* 1970, *12*, 211–227.
3. Duffin, R.J.; Peterson, E.L. Reversed geometric programs treated by harmonic means. *Indiana Univ. Math. J.* 1972, *22*, 531–550.
4. Rao, S.S. *Engineering Optimization: Theory and Practice*; John Wiley & Sons: Hoboken, NJ, USA, 2019.
5. Liu, S.T. Fuzzy measures for profit maximization with fuzzy parameters. *J. Comput. Appl. Math.* 2011, *236*, 1333–1342.
6. Lu, T.; Liu, S.T. Fuzzy nonlinear programming approach to the evaluation of manufacturing processes. *Eng. Appl. Artif. Intell.* 2018, *72*, 183–189.
7. Ahmad, F.; Adhami, A.Y. Total cost measures with probabilistic cost function under varying supply and demand in transportation problem. *Opsearch* 2019, *56*, 583–602.
8. Chakraborty, D.; Chatterjee, A.; Aishwaryaprajna. Multi-objective Fuzzy Geometric Programming Problem Using Fuzzy Geometry. In *Trends in Mathematics and Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 123–129.
9. Garg, H.; Rani, M.; Sharma, S.; Vishwakarma, Y. Intuitionistic fuzzy optimization technique for solving multi-objective reliability optimization problems in interval environment. *Expert Syst. Appl.* 2014, *41*, 3157–3167.
10. Islam, S.; Roy, T.K. Modified geometric programming problem and its applications. *J. Appl. Math. Comput.* 2005, *17*, 121–144.
11. Islam, S.; Roy, T.K. A new fuzzy multi-objective programming: Entropy based geometric programming and its application of transportation problems. *Eur. J. Oper. Res.* 2006, *173*, 387–404.
12. Islam, S.; Mandal, W.A. Preliminary Concepts of Geometric Programming (GP) Model. In *Fuzzy Geometric Programming Techniques and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–25.
13. Islam, S.; Mandal, W.A. Geometric Programming Problem Under Uncertainty. In *Fuzzy Geometric Programming Techniques and*

Applications; Springer: Berlin/Heidelberg, Germany, 2019; pp. 287–330.

14. Islam, S.; Mandal, W.A. Fuzzy Unconstrained Geometric Programming Problem. In *Fuzzy Geometric Programming Techniques and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 133–153.
15. Islam, S.; Mandal, W.A. Intuitionistic and Neutrosophic Geometric Programming Problem. In *Fuzzy Geometric Programming Techniques and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 331–355.
16. Khorsandi, A.; Cao, B.Y.; Nasseri, H. A New Method to Optimize the Satisfaction Level of the Decision Maker in Fuzzy Geometric Programming Problems. *Mathematics* 2019, *7*, 464.
17. Mahapatra, G.; Roy, T.K. Fuzzy multi-objective mathematical programming on reliability optimization model. *Appl. Math. Comput.* 2006, *174*, 643–659.
18. Negi, G.; Kumar, A.; Pant, S.; Ram, M. Optimization of complex system reliability using hybrid grey wolf optimizer. *Decis. Mak. Appl. Manag. Eng.* 2021, *4*, 241–256.
19. Roustaee, M.; Kazemi, A. Multi-objective stochastic operation of multi-microgrids constrained to system reliability and clean energy based on energy management system. *Electr. Power Syst. Res.* 2021, *194*, 106970.
20. Zolfaghari, S.; Mousavi, S.M. A novel mathematical programming model for multi-mode project portfolio selection and scheduling with flexible resources and due dates under interval-valued fuzzy random uncertainty. *Expert Syst. Appl.* 2021, *182*, 115207.
21. Sedaghat, N.; Ardakan, M.A. G-mixed: A new strategy for redundant components in reliability optimization problems. *Reliab. Eng. Syst. Saf.* 2021, *216*, 107924.
22. Meng, Z.; Ren, S.; Wang, X.; Zhou, H. System reliability-based design optimization with interval parameters by sequential moving asymptote method. *Struct. Multidiscip. Optim.* 2021, *63*, 1767–1788.
23. Kugele, A.S.H.; Ahmed, W.; Sarkar, B. Geometric programming solution of second degree difficulty for carbon ejection controlled reliable smart production system. *RAIRO Oper. Res.* 2022, *56*, 1013–1029.

24. Son, Y.; Mukherjee, S.; Mallik, R.; Majmunovi'c, B.; Dutta, S.; Johnson, B.; Maksimović, D.; Seo, G.S. Levelized Cost of Energy-Oriented Modular String Inverter Design Optimization for PV Generation System Using Geometric Programming. *IEEE Access* 2022, *10*, 27561–27578.
25. Shen, K.; Yu, W.; Chen, X.; Khosravirad, S.R. Energy Efficient HARQ for Ultrareliability via a Novel Outage Probability Bound and Geometric Programming. *IEEE Trans. Wirel. Commun.* 2022.
26. Rajamony, R.; Wang, S.; Navaratne, R.; Ming, W. Multi-objective design of single-phase differential buck inverters with active power decoupling. *IEEE Open J. Power Electron.* 2022, *3*, 105–114.
27. Singh, G.; Singh, A.K. Optimizing multi-VM migration by allocating transfer and compression rate using geometric programming. *Simul. Model. Pract. Theory* 2021, *106*, 102201.
28. Server, N. State-of-the-Art Solvers for Numerical Optimization. 2016. Available online: https://neos-server.org/neos/ (accessed on 22 June 2022).
29. Kundu, T.; Islam, S. Neutrosophic goal geometric programming problem and its application to multi-objective reliability optimization model. *Int. J. Fuzzy Syst.* 2018, *20*, 1986–1994.
30. Ahmad, F.; Adhami, A.Y. Spherical Fuzzy Linear Programming Problem. In *Decision Making with Spherical Fuzzy Sets*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 455–472.

Chapter

13

A COMPREHENSIVE REVIEW OF ISOGEOMETRIC TOPOLOGY OPTIMIZATION: METHODS, APPLICATIONS AND PROSPECTS

Jie Gao[1,2], Mi Xiao[3] , Yan Zhang[4] , and Liang Gao[3]

[1]Department of Engineering Mechanics, School of Aerospace Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2] Hubei Key Laboratory for Engineering Structural Analysis and Safety Assessment, Huazhong University of Science and Technology, Wuhan 430074, China
[3] The State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[4] School of Machinery and Automation, Wuhan University of Science and Technology, Wuhan 430081, China

ABSTRACT

Topology Optimization (TO) is a powerful numerical technique to determine the optimal material layout in a design domain, which has accepted

Citation: (APA): Gao, J., Xiao, M., Zhang, Y., & Gao, L. (2020). A comprehensive review of isogeometric topology optimization: methods, applications and prospects. Chinese Journal of Mechanical Engineering, 33(1), 1-14. (14 pages).

considerable developments in recent years. The classic Finite Element Method (FEM) is applied to compute the unknown structural responses in TO. However, several numerical deficiencies of the FEM significantly influence the effectiveness and efficiency of TO. In order to eliminate the negative influence of the FEM on TO, IsoGeometric Analysis (IGA) has become a promising alternative due to its unique feature that the Computer-Aided Design (CAD) model and Computer-Aided Engineering (CAE) model can be unified into a same mathematical model. In the paper, the main intention is to provide a comprehensive overview for the developments of Isogeometric Topology Optimization (ITO) in methods and applications. Finally, some prospects for the developments of ITO in the future are also presented.

INTRODUCTION

Structural optimization [1] has attracted considerable attentions among researchers ranging from theoretical research to engineering applications, which aims to solve the optimal design of the load-carrying structures with the reasonable structural features, like the connectivity of holes, the shapes of boundaries. Overall speaking, structural optimization mainly contains three components as far as the design stage, presented in Figure 1, namely the conceptual design stage of Topology Optimization (TO), the basic design stage of shape optimization and the detailed design stage of size optimization. One of them, TO, has been identified as an important but with more challenges sub-discipline, and the main intention of TO is to seek for the optimal material layout with the expected structural performance in a design domain without the prior knowledge subject to several pre-defined constraints [2].

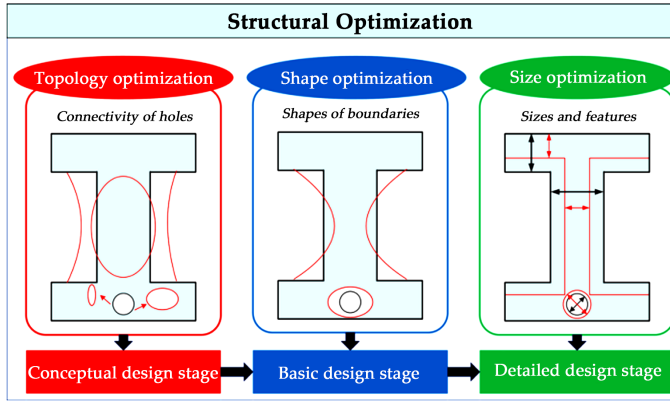


Figure 1: Structural optimization.

As we know, TO originates from a pioneering work [3] that discusses the frame-structures under the limits of economy of materials. Cheng and Olhoff [4, 5] addressed the optimal design of solid elastic plates, which is considered as the seminar work for the structural optimization of continuum structures and attracts a wide of discussions in the last three decades. In 1988, Bendsøe and Kikuchi [6] used the homogenization approach to optimize the structural topology by gradually changing the sizes and orientations of holes in a design domain. After that, TO has accepted a myriad of discussions ranging from the developments of TO methods to the applications of different problems, and the details can refer to some comprehensive reviews of TO [7,8,9,10,11]. Up to now, there are several different topology optimization methods with the unique positive features which have been proposed in recent years.

The developed TO methods can be mainly divided into two branches as far as the representation model of the structural topology, including Material Description Models (MDMs) and Boundary Description Models (BDMs). In the first branch of TO methods, MDMs discrete the design domain to be a series of designable points or elements with the densities, namely the density-based TO methods. The density in each designable point or element determines the non/existence of material at the corresponding location in a design domain. This branch mainly contains the Solid Isotropic Material with Penalization (SIMP) method [12, 13], and the Evolutionary Structural Optimization (ESO) method [14]. However, the second branch of TO methods uses the BDMs to display the structural topology, where a higher-dimensional function in an implicit or explicit form is constructed for the evolvement of topology in the design and structural boundaries are defined by the iso-contour/surface of the function. In this branch, the Level Set Method (LSM) [15,16,17], the phase field method [18, 19], the recently proposed Moving Morphable Components/Voids (MMC/V) method [20,21,22,23] and the Bubble method [24, 25] have been obtained considerable discussions. These developed TO methods have been also applied to address several different numerical problems, like the dynamic optimization [26,27,28], compliant mechanisms [29, 30], stress problems [31,32,33], robust designs [34,35,36], materials design [37,38,39,40,41], concurrent topology optimization [42,43,44,45,46,47,48].

In the previously mentioned TO works, the classic Finite Element Method (FEM) [49] is applied to solve the unknown structural responses in numerical analysis. However, it is known that the FEM features several deficiencies in numerical analysis, like (1) the finite element mesh is just an

approximant of the structural geometry, rather than the exact representation; (2) The neighboring finite elements have the low-order (C0) continuity of the structural responses, and the deficiency also exists in the higher-order finite elements; (3) The lower efficiency to gain a high quality of the finite element mesh. These drawbacks mainly stem from the use of different mathematical languages in geometric model and numerical analysis model: spline basis functions are used in the former whereas Lagrangian and Hermitian polynomials in the latter. Meanwhile, in TO, the optimized designs generally need the additional post-processing to meet the requirements of the practical engineering structures, so that the communication with CAD systems is compulsory. On the other side, these three deficiencies might cause the high possibility of the occurrence of numerical issues in TO. Recently, a promising and powerful alternative of the FEM, termed by the IsoGeometric Analysis (IGA), is proposed by Hughes and his co -workers [50, 51] to perform the numerical analysis, which can completely remove the above limitations of FEM. In IGA, the core is that the same spline information including control points and spline basis functions is simultaneously applied into the representation of the structural geometry and solve the numerical analysis. The geometrical model and numerical analysis model are kept consistent in IGA. This such unification of the mathematical model in structural geometry and numerical analysis can offer benefits for the later optimization to resolve the above numerical issues occurred in TO.

Since the developments of IGA to eliminate the defects of the conventional FEM, several researchers have devoted to developing new TO methods and discussing their applications using IGA, rather the FEM. To the best knowledge of the authors, the first work introducing IGA into topology optimization might go back to Ref. [52], which discussed the shape optimization using IGA and its extension to the topological design. Later, an extensive work [53] used the trimmed spline surfaces to present structural boundaries and then proposed a novel Isogeometric Topology Optimization (ITO) framework based on TO and IGA, which opens up a new window for the development of TO in the future. After that, many research works have been performed to sufficiently consider the positive features of IGA into TO, which can develop more and more efficient and effective ITO methods for many numerical problems. Up to now, two publications present reviews for the IGA into structural optimization [54, 55]. However, these two papers mostly focus on the descriptions about the introducing of IGA into shape optimization and its developments, and the discussions about the IGA into topology optimization are limited in these papers. Moreover,

the considerations of IGA to replace the classic FEM in TO have obtained more and more attentions among many researchers in recent years. It is compulsory to provide an overview for the developments of ITO methods and their applications, which can provide more better research orientations and suggestions for the newcomer in the field of TO or ITO, also other readers who have interests in this field.

The rest of this paper is organized as follows: a brief description about the ITO methods in different types is presented in Section 2, and Section 3 provides the discussions about the applications of the ITO methods in different numerical problems. In Section 4, some prospects about the ITO in methods and applications are also presented. Finally, the paper ends with some concluded remarks in Section 5.

ISOGEOMETRIC TOPOLOGY OPTIMIZATION (ITO) METHODS

As already discussed in Introduction, Seo et al. [52, 53] firstly implemented the ITO using the trimmed spline surfaces and IGA, where the trimmed surface analysis treats topologically complex spline surfaces using trimming information provided by CAD systems and it is also used for calculating structural response analysis and sensitivity calculation in TO. The spline surface and trimming curves are applied to represent the outer and inner boundaries of geometrical design models, in which the coordinates of control points of a spline surface and those of trimming curves work as design variables in TO. In the design, this ITO framework deal with the inner front creation and inner front merging. When considering the complex structures in the optimization, the number of the trimming curves will increase, and a highly prohibitive computational cost might be caused.

After that, the development of ITO starts to focus on how to construct a more efficient and effective ITO method based on the previous TO methods and IGA. Up to now, many different ITO methods have been developed. According to the classifications of TO methods already discussed in Introduction, we still divide discussions about ITO methods into two different branches, namely MDMs-based ITO methods and BDMs-based ITO methods. In the first branch using MDMs, the development of ITO methods strongly depends on the “density”, namely the density-based ITO methods. As far as the second branch using the BDMs, the previous research works mostly develop ITO methods using the level set or MMC/V, namely the level set-based ITO methods and MMC/V-based ITO methods. Hence,

we will provide the detailed discussions about the ITO methods in three different types, including the density-based, level set-based and MMC/V-based, respectively.

Density-Based

As we know, the homogenization approach is earlier used to realize the optimization of structural topology, which will introduces several numerical difficulties in the design. After that, several improvements are also discussed. One of them, the Solid Isotropic Material with Penalization (SIMP) method, can be viewed as a powerful alternative, which has accepted more and more attentions owing to its conceptual clarity and easy numerical implementation [12, 13]. The basic intention of topology optimization to search the continuous material distribution is fully converted into seeking for the reasonable spatial arrangement of densities of finite elements. It is well-known that some numerical artifacts are also occurred in the optimized solutions, like the checkerboards, "zig-zag" or wavy structural boundaries and mesh-dependency [56,57,58], and several works reveal that these issues mainly stem from the strong dependency on finite elements in SIMP method [59,60,61]. Hence, some alternative variants of SIMP are also developed to eliminate the numerical difficulties and produce the distinct material interface, like introducing the densities at elementary nodes [62,63,64]. A comprehensive review about the SIMP method can refer to [9, 65]. Here, we provide a general mathematical model of the SIMP as far as the classic compliance minimization problem, given as:

$$\begin{aligned} &\underset{\boldsymbol{\rho}}{Min}:\ c(\boldsymbol{\rho}) = \mathbf{U}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{U} = \sum_{e=1}^{N} \rho_e^{p} u_e^{\mathrm{T}} k_0 u_e \\ &S.t.: \begin{cases} G(\boldsymbol{\rho}) = \sum_{e=1}^{N} \rho_e \upsilon_0 - V_0 \leq 0 \\ \boldsymbol{K}\boldsymbol{U} = \mathbf{F} \\ 0 < \rho_{\min} \leq \boldsymbol{\rho} \leq 1 \end{cases} \end{aligned}, \tag{1}$$

where c is the objective function, defined by the structural compliance, $\boldsymbol{\rho}$ is a vector containing a series of design variables, namely the element densities. ρ_{e} denotes the e_{th} element density, and p is the penalty parameter to enforce element densities to be 0 or 1. $\boldsymbol{U}$ is the global displacement field and $\boldsymbol{K}$ is the global stiffness matrix. u_e is the element displacement, and k_0 is the element stiffness matrix. υ_0 is the elementary volume fraction, and V_0 is allowable material volume fraction. $\rho_{\min}$ is the minimal value of

design variables. N is the total number of element densities. Hence, in the SIMP method, the design aims to find a reasonable layout of $\boldsymbol{\rho}$ in the design domain with the expected structural compliance c, subject to the material volume fraction V_0.

In 2011, Kumar and Parthasarathy [66] constructed B-spline finite elements for the density representation function and the displacement field in the design domain to eliminate the numerical artifacts of traditional elements, who reveal B-spline basis functions feature a smoothing effect to remove the mesh dependency, similarly to the density filtering schemes. Later, Hassani et al. [67] firstly developed an ITO method for structural compliance problem, where densities are defined at control points and Non-Uniform Rational B-Spline (NURBS) basis functions are combined with the pre-defined densities at control points to develop the density distribution function for the representation of structural topology. As shown in Figure 2, we provide some numerical results. We can easily find that although several numerical artifacts of SIMP can be successfully removed using the current ITO method, some new deficiencies are also shown in the optimized designs, like the blur and wavy structural boundaries. In the viewpoint of the authors, the current work opens up the combination of SIMP and IGA, which verify the feasibility of the introducing of IGA into SIMP. However, several new numerical artifacts are introduced. Meanwhile, the work directly employs the densities at control points to approximately represent the structural topology. It is suitable for the rectangular design domain, but it might introduce errors in the optimization for the curved structures. The main cause is that some parts of the control points are not at the curved design domain.

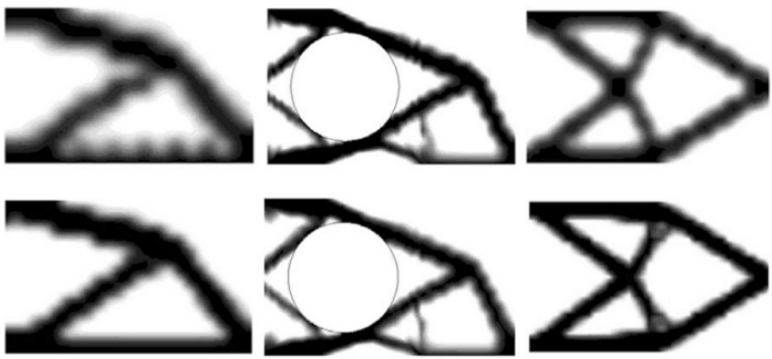


Figure 2: Some numerical results in Ref. [67].

After that, Qian [68] developed a B-spline space for the topology optimization. In this work, an arbitrarily shaped design domain is embedded into a rectangular domain, which can sufficiently employ the tensor-product feature of B-splines to develop the density field for the representation of the structural topology in the design domain. The author reveals that the B-spline representation of the topology can offer an intrinsic filter for the topology optimization, which can effectively remove numerical artifacts and control minimal feature length in the optimized designs. Moreover, the B-spline space can decouple the representation of the density distribution from the finite element analysis, which avoids the re-meshing of the design domain in the multi-resolution. We also provide some numerical results in Ref. [68], as shown in Figure 3. As we can easily see, the structural features are similar to numerical results of SIMP, like the "zig-zag" or wavy boundaries. The main reason is that the final representation of the structural topology is still based on element densities which are defined by the B-spline density representation using control densities. The spatial distribution of element densities in the design domain has the intrinsic feature, namely "zig-zag". Meanwhile, the mapping from the densities at control points to element densities will increase the existence of intermediate densities. Hence, the structural boundaries of the optimized topology are still featured with a "zig-zag" or wavy shape, which still need the additional post-processing to smooth structural boundaries for the latter manufacturing.

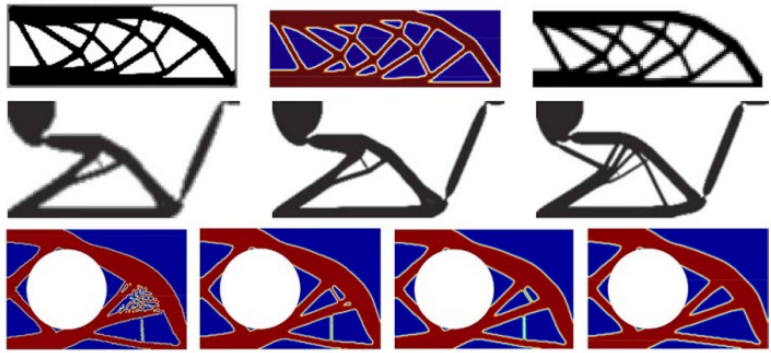


Figure 3: Some numerical results in Ref. [68].

Later, Gao et al. [69] constructed an enhanced density distribution function to develop a new ITO method. In the construction of the density distribution function, two steps are involved: (1) Smoothness: the Shepard function is firstly employed to improve the overall smoothness of the densities

pre-defined at control points. (2) Continuity: the NURBS basis functions are linearly combined with the smoothed control densities to construct the density distribution function. In each optimization iteration, the density distribution function to represent the structural topology will be advanced. As shown in Figure 4, some numerical results are also given. As we can see, an enhanced density distribution function can offer more benefits for the optimization and the representation of the structural topology. However, it should be noted that the structural boundaries are represented by the iso-contour of the density distribution function with the iso-value (0.5) of the density. It originates from the level set method, and the reasonability of the definition of the structural boundaries at the iso-contour/surface of the density distribution function is also discussed. We can easily find that the post-processing scheme is very simple, heuristic but efficient. However, it also introduces some errors in the evaluation of structural performance of the optimized designs.

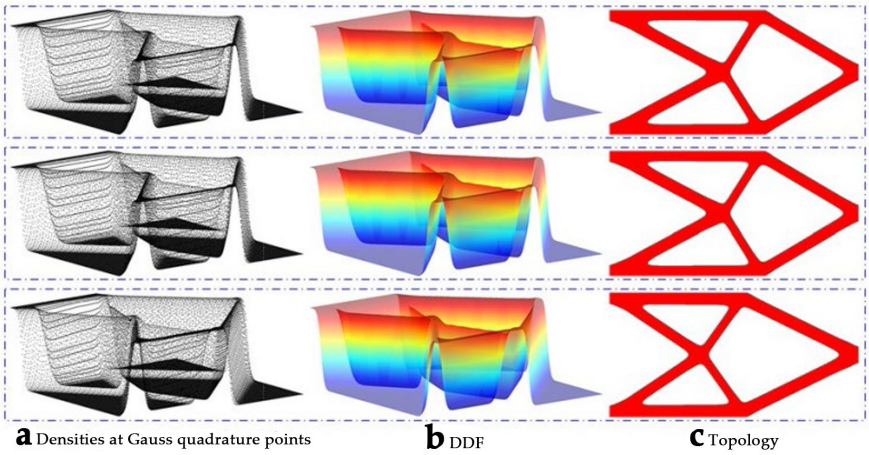


Figure 4: Some numerical results in Ref. [69].

Lieu and Lee [70] developed a multiresolution scheme to topology optimization using the framework of IGA, where a variable parameter space is defined for the implementation of multiresolution TO using SIMP method. Then, they inherited the multiresolution ITO framework [70], and applied it to discuss the multi-material topology optimization problem [71], in which the alternating active-phase algorithm [72] for the multi-material topology optimization is directly used in the multiresolution ITO framework. Wang et al. [73] discussed the multiscale ITO for periodic lattice materials, in which the asymptotic homogenization is applied for the calculation of mechanical

properties for lattice materials with uniform and graded relative density respectively. Taheri et al. [74] also studied the application of the ITO to the multi-material topology optimization problem and the design of functionally graded structures, where the multi-material interpolation scheme proposed by Stegmann and Lund [75] to realize the discrete material optimization is directly used. Liu et al. [76] also addressed the stress-constrained topology optimization problem of plane stress and bending of thin plates using the ITO framework, where two stability transformation methods are developed to stabilize the optimization using the P-norm function for global stress constraint. Later, Gao et al. [77] proposed a NURBS-based Multi-Material Interpolation (N-MMI) model in the ITO method [69] to develop a Multi-material ITO (M-ITO) method. Then Gao et al. [78] employed the ITO method to study the design of auxetic metamaterials and the M-ITO method to discuss the optimization of auxetic composites, where a series of novel and interesting material microstructures with the auxetic property can be found. Xu et al. [79] also applied the ITO method to study the rational design of ultra-lightweight architected materials. The topology optimization of the spatially graded hierarchical structures is also discussed in the framework of ITO [80]. Xie et al. [81] also proposed a truncated hierarchical B-spline–based topology optimization to address topology optimization for both minimum compliance and compliant mechanism. Wang et al. [82] discussed the numerical efficiency of the ITO method and employed the multilevel mesh, MGCG and local-update strategy to improve the computational efficiency by mesh scale reduction, solving acceleration and design variables reduction. Zhao et al. [83] also addressed the T-Splines Based ITO method for the design domains with arbitrarily shape, where the arbitrarily shaped design domains is directly obtained from CAD and defined by a single T-spline surface. The T-spline can overcome the topological limitations of NURBS. However, it also introduces an important problem that how many control points should be arranged in the local structural features. The basic feature of TO is that we do not known the final optimized design without the prior knowledge. Hence, a uniform initial design is much better for the latter optimization, which can offer the equal opportunity for the advancement of each point in the design domain and avoid the occurrence of the local optimum design. However, when using T-splines to model the geometry and analysis, a non-uniform IGA mesh will occur and also a control lattice with nonuniform features will be utilized, which will introduce some numerical issues in the latter optimization.

Level Set-Based

It is known that Level Set Method (LSM) is numerical technique to track the interface and shape, which has been extensively used in many disciplines. The core of the LSM is to define a level set function with a higher-dimension to represent the structure, where the zero-level set is employed to represent the structural boundaries. The level set function with the negative values are applied to display the voids, and the solids in the design domain are represented by the level set function with the positive values, namely the implicit boundary representation model. Hence, the evolvement of the level set function can describe the advancing of the structural topology in the design domain.

As already discussed in Section 1, Sethian and Wiegmann [15] firstly employed the level set function to represent the structure topology and used structural stress to develop the evolving mechanism. After that, Wang et al. [16] innovatively developed the level-set topology optimization framework, where the upwind scheme and the finite difference method are utilized to solve the H-J PDEs to advance the structural topology. Allaire et al. [17] developed a level-set topology optimization method based on the classical shape derivatives in the level-set method for front propagation. Compared to MDMs, we can easily find that the level-set topology optimization is actually a shape optimization method but with a superior capability to implement the shape and topology optimization. The optimized topologies will have the smooth structural boundaries and distinct interfaces, and the LSM will feature several inherent physical merits: (1) a smooth and distinct boundary description for the optimized design, (2) the shape fidelity and higher topological flexibility during the optimization, (3) the shape and topology optimization are performed simultaneously and (4) a physical meaning solution of the H-J PDEs. The mathematical model of the level-set based TO method for the structural compliance problem can read as:

$$\begin{aligned} &\underset{(u,\Phi)}{Min}: \quad J(u,\Phi)=\int_D C_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u)H(\Phi)\mathrm{d}\Omega \\ &S.t.:\begin{cases} G(u,\Phi)=\int_D H(\Phi)\mathrm{d}\Omega - V_0 \leqslant 0 \\ a(u,\upsilon,\Phi)=l(\upsilon,\Phi) \quad \forall \upsilon \in \boldsymbol{U} \\ a(u,\upsilon,\Phi)=\int_D C_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(\upsilon)H(\Phi)\mathrm{d}\Omega \\ l(\upsilon,\Phi)=\int_D p\upsilon H(\Phi)\mathrm{d}\Omega+\int_\Gamma \tau\upsilon\,\mathrm{d}\Gamma \end{cases} \end{aligned}, \tag{2}$$

where J is the objective function, defined by the structural compliance problem. u denotes the global displacement field in design domain, and Φ is the level set function with a higher dimension to represent the structural topology. D is the reference domain, and Ω is the design domain containing all admissible shapes. H is the Heaviside function which serves as a characteristic function. G is the volume constraint function. V_0 is the allowable material consumption. The elastic equilibrium equation is stated in the weak variational form, in which a is the bilinear energy function and l is the linear load function. v is the virtual displacement field, which belongs to the kinematically admissible displacement space $\boldsymbol{U}$. As shown in Figure 5, a 3D level set function with the corresponding 2D structural design domain is given.

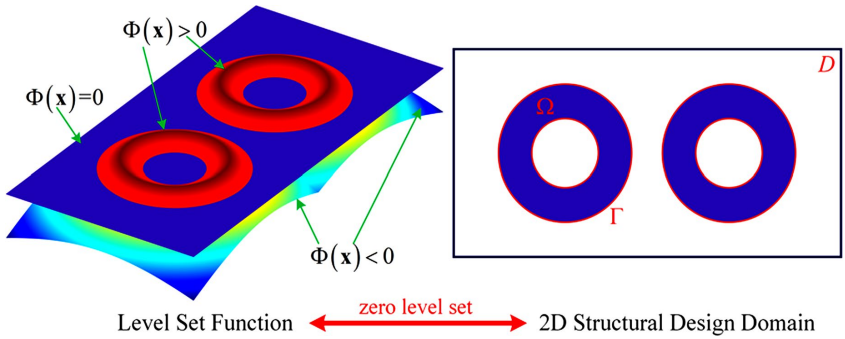


Figure 5: A 3D level set function and 2D design domain.

In 2012, Shojaee et al. [84] discussed the composition of IGA with LSM to develop a level set-based ITO framework for the structural topology optimization, where the Radial Basis Function (RBF) is applied to parametrize the level set function. The corresponding numerical results are shown in Figure 6(a). In Ref. [84], the level set function is constructed by the RBF to show the topology, and IGA uses the NURBS basis functions to develop the analysis model. In actual, we can easily obtain that the geometric model and analysis model are not in an integrated mathematical language. Later, Wang et al. [85] also proposed a parametrized level set-based ITO method using parametric level set method and IGA, where the same NURBS basis functions are used to parameterize the level set function and construct the solution space of numerical analysis. The geometric model and the analysis model of the structural topology can be unified, which coincides with the core of IGA. The numerical results of [85] are also presented in Figure 6(b). Then, Wang et al. [86] discussed the topology optimization

for geometrically constrained design domains using the proposed level set-based ITO method, where the fast point-in-polygon algorithm and trimmed elements are utilized for ITO with the arbitrary geometric constraints. As shown in Figure 6(c), the corresponding numerical results are also given. Xia et al. [87] implemented Graphics Processing Units (GPU) parallel strategy for the level set-based ITO method to improve numerical efficiency. After that, Ghasemi et al. [88] also developed a level set-based ITO method but for the optimization of flexoelectric materials, where the NURBS-based IGA elements are successfully employed to model the flexoelectric effect in dielectrics and the energy conversion efficiency of flexoelectric micro and nanostructures is improved. Moreover, the point wise density mapping is directly used in the weak form of the governing equations and the adjoint sensitivity technique is applied to compute the derivative. Jahangiry et al. [89] also discussed the application of IGA in the structural level set topology optimization to develop a new level set-based ITO framework, where the control mesh is gradually updated in the optimization iterations, and then the authors also discussed the application of the new level set-based ITO framework in the topology optimization of the concentrated heat flow and uniformly distributed heat generation systems [90]. Lee et al. [91] also implemented the isogeometric topological shape optimization using dual evolution with boundary integral equation and level sets, where the implicit geometry using the level sets is transformed into the parametric NURBS curves by minimizing the difference of velocity fields in both representations. Xu et al. [92] employed the level set-based ITO method in Ref. [85] to discuss the design of vibrating structures to maximize the fundamental eigenfrequency and avoid resonance, and the related numerical results are shown in Figure 7(a). Yu et al. [93] also employed the level set-based ITO method in Ref. [85] to implement the multiscale topology optimization using the unified microstructural skeleton, where a prototype microstructure is defined to obtain a series of graded microstructures. Figure 7(b) shows the related numerical results. In Ref. [94], a level set-based ITO method was proposed for topology optimization to control the high-frequency electromagnetic wave propagation in a domain with periodic microstructures, where the high-frequency homogenization method is used to characterize the macroscopic high-frequency waves in periodic heterogeneous media. The corresponding numerical results are also given in Figure 7(c).

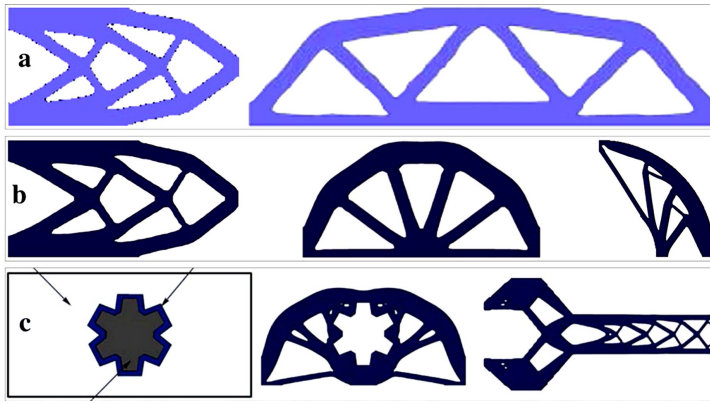


Figure 6: Some numerical results.

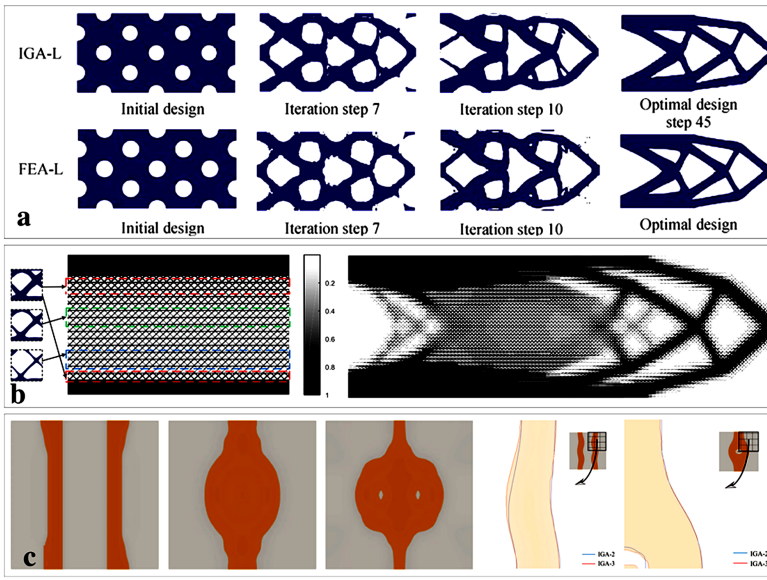


Figure 7: Some numerical results.

MMC/V-Based

Compared to the density-based and level set-based TO methods, MMC/V has implemented the topology optimization in an explicit and geometrical way. MMC/V can incorporate more geometry and mechanical information into topology optimization directly. Since the seminar work of MMC proposed by Guo et al. [20], it have been accepted more and more attentions

in not only theoretical research but also engineering applications. Zhang et al. [95] developed a new MMC-based topology optimization method, where the ersatz material model is utilized through projecting the topological description functions of the components. Later, Guo et al. [21] studied the explicit structural topology optimization based on moving morphable components (MMC) with curved skeletons. In Refs. [22, 23], the B-spline curves are used to describe the boundaries of moving morphable voids (MMVs) to develop the MMV-based topology optimization method.

In 2017, Hou et al. [96] firstly proposed an MMC-based ITO method, where NURBS basis functions are applied to construct the NURBS patch to represent the geometries of structural components using explicit design parameters and the same functions are also applied into the latter IGA. As already indicated in Ref. [96], the proposed MMC-based ITO method can naturally inherit the explicitness of the MMC-based TO method, and also embraces the merits of IGA, such as a tighter link with Computer-Aided Design (CAD) and higher-order continuity of the basis functions. The numerical results are displayed in Figure 8(a). Xie et al. [97] also developed a new MMC-based ITO method based on R-functions and collocation schemes, in which the R-functions are used to construct the topology description functions to overcome the C1 discontinuity problem of the overlapping regions of components. As given in Ref. [97] to discuss the efficiency of the proposed method, the numerical results show that the current method can improve the convergence rate in a range of 17%–60% for different cases in both FEM and IGA frameworks. This proposed MMC-based ITO method was applied to the topology optimization for the symmetric structures using energy penalization method [98]. After that, Xie et al. [99] proposed a new MMC-based ITO method using a hierarchical B-spline which can implement the adaptive IGA to efficiently and accurately assess the structural performance. As far as the MMV-based ITO method, Zhang et al. [100] proposed a new MMV-based ITO method, in which the MMV-based topology optimization framework is seamlessly integrated into IGA by using TSA (trimming surface analysis) technique. Comparatively speaking, the current MMV-based ITO method can flexibly control the structural geometry/topology. Meanwhile, it can also prevent the occurrence of self-intersection and jagged boundaries. The related numerical results are also shown in Figure 8(c). Later, Gai et al. [101] also studied the development of the MMV-based ITO method, where the closed B-spline boundary curves are utilized to model the MMVs to represent the structural topology. Du et

al. [102] discussed the application of the MMC-based ITO method in the multiresolution topology optimization problem.

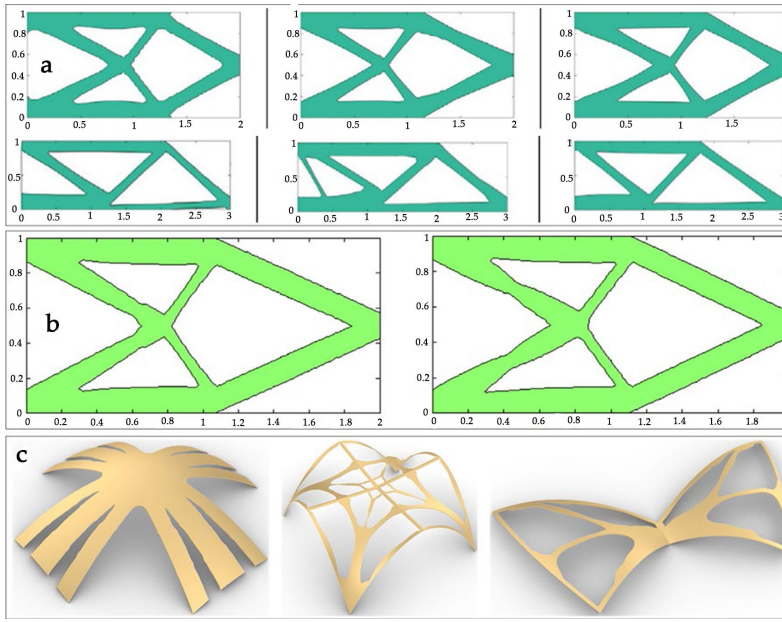


Figure 8: Some numerical results of the MMC/V-based ITO works.

Other Types

Besides the previously mentioned works, the ITO methods are also developed based on other TO methods. Dedè et al. [103] proposed a phase field-based ITO method, where the optimal design can be obtained by the steady state of the phase transition described by the generalized Cahn–Hilliard equation. The numerical solutions are presented in Figure 9(a). Yin et al. [104] developed an ITO method based on the scheme of Bi-directional Evolutionary Structural Optimization (BESO), namely the BESO-based ITO method. Sahithi et al. [105] studied the evolutionary algorithms to realize the ITO of continuum Structures using the parallel computing, where the evolutionary optimization process and metaheuristics are used to optimize the layout of material in the design domain, and the related numerical results are shown in Figure 9(b).

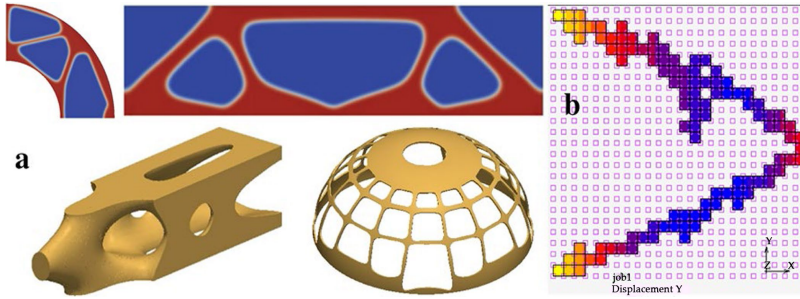


Figure 9: some numerical results.

APPLICATIONS OF ITO

In Section 2, we give a comprehensive review about the development of the ITO methods considering three components: the density-based ITO, level set-based ITO and MMC/V-based ITO. In the development of the ITO methods, the applications of the ITO methods are also involved into many numerical optimization problems, like the classic structural compliance problem with the single-material [67,68,69, 85, 89, 96, 97], the multi-material topology optimization problem [71, 74, 77, 106], the trimmed spline surfaces [53, 86, 107], the functional graded structures [74, 80].

In this section, we review the applications of the ITO in three important numerical optimization problems, including mechanical metamaterials, the splines used in IGA and the computational efficiency.

Mechanical Metamaterials

Mechanical metamaterials are a kind of artificial materials with counterintuitive mechanical properties that are obtained by the topology of their unit cell instead of the properties of each component [108]. Generally speaking, mechanical metamaterials are always associated with four elastic constants: Young's modulus, shear modulus, bulk modulus and Poisson's ratio. The corresponding subtypes of mechanical metamaterials mainly contains acoustic metamaterials, auxetic metamaterials, etc.

As already discussed in the definition of mechanical metamaterials, the effective macroscopic properties of materials strongly depend on the micro-architecture that are homogeneously arranged in the bulk material, rather than constituent properties of the base material. This feature of mechanical metamaterials can offer the high possibility for the applications of topology

optimization to seek for a series of novel metamaterial microstructures with the promising macroscopic properties. Since the homogenization theory is developed to predict macroscopic effective properties [109], an inverse homogenization procedure is proposed for the optimization of a base unit cell with the negative Poisson ratio using topology optimization [110]. Later, this work is inspired and extended to the topology optimization of the rationally artificial materials with the extreme or novel properties [111], particularly for auxetic metamaterials with the Negative Possion's Ratio (NPRs) behavior.

The earlier work introducing the IGA into the design of mechanical metamaterials can go back to Ref. [112], in which the IGA-based shape optimization is developed for the design of smoothed petal auxetic structures via computational periodic homogenization. The authors also discussed the optimal form and size feature of planar isotropic petal-shaped auxetic structures with the tunable effective properties using the IGA-based shape optimization [113]. The IGA-based shape optimization for periodic material microstructures using the inverse homogenization was also studied in Ref. [114]. The introducing of IGA into topology optimization for the rational design of auxetic metamaterials can track to Ref. [78], which used the SIMP-based ITO method proposed in Ref. [69] and also numerically implemented the energy-based homogenization method to evaluate the effective macroscopic properties using IGA with the imposing of the periodic boundary formulation on the base material unit cell. A reasonable ITO formulation for auxetic metamaterials with the re-entrant and chiral deformation mechanisms is developed, and several optimized design are shown in Figure 10(a). Later, the authors also discussed the computational design of auxetic composites via an IGA-based M-ITO method developed in Ref. [77], where an appropriate objective function with a weight parameter is also defined for the controlling of the generation of different deformation mechanisms with the re-entrant and chiral in auxetic composite microstructures [115]. The related numerical optimized microstructures with the auxetic are also shown in Figure 10(b). Later, Nguyen et al. [116] also discussed the design of auxetic metamaterials using the level set-based ITO method, where the reduced order model is utilized to reduce the computational degree of the linearly elastic equilibrium equation to improve the computational efficiency. Similarly, a series of novel and interesting auxetic microstructures in 2D and 3D, shown in Figure 10(c). Xu et al. [79] also utilized the density-based ITO method to discuss the rational design of

ultra-lightweight architected materials with the extreme bulk modulus and extreme shear modulus, and a series of novel 3D ultra-lightweight architected material microstructures can be found. Nishi et al. [94] utilized the LSM-based ITO method to discuss the design of periodic microstructures in anisotropic metamaterials to control high-frequency electromagnetic wave, in which anisotropic metamaterials with the hyperbolic and bidirectional dispersion properties at the macroscale can be obtained.

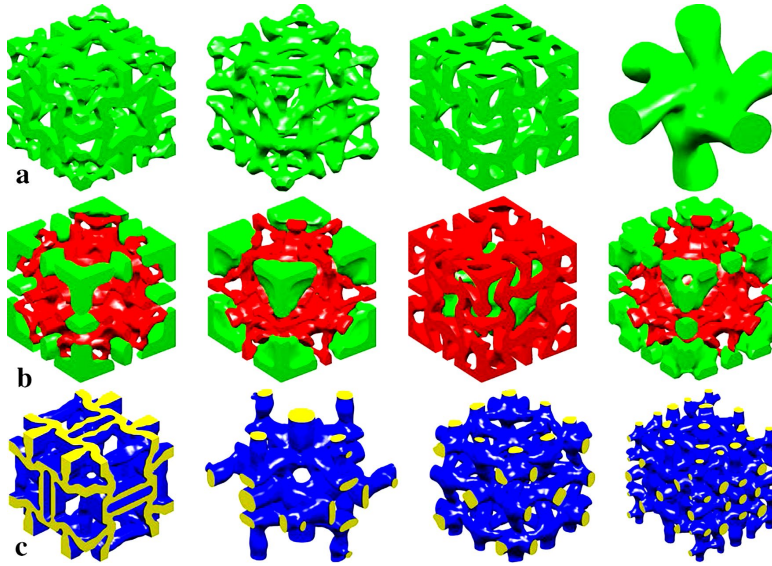


Figure 10: Some optimized design of auxetic microstructures.

Splines

In the development of the ITO method, a key in IGA is to the spline. In the earlier ITO works, the trimmed spline surfaces are employed to represent the structural topology. The outer and inner structural boundaries of the geometry are represented by a spline surface and trimming curves, in which design variables are the coordinates of control points of a spline surface and those of trimming curves [52]. This basic numerical technique is inherited in the later work [53, 100, 107], where the trimmed surface analysis is employed for the structural response analysis and sensitivity calculation in the optimization. A basic numerical scheme for the merging of the inner is shown in Figure 11(a).

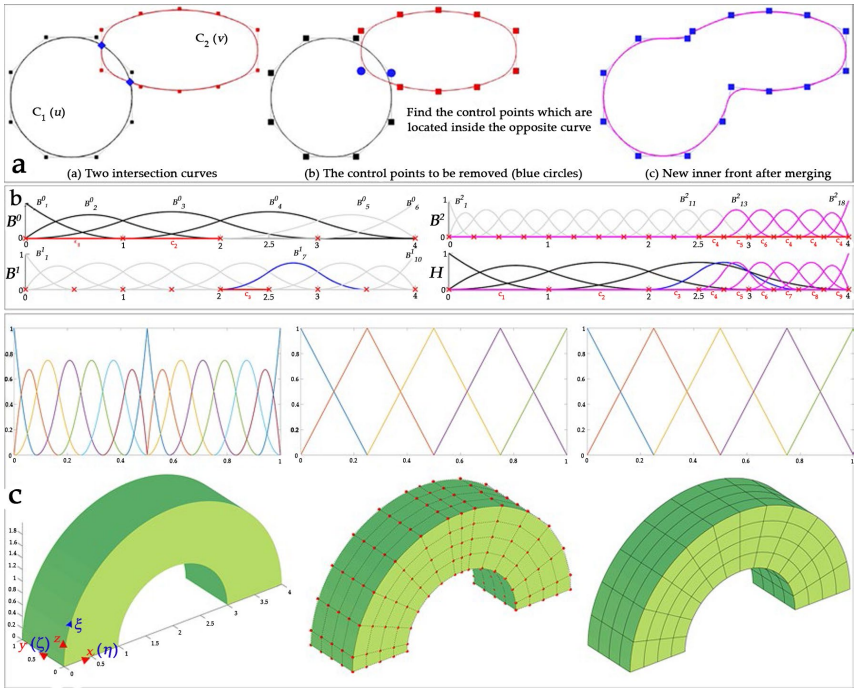


Figure 11: Illustrations of different spline schemes.

Later, the B-spline is employed in the construction of the geometrical model and B-spline basis functions are applied to develop the solution space in the IGA. Meanwhile, the B-spline-based IGA is introduced in the topology optimization. Qian [68] constructed a B-spline space for the topology optimization, where an arbitrarily shaped domain can be embedded into a rectangular domain modelled by the tensor-product B-splines. Some researchers studied the role of the B-spline in the topology optimization without using the IGA to solve the structural responses [117], where the free-form curve of closed B-splines is introduced as basic design primitives to realize topology optimization with small number of design variables. Then, the B-spline multi-parameterization method is proposed for topology optimization of thermoelastic structures [106]. After that, the hierarchical spline is applied into the development of the MMC-based ITO method, in which the adaptive IGA is implemented by the hierarchical B-spline to efficiently and accurately assess the structural performance [99]. Xie et al. [81] developed a truncated hierarchical B-spline–based topology optimization. It should be indicated that sensitivity and density filters with a lower bound can be adaptively consistent with the hierarchical levels

of active elements to remove the checkboard pattern and reduce the gray transition area. A basic illustration of the hierarchical B-spline is given in Figure 11(b).

Comparatively speaking, NURBS, working as a mathematical model commonly used in computer graphics for generating and representing curves and surfaces, is also mostly employed in the development of the ITO method in three types. Wang et al. [85] developed the level set-based ITO method using NURBS, in which NURBS is firstly applied to parametrize the level set function to represent the structural topology and then construct the solution space in IGA to solve the unknown structural responses. Gao et al. [69] also employed NURBS to develop an enhanced density distribution function with the sufficient smoothness and continuity to represent the structural topology, and the same NURBS basis functions are also used in IGA. Hou et al. [96] used NURBS to construct the MMCs for the representation of the geometries of structural components (a subset of the design domain) with use of explicit design parameters, and the NURBS-based IGA is also applied to solve the structural responses. A basis description about the NURBS for the representation of the structural geometry is shown in Figure 11(c). Besides the above discussed splines, the T-spline is also used in the ITO method for the topology optimization, and the T-spline-based ITO method is developed to realize the optimization of design domain with arbitrary shapes [83] to eliminate the complexity of the multi-patch NURBS for the structural geometry. In actual, it will introduce an important problem that how many control points should be arranged in the representation of structural local features, which will have a significant effect on the latter topology optimization.

Computational Cost

Although computer has gained a great number of developments in recent years, the computational cost of topology optimization is still a prohibitive problem, especially for the common laptop. In order to improve the computational efficiency of the ITO in numerical implementations, several research works have been implemented in recent years. The most method is the use of multiresolution scheme in numerical calculation of the topology optimization [118]. In the multiresolution topology optimization, three distinct meshes are defined for the optimization: (1) a displacement mesh for the finite element analysis; (2) a design variable mesh for the optimization; and (3) a density or level set mesh to display the material distribution. The

basic idea is that topology optimization can achieve the higher-resolution designs but with a lower computation cost as well. Lieu et al. [70] developed a multiresolution ITO method using SIMP to improve computational efficiency, and then applied it to address the multi-material topology optimization problem [71]. Du et al. [102] also utilized the multiresolution scheme in the MMV-based ITO method to reduce the computational cost. A simple illustration of multiresolution scheme is shown in Figure 12. Wang et al. [82] also improved the computational efficiency in three aspects: namely the mesh scale reduction, solving acceleration and design variables reduction, and the ITO method is developed using multilevel mesh, multigrid conjugate gradient method and local-update strategy. As already given in numerical results, the current proposed method can successfully reduce 37%–93% computational time compared to previous works. The GPU parallel strategy is also employed in the parameterized LSM-based ITO method to reduce the computational cost [87], where the parallel implementations are utilized in the initial design domain, IGA, sensitivity analysis and design variable update.

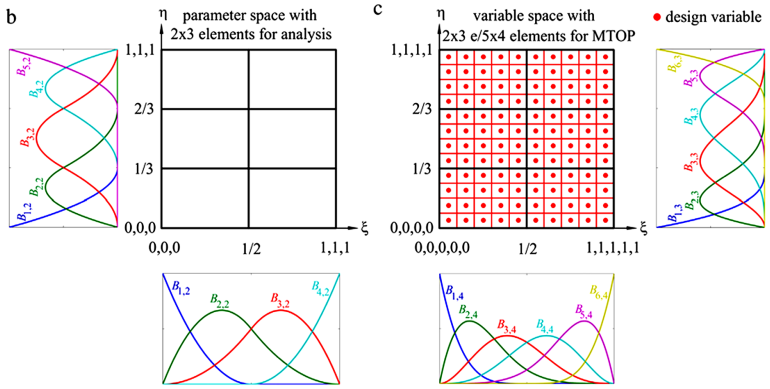


Figure 12: A simple illustration of multiresolution scheme in TO [70].

PROSPECTS

In this Section, we will provide three main directions for the development of ITO in the future, including the Data-driven ITO, ITO for additive manufacturing and ITO considering the advantages of IGA in several problems. The details are given as follows.

(1) *Data-driven ITO*: It is known that the application of topology optimization for the complex engineering materials is very

difficult due to the complexity. In recent years, the big data and machine learning have been becoming popularly in the field of computational mechanics, which can provide new windows for the topology optimization for complex problems. For example, the deep neural network is employed to approximate the field of variables to solve the boundary value equations in strong or weak forms [119]. In the work, the developed data-driven neural networks can efficiently reduce the computational costs. Meanwhile, the data-driven isogeometric shape optimization for auxetic microstructures is also studied [120]. Hence, in the future work of the ITO, the data-driven ITO method and its applications in several numerical problems will be the promising research topic.

(2) *ITO for additively manufacturing*: In recent years, additive manufacturing technique, a layer-by-layer manner to fabricate structures, has accepted great attentions and been becoming a powerful alternative to the conventional fabrication methods, like the machining and casting, due to its merits to manufacture the structures with specific features, like the cavity. Hence, additive manufacturing can offer the higher flexibility and efficiency for the fabrication of structures. The topology optimization design for additive manufacturing has proposed in recent years, and the comprehensive reviews for this topic can refer to Refs. [121, 122]. IGA has the positive feature to unify Computer-Aided Design (CAD) model and Computer-Aided Engineering (CAE) into a same mathematical language, so that the ITO can offer more possibility for engineering structures from the conceptual design phase to the last manufacturing into an integrated process, if the development of the ITO can consider the additive manufacturing. This unification will significantly reduce the financial cost of the product design. Hence, in the future, the ITO for additive manufacturing will be also a hot research topic.

(3) *The advantages of IGA in several problems*: IGA has the compelling benefits in the field of shell and plate overall conventional approaches [123, 124], because the smoothness of NURBS basis functions can offer a straightforward manner to construct the plate/shell elements, particularly for the thin

shells and rotation-free. Meanwhile, the smoothness of NURBS basis functions can also offer more benefits for the analysis of fluids [125] and the fluid-structure interaction problems [126]. In addition, due to the ease of construction of the higher-order basis functions, IGA with more success can be utilized to solve PDEs with the forth-order (or higher) derivatives, for example the Hill–Cahnard equation [127]. Hence, in the future, the considerations of the ITO in the mentioned above numerical problems might be more meaningful for the development of this field.

CONCLUSIONS

In the current paper, we offer a comprehensive review for the Isogeometric Topology Optimization (ITO) in methods and applications. Firstly, we mainly divide the descriptions of ITO methods into three aspects, including the density-based ITO methods, level set-based ITO methods and MMC/V-based ITO methods. The corresponding discussion for each classification is clearly provided, and the development trajectory in each classification is also given. Secondly, the descriptions of the applications of ITO mainly focus on three components, namely the ITO for mechanical metamaterials, the splines in ITO and the computational cost of ITO. Finally, we also provide some prospects for the developments of the ITO methods and applications in the future, which contains the data-driven ITO to considerably reduce the computation cost, the ITO for additive manufacturing to consider the manufacturing problems into the initial conceptual design phase and the ITO considering the advantages of IGA in several problems.

REFERENCES

1. R T Haftka, Z Gürdal. *Elements of structural optimization*. Berlin: Springer, 2012.
2. M P Bendsøe, O Sigmund. *Topology optimization: Theory, methods and applications*. Springer, 2003.
3. A G M Michell. The limits of economy of material in frame-structures. London, Edinburgh, *Dublin Philosophical Magazine and Journal of Science*, 1904(8): 589–597.
4. K-T Cheng, N Olhoff. An investigation concerning optimal design of solid elastic plates. *International Journal of Solids and Structures*, 1981(17): 305–323.
5. K-T Cheng, N Olhoff. Regularized formulation for optimal design of axisymmetric plates. *International Journal of Solids and Structures*, 1982(18): 153–169.
6. M P Bendsøe, N Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 1988(71): 197–224.
7. GIN Rozvany. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, 2009(37): 217–237.
8. X Huang, Y-MM Xie. A further review of ESO type methods for topology optimization. *Structural and Multidisciplinary Optimization*, 2010(41): 671–683.
9. O Sigmund, K Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 2013(48): 1031–1055.
10. van Dijk NP, K Maute, M Langelaar et al. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 2013(48): 437–472.
11. J D Deaton, R V Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 2014(49): 1–38.
12. M Zhou, GIN Rozvany. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 1991(89): 309–336.
13. M P Bendsøe, O Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 1999(69): 635–654.

14. Y M Xie, G P Steven. A simple evolutionary procedure for structural optimization. *Computers & Structures*, 1993(49): 885–969.
15. J A Sethian, A Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics*, 2000(163): 489–528.
16. M Y Wang, X Wang, D Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 2003(192): 227–246.
17. G Allaire, F Jouve, AM Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 2004(194): 363–393.
18. MY Wang, S Zhou. Phase field: a variational method for structural topology optimization. *Computer Modeling in Engineering & Sciences*, 2004(6): 547–566.
19. A Takezawa, S Nishiwaki, M Kitamura. Shape and topology optimization based on the phase field method and sensitivity analysis. *Journal of Computational Physics*, 2010(229): 2697–2718.
20. X Guo, W Zhang, W Zhong. Doing topology optimization explicitly and geometrically—a mew moving morphable components based framework. *Journal of Applied Mechanics*, 2014(81): 081009.
21. X Guo, W Zhang, J Zhang et al. Explicit structural topology optimization based on moving morphable components (MMC) with curved skeletons. *Computer Methods in Applied Mechanics and Engineering*, 2016(310): 711–748.
22. W Y Yang, W S Zhang, X Guo. Explicit structural topology optimization via Moving Morphable Voids (MMV) approach. *2016 Asian Congr. Struct. Multidiscip. Optim.* Nagasaki, Japan, 2016: 98.
23. W Zhang, W Yang, J Zhou et al. Structural topology optimization through explicit boundary evolution. *Journal of Applied Mechanics*, 2017: 84.
24. H A Eschenauer, V V Kobelev, A Schumacher. Bubble method for topology and shape optimization of structures. *Structural Optimization*, 1994(8): 42–51.
25. S Cai, W Zhang. An adaptive bubble method for structural shape and topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 2020(360): 112778.

26. A R Díaaz, N Kikuchi. Solutions to shape and topology eigenvalue optimization problems using a homogenization method. *International Journal for Numerical Methods in Engineering*, 1992(35): 1487–1502.
27. J Du, N Olhoff. Topological design of freely vibrating continuum structures for maximum values of simple and multiple eigenfrequencies and frequency gaps. *Structural and Multidisciplinary Optimization*, 2007(34): 91–110.
28. J Gao, Z Luo, H Li, et al. Dynamic multiscale topology optimization for multi-regional micro-structured cellular composites. *Composite Structures*, 2019(211): 401–417.
29. L Yin, G K Ananthasuresh. Topology optimization of compliant mechanisms with multiple materials using a peak function material interpolation scheme. *Structural and Multidisciplinary Optimization*, 2001(23): 49–62.
30. S Chu, L Gao, M Xiao, et al. Stress-based multi-material topology optimization of compliant mechanisms. *International Journal for Numerical Methods in Engineering*, 2018(113): 1021–1044.
31. G Allaire, J F ouve. Minimum stress optimal design with the level set method. *Engineering Analysis with Boundary Elements*, 2008(32): 909–918.
32. S Chu, L Gao, M Xiao, et al. A new method based on adaptive volume constraint and stress penalty for stress-constrained topology optimization. *Structural and Multidisciplinary Optimization*, 2018(57): 1163–1185.
33. K Long, X Wang, H Liu. Stress-constrained topology optimization of continuum structures subjected to harmonic force excitation using sequential quadratic programming. *Structural and Multidisciplinary Optimization*, 2019(59): 1747–1759.
34. X Zhang, A Takezawa, Z Kang. Robust topology optimization of vibrating structures considering random diffuse regions via a phase-field method. *Computer Methods in Applied Mechanics and Engineering*, 2019(344): 766–797.
35. J Zheng, Z Luo, C Jiang, et al. Robust topology optimization for concurrent design of dynamic structures under hybrid uncertainties. *Mechanical Systems and Signal Processing*, 2019(120): 540–559.
36. Y Zheng, M Xiao, L Gao, et al. Robust topology optimization for periodic structures by combining sensitivity averaging with a

semianalytical method. *International Journal for Numerical Methods in Engineering*, 2019(117): 475–497.

37. J Gao, H Li, L Gao, et al. Topological shape optimization of 3D micro-structured materials using energy-based homogenization method. *Advances in Engineering Software*, 2018(116): 89–102.
38. Y Wang, J Gao, Z Luo, et al. Level-set topology optimization for multimaterial and multifunctional mechanical metamaterials. *Engineering Optimization*, 2017(49): 22–42.
39. K Long, X Du, S Xu, et al. Maximizing the effective Young's modulus of a composite material by exploiting the Poisson effect. *Composite Structures*, 2016(153): 593–600.
40. L Xia, P Breitkopf. Design of materials using topology optimization and energy-based homogenization approach in Matlab. *Structural and Multidisciplinary Optimization*, 2015(52): 1229–1241.
41. Y Zhang, H Li, M Xiao, et al. Concurrent topology optimization for cellular structures with nonuniform microstructures based on the kriging metamodel. *Structural and Multidisciplinary Optimization*, 2019(59): 1273–1299.
42. L Xia, P Breitkopf. Concurrent topology optimization design of material and structure within FE2 nonlinear multiscale analysis framework. *Computer Methods in Applied Mechanics and Engineering*, 2014(278): 524–542.
43. H Li, Z Luo, N Zhang, et al. Integrated design of cellular composites using a level-set topology optimization method. *Computer Methods in Applied Mechanics and Engineering*, 2016(309): 453–475.
44. Y Wang, F Chen, M Y Wang. Concurrent design with connectable graded microstructures. *Computer Methods in Applied Mechanics and Engineering*, 2017(317): 84–101.
45. H Li, Z Luo, L Gao, et al. Topology optimization for functionally graded cellular composites with metamaterials by level sets. *Computer Methods in Applied Mechanics and Engineering*, 2018(328): 340–364.
46. J Gao, Z Luo, H Li, et al. Topology optimization for multiscale design of porous composites with multi-domain microstructures. *Computer Methods in Applied Mechanics and Engineering*, 2019(344): 451–476.
47. J Gao, Z Luo, L Xia, et al. Concurrent topology optimization of multiscale composite structures in Matlab. *Structural and Multidisciplinary Optimization*, 2019(60): 2621–2651.

48. Y Zhang, M Xiao, L Gao, et al. Multiscale topology optimization for minimizing frequency responses of cellular composites with connectable graded microstructures. *Mechanical Systems and Signal Processing*, 2020(135): 106369.
49. TJR Hughes. *The finite element method: linear static and dynamic finite element analysis.* Courier Corporation, 2012.
50. J A Cottrell, T J R Hughes, Y Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. 2009.
51. T J R Hughes, J A Cottrell, Y Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 2005(194): 4135–4195.
52. Y-D Seo, H-J Kim, S-K Youn. Shape optimization and its extension to topological design based on isogeometric analysis. *International Journal of Solids and Structures*, 2010(47): 1618–1640.
53. Y-D Seo, H-J Kim, S-K Youn. Isogeometric topology optimization using trimmed spline surfaces. *Computer Methods in Applied Mechanics and Engineering*, 2010(199): 3270–3296.
54. Y Wang, Z Wang, Z Xia, et al. Structural design optimization using isogeometric analysis: a comprehensive review. *Computer Modeling in Engineering & Sciences*, 2018(117): 455–507.
55. Hongliang Liu, Xuefeng Zhu, Dixiong Yang. Research advances in isogeometric analysis-based optimum design of structure. *Chinese Journal of Solid Mechanics*, 2018(39): 248–267. (in Chinese)
56. D Aíaz, S O igmund. Checkerboard patterns in layout optimization. *Structural Optimization*, 1995(10): 40–45.
57. O Sigmund, J Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 1998(16): 68–75.
58. O Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 2007(33): 401–424.
59. K Matsui, K Terada. Continuous approximation of material distribution for topology optimization. *International Journal for Numerical Methods in Engineering*, 2004(59): 1925–1944.

60. S F Rahmatalla, S CC. A Q4/Q4 continuum structural topology optimization implementation. *Structural and Multidisciplinary Optimization*, 2004(27): 130–135.
61. G H Paulino, C H Le. A modified Q4/Q4 element for topology optimization. *Structural and Multidisciplinary Optimization*, 2009(37): 255–264.
62. Z Kang, Y Wang. Structural topology optimization based on non-local Shepard interpolation of density field. *Computer Methods in Applied Mechanics and Engineering*, 2011(200): 3515–3525.
63. Z Kang, Y Wang. A nodal variable method of structural topology optimization based on Shepard interpolant. *International Journal for Numerical Methods in Engineering*, 2012(90): 329–342.
64. Z Luo, N Zhang, Y Wang, et al. Topology optimization of structures using meshless density variable approximants. *International Journal for Numerical Methods in Engineering*, 2013(93): 443–464.
65. F Wang, B S Lazarov, O Sigmund. On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, 2011(43): 767–784.
66. A V Kumar, A Parthasarathy. Topology optimization using B-spline finite elements. *Structural and Multidisciplinary Optimization*, 2011(44): 471.
67. B Hassani, M Khanzadi, S M Tavakkoli. An isogeometrical approach to structural topology optimization by optimality criteria. *Structural and Multidisciplinary Optimization*, 2012(45): 223–233.
68. X Qian. Topology optimization in B-spline space. *Computer Methods in Applied Mechanics and Engineering*, 2013(265): 15–35.
69. J Gao, L Gao, Z Luo, et al. Isogeometric topology optimization for continuum structures using density distribution function. *International Journal for Numerical Methods in Engineering*, 2019(119): 991–1017.
70. Q X Lieu, J Lee. Multiresolution topology optimization using isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 2017(112): 2025–2047.
71. QX Lieu, J Lee. A multi-resolution approach for multi-material topology optimization based on isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 2017(323): 272–302.
72. R Tavakoli, S M Mohseni. Alternating active-phase algorithm for multimaterial topology optimization problems: A 115-line MATLAB

implementation. *Structural and Multidisciplinary Optimization*, 2014(49): 621–642.

73. Y Wang, H Xu, D Pasini. Multiscale isogeometric topology optimization for lattice materials. *Computer Methods in Applied Mechanics and Engineering*, 2017(316): 568–585.
74. A H Taheri, K Suresh. An isogeometric approach to topology optimization of multi-material and functionally graded structures. *International Journal for Numerical Methods in Engineering*, 2017(109): 668–696.
75. J Stegmann, E Lund. Discrete material optimization of general composite shell structures. *International Journal for Numerical Methods in Engineering*, 2005(62): 2009–2027.
76. H Liu, D Yang, P Hao, et al. Isogeometric analysis based topology optimization design with global stress constraint. *Computer Methods in Applied Mechanics and Engineering*, 2018(342): 625–652.
77. J Gao, Z Luo, M Xiao, et al. A NURBS-based Multi-Material Interpolation (N-MMI) for isogeometric topology optimization of structures. *Applied Mathematical Modelling*, 2020(81): 818–843.
78. J Gao, H Xue, L Gao, et al. Topology optimization for auxetic metamaterials based on isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 2019(352): 211–236.
79. J Xu, L Gao, M Xiao, et al. Isogeometric topology optimization for rational design of ultra-lightweight architected materials. *International Journal of Mechanical Sciences*, 2020(166): 105103.
80. M Xu, L Xia, S Wang, et al. An isogeometric approach to topology optimization of spatially graded hierarchical structures. *Composite Structures*, 2019(225): 111171.
81. X Xie, S Wang, Y Wang et al. Truncated hierarchical B-spline–based topology optimization. *Structural and Multidisciplinary Optimization*, 2020(62): 83–105.
82. Y Wang, Z Liao, M Ye, et al. An efficient isogeometric topology optimization using multilevel mesh, MGCG and local-update strategy. *Advances in Engineering Software*, 2020(139): 102733.
83. G Zhao, Y J ang, W Wang, et al. T-Splines based isogeometric topology optimization with arbitrarily shaped design domains. *Computer Modeling in Engineering & Sciences*, 2020(123): 1033–1059.
84. S Shojaee, M Mohamadianb, N Valizadeh. Composition of isogeometric analysis with level set method for structural topology

optimization. *International Journal of Optimization in Civil Engineering*, 2012(2): 47–70.

85. Y Wang, D J Benson. Isogeometric analysis for parameterized LSM-based structural topology optimization. *Computational Mechanics*, 2016(57): 19–35.
86. Y Wang, D J Benson. Geometrically constrained isogeometric parameterized level-set based topology optimization via trimmed elements. *Frontiers in Mechanical Engineering*, 2016(11): 328–343.
87. Z Xia, Y Wang, Q Wang, et al. GPU parallel strategy for parameterized LSM-based topology optimization using isogeometric analysis. *Structural and Multidisciplinary Optimization*, 2017: 1–22.
88. H Ghasemi, H S Park, T Rabczuk. A level-set based IGA formulation for topology optimization of flexoelectric materials. *Computer Methods in Applied Mechanics and Engineering*, 2017(313): 239–258.
89. H A Jahangiry, S M Tavakkoli. An isogeometrical approach to structural level set topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 2017(319): 240–257.
90. H A Jahangiry, A Jahangiri. Combination of Isogeometric analysis and level-set method in topology optimization of heat-conduction systems. *Applied Thermal Engineering*, 2019(161): 114134.
91. S-W Lee, M Yoon, S Cho. Isogeometric topological shape optimization using dual evolution with boundary integral equation and level sets. *Computer-Aided Design*, 2017(82): 88–99.
92. M Xu, S Wang, X Xie. Level set-based isogeometric topology optimization for maximizing fundamental eigenfrequency. *Frontiers in Mechanical Engineering*, 2019(14): 222–234.
93. C Yu, Q Wang, C Mei, et al. Multiscale isogeometric topology optimization with unified structural skeleton. *Computer Modeling in Engineering & Sciences*, 2020(122): 779–803.
94. S Nishi, T Yamada, K Izui, et al. Isogeometric topology optimization of anisotropic metamaterials for controlling high-frequency electromagnetic wave. *International Journal for Numerical Methods in Engineering*. 2020(121): 1218–1247.
95. W Zhang, J Yuan, J Zhang, et al. A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model. *Structural and Multidisciplinary Optimization*, 2016(53): 1243–1260.

96. W Hou, Y Gai, X Zhu, et al. Explicit isogeometric topology optimization using moving morphable components. *Computer Methods in Applied Mechanics and Engineering*, 2017(326): 694–712.

97. X Xie, S Wang, M Xu, et al. A new isogeometric topology optimization using moving morphable components based on R-functions and collocation schemes. *Computer Methods in Applied Mechanics and Engineering*, 2018(339): 61–90.

98. X Xie, S Wang, M Ye, et al. Isogeometric topology optimization based on energy penalization for symmetric structure. *Frontiers in Mechanical Engineering*, 2020(15): 100–122.

99. X Xie, S Wang, M Xu, et al. A hierarchical spline based isogeometric topology optimization using moving morphable components. *Computer Methods in Applied Mechanics and Engineering*, 2020(360): 112696.

100. W Zhang, D Li, P Kang et al. Explicit topology optimization using IGA-based moving morphable void (MMV) approach. *Computer Methods in Applied Mechanics and Engineering*, 2020(360): 112685.

101. Y Gai, X Zhu, Y J Zhang, et al. Explicit isogeometric topology optimization based on moving morphable voids with closed B-spline boundary curves. *Structural and Multidisciplinary Optimization*, 2020(61): 963–982.

102. B Du, Y Zhao, W Yao, et al. Multiresolution isogeometric topology optimisation using moving morphable voids. *Computer Modeling in Engineering & Sciences*, 2020(122): 1119–1140.

103. L Dedè, M J Borden, T J R Hughes. Isogeometric analysis for topology optimization with a phase field model. *Archives of Computational Methods in Engineering*, 2012(19): 427–65.

104. L Yin, F Zhang, X Deng, et al. Isogeometric bi-directional evolutionary structural optimization. *IEEE Access*, 2019(7): 91134–91145.

105. NSS Sahithi, KNV Chandrasekhar, TM Rao. A comparative study on evolutionary algorithms to perform isogeometric topology optimisation of continuum structures using parallel computing. *Journal of Aerospace Engineering & Technology*, 2018(8): 51–58.

106. X Zhao, W Zhang, T Gao, et al. A B-spline multi-parameterization method for multi-material topology optimization of thermoelastic structures. *Structural and Multidisciplinary Optimization*, 2020(61): 923–942.

107. P Kang, S-K Youn. Isogeometric topology optimization of shell structures using trimmed NURBS surfaces. *Finite Elements in Analysis and Design*, 2016(120): 18–40.

108. X Yu, J Zhou, H Liang, et al. Mechanical metamaterials associated with stiffness, rigidity and compressibility: A brief review. *Progress in Materials Science*, 2018(94): 114–173.

109. JMJ Guedes, N Kikuchi. Preprocessing and postprocessing for materials based on the homogenization method with adaptive finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 1990(83): 143–198.

110. O Sigmund. Materials with prescribed constitutive parameters: An inverse homogenization problem. *International Journal of Solids and Structures*, 1994(31): 2313–2329.

111. M Osanov, J K Guest. Topology optimization for architected materials design. *Annual Review of Materials Research*, 2016(46): 211–233.

112. Z-P Wang, L H Poh, J Dirrenberger, et al. Isogeometric shape optimization of smoothed petal auxetic structures via computational periodic homogenization. *Computer Methods in Applied Mechanics and Engineering*, 2017(323): 250–271.

113. Z-P Wang, L H Poh. Optimal form and size characterization of planar isotropic petal-shaped auxetics with tunable effective properties using IGA. *Composite Structures*, 2018(201): 486–502.

114. J K Lüdeker, O Sigmund, B Kriegesmann. Inverse homogenization using isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 2020(368): 113170.

115. J Gao, M Xiao, L Gao, et al. Isogeometric topology optimization for computational design of re-entrant and chiral auxetic composites. *Computer Methods in Applied Mechanics and Engineering*, 2020(362): 112876.

116. C Nguyen, X Zhuang, L Chamoin, et al. Three-dimensional topology optimization of auxetic metamaterial using isogeometric analysis and model order reduction. *Computer Methods in Applied Mechanics and Engineering*, 2020(371): 113306.

117. W Zhang, L Zhao, T Gao, et al. Topology optimization with closed B-splines and Boolean operations. *Computer Methods in Applied Mechanics and Engineering*, 2017(315): 652–670.

118. T H Nguyen, G H Paulino, J Song, et al. A computational paradigm for multiresolution topology optimization (MTOP). *Structural and Multidisciplinary Optimization*, 2010(41): 525–539.

119. E Samaniego, C Anitescu, S Goswami, et al. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 2020(362): 112790.

120. Y Wang, Z Liao, S Shi, et al. Data-driven structural design optimization for petal-shaped auxetics using isogeometric analysis. *Computer Modeling in Engineering & Sciences*, 2020(122): 433–458.

121. J Liu, A T Gaynor, S Chen, et al. Current and future trends in topology optimization for additive manufacturing. *Structural and Multidisciplinary Optimization*, 2018(57): 2457–2483.

122. L Meng, W Zhang, D Quan, et al. From topology optimization design to additive manufacturing: Today's success and tomorrow's roadmap. *Archives of Computational Methods in Engineering*, 2019:1–26.

123. D J Benson, Y Bazilevs, MC Hsu et al. Isogeometric shell analysis: The Reissner–Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 2010(199): 276–289.

124. DJ Benson, Y Bazilevs, M-C Hsu, et al. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 2011(200): 1367–1378.

125. H Gomez, TJR Hughes, X Nogueira, et al. Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations. *Computer Methods in Applied Mechanics and Engineering*, 2010(199): 1828–1840.

126. Y Bazilevs, VM Calo, TJR Hughes, et al. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 2008(43): 3–37.

127. H Gómez, V M Calo, Y Bazilevs, et al. Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 2008(197): 4333–4352.

Chapter

14

ANALYSIS AND COMPUTATIONS OF OPTIMAL CONTROL PROBLEMS FOR BOUSSINESQ EQUATIONS

Andrea Chierici [1], Valentina Giovacchini [2], and Sandro Manservisi [2]

[1]Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409, USA
[2]Laboratory of Montecuccolino, Department of Industrial Engineering, University of Bologna, Via dei Colli 16, 40136 Bologna, Italy

ABSTRACT

The main purpose of engineering applications for fluid with natural and mixed convection is to control or enhance the flow motion and the heat transfer. In this paper, we use mathematical tools based on optimal control theory to show the possibility of systematically controlling natural and mixed convection flows. We consider different control mechanisms such

Citation: (APA): Chierici, A., Giovacchini, V., & Manservisi, S. (2022). Analysis and Computations of Optimal Control Problems for Boussinesq Equations. Fluids, 7(6), 203. (27 pages).

as distributed, Dirichlet, and Neumann boundary controls. We introduce mathematical tools such as functional spaces and their norms together with bilinear and trilinear forms that are used to express the weak formulation of the partial differential equations. For each of the three different control mechanisms, we aim to study the optimal control problem from a mathematical and numerical point of view. To do so, we present the weak form of the boundary value problem in order to assure the existence of solutions. We state the optimization problem using the method of Lagrange multipliers. In this paper, we show and compare the numerical results obtained by considering these different control mechanisms with different objectives.

Keywords: optimal control; natural convection; mixed convection; Lagrange multipliers method; Boussinesq equations

INTRODUCTION

The optimization of complex systems in engineering is a crucial aspect that encourages and promotes research in the optimal control field. Optimization problems have three main ingredients: objectives, controls, and constraints. The first ingredient is the objective of interest in engineering applications, namely, flow matching, drag minimization, and enhancing or reducing turbulence. A quadratic functional minimization usually defines this objective. The controls can be chosen for large classes of design parameters. Examples are boundary controls such as injection or suction of fluid [1] and heating or cooling temperature controls [2,3,4], distributed controls such as heat sources or magnetic fields [5], and shape controls such as geometric domains [6]. Finally, a specific set of partial differential equations for the state variables defines the constraints. A typical optimization problem consists of finding state and control variables that minimize the objective functional and satisfy the imposed constraints [7]. In [7], the interested reader can find time-dependent and stochastic (input data polluted by random noise) analyses of optimal control theory that broaden the perspective of this work, here limited to stationary equations. Of course, the stochastic and optimal control time-dependent approach requires larger computational resources that severely limit real-life applications.

In this paper, we focus on engineering applications where fluid natural convection plays a main role. In these cases, buoyancy forces have a strong influence on the flow. Applications for natural convection optimal design

are crucial in many contexts, ranging from semiconductor production processes, where buoyancy forces can control the crystal growth, to thermal hydraulics of lead-cooled fast reactors (LFR), where emergency cooling is guaranteed by natural convection. In the design of engineering devices such as heat exchangers, nuclear cores, and primary or secondary circuit pipes, optimization techniques can be used to achieve specified objectives such as desired wall temperatures or wall-normal heat fluxes, target mean temperatures, velocity profiles, or turbulence enhancements/reductions. The thermodynamic properties of lead allow a high level of natural circulation cooling in the primary system of an LFR. For core cooling, LFR design enhances strong natural circulation during plant operations and shutdown conditions [8]. Within this framework, we aim to study optimal control problems for mixed and natural convection.

In the past few years, the mathematical analysis of the optimal control of Navier–Stokes and energy equations has made considerable progress. The optimization of the heat transfer in forced convection flows can be found in many studies, mainly where the coupling between the Navier–Stokes and energy equations ignores density variations (see for example [2,4] and citations therein). In the case of natural or mixed convection flows, several authors have studied the mathematical analysis of the optimal control for the Oberbeck–Boussinesq system, focusing on stationary distributed and boundary thermal controls (see for example [3,5,9,10,11,12]). The solvability of the stationary boundary control problem for the Boussinesq equation is studied in [13,14], considering as boundary controls the velocity, the temperature, and the heat flux. Recently, new approaches to the study of the optimal control of Boussinesq equations have been proposed [15,16,17]. In [15], the solvability of an optimal control problem for steady non-isothermal incompressible creeping flows was proven. The temperature and the pressure in a flat portion of the local Lipschitz boundary played the role of controls. In [16], the optimal Neumann control problem for non-isothermal steady flows in low-concentration aqueous polymer solutions was considered, and sufficient conditions for the existence of optimal solutions were established. The problem of the optimal start control for unsteady Boussinesq equations was investigated in [17] to prove their solvability.

The main aim of this paper is to show the possibility of systematically controlling natural and mixed convection flows using mathematical tools based on optimal control theory. We consider three different control mechanisms: distributed, Dirichlet, and Neumann boundary controls. The solvability of the stationary optimal control problem for the Boussinesq

equations has already been widely investigated in previous studies, considering as controls the forces and heat sources acting on the domain, together with the velocity, pressure, heat flux, and temperature on a portion of the boundary [3,5,9,10,11,12,13,14,15,16,17]. However, only a few studies show the numerical results of the optimal control problem for the Boussinesq equation and consider only a single control mechanism [2,11,14]. Thus, while the theoretical analysis of these control problems has been widely presented in previous studies, the implementation through an efficient numerical algorithm in a finite element code of the obtained optimality systems represents the novelty of this work. This paper aims to review the main thermal control mechanisms, showing and comparing the numerical results obtained for the different control mechanisms, objectives, and penalization parameters.

In Section 2, we first introduce the required mathematical tools such as functional spaces and their norms together with bilinear and trilinear forms that are used to express the weak formulation of the partial differential equations. In Section 3, the general forms of the optimal control problem and of the objective functional are presented. For each of the three different control mechanisms, we aim to study the optimal control problem from a mathematical point of view. To do so, we present the weak form of the boundary value problem, in order to prove the existence of solutions. We state the optimization problem and the existence of its solution using the method of Lagrange multipliers. Moreover, we present a numerical algorithm for each control type, in order to successfully solve the optimization system arising from the optimization problem. Numerical results are then presented in Section 4, considering the three thermal control mechanisms with different objectives for the temperature and velocity fields. The importance of the choice of the penalization parameter λ is taken into account, and the results are discussed for different values of the penalization parameter.

NOTATION

We use the standard notation $H^s(O)$ for a Sobolev space of order s with respect to the set O, which can be the flow domain $\Omega \subset \mathbb{R}^n$, with n=2,3, or its boundary Γ. We remark that $H^0(O)=L^2(O)$. Corresponding Sobolev spaces of vector-valued functions are denoted by $H^s(O)$. In particular, we denote the space $H^1(\Omega)$ by $\{v_i \in L^2(\Omega) || \partial v_i/\partial x_j \in L^2(\Omega) \text{for } i,j=1,\cdots,n\}$ and the subspace

$\mathbf{H}^1_{\Gamma_j}(\Omega)$ by $\{\mathbf{v} \in \mathbf{H}^1(\Omega) | \mathbf{v} = \mathbf{0} \text{ on } \Gamma_j\}$, where Γ_j is a subset of Γ. In addition, we write $\mathbf{H}^1_0(\Omega) = \mathbf{H}^1_\Gamma(\Omega)$. The dual space of $\mathbf{H}^1_{\Gamma_s}(\Omega)$ is denoted by $\mathbf{H}^{1*}_{\Gamma_s}(\Omega)$. In particular, the dual spaces of $\mathbf{H}^1(\Omega)$ and $\mathbf{H}^1_0(\Omega)$ are $\mathbf{H}^{1*}(\Omega)$ and $H^{-1}(\Omega)$, respectively. We define the space of square integrable functions having zero mean over Ω as

$$L^2_0(\Omega) = \left\{ q \in L^2(\Omega) \Big| \int_\Omega q d\mathbf{x} = 0 \right\},$$

and the solenoidal spaces as

$$\mathbf{V} = \{\mathbf{v} \in \mathbf{H}^1(\Omega) \,|\, \nabla \cdot \mathbf{v} = 0\}, \quad \mathbf{V}_0 = \{\mathbf{v} \in \mathbf{H}^1_0(\Omega) \,|\, \nabla \cdot \mathbf{v} = 0\}.$$

The norms of the functions belonging to $H^m(O)$ are denoted by $\|\cdot\|_{m,O}$. For $(fg) \in L^1(O)$ and $(u \cdot v) \in L^1(O)$, we define the scalar product as

$$(f,g)_{\mathcal{O}} = \int_{\mathcal{O}} f g d\mathbf{x}, \qquad (\mathbf{u},\mathbf{v})_{\mathcal{O}} = \int_{\mathcal{O}} \mathbf{u} \cdot \mathbf{v} d\mathbf{x}.$$

Whenever possible, we will neglect the domain label. Thus, the inner product in $L^2(\Omega)$ and $L^2(\Omega)$ are both denoted by $(\cdot,\cdot)$. This notation will also be employed to denote pairings between Sobolev spaces and their duals.

For the description of the Boussinesq system, we use the bilinear forms

$$a(\mathbf{u},\mathbf{v}) = \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} d\mathbf{x} \qquad \forall \mathbf{u},\mathbf{v} \in \mathbf{H}^1(\Omega), \tag{1}$$

$$a(T,\theta) = \int_\Omega \nabla T \cdot \nabla \theta d\mathbf{x} \qquad \forall T,\theta \in H^1(\Omega), \tag{2}$$

$$b(\mathbf{u},q) = -\int_\Omega q \nabla \cdot \mathbf{u} d\mathbf{x} \qquad \forall q \in L^2_0(\Omega), \forall \mathbf{u} \in \mathbf{H}^1(\Omega), \tag{3}$$

and the trilinear forms

$$c(\mathbf{w},\mathbf{u},\mathbf{v}) = \int_\Omega \mathbf{w} \cdot \nabla \mathbf{u} \cdot \mathbf{v} d\mathbf{x} \qquad \forall \mathbf{w},\mathbf{u},\mathbf{v} \in \mathbf{H}^1(\Omega), \tag{4}$$

$$c(\mathbf{w},T,\theta) = \int_\Omega \mathbf{w} \cdot \nabla T \theta d\mathbf{x} \qquad \forall \mathbf{w} \in \mathbf{H}^1(\Omega), \forall T,\theta \in H^1(\Omega). \tag{5}$$

These forms are continuous [18]. Note that, for all $u \in V$, $v \in H^1(\Omega)$ and $T \in H^1(\Omega)$, we have $c(u,v,v)=0$ and $c(u,T,T)=0$.

OPTIMAL CONTROL OF BOUSSINESQ EQUATIONS

In this paper, we study optimal control problems for stationary incompressible flows in mixed or natural convection regimes. In these engineering applications, the dependence on the temperature field cannot be neglected

in the Navier–Stokes equation. Thus, the temperature and velocity fields are mutually dependent through buoyancy forces and advection. These flows are defined by the following Boussinesq equations:

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{6}$$

$$\mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nu \Delta \mathbf{u} = \mathbf{f} - \beta \mathbf{g} T \qquad \text{in } \Omega, \tag{7}$$

$$\mathbf{u} \cdot \nabla T - \alpha \Delta T = Q \qquad \text{in } \Omega, \tag{8}$$

where Ω is a bounded open set in $\mathbb{R}^d$, d=2 or 3 with smoothing as necessary at boundary Γ. The operator Δ defines the Laplace operator $\nabla \cdot \nabla = \nabla^2 = \Delta$. In (6)–(8), u, p, and T denote the velocity, pressure, and temperature fields, while f is a body force, Q is a heat source, and g is the gravitational acceleration. The fluid thermal diffusivity, kinematic viscosity, and coefficient of expansion are defined by α, ν, and β, respectively. The system (6)–(8) is closed, with appropriate boundary conditions on ∂Ω. For the velocity, we set Dirichlet boundary conditions, while for the temperature field we consider a mixed boundary condition defined as

$$\begin{aligned} \mathbf{u} &= \mathbf{w} && \text{on } \partial\Omega, \\ T &= g_t && \text{on } \Gamma_d, \\ \alpha \nabla T \cdot \mathbf{n} &= g_{t,n} && \text{on } \Gamma_n. \end{aligned} \tag{9}$$

We denote by Γ_d and Γ_n the boundaries where Dirichlet and Neumann boundary conditions are applied, with $\Gamma_d \cup \Gamma_n = \Gamma = \partial\Omega$.

We formulate this control problem as a constrained minimization of the following objective functional:

$$\mathcal{T}(\mathbf{u}, T) = \frac{\alpha_u}{2} \int_{\Omega_d} |\mathbf{u} - \mathbf{u}_d|^2 d\mathbf{x} + \frac{\alpha_T}{2} \int_{\Omega_d} |T - T_d|^2 d\mathbf{x}, \tag{10}$$

subject to the Boussinesq Equations (6)–(8) imposed as constraints. In (10), the functions u_d and T_d are the given desired velocity and temperature distributions. The terms in the functional (10) measure the $L^2(\Omega)$ distance between the velocity u and the target field u_d and/or between the temperature T and the target field T_d. The non-negative penalty parameters α_u and α_T can be used to change the relative importance of the terms appearing in the definition of the functional. If $\alpha_u = 0$, we have as the objective a temperature matching case; if $\alpha_T = 0$, we consider a velocity matching case.

The control can be a volumetric heat source, a boundary temperature, or a heat flux. In all these cases, the control has to be limited to avoid unbounded solutions. To do so, we can add a constraint limiting the value of the admissible control, or we can penalize the objective functional T by adding a regularization term. With this second approach, we do not need to impose any a priori constraints on the size of the control. Let c be the control belonging to a Hilbert space $H^s(O)$. We can then define a cost functional

$$\mathcal{J}(\mathbf{u},T,c)=\frac{\alpha_u}{2}\int_{\Omega_d}|\mathbf{u}-\mathbf{u}_d|^2d\mathbf{x}+\frac{\alpha_T}{2}\int_{\Omega_d}|T-T_d|^2d\mathbf{x}+\lambda||c||_{H^s(\mathcal{O})}\,, \tag{11}$$

where the last term contains the $H^s(O)$-norm of the control c penalized with a parameter λ. The value of the parameter λ is used to change the relative importance of the objective and cost terms.

Dirichlet Boundary Control

In a Dirichlet boundary control problem, we aim to control the fluid state acting on the temperature on a portion of the boundary $\Gamma_c \subseteq \Gamma_d$. The boundary condition reported in (9) can be written in this case as

$$\mathbf{u}=\mathbf{w}\text{ on }\partial\Omega\,,\quad T=g_t\text{ on }\Gamma_i\,,\quad T=g_t+T_c\text{ on }\Gamma_c\,,\quad \alpha\nabla T\cdot\mathbf{n}=g_{t,n}\text{ on }\Gamma_n\,, \tag{12}$$

where $\Gamma_i=\Gamma_d\backslash\Gamma_c$. In (12), g_t, $g_{t,n}$, and w are given functions, while T_c is the control. Thus, Γ_c and Γ_i denote the portions of Γ_d where temperature control is and is not applied, respectively. By considering Equation (11) with s=1, the cost functional is given as follows:

$$\mathcal{J}(\mathbf{u},T,T_c)=\frac{\alpha_u}{2}\int_{\Omega_d}|\mathbf{u}-\mathbf{u}_d|^2d\mathbf{x}+\frac{\alpha_T}{2}\int_{\Omega_d}|T-T_d|^2d\mathbf{x}+\frac{\lambda}{2}\int_{\Gamma_c}(|T_c|^2+|\nabla_s T_c|^2)d\mathbf{x}\,, \tag{13}$$

where ∇_s denotes the surface gradient operator, i.e., $\nabla_s f:=\nabla f-n(n\cdot\nabla f)$. The cost contribution measures the $H^1(\Gamma_c)$-norm of the control T_c.

Weak Formulation and Lagrange Multiplier Approach

The weak form of the boundary value problem (6)–(8) and (12) is given as follows: find $(\mathbf{u},p,T)\in\mathbf{H}^1(\Omega)\times L_0^2(\Omega)\times H^1(\Omega)$ such that

$$\begin{aligned}
&va(\mathbf{u},\mathbf{v})+c(\mathbf{u},\mathbf{u},\mathbf{v})+b(\mathbf{v},p)=(\mathbf{f},\mathbf{v})-\beta(\mathbf{g}T,\mathbf{v}) && \forall\mathbf{v}\in\mathbf{H}_0^1(\Omega)\,,\\
&b(\mathbf{u},q)=0 && \forall q\in L_0^2(\Omega)\,,\\
&\alpha a(T,\varphi)+c(\mathbf{u},T,\varphi)=(Q,\varphi)+(g_{t,n},\varphi)_{\Gamma_n} && \forall\varphi\in H^1_{\Gamma_d}(\Omega)\,,\\
&(T,s_T)_{\Gamma_d}=(g_t,s_T)_{\Gamma_d}+(T_c,s_T)_{\Gamma_c} && \forall s_T\in H^{-1/2}(\Gamma_d)\,.
\end{aligned} \tag{14}$$

The existence of the solution of the system (14) has been proved in [3]. Note that the normal heat flux on Γ_d can be computed from T as

$$q_n = -\alpha \nabla T \cdot \mathbf{n}|_{\Gamma_d}. \tag{15}$$

Now, we state the optimal control problem. We look for a $(u,p,T,T_c) \in H^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times H_0^1(\Gamma_c)$ such that the cost functional (13) is minimized subject to the constraints (14). The admissible set of states and controls is

$$\begin{aligned} \mathcal{U}_{ad} = \{ &(\mathbf{u}, p, T, T_c) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times H_0^1(\Gamma_c) : \\ &\mathcal{J}(\mathbf{u}, T, T_c) < \infty \text{ and (14) is satified.} \} \end{aligned} \tag{16}$$

Then, $(\hat{\mathbf{u}}, \hat{p}, \hat{T}, \hat{T}_c) \in \mathcal{U}_{ad}$ is called an optimal solution if there exists $\varepsilon > 0$ such that

$$\begin{aligned} &\mathcal{J}(\hat{\mathbf{u}}, \hat{p}, \hat{T}, \hat{T}_c,) \le \mathcal{J}(\mathbf{u}, p, T, T_c) \quad \forall (\mathbf{u}, p, T, T_c) \in \mathcal{U}_{ad} \text{ satisfying} \\ &\|\mathbf{u} - \hat{\mathbf{u}}\|_1 + \|p - \hat{p}\|_0 + \|T - \hat{T}\|_1 + \|T_c - \hat{T}_c\|_{1,\Gamma_c} < \varepsilon \end{aligned} \tag{17}$$

The existence of at least one optimal solution $(\hat{\mathbf{u}}, \hat{p}, \hat{T}, \hat{T}_c) \in \mathcal{U}_{ad}$ was proven in [3].

We use the method of Lagrange multipliers to turn the constrained optimization problem (16) into an unconstrained one. We first show that suitable Lagrange multipliers exist. We summarize all the equations and the functional in two mappings and study their differential properties. It is convenient to define the following functional spaces:

$$\mathbf{B}_1 = \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times H_0^1(\Gamma_c) \times H^{-\frac{1}{2}}(\Gamma_d), \tag{18}$$

$$\mathbf{B}_2 = \mathbf{H}^{-1}(\Omega) \times L_0^2(\Omega) \times H_{\Gamma_i}^{1*}(\Omega) \times H^{\frac{1}{2}}(\Gamma_d), \tag{19}$$

$$\mathbf{B}_3 = \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times H_0^1(\Gamma_c) \times H^{-\frac{1}{2}}(\Gamma_d). \tag{20}$$

Let $M: B_1 \to B_2$ denote the generalized constraint equations, namely, $M(z) = l$ for $z=(u,p,T,T_c,q_n) \in B_1$ and $l=(l_1,l_2,l_3,l_4) \in B_2$ if and only if

$$\begin{aligned} &\nu a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) - (\mathbf{f}, \mathbf{v}) + \beta(\mathbf{g}T, \mathbf{v}) = (\mathbf{l}_1, \mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ &b(\mathbf{u}, q) = (l_2, q) && \forall q \in L_0^2(\Omega), \\ &\alpha a(T, \varphi) + c(\mathbf{u}, T, \varphi) - (Q, \varphi) - (g_{t,n}, \varphi)_{\Gamma_n} - (q_n, \varphi)_{\Gamma_c} = (l_3, \varphi) && \forall \varphi \in H_{\Gamma_i}^1(\Omega), \\ &(T, s_T)_{\Gamma_d} - (g_t, s_T)_{\Gamma_d} - (T_c, s_T)_{\Gamma_c} = (l_4, s_T)_{\Gamma_d} && \forall s_T \in H^{-1/2}(\Gamma_d). \end{aligned} \tag{21}$$

Thus, the constraint (14) can be expressed as M(z)=0. Let $(\hat{\mathbf{u}}, \hat{p}, \hat{T}, \hat{T}_c) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times H_0^1(\Gamma_c)$ denote an optimal solution in the sense of (17). Then, consider the nonlinear operator $N : \mathbf{B}_1 \to \mathbb{R} \times \mathbf{B}_2$ defined by

$$N(\mathbf{u}, p, T, T_c, q_n) = \begin{pmatrix} \mathcal{J}(\mathbf{u}, T, T_c) - \mathcal{J}(\hat{\mathbf{u}}, \hat{T}, \hat{T}_c) \\ M(\mathbf{u}, p, T, T_c, q_n) \end{pmatrix}. \tag{22}$$

Given $\mathbf{z} = (\mathbf{u}, p, T, T_c, q_n) \in \mathbf{B}_1$, the operator M′(z):$B_3 \to B_2$ may be defined as $M'(\mathbf{z}) \cdot \tilde{\mathbf{z}} = \tilde{\mathbf{l}}$ for $\tilde{\mathbf{z}} = (\tilde{\mathbf{u}}, \tilde{p}, \tilde{T}, \tilde{T}_c, \tilde{q}_n) \in \mathbf{B}_3$ and $\tilde{\mathbf{l}} = (\tilde{l}_1, \tilde{l}_2, \tilde{l}_3, \tilde{l}_4) \in \mathbf{B}_2$ if and only if

$$\begin{aligned}
&\nu a(\tilde{\mathbf{u}}, \mathbf{v}) + c(\tilde{\mathbf{u}}, \mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \tilde{\mathbf{u}}, \mathbf{v}) + b(\mathbf{v}, \tilde{p}) + \beta(\mathbf{g}\tilde{T}, \mathbf{v}) = (\tilde{l}_1, \mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\
&b(\tilde{\mathbf{u}}, q) = (\tilde{l}_2, q) && \forall q \in L_0^2(\Omega), \\
&\alpha a(\tilde{T}, \varphi) + c(\tilde{\mathbf{u}}, T, \varphi) + c(\mathbf{u}, \tilde{T}, \varphi) - (\tilde{q}_n, \varphi)_{\Gamma_c} = (\tilde{l}_3, \varphi) && \forall \varphi \in H_{\Gamma_i}^1(\Omega), \\
&(\tilde{T}, s_T)_{\Gamma_d} - (\tilde{T}_c, s_T)_{\Gamma_c} = (\tilde{l}_4, s_T)_{\Gamma_d} && \forall s_T \in H^{-1/2}(\Gamma_d).
\end{aligned} \tag{23}$$

The operator $N'(\mathbf{z}) : \mathbf{B}_3 \to \mathbb{R} \times \mathbf{B}_2$ may be defined as $N'(\mathbf{z}) \cdot \tilde{\mathbf{z}} = (\tilde{a}, \tilde{\mathbf{l}})$ for $\tilde{a} \in \mathbb{R}$ if and only if

$$\begin{aligned}
&\alpha_u(\mathbf{u} - \mathbf{u}_d, \tilde{\mathbf{u}})_{\Omega_d} + \alpha_T(T - T_d, \tilde{T})_{\Omega_d} + \lambda(T_c, \tilde{T}_c)_{\Gamma_c} + \\
&\quad + \lambda(\nabla_s T_c, \nabla_s \tilde{T}_c)_{\Gamma_c} = \tilde{a} \\
&\nu a(\tilde{\mathbf{u}}, \mathbf{v}) + c(\tilde{\mathbf{u}}, \mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \tilde{\mathbf{u}}, \mathbf{v}) + b(\mathbf{v}, \tilde{p}) + \beta(\mathbf{g}\tilde{T}, \mathbf{v}) = (\tilde{l}_1, \mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\
&b(\tilde{\mathbf{u}}, q) = (\tilde{l}_2, q) && \forall q \in L_0^2(\Omega), \\
&\alpha a(\tilde{T}, \varphi) + c(\tilde{\mathbf{u}}, T, \varphi) + c(\mathbf{u}, \tilde{T}, \varphi) - (\tilde{q}_n, \varphi)_{\Gamma_c} = (\tilde{l}_3, \varphi) && \forall \varphi \in H_{\Gamma_i}^1(\Omega), \\
&(\tilde{T}, s_T)_{\Gamma_d} - (\tilde{T}_c, s_T)_{\Gamma_c} = (\tilde{l}_4, s_T)_{\Gamma_d} && \forall s_T \in H^{-1/2}(\Gamma_d).
\end{aligned}$$

The differential operator M′ is characterized by non-coercive elliptic equations. The advection terms in these equations are driven by the velocity field u∈H^1(Ω). Thus, the existence result for this class of equations is not trivial and cannot be obtained in the Lax–Milgram setting. However, by using a Leray–Schauder topological degree argument, we can introduce the following statements.

Let $\Omega \subset \mathbb{R}^n$ be a bounded open subset with boundary Γ. Let $\Gamma_d \subset \Gamma$ be a set with positive measure and $\Gamma_n \subseteq \Gamma \setminus \Gamma_d$. Consider

$$\begin{aligned}
-\nabla \cdot (A^T \nabla y) + (\mathbf{u} \cdot \nabla) y + by &= f && \text{in } \Omega, \\
y &= y_1 && \text{on } \Gamma_d, \\
A^T \nabla y \cdot \mathbf{n} &= y_n && \text{on } \Gamma_n,
\end{aligned} \tag{24}$$

with $b \in L^{n_*/2}(\Omega)$, $b \geq 0$ a.e. on Ω, $\mathbf{u} \in \mathbf{L}^{n_*}(\Omega)$, and $f \in H^{1*}_{\Gamma_D}(\Omega)$, where $n_*=n$ when $n\geq 3$, $n_* \in]2,\infty[$ when $n=2$. Based on the Leray–Schauder topological degree argument in [19], if A is a function which satisfies these two properties and:

- $\exists \alpha_A>0$ such that $A(x)\xi\cdot\xi \geq \alpha_A|\xi|^2$ for a.e. $x\in\Omega$ and for all $\xi \in \mathbb{R}^n$;
- $\exists \Lambda_A>0$ such that $|A(x)|\leq \Lambda_A$ for a.e. $x\in\Omega$;

then, there exists a unique solution $y\in H^1(\Omega)$ of (24).

Furthermore, let $z_0 \in B_1$. Then we have that the operator $M'(z_0)$ has closed range in B_2 and the operator $N'(z_0)$ has closed range but is not in $\mathbb{R}\times B_2$. This allows us to find the Lagrange multipliers and the final optimality system. Let $\hat{\mathbf{z}} = (\hat{\mathbf{u}}, \hat{p}, \hat{T}, \hat{T}_c, \hat{q}_n) \in \mathbf{H}^1(\Omega) \times L^2_0(\Omega) \times H^1(\Omega) \times H^1_0(\Gamma_c) \times H^{-1/2}(\Gamma_c)$ denote an optimal solution in the sense of (17). Then, there exists a nonzero Lagrange multiplier $(\Lambda, \hat{\mathbf{u}}_a, \hat{p}_a, \hat{T}_a, \hat{q}_a) \in \mathbb{R} \times \mathbf{B}_2^*$ satisfying the Euler equations

$$\Lambda \mathcal{J}'(\hat{\mathbf{u}}, \hat{T}, \hat{T}_c)\cdot \tilde{\mathbf{z}} + \langle (\hat{\mathbf{u}}_a, \hat{p}_a, \hat{T}_a, \hat{q}_a), M'(\hat{\mathbf{z}})\cdot\tilde{\mathbf{z}}\rangle = 0\,, \quad \forall \tilde{\mathbf{z}} \in \mathbf{B}_3\,, \tag{25}$$

where $\langle\cdot,\cdot\rangle$ denotes the duality pairing between $\mathbf{B}_2$ and $\mathbf{B}_2^*$. For details on all the theoretical procedure regarding the existence of the Lagrange multipliers, the interested reader can consult [20].

The Optimality System

Now, we derive the optimality system using (25), and we drop the $(\cdot\hat{})$ notation for the optimal solution. The Euler Equation (25) are equivalent to

$$\begin{aligned}
&\alpha_u \Lambda(\mathbf{u}-\mathbf{u}_d, \tilde{\mathbf{u}})_{\Omega_d} + \alpha_T \Lambda(T-T_d, \tilde{T})_{\Omega_d} + \Lambda\lambda(T_c, \tilde{T}_c)_{\Gamma_c} + \Lambda\lambda(\nabla_s T_c, \nabla_s \tilde{T}_c)_{\Gamma_c} + \\
&\quad + b(\tilde{\mathbf{u}}, p_a) + \nu a(\tilde{\mathbf{u}}, \mathbf{u}_a) + c(\tilde{\mathbf{u}}, \mathbf{u}, \mathbf{u}_a) + b(\mathbf{u}_a, \tilde{p}) + c(\mathbf{u}, \tilde{\mathbf{u}}, \mathbf{u}_a) + \beta(\mathbf{g}\tilde{T}, \mathbf{u}_a) + \\
&\quad + \alpha a(\tilde{T}, T_a) + c(\tilde{\mathbf{u}}, T, T_a) + c(\mathbf{u}, \tilde{T}, T_a) - (\tilde{q}_n, T_a)_{\Gamma_c} + (\tilde{T}, q_a)_{\Gamma_d} - (\tilde{T}_c, q_a)_{\Gamma_c} = 0\,.
\end{aligned} \tag{26}$$

By extracting the terms involved in the same variation and setting $\Lambda=-1$, we obtain the following equations:

$$
\begin{aligned}
&\nu a(\tilde{\mathbf{u}}, \mathbf{u}_a) + c(\mathbf{u}, \tilde{\mathbf{u}}, \mathbf{u}_a) + c(\tilde{\mathbf{u}}, \mathbf{u}, \mathbf{u}_a) + b(\tilde{\mathbf{u}}, p_a) = \\
&\qquad = \alpha_u(\mathbf{u} - \mathbf{u}_d, \tilde{\mathbf{u}})_{\Omega_d} - c(\tilde{\mathbf{u}}; T, T_a) \qquad \forall \tilde{\mathbf{u}} \in \mathbf{H}_0^1(\Omega)\,, \\
&b(\mathbf{u}_a, \tilde{p}) = 0 \qquad \forall \tilde{p} \in L_0^2(\Omega)\,, \\
&\alpha a(\tilde{T}, T_a) + c(\mathbf{u}, \tilde{T}, T_a) + (\tilde{T}, q_a)_{\Gamma_c} = \\
&\qquad = -(\beta \mathbf{g}\tilde{T}, \mathbf{u}_a) + \alpha_T(T - T_d, \tilde{T})_{\Omega_d} \qquad \forall \tilde{T} \in H^1_{\Gamma_i}(\Omega)\,, \\
&(T_a, \tilde{q}_n)_{\Gamma_c} = 0\,, \qquad \forall \tilde{q}_n \in H^{-1/2}(\Gamma_c)\,,
\end{aligned}
\tag{27}
$$

and the control equation

$$
\lambda(T_c, \tilde{T}_c)_{\Gamma_c} + \lambda(\nabla_s T_c, \nabla_s \tilde{T}_c)_{\Gamma_c} + (q_a, \tilde{T}_c)_{\Gamma_c} = 0 \qquad \forall \tilde{T}_c \in H_0^1(\Gamma_c)\,, \tag{28}
$$

with $q_a = -\alpha \nabla T_a \cdot \mathbf{n}|_{\Gamma_c}$ on Γ_c. The necessary conditions for an optimum are that Equations (14) and (27) are satisfied. This system of equations is called the optimality system. By applying integration by parts, it is easy to show that the system constitutes a weak formulation of the boundary value problem for the state equations

$$
\begin{aligned}
&\mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nu \Delta \mathbf{u} = \mathbf{f} - \beta \mathbf{g} T\,, \\
&\nabla \cdot \mathbf{u} = 0\,, \\
&\mathbf{u} \cdot \nabla T - \alpha \Delta T = Q\,, \\
&\mathbf{u} = \mathbf{w} \text{ on } \Gamma\,, \quad \alpha \nabla T \cdot \mathbf{n}|_{\Gamma_n} = g_{t,n} \text{ on } \Gamma_n\,, \quad T = g_t \text{ on } \Gamma_i\,, \quad T = g_t + T_c \text{ on } \Gamma_c\,,
\end{aligned}
\tag{29}
$$

the adjoint equations

$$
\begin{aligned}
&\mathbf{u}_a \cdot (\nabla \mathbf{u})^T - \mathbf{u} \cdot \nabla \mathbf{u}_a + \nabla p_a - \nu \Delta \mathbf{u}_a = -T \nabla T_a + \alpha_u(\mathbf{u} - \mathbf{u}_d)\,, \\
&\nabla \cdot \mathbf{u}_a = 0\,, \\
&-\alpha \Delta T_a - \mathbf{u} \cdot \nabla T_a = -\beta \mathbf{g} \cdot \mathbf{u}_a + \alpha_T(T - T_d)\,, \\
&\mathbf{u}_a = 0 \text{ on } \Gamma\,, \quad \nabla T_a \cdot \mathbf{n}|_{\Gamma_n} = 0 \text{ on } \Gamma_n\,, \quad T_a = 0 \text{ on } \Gamma_d\,,
\end{aligned}
\tag{30}
$$

and the control equation

$$
\begin{aligned}
&-\Delta_s T_c + T_c - \frac{\alpha \nabla T_a \cdot \mathbf{n}|_{\Gamma_c}}{\lambda} = 0 \text{ on } \Gamma_c\,, \\
&T_c = 0 \text{ on } \partial\Gamma_c\,,
\end{aligned}
\tag{31}
$$

where Δ_s denotes the surface Laplacian. The optimality system in the strong form consists of the Boussinesq system (29), the adjoint of the Boussinesq Equation (30), and the control Equation (31).

Numerical Algorithm

The optimality system consists of three groups of equations: the state Equation (14), the adjoint state Equation (27), and the optimality conditions for T_c (28). Due to the nonlinearity and large dimension of this system, a one-shot solver cannot be implemented. We may construct an iterative method to iterate among the three groups of equations so that at each iteration we are dealing with a smaller-sized system of equations. We consider a gradient method for the solution of the optimality problem, and the gradient of the functional is determined with the help of the solution of the adjoint system.

Let us consider the gradient method for the following minimization problem: find $T_c \in H_0^1(\Gamma_c)$ such that $\mathcal{F}(T_c) := \mathcal{J}(\mathbf{u}(T_c), T(T_c), T_c)$ is minimized. Given $T_c^{(0)}$, we can define the sequence

$$T_c^{(n+1)} = T_c^{(n)} - \rho^{(n)} \frac{d\mathcal{F}(T_c^{(n)})}{dT_c^{(n)}}, \tag{32}$$

recursively, where $\rho^{(n)}$ is a variable step size. Let $\hat{T}_c$ be a solution of the minimization problem. Thus, the following necessary condition holds

$$\frac{d\mathcal{F}(\hat{T}_c)}{d\hat{T}_c} = \frac{d\mathcal{J}(\mathbf{u}(\hat{T}_c), T(\hat{T}_c), \hat{T}_c)}{d\hat{T}_c} = 0, \tag{33}$$

and at the optimum state the equality $T_c^{(n+1)} = T_c^{(n)}$ holds. For each fixed T_c, the Gâteaux derivative $(d\mathcal{F}(T_c)/dT_c) \cdot \widetilde{T}_c$ for every direction $\widetilde{T}_c \in H^1(\Gamma_c)$ may be computed as

$$\frac{d\mathcal{F}(T_c)}{dT_c} \cdot \widetilde{T}_c = \lambda(\nabla_s T_c, \nabla_s \widetilde{T}_c)_{\Gamma_c} + \lambda(T_c, \widetilde{T}_c)_{\Gamma_c} + (\widetilde{T}_c, q_a)_{\Gamma_c}, \tag{34}$$

or

$$\frac{d\mathcal{F}(T_c)}{dT_c} = -\lambda \Delta_s T_c + \lambda T_c + q_a. \tag{35}$$

Therefore, by combining (32) and (35), we implemented the following optimization algorithm.

(a) Initial step:

- choose tolerance τ and $T_c^{(0)}$; set n=0 and $\rho^{(0)}$=1;
- solve for $(u^{(0)},p^{(0)},T^{(0)})$ from (14) with $T_c=T^{(0)}c$;
- evaluate $\mathcal{J}^{(0)} = \mathcal{J}(\mathbf{u}^{(0)}, T^{(0)}, T_c^{(0)})$ using (13).

(b) Main loop:

- set n=n+1;
- solve for $(\mathbf{u}_a^{(n)}, p_a^{(n)}, T_a^{(n)})$ from (27);
- solve for $T_c^{(n)}$ from

$$T_c^{(n)} = T_c^{(n-1)} - \rho^{(n)}\Big(- \Delta_s T_c^{(n-1)} + T_c^{(n-1)} + \frac{\alpha}{\lambda}\nabla T_a^{(n)} \cdot \mathbf{n}|_{\Gamma_c}\Big), \tag{36}$$

or

$$-\Delta_s T_c^{(n)} + T_c^{(n)} = -\Delta_s T_c^{(n-1)} + T_c^{(n-1)} - \rho^{(n)}\Big(- \Delta_s T_c^{(n-1)} + T_c^{(n-1)} + \frac{\alpha}{\lambda}\nabla T_a^{(n)} \cdot \mathbf{n}|_{\Gamma_c}\Big); \tag{37}$$

- solve for $(\mathbf{u}^{(n)}, p^{(n)}, T^{(n)})$ from (14) with $T_c = T_c^{(n)}$;
- evaluate $\mathcal{J}^{(n)} = \mathcal{J}(\mathbf{u}^{(n)}, T^{(n)}, T_c^{(n)})$ using (13);

(i) if $\mathcal{J}^{(n)} > \mathcal{J}^{(n-1)}$, set $\rho^{(n)} = 0.5\rho^{(n)}$ and go to step (b) 3.;

(ii) if $\mathcal{J}^{(n)} < \mathcal{J}^{(n-1)}$, set $\rho^{(n+1)} = 1$ and go to step (b) 1.;

(iii) if $|\mathcal{J}^{(n)} - \mathcal{J}^{(n-1)}|/|\mathcal{J}^{(n)}| < \tau$ stop.

In this algorithm, we propose two different methods, given by (36) and (37), for the control update. With the method in (37), we enforce the belonging of T_c to $H_0^1(\Gamma_c)$ and we give more regularity to the control. The convergence of the algorithm was proven in [3]. The finite element discretization of the optimality system and an estimation of the approximation error were analyzed in [11].

Neumann Boundary Control

In a Neumann boundary control problem, we aim to control the state by acting on the heat flux on a portion of the boundary $\Gamma_c \subseteq \Gamma_n$. The general boundary conditions reported in (9) can be written in this case as

$$\mathbf{u} = \mathbf{w} \text{ on } \partial\Omega, \quad T = g_t \text{ on } \Gamma_d, \quad \alpha\nabla T \cdot \mathbf{n} = g_{t,n} \text{ on } \Gamma_i, \quad \alpha\nabla T \cdot \mathbf{n} = h \text{ on } \Gamma_c, \tag{38}$$

where $\Gamma_i=\Gamma_n\backslash\Gamma_c$. In (12), g_t, $g_{t,n}$, and w are given functions, while h is the control. Thus, Γ_i and Γ_c denote the portions of Γ_n where the control is applied or not, respectively.

The cost functional is given as follows:

$$\mathcal{J}(\mathbf{u},T,h)=\frac{\alpha_u}{2}\int_{\Omega_d}|\mathbf{u}-\mathbf{u}_d|^2d\mathbf{x}+\frac{\alpha_T}{2}\int_{\Omega_d}|T-T_d|^2d\mathbf{x}+\frac{\lambda}{2}\int_{\Gamma_c}|h|^2d\mathbf{x}. \tag{39}$$

The cost contribution measures the $L^2(\Gamma_c)$-norm of the control h.

Weak Formulation and Lagrange Multiplier Approach

The weak form of the boundary value problem (6)–(8) and (38) is given as follows: find $(\mathbf{u},p,T)\in\mathbf{H}^1(\Omega)\times L_0^2(\Omega)\times H^1(\Omega)$ such that

$$\begin{aligned}
&\nu a(\mathbf{u},\mathbf{v})+c(\mathbf{u},\mathbf{u},\mathbf{v})+b(\mathbf{v},p)=(\mathbf{f},\mathbf{v})-\beta(\mathbf{g}T,\mathbf{v}) && \forall\mathbf{v}\in\mathbf{H}_0^1(\Omega),\\
&b(\mathbf{u},q)=0 && \forall q\in L_0^2(\Omega),\\
&\alpha a(T,\varphi)+c(\mathbf{u},T,\varphi)=(Q,\varphi)+(g_{t,n},\varphi)_{\Gamma_i}+(h,\varphi)_{\Gamma_c} && \forall\varphi\in H^1_{\Gamma_d}(\Omega).
\end{aligned} \tag{40}$$

The existence of the solution of the system (40) was proved in [9] (see Preposition 2.3).

Now, we state the optimal control problem: we look for a $(u,p,T,h)\in H^1(\Omega)\times L_0^2(\Omega)\times H^1(\Omega)\times L^2(\Gamma_c)$ such that the cost functional (39) is minimized subject to the constraints (40). The admissible set of states and controls is

$$\begin{aligned}
\mathcal{U}_{ad}=\{&(\mathbf{u},p,T,h)\in\mathbf{H}^1(\Omega)\times L_0^2(\Omega)\times H^1(\Omega)\times L^2(\Gamma_c):\\
&\mathcal{J}(\mathbf{u},T,h)<\infty \text{ and (40) is satified.}\}
\end{aligned} \tag{41}$$

Then, $(\hat{\mathbf{u}},\hat{p},\hat{T},\hat{h})\in\mathcal{U}_{ad}$ is called an optimal solution if there exists $\varepsilon>0$ such that

$$\begin{aligned}
&\mathcal{J}(\hat{\mathbf{u}},\hat{p},\hat{T},\hat{h})\le\mathcal{J}(\mathbf{u},p,T,h)\quad\forall(\mathbf{u},p,T,h)\in\mathcal{U}_{ad}\text{ satisfying}\\
&\qquad\|\mathbf{u}-\hat{\mathbf{u}}\|_1+\|p-\hat{p}\|_0+\|T-\hat{T}\|_1+\|h-\hat{h}\|_{0,\Gamma_c}<\varepsilon.
\end{aligned} \tag{42}$$

The existence of at least one optimal solution $(\hat{\mathbf{u}},\hat{p},\hat{T},\hat{h})\in\mathcal{U}_{ad}$ was proved in [9].

In addition, for the Neumann control, we consider all the constraint equations and the functional to study their differential properties. We define the following functional spaces:

$$\mathbf{B}_1=\mathbf{H}^1(\Omega)\times L_0^2(\Omega)\times H^1(\Omega)\times L^2(\Gamma_c), \tag{43}$$

$$\mathbf{B}_2 = \mathbf{H}^{-1}(\Omega) \times L_0^2(\Omega) \times H_{\Gamma_d}^{1*}(\Omega), \tag{44}$$

$$\mathbf{B}_3 = \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \times H_{\Gamma_d}^1(\Omega) \times L^2(\Gamma_c). \tag{45}$$

Let $M:B_1 \to B_2$ denote the generalized constraint equations, namely, $M(z)=l$ for $z=(u,p,T,h)\in B_1$ and $l=(l_1,l_2,l_3)\in B_2$ if and only if

$$\begin{aligned} &\nu a(\mathbf{u},\mathbf{v}) + c(\mathbf{u},\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) - (\mathbf{f},\mathbf{v}) + \beta(\mathbf{g}T,\mathbf{v}) = (\mathbf{l}_1,\mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ &b(\mathbf{u},q) = (l_2,q) && \forall q \in L_0^2(\Omega), \\ &\alpha a(T,\varphi) + c(\mathbf{u},T,\varphi) - (Q,\varphi) - (g_{t,n},\varphi)_{\Gamma_i} - (h,\varphi)_{\Gamma_c} = (l_3,\varphi) && \forall \varphi \in H_{\Gamma_d}^1(\Omega). \end{aligned} \tag{46}$$

Thus, the constraints (40) can be expressed as $M(\mathbf{z}) = \mathbf{0}$. Let $(\hat{\mathbf{u}},\hat{p},\hat{T},\hat{h}) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times L^2(\Gamma_c)$ denote an optimal solution in the sense of (42). Then, consider the nonlinear operator $N : \mathbf{B}_1 \to \mathbb{R} \times \mathbf{B}_2$ defined by

$$N(\mathbf{u},p,T,h) = \begin{pmatrix} \mathcal{J}(\mathbf{u},T,h) - \mathcal{J}(\hat{\mathbf{u}},\hat{T},\hat{h}) \\ M(\mathbf{u},p,T,h) \end{pmatrix}. \tag{47}$$

Given $z=(u,p,T,h)\in B_1$, the operator $M'(z):B_1 \to B_2$ may be defined as $M'(\mathbf{z})\cdot\tilde{\mathbf{z}} = \tilde{\mathbf{l}}$ for $\tilde{\mathbf{z}} = (\tilde{\mathbf{u}},\tilde{p},\tilde{T},\tilde{h}) \in \mathbf{B}_3$ and $\tilde{\mathbf{l}} = (\tilde{\mathbf{l}}_1,\tilde{l}_2,\tilde{l}_3) \in \mathbf{B}_2$ if and only if

$$\begin{aligned} &\nu a(\tilde{\mathbf{u}},\mathbf{v}) + c(\tilde{\mathbf{u}},\mathbf{u},\mathbf{v}) + c(\mathbf{u},\tilde{\mathbf{u}},\mathbf{v}) + b(\mathbf{v},\tilde{p}) + \beta(\mathbf{g}\tilde{T},\mathbf{v}) = (\tilde{\mathbf{l}}_1,\mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ &b(\tilde{\mathbf{u}},q) = (\tilde{l}_2,q) && \forall q \in L_0^2(\Omega), \\ &\alpha a(\tilde{T},\varphi) + c(\tilde{\mathbf{u}},T,\varphi) + c(\mathbf{u},\tilde{T},\varphi) - (\tilde{h},\varphi)_{\Gamma_c} = (\tilde{l}_3,\varphi) && \forall \varphi \in H_{\Gamma_d}^1(\Omega). \end{aligned} \tag{48}$$

The operator $N'(\mathbf{z}) : \mathbf{B}_3 \to \mathbb{R} \times \mathbf{B}_2$ may be defined as $N'(\mathbf{z})\cdot\tilde{\mathbf{z}} = (\tilde{a},\tilde{\mathbf{l}})$ for $\tilde{a} \in \mathbb{R}$ if and only if

$$\begin{aligned} &\alpha_u(\mathbf{u}-\mathbf{u}_d,\tilde{\mathbf{u}})_{\Omega_d} + \alpha_T(T-T_d,\tilde{T})_{\Omega_d} + \lambda(h,\tilde{h})_{\Gamma_c} = \tilde{a} \\ &\nu a(\tilde{\mathbf{u}},\mathbf{v}) + c(\tilde{\mathbf{u}},\mathbf{u},\mathbf{v}) + c(\mathbf{u},\tilde{\mathbf{u}},\mathbf{v}) + b(\mathbf{v},\tilde{p}) + \beta(\mathbf{g}\tilde{T},\mathbf{v}) = (\tilde{\mathbf{l}}_1,\mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ &b(\tilde{\mathbf{u}},q) = (\tilde{l}_2,q) && \forall q \in L_0^2(\Omega), \\ &\alpha a(\tilde{T},\varphi) + c(\tilde{\mathbf{u}},T,\varphi) + c(\mathbf{u},\tilde{T},\varphi) - (\tilde{h},\varphi)_{\Gamma_c} = (\tilde{l}_3,\varphi) && \forall \varphi \in H_{\Gamma_d}^1(\Omega). \end{aligned} \tag{49}$$

Let $z_0\in B_1$. Then, we have that the operator $M'(z_0)$ has closed range in B_2 and the operator $N'(z_0)$ has closed range but is not in $\mathbb{R}\times B2$. This follows standard techniques (see [21]). Therefore, let $\hat{\mathbf{z}} = (\hat{\mathbf{u}},\hat{p},\hat{T},\hat{h}) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times L^2(\Gamma_c)$ denote an optimal solution satisfying (42). Then, there exists a nonzero Lagrange multiplier $(\Lambda,\hat{\mathbf{u}}_a,\hat{p}_a,\hat{T}_a) \in \mathbb{R} \times \mathbf{B}_2^*$ satisfying the Euler equations

$$\Lambda\mathcal{J}'(\hat{\mathbf{u}},\hat{T},\hat{h})\cdot\tilde{\mathbf{z}} + \langle(\hat{\mathbf{u}}_a,\hat{p}_a,\hat{T}_a), M'(\hat{\mathbf{z}})\cdot\tilde{\mathbf{z}}\rangle = 0, \quad \forall \tilde{\mathbf{z}} \in \mathbf{B}_3, \tag{50}$$

where $\langle\cdot,\cdot\rangle$ denotes the duality pairing between B_2 and $\mathbf{B}_2^*$.

The Optimality System

We drop the $\widehat{(\cdot)}$ notation for the optimal solution and derive now the optimality system using (50). The Euler Equation (50) are equivalent to

$$\begin{aligned}\alpha_u\Lambda(\mathbf{u}-\mathbf{u}_d,\tilde{\mathbf{u}})_{\Omega_d}+\alpha_T\Lambda(T-T_d,\tilde{T})_{\Omega_d}+\Lambda\lambda(h,\tilde{h})_{\Gamma_c}+b(\tilde{\mathbf{u}},p_a)+\nu a(\tilde{\mathbf{u}},\mathbf{u}_a)+\\ +c(\tilde{\mathbf{u}},\mathbf{u},\mathbf{u}_a)+b(\mathbf{u}_a,\tilde{p})+c(\mathbf{u},\tilde{\mathbf{u}},\mathbf{u}_a)+\beta(\mathbf{g}\tilde{T},\mathbf{u}_a)+\alpha a(\tilde{T},T_a)+\\ +c(\tilde{\mathbf{u}},T,T_a)+c(\mathbf{u},\tilde{T},T_a)-(\tilde{h},T_a)_{\Gamma_c}=0\,.\end{aligned} \tag{51}$$

By extracting the terms involved in the same variation and setting $\Lambda=-1$, we obtain the following equations:

$$\begin{aligned}&\nu a(\tilde{\mathbf{u}},\mathbf{u}_a)+c(\mathbf{u},\tilde{\mathbf{u}},\mathbf{u}_a)+c(\tilde{\mathbf{u}},\mathbf{u},\mathbf{u}_a)+b(\tilde{\mathbf{u}},p_a)d=\\ &\quad=\alpha_u(\mathbf{u}-\mathbf{u}_d,\tilde{\mathbf{u}})_{\Omega_d}-c(\tilde{\mathbf{u}};T,T_a) && \forall\tilde{\mathbf{u}}\in\mathbf{H}_0^1(\Omega)\,,\\ &b(\mathbf{u}_a,\tilde{p})=0 && \forall\tilde{p}\in L_0^2(\Omega)\,,\\ &\alpha a(\tilde{T},T_a)+c(\mathbf{u},\tilde{T},T_a)=-(\beta\mathbf{g}\tilde{T},\mathbf{u}_a)+\alpha_T(T-T_d,\tilde{T})_{\Omega_d} && \forall\tilde{T}\in H^1_{\Gamma_d}(\Omega)\,,\end{aligned} \tag{52}$$

and the control equation

$$\lambda(h,\tilde{h})_{\Gamma_c}+(\tilde{h},T_a)_{\Gamma_c}=0 \qquad\qquad \forall\tilde{h}\in L^2(\Gamma_c)\,. \tag{53}$$

The necessary conditions for an optimum are that Equations (40) and (52) are satisfied. This system of equations is called the optimality system. Integrations by parts may be used to show that the system constitutes a weak formulation of the boundary value problem

$$\begin{aligned}&\mathbf{u}\cdot\nabla\mathbf{u}+\nabla p-\nu\Delta\mathbf{u}=\mathbf{f}-\beta\mathbf{g}T\,,\\ &\nabla\cdot\mathbf{u}=0\,,\\ &\mathbf{u}\cdot\nabla T-\alpha\Delta T=Q\,,\\ &\mathbf{u}=\mathbf{w}\text{ on }\Gamma\,,\quad \alpha\nabla T\cdot\mathbf{n}|_{\Gamma_i}=g_{t,n}\text{ on }\Gamma_i\,,\quad \alpha\nabla T\cdot\mathbf{n}|_{\Gamma_c}=h\text{ on }\Gamma_c\,,\quad T=g_t\text{ on }\Gamma_d\,,\end{aligned} \tag{54}$$

the adjoint equations

$$\begin{aligned}&\mathbf{u}_a\cdot(\nabla\mathbf{u})^T-\mathbf{u}\cdot\nabla\mathbf{u}_a+\nabla p_a-\nu\Delta\mathbf{u}_a=-T\nabla T_a+\alpha_u(\mathbf{u}-\mathbf{u}_d)\,,\\ &\nabla\cdot\mathbf{u}_a=0\,,\\ &-\alpha\Delta T_a-\mathbf{u}\cdot\nabla T_a=-\beta\mathbf{g}\cdot\mathbf{u}_a+\alpha_T(T-T_d)\,,\\ &\mathbf{u}_a=0\text{ on }\Gamma\,,\quad \nabla T_a\cdot\mathbf{n}|_{\Gamma_n}=0\text{ on }\Gamma_n\,,\quad T_a=0\text{ on }\Gamma_d\,,\end{aligned} \tag{55}$$

and the control equation

$$h=-\frac{T_a}{\lambda}\quad\text{on }\Gamma_c\,. \tag{56}$$

The optimality system in the strong form consists of the Boussinesq system (54), the adjoint of Boussinesq Equation (55), and the control Equation (56).

Numerical Algorithm

We consider the gradient method for the following minimization problem: find $h \in L^2(\Gamma_c)$ such that $\mathcal{F}(h) := \mathcal{J}(\mathbf{u}(h), T(h), h)$ is minimized. Given $h^{(0)}$, we can define the sequence

$$h^{(n+1)} = h^{(n)} - \rho^{(n)} \frac{d\mathcal{F}(h^{(n)})}{dh^{(n)}}, \tag{57}$$

recursively, where $\rho^{(n)}$ is a variable step size. For each fixed T_c, the Gâteaux derivative $(d\mathcal{F}(h)/dh) \cdot \widetilde{h}$ for every direction $\widetilde{h} \in L^2(\Gamma_c)$ may be computed as

$$\frac{d\mathcal{F}(h)}{dh} \cdot \widetilde{h} = \lambda (h, \widetilde{h})_{\Gamma_c} + (\widetilde{h}, T_a)_{\Gamma_c}, \tag{58}$$

or

$$\frac{d\mathcal{F}(h)}{dh} = h + \frac{T_a}{\lambda}. \tag{59}$$

The optimization algorithm is then given as follows

(a) Initial step:

1. choose tolerance τ and $h^{(0)}$; set n=0 and $\rho^{(0)}=1$;
2. solve for $(u^{(0)},p^{(0)},T^{(0)})$ from (40) with $h=h^{(0)}$;
3. evaluate $J^{(0)}=J(u^{(0)},T^{(0)},h^{(0)})$ using (39).

(b) Main loop:

4. set n=n+1;
5. solve for $(\mathbf{u}_a^{(n)}, p_a^{(n)}, T_a^{(n)})$ from (52);
6. solve for $h^{(n)}$ from

$$h^{(n)} = h^{(n-1)} - \rho^{(n)} \left(h^{(n-1)} + \frac{T_a^{(n)}}{\lambda} \right); \tag{60}$$

7. solve for $(u^{(n)},p^{(n)},T^{(n)})$ from (40) with $h=h^{(n)}$;
8. evaluate $\mathcal{J}^{(n)} = \mathcal{J}(\mathbf{u}^{(n)}, T^{(n)}, h^{(n)})$ using (39);

(i) if $\mathcal{J}^{(n)} > \mathcal{J}^{(n-1)}$, set $\rho^{(n)} = 0.5\rho^{(n)}$ and go to step (b) 3.;

(ii) if $\mathcal{J}^{(n)} < \mathcal{J}^{(n-1)}$, set $\rho^{(n+1)} = 1$ and go to step (b) 1.;
(iii) if $|\mathcal{J}^{(n)} - \mathcal{J}^{(n-1)}|/|\mathcal{J}^{(n)}| < \tau$ stop.

Distributed Control

A distributed control problem aims to control the flow state using a heat source acting on the domain Ω as a control mechanism. In (8), the heat source Q is the control of the optimal control problem. The boundary conditions are those reported in (9), where w, g_t, and $g_{t,n}$ are given functions. The cost functional is formulated as

$$\mathcal{J}(\mathbf{u}, T, Q) = \frac{\alpha_u}{2}\int_{\Omega_d} |\mathbf{u} - \mathbf{u}_d|^2 d\mathbf{x} + \frac{\alpha_T}{2}\int_{\Omega_d} |T - T_d|^2 d\mathbf{x} + \frac{\lambda}{2}\int_{\Omega} |Q|^2 d\mathbf{x}, \tag{61}$$

where the cost contribution measures the $L^2(\Omega)$-norm of the control Q.

Weak Formulation and Lagrange Multiplier Approach

The weak form of the boundary value problem (6)–(9) is given as follows: find $(\mathbf{u}, p, T) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega)$ such that

$$\begin{aligned} &\nu a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = (\mathbf{f}, \mathbf{v}) - \beta(\mathbf{g}T, \mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ &b(\mathbf{u}, q) = 0 && \forall q \in L_0^2(\Omega), \\ &\alpha a(T, \varphi) + c(\mathbf{u}, T, \varphi) = (Q, \varphi) + (g_{t,n}, \varphi)_{\Gamma_n} && \forall \varphi \in H_{\Gamma_d}^1(\Omega). \end{aligned} \tag{62}$$

The existence of the solution of the system (62) can be proved as in [9].

We now state the optimal control problem. We look for a (u, p, T, Q)∈H1(Ω)×L20(Ω)×H1(Ω)×L2(Ω) such that the cost functional (61) is minimized subject to the constraints (62). The admissible set of states and controls is

$$\begin{aligned} \mathcal{U}_{ad} = \{&(\mathbf{u}, p, T, Q) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times L^2(\Omega) : \\ &\mathcal{J}(\mathbf{u}, T, Q) < \infty \text{ and (62) is satified.}\} \end{aligned} \tag{63}$$

Then (u^,p^,T^,Q^)∈Uad is called an optimal solution if there exists ε>0 such that

$$\begin{aligned} &\mathcal{J}(\hat{\mathbf{u}}, \hat{p}, \hat{T}, \hat{Q}) \le \mathcal{J}(\mathbf{u}, p, T, Q) \quad \forall (\mathbf{u}, p, T, Q) \in \mathcal{U}_{ad} \text{ satisfying} \\ &\qquad \|\mathbf{u} - \hat{\mathbf{u}}\|_1 + \|p - \hat{p}\|_0 + \|T - \hat{T}\|_1 + \|Q - \hat{Q}\|_0 < \varepsilon. \end{aligned} \tag{64}$$

The existence of at least one optimal solution (u^,p^,T^,Q^)∈Uad can be proved as in [9].

We define the following functional spaces:

$$\mathbf{B}_1 = \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times L^2(\Omega), \tag{65}$$

$$\mathbf{B}_2 = \mathbf{H}^{-1}(\Omega) \times L_0^2(\Omega) \times H_{\Gamma_d}^{1*}(\Omega), \tag{66}$$

$$\mathbf{B}_3 = \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \times H_{\Gamma_d}^1(\Omega) \times L^2(\Omega). \tag{67}$$

Let $M:B_1 \to B_2$ denote the generalized constraint equations, namely, $M(z)=l$ for $z=(u,p,T,Q)\in B_1$ and $l=(l_1,l_2,l_3)\in B_2$ if and only if

$$\begin{aligned} &\nu a(\mathbf{u},\mathbf{v}) + c(\mathbf{u},\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) - (\mathbf{f},\mathbf{v}) + \beta(\mathbf{g}T,\mathbf{v}) = (\mathbf{l}_1,\mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ &b(\mathbf{u},q) = (l_2,q) && \forall q \in L_0^2(\Omega), \\ &\alpha a(T,\varphi) + c(\mathbf{u},T,\varphi) - (Q,\varphi) - (g_{t,n},\varphi)_{\Gamma_n} = (l_3,\varphi) && \forall \varphi \in H_{\Gamma_d}^1(\Omega). \end{aligned} \tag{68}$$

Thus, the constraints (62) can be expressed as $M(z)=0$. Let $(\hat{\mathbf{u}},\hat{p},\hat{T},\hat{Q}) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times L^2(\Omega)$ denote an optimal solution in the sense of (64). Then, consider the nonlinear operator $N : \mathbf{B}_1 \to \mathbb{R} \times \mathbf{B}_2$ defined by

$$N(\mathbf{u},p,T,Q) = \begin{pmatrix} \mathcal{J}(\mathbf{u},T,Q) - \mathcal{J}(\hat{\mathbf{u}},\hat{T},\hat{Q}) \\ M(\mathbf{u},p,T,Q) \end{pmatrix}. \tag{69}$$

Given $z=(u,p,T,Q)\in B_1$, the operator $M'(z):B_1 \to B_2$ may be defined as $M'(\mathbf{z})\cdot\tilde{\mathbf{z}} = \tilde{\mathbf{l}}$ for $\tilde{\mathbf{z}} = (\tilde{\mathbf{u}},\tilde{p},\tilde{T},\tilde{Q}) \in \mathbf{B}_3$ and $\tilde{\mathbf{l}} = (\tilde{\mathbf{l}}_1,\tilde{l}_2,\tilde{l}_3) \in \mathbf{B}_2$ if and only if

$$\begin{aligned} &\nu a(\tilde{\mathbf{u}},\mathbf{v}) + c(\tilde{\mathbf{u}},\mathbf{u},\mathbf{v}) + c(\mathbf{u},\tilde{\mathbf{u}},\mathbf{v}) + b(\mathbf{v},\tilde{p}) + \beta(\mathbf{g}\tilde{T},\mathbf{v}) = (\tilde{\mathbf{l}}_1,\mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ &b(\tilde{\mathbf{u}},q) = (\tilde{l}_2,q) && \forall q \in L_0^2(\Omega), \\ &\alpha a(\tilde{T},\varphi) + c(\tilde{\mathbf{u}},T,\varphi) + c(\mathbf{u},\tilde{T},\varphi) - (\tilde{Q},\varphi) = (\tilde{l}_3,\varphi) && \forall \varphi \in H_{\Gamma_d}^1(\Omega). \end{aligned} \tag{70}$$

The operator $N'(\mathbf{z}) : \mathbf{B}_3 \to \mathbb{R} \times \mathbf{B}_2$ may be defined as $N'(\mathbf{z})\cdot\tilde{\mathbf{z}} = (\tilde{a},\tilde{\mathbf{l}})$ for $\tilde{a} \in \mathbb{R}$ if and only if

$$\begin{aligned} &\alpha_u(\mathbf{u}-\mathbf{u}_d,\tilde{\mathbf{u}})_{\Omega_d} + \alpha_T(T-T_d,\tilde{T})_{\Omega_d} + \lambda(Q,\tilde{Q}) = \tilde{a} \\ &\nu a(\tilde{\mathbf{u}},\mathbf{v}) + c(\tilde{\mathbf{u}},\mathbf{u},\mathbf{v}) + c(\mathbf{u},\tilde{\mathbf{u}},\mathbf{v}) + b(\mathbf{v},\tilde{p}) + \beta(\mathbf{g}\tilde{T},\mathbf{v}) = (\tilde{\mathbf{l}}_1,\mathbf{v}) && \forall \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ &b(\tilde{\mathbf{u}},q) = (\tilde{l}_2,q) && \forall q \in L_0^2(\Omega), \\ &\alpha a(\tilde{T},\varphi) + c(\tilde{\mathbf{u}},T,\varphi) + c(\mathbf{u},\tilde{T},\varphi) - (\tilde{Q},\varphi) = (\tilde{l}_3,\varphi) && \forall \varphi \in H_{\Gamma_d}^1(\Omega). \end{aligned} \tag{71}$$

Let $z_0\in B_1$. We have that the operator $M'(z_0)$ has closed range in B_2 and the operator $N'(z_0)$ has closed range but is not in $\mathbb{R}\times\mathbf{B}_2$ [21].

Similarly to the other controls presented in previous sections, let $\hat{\mathbf{z}} = (\hat{\mathbf{u}},\hat{p},\hat{T},\hat{Q}) \in \mathbf{H}^1(\Omega) \times L_0^2(\Omega) \times H^1(\Omega) \times L^2(\Omega)$ denote an

optimal solution in the sense of (64). Then, there exists a nonzero Lagrange multiplier $(\Lambda, \hat{\mathbf{u}}_a, \hat{p}_a, \hat{T}_a) \in \mathbb{R} \times \mathbf{B}_2^*$ satisfying the Euler equations

$$\Lambda \mathcal{J}'(\hat{\mathbf{u}}, \hat{T}, \hat{Q}) \cdot \tilde{\mathbf{z}} + \langle (\hat{\mathbf{u}}_a, \hat{p}_a, \hat{T}_a), M'(\hat{\mathbf{z}}) \cdot \tilde{\mathbf{z}} \rangle = 0 \quad \forall \tilde{\mathbf{z}} \in \mathbf{B}_3 , \tag{72}$$

where $\langle \cdot, \cdot \rangle$ denotes the duality pairing between B2 and B∗2. The interested reader can consult [21] on the existence of the Lagrange multiplier.

Optimality System

As in the previous case, we drop the $(\cdot\hat{})$ notation for the optimal solution and derive the optimality system using the Euler Equation (72)

$$\begin{aligned} &\alpha_u \Lambda(\mathbf{u} - \mathbf{u}_d, \tilde{\mathbf{u}})_{\Omega_d} + \alpha_T \Lambda(T - T_d, \tilde{T})_{\Omega_d} + \Lambda \lambda(Q, \tilde{Q}) + b(\tilde{\mathbf{u}}, p_a) + \nu a(\tilde{\mathbf{u}}, \mathbf{u}_a) + \\ &\quad + c(\tilde{\mathbf{u}}, \mathbf{u}, \mathbf{u}_a) + b(\mathbf{u}_a, \tilde{p}) + c(\mathbf{u}, \tilde{\mathbf{u}}, \mathbf{u}_a) + \beta(\mathbf{g}\tilde{T}, \mathbf{u}_a) + \alpha a(\tilde{T}, T_a) + c(\tilde{\mathbf{u}}, T, T_a) + \\ &\quad + c(\mathbf{u}, \tilde{T}, T_a) - (\tilde{Q}, T_a) = 0 . \end{aligned} \tag{73}$$

We extract the terms involved in the same variation, set Λ=−1, and obtain the following equations:

$$\begin{aligned} &\nu a(\tilde{\mathbf{u}}, \mathbf{u}_a) + c(\mathbf{u}, \tilde{\mathbf{u}}, \mathbf{u}_a) + c(\tilde{\mathbf{u}}, \mathbf{u}, \mathbf{u}_a) + b(\tilde{\mathbf{u}}, p_a) = \\ &\quad = \alpha_u (\mathbf{u} - \mathbf{u}_d, \tilde{\mathbf{u}})_{\Omega_d} - c(\tilde{\mathbf{u}}; T, T_a) && \forall \tilde{\mathbf{u}} \in \mathbf{H}_0^1(\Omega) , \\ &b(\mathbf{u}_a, \tilde{p}) = 0 && \forall \tilde{p} \in L_0^2(\Omega) , \\ &\alpha a(\tilde{T}, T_a) + c(\mathbf{u}, \tilde{T}, T_a) = -(\beta \mathbf{g}\tilde{T}, \mathbf{u}_a) + \alpha_T (T - T_d, \tilde{T})_{\Omega_d} && \forall \tilde{T} \in H_{\Gamma_d}^1(\Omega) , \end{aligned} \tag{74}$$

and the control equation

$$\lambda(Q, \tilde{Q}) + (\tilde{Q}, T_a) = 0 , \quad \forall \tilde{Q} \in L^2(\Omega) . \tag{75}$$

The necessary conditions for an optimum are defined by Equations (62) and (74). This system of equations is the optimality system. We can use integrations to show that the system constitutes a weak formulation of the boundary value problem for state equations

$$\begin{aligned} &\mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nu \Delta \mathbf{u} = \mathbf{f} - \beta \mathbf{g} T , \\ &\nabla \cdot \mathbf{u} = 0 , \\ &\mathbf{u} \cdot \nabla T - \alpha \Delta T = Q , \\ &\mathbf{u} = \mathbf{w} \text{ on } \Gamma , \quad \alpha \nabla T \cdot \mathbf{n}|_{\Gamma_n} = g_{t,n} \text{ on } \Gamma_n , \quad T = g_t \text{ on } \Gamma_d , \end{aligned} \tag{76}$$

the adjoint equations

$$\begin{aligned}
&\mathbf{u}_a \cdot (\nabla \mathbf{u})^T - \mathbf{u} \cdot \nabla \mathbf{u}_a + \nabla p_a - \nu \Delta \mathbf{u}_a = -T \nabla T_a + \alpha_u (\mathbf{u} - \mathbf{u}_d), \\
&\nabla \cdot \mathbf{u}_a = 0, \\
&-\alpha \Delta T_a - \mathbf{u} \cdot \nabla T_a = -\beta \mathbf{g} \cdot \mathbf{u}_a + \alpha_T (T - T_d), \\
&\mathbf{u}_a = 0 \text{ on } \Gamma, \quad \nabla T_a \cdot \mathbf{n}|_{\Gamma_n} = 0 \text{ on } \Gamma_n, \quad T_a = 0 \text{ on } \Gamma_d,
\end{aligned} \tag{77}$$

and the control equation

$$Q = -\frac{T_a}{\lambda} \quad \text{in } \Omega. \tag{78}$$

Therefore, the optimality system in the strong form consists of the Boussinesq system (76), the adjoint of Boussinesq Equation (77), and the control Equation (78).

Numerical Algorithm

Let us consider the gradient method for the following minimization problem: find $Q \in L^2(\Omega)$ such that $\mathcal{F}(Q) := \mathcal{J}(\mathbf{u}(Q), T(Q), Q)$ is minimized. Given $Q^{(0)}$, we can define the sequence

$$Q^{(n+1)} = Q^{(n)} - \rho^{(n)} \frac{d\mathcal{F}(Q^{(n)})}{dQ^{(n)}}, \tag{79}$$

recursively, where $\rho^{(n)}$ is a variable step size. Let $\hat{Q}_c$ be a solution of the minimization problem. Thus, at the optimum $d\mathcal{F}(\hat{Q})/d\hat{Q} = 0$ and $Q^{(n+1)} = Q^{(n)}$. The Gâteaux derivative $(d\mathcal{F}(Q)/dQ) \cdot \widetilde{Q}$ for every direction $\widetilde{Q} \in L^2(\Omega)$ may be computed as

$$\frac{d\mathcal{F}(Q)}{dQ} \cdot \widetilde{Q} = \lambda(Q, \widetilde{Q}) + (\widetilde{Q}, T_a). \tag{80}$$

Thus, the Gâteaux derivative may be computed as

$$\frac{d\mathcal{F}(Q)}{dQ} = Q + \frac{T_a}{\lambda}. \tag{81}$$

The optimization algorithm is then given as follows.

(a) Initial step:

1. choose tolerance τ and $Q^{(0)}$; set n=0 and $\rho^{(0)}=1$;
2. solve for $(u^{(0)}, p^{(0)}, T^{(0)})$ from (62) with $Q=Q^{(0)}$;
3. evaluate $\mathcal{J}^{(0)} = \mathcal{J}(\mathbf{u}^{(0)}, T^{(0)}, Q^{(0)})$ using (61).

(b) Main loop:

4. set n=n+1;
5. solve for $(\mathbf{u}_a^{(n)}, p_a^{(n)}, T_a^{(n)})$ from (74);
6. solve for Q(n) from

$$Q^{(n)} = Q^{(n-1)} - \rho^{(n)}\left(Q^{(n-1)} + \frac{T_a^{(n)}}{\lambda}\right); \tag{82}$$

7. solve for $(\mathbf{u}^{(n)}, p^{(n)}, T^{(n)})$ from (40) with $Q=Q^{(n)}$;
8. evaluate $\mathcal{J}^{(n)} = \mathcal{J}(\mathbf{u}^{(n)}, T^{(n)}, Q^{(n)})$ using (39);

(i) if $\mathcal{J}^{(n)} > \mathcal{J}^{(n-1)}$, set $\rho^{(n)} = 0.5\rho^{(n)}$ and go to step (b) 3.;

(ii) if $\mathcal{J}^{(n)} < \mathcal{J}^{(n-1)}$, set $\rho^{(n+1)} = 1$ and go to step (b) 1.;

(iii) if $|\mathcal{J}^{(n)} - \mathcal{J}^{(n-1)}|/|\mathcal{J}^{(n)}| < \tau$ stop.

NUMERICAL RESULTS

In this section, we report some numerical results obtained by using the mathematical models shown in the previous sections. The main difference between the three control problems is in the nature of the control equations. For Neumann and distributed controls, the control equation is an algebraic equation that states that the control is proportional to the adjoint temperature (see Equations (56) and (78)). In contrast, when we have a Dirichlet boundary control, the control equation is a partial differential equation with the normal adjoint temperature gradient as source term, as reported in (31). Thus, the adjoint temperature Ta plays a key role in all three control mechanisms, as does the regularization parameter λ that appears in the denominator of the source terms. The adjoint temperature Ta depends on the objectives of the velocity and temperature fields. When the objective relates to the temperature field, the dependence is direct through the term $\alpha_T(T-T_d)$ appearing on the right-hand side of the adjoint temperature Equations (27), (52), and (74). If the objective relates to the velocity field, the control mechanism is indirect, since the term $\alpha_u(u-u_d)$ acts as a source in the adjoint velocity equation. In turn, the adjoint velocity appears in the source term of the adjoint temperature $\beta_g \cdot u_a$.

The geometry considered for all the simulations is a square cavity with L=0.01m. The domain $\Omega=[0,L]\times[0,L]\in \mathbb{R}^2$ is shown in Figure 1. We consider liquid lead with the properties reported in Table 1. We discretize the numerical problem in a finite element framework, and we consider a 20×20 uniform quadrangular mesh formed by biquadratic elements. The simulations were performed using the in-house finite element multigrid code FEMuS developed at the University of Bologna [22]. The code is based on a

C++ main program that handles several external open-source libraries such as the MPI and PETSc libraries.

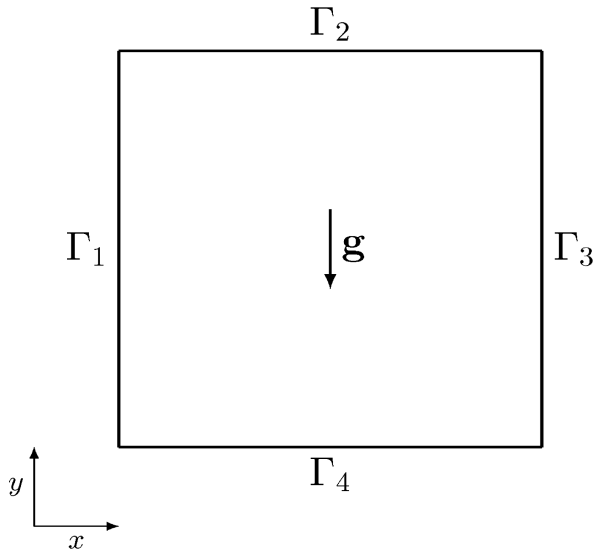


Figure 1: Computational domain for the optimal control of Boussinesq equations, where g is the gravity vector and Γ_1, Γ_2, Γ_3, and Γ_4 are the boundaries.

Table 1: Boussinesq control: physical properties employed for the numerical simulations

Property	Symbol	Value	Units
Viscosity	μ	0.00181	Pa s
Density	ρ	10,340	kg/m^3
Thermal conductivity	λ	10.72	W/(mK)
Specific heat	c	145.75	J/(kgK)
Coefficient of expansion	β	2.5684×10^{-4}	K^{-1}

Dirichlet Boundary Control

We now show the numerical results for the Dirichlet boundary control. The boundary conditions are reported in (12), where $\Gamma_d=\Gamma_1 \cup \Gamma_3$, $\Gamma_i=\Gamma_3$, $\Gamma_c=\Gamma_1$ and $\Gamma_n=\Gamma_2 \cup \Gamma_4$. We set f=0 and Q=0 in (14), and $g_u=0$, $g_{t,n}=0$, $g_t=493K$ on Γ_3, and $g_t=503K$ on Γ_1 in (12). For the reference case, we set $T^{(0)}c=0$. Then, on $\Gamma_c=\Gamma_1$ we have $T^{(0)}=g_t$. In Figure 2a,b, we show the temperature and velocity contours, respectively, of the numerical solution when the control algorithm is not applied. Lead flows in the cavity and forms a clockwise vortex due

to buoyancy forces caused by the heated cavity wall. The bulk velocity is U_b=0.008765m/s. The Richardson number, computed as $Ri = gL\Delta T\beta/U_b^2$, is equal to 3.28. The Grashof number is Gr=RiRe^2=8.2×10^5. Lastly, the Rayleigh number is given by Ra=GrPr=2×104. The results shown in Figure 2 follow the typical features of temperature and velocity profiles for Ra≈10^4, i.e., isotherms departing from the vertical position with the formation of a central elliptic clockwise vortex [23].

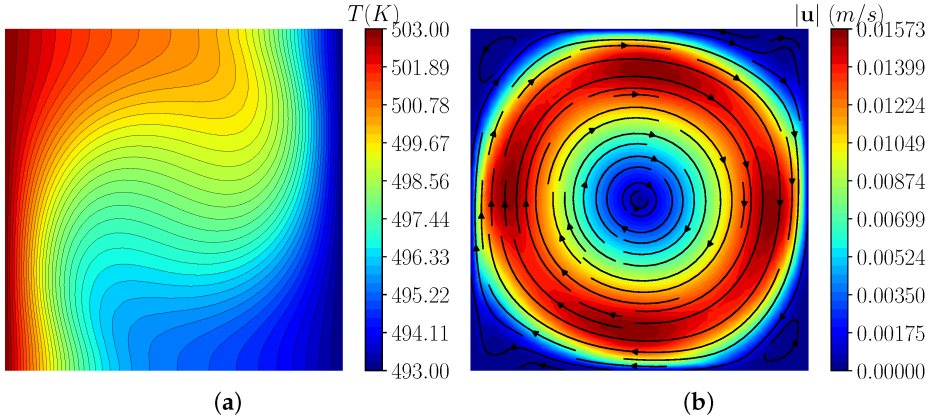


Figure 2: Uncontrolled solution: contours of the temperature field T (a); contours and streamlines of the velocity field u (b). The velocity magnitude is indicated by |u|.

Temperature Matching Case

Firstly, we aim to test the optimization algorithm with a temperature matching case. Let (13) be the objective functional with α_u=0, α_T=1, and Ω_d=[0.45L;0.55L]×[0.75L;0.85L]. The region Ω_d is indicated in Figure 3a. We set T_d=450K. Then, in Ω_d we aim at obtaining cooler fluid than in the reference case reported in Figure 2a. We consider four different values of the regularization parameter λ, namely, 10^{-5},10^{-6},10^{-7}, and 10^{-8}. The reference objective functional is $J^{(0)}$=0.001250. For the numerical simulations, we use the algorithm for Dirichlet boundary problems presented in the previous sections, and we choose (37) for the update algorithm of the control.

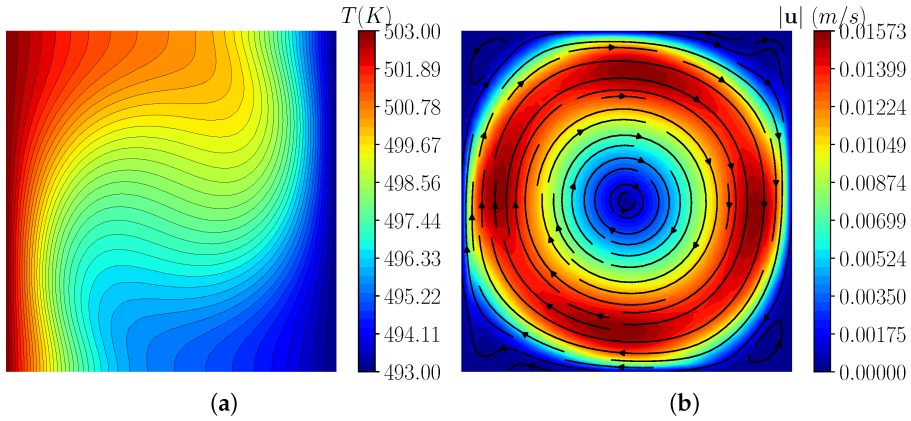


Figure 3: Temperature matching case with Dirichlet boundary control: optimal solution for $\lambda=10^{-7}$. Contours of the temperature field T (a); contours and streamlines of the velocity field u (b). The velocity magnitude is indicated by |u|, and Ω_d is the region where the objective is set.

The contours of the optimal solution in terms of temperature and velocity fields are shown in Figure 3a,b, respectively, for $\lambda=10^{-7}$. The region Ω_d, where the objective is set, is highlighted with a black square in Figure 3a. From the contours, we can see that the optimal temperature field assumes values close to the target temperature T_d=450K. To achieve the objective, the temperature on the left wall decreases with respect to the reference case. For this reason, the motion changes, and we obtain a counterclockwise vortex, as depicted by the streamlines in Figure 3b.

In Table 2, we report the objective functional values $J^{(n)}$ corresponding to its optimal state for each numerical simulation. We also report the value of the reference objective functional $J^{(0)}$ and the percentage reduction for each case evaluated as $(J^{(0)}-J^{(n)})/J^{(0)}$. In addition, the number of iterations n of the optimization algorithm is included in Table 2. The lowest value of λ results in the lowest functional value of $J^{(10)}=1.979\times10^{-6}$ and the greatest percentage reduction.

Table 2: Temperature matching case with Dirichlet boundary control: objective functional, percentage reduction, and number of iterations of the optimization algorithm for different λ values

λ	10^{-5}	10^{-6}	10^{-7}	10^{-8}	Reference
$\mathcal{J}^{(n)} \times 10^6$	3.110	2.179	2.091	1.979	1250
% Reduction	99.75	99.82	99.83	99.84	0
Iterations n	6	5	6	10	0

Temperature profiles along the boundary Γ_c are reported in Figure 4a for the different values of the regularization parameter λ. As λ decreases, the minima of the profiles move towards y/L=1. In Figure 4b, the temperature is plotted along a line at y/L=0.8 for 0.45<x/L<0.55 in the region Ω_d. We can see that for the lowest values of λ, the optimal solutions tend to the target profile T_d. The case $\lambda=10^{-5}$ is the farthest from the objective, as we can also deduce from the functional values reported in Table 2.

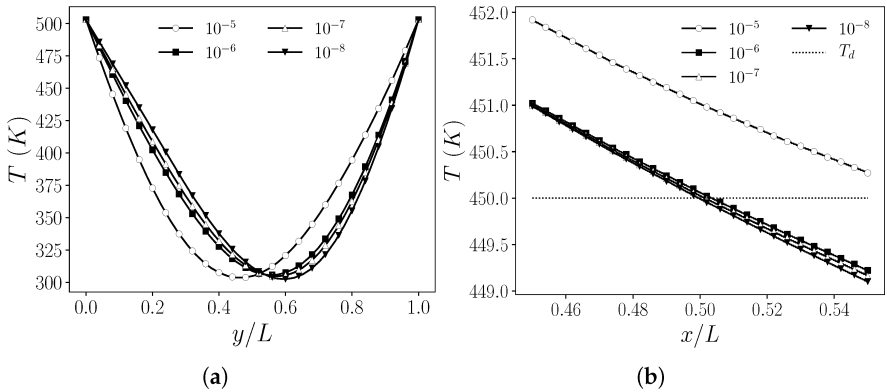


Figure 4: Temperature matching case with Dirichlet boundary control: temperature T profiles plotted against y/L along the controlled boundary Γ_c (a); temperature T profiles plotted against x/L on the region Ω_d along the line y/L=0.8 (b). Numerical results for $\lambda=10^{-5},10^{-6},10^{-7}$, and 10^{-8}. The target value T_d is shown as a dotted line.

Velocity Matching Case 1

The second test for the Dirichlet optimal control is a velocity matching case. The objective functional is the one reported in Equation (13), setting $\alpha_u=1$, $\alpha_T=0$ and $\Omega_d=[0.15L;0.25L]\times[0.45L;0.55L]$. The region Ω_d is represented in Figure 5c. We aim to control the *y*-component of the velocity with $v_d=0.05$m/s. In the reference case, the mean value of v over Ω_d is 0.0159m/s, but we aim to accelerate the fluid near the controlled boundary Γ_c in order to

enhance the velocity on Ω_d. We consider different values of the regularization parameter λ, i.e., 10^{-10},10^{-11},10^{-12}, 10^{-13}, and 10^{-14}. The considered values are lower than those used for the temperature matching test. We also tested higher values of the regularization parameter, but the control was ineffective in those cases. Indeed, it is easier to achieve an objective on the temperature field than on the velocity field, since the control parameter T_c (or h or Q) depends directly on the adjoint temperature but indirectly on the adjoint velocity. The value of the reference objective functional is $J^{(0)}=7.011\times10^{-10}$.

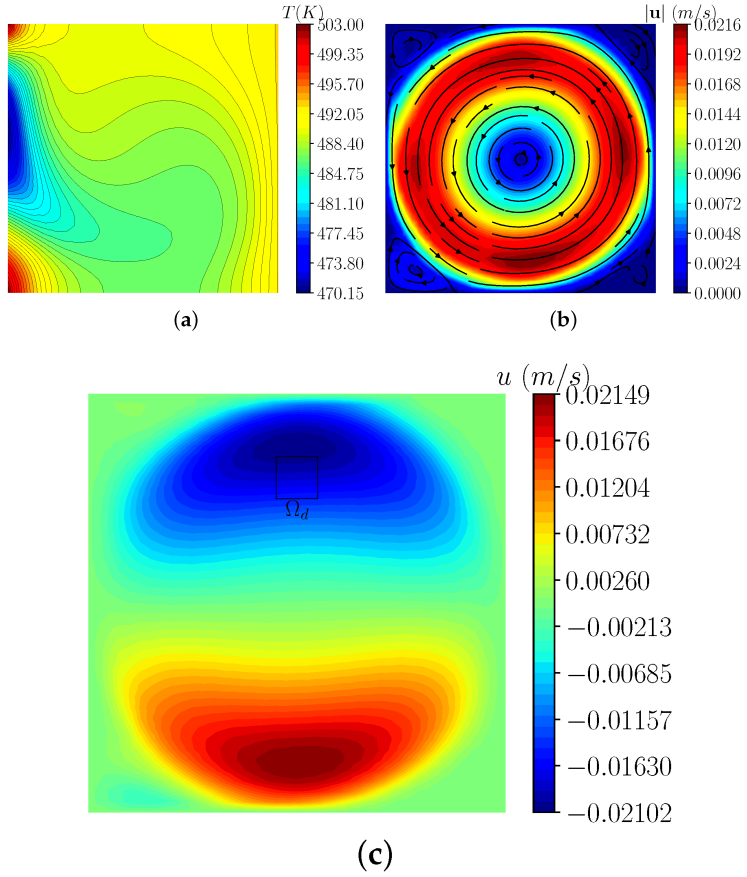


Figure 5: Velocity matching case with Dirichlet boundary control—Case 1: optimal solution for $\lambda=10^{-13}$. Contours of the temperature field T (a); contours and streamlines of the velocity field u (b); contours of the y-component of the velocity field v (c). The velocity magnitude is indicated by |u|, and Ω_d is the region where the objective is set.

In Figure 5, the optimal solution obtained with $\lambda=10^{-13}$ is reported. In Figure 5a, the contours of the optimal temperature field are shown. Along Γ_c, the temperature shows a sharp variation. At the bottom of Γ_c, we have a maximum for the temperature, while at the top is the minimum temperature value. The fluid is heated and is accelerated to the desired velocity in the region Ω_d. The resulting velocity field is shown in Figure 5b, where contours of the velocity magnitude and streamlines are shown. The contours of the *y*-component of the velocity are shown in Figure 5c, where the region Ω_d is highlighted.

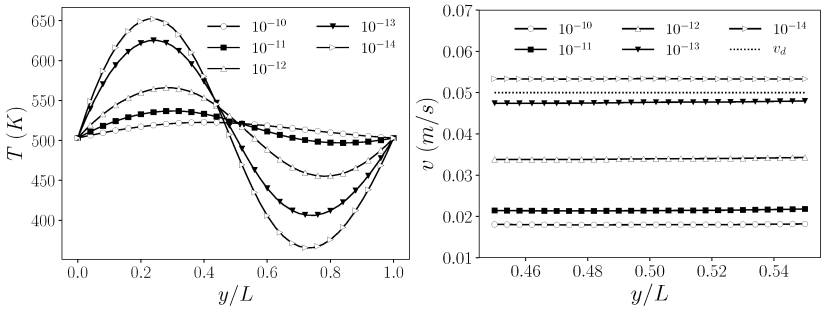
In Table 3, we report the objective functional values $J^{(n)}$, the number of iterations *n* of the optimization algorithm, and the percentage reduction with respect to the reference $J^{(0)}$. For the highest values of λ, the control is poor, and the functional is quite similar to the reference value. However, we can observe a strong functional reduction for the cases with $\lambda \leq 10^{-13}$.

Table 3: Velocity matching case with Dirichlet boundary control. Case 1: objective functional, percentage reduction, and number of iterations of the optimization algorithm for different λ values

λ	10^{-10}	10^{-11}	10^{-12}	10^{-13}	10^{-14}	Reference
$\mathcal{J}^{(n)} \times 10^{12}$	586.3	413.6	137.4	9.767	8.796	701.1
% Reduction	16.4	41.01	80.40	98.61	98.74	0
Iterations *n*	5	5	4	6	5	0

Temperature profiles along the boundary Γ_c are reported in Figure 6a for different values of the regularization parameter λ. For $\lambda=10^{-10}$, the profile only has a stationary point at $y/L \approx 0.5$. For lower values of λ, there is a change of concavity in the temperature profiles and an inflection point at $y/L \approx 0.5$. As λ decreases, the maximum is located at $0.2<y/L<0.4$ and its value increases, while the minimum is located at $0.6<y/L<0.8$ and its value decreases. As expected, with low values of the regularization parameter, the $H^{-1}(\Gamma_c)$-norm of the control has less weight in the objective functional, and more irregular functions are accepted as optimal solutions. In Figure 6b, the *y*-component of the velocity is plotted along a line at $x/L=0.2$ for $0.45<y/L<0.55$ in the region Ω_d. The velocity profile is reported for all values of λ, together with the target velocity profile v_d. For the lowest values of λ (10^{-13}, 10^{-14}), the optimal solutions tend to the target profile v_d, while the highest

values of λ (10^{-10},10^{-11}, 10^{-12}) lead to the solutions farthest from the objective, as can be deduced from the functional values in Table 3. However, when λ is small (10^{-13}, 10^{-14}), the maximum temperature value increases (from 503K up to 650K) and the minimum value decreases (from 503K down to 400K). This large variation is due to the fact that the target v_d is quite far from the reference case, and the temperature over Γ_c must change considerably to reach the objective.



(**a**) (**b**)

Figure 6: Velocity matching case with Dirichlet boundary control—Case 1: temperature profiles *T* plotted against y/L on the controlled boundary Γ_c (a); *y*-component of the velocity *v* profiles plotted against y/L on the region Ω_d along the line x/L=0.2 (b). Numerical results for λ=10^{-10},10^{-11},10^{-12},10^{-13}, and 10^{-14}. The target value v_d is shown as a dotted line.

Velocity Matching Case 2

A second case for the velocity matching test is now considered. The objective is set on the *x*-component of the velocity field, where we aim to achieve a counterclockwise flow. Let us consider Ω_d=[0.45L;0.55L]×[0.75L;0.85L]. This region is highlighted in Figure 7c. In the reference case, the mean value of *u* on Ω_d is set to 0.0129m/s. Then, we set a uniform value u_d=−0.02m/s as a target profile. The simulations are performed considering different values of λ, namely, 10^{-10},10^{-11}, and 10^{-12}. The reference objective functional is $J^{(0)}$=5.425×10−10.

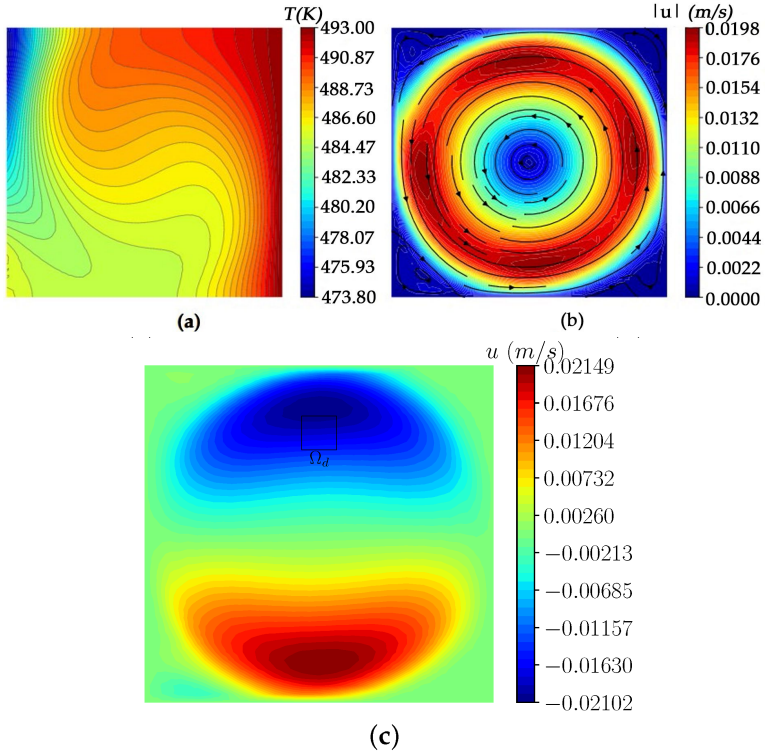


Figure 7: Velocity matching case with Dirichlet boundary control—Case 2: optimal solution for $\lambda=10^{-11}$. Contours of the temperature field T (a); contours and streamlines of the velocity field u (b); contours of the x-component of the velocity field u (c). The velocity magnitude is indicated by |u|, and Ω_d is the region where the objective is set.

The optimal temperature and velocity fields obtained with $\lambda=10^{-11}$ are reported in Figure 7. In Figure 7a, the contours of the optimal temperature field are shown. The resulting velocity field is shown in Figure 7b, where contours of the velocity magnitude and streamlines are reported. We can observe that a counterclockwise flow is driven by the buoyancy forces. The contours of the x-component of the velocity are represented in Figure 7c, where the region Ω_d is highlighted. We also report the optimal solution obtained with $\lambda=10^{-12}$ in Figure 8. In this case, the solution is quite unexpected. Figure 8a shows the contours of the optimal temperature field. At the bottom of the left wall ($\Gamma_c=\Gamma_1$), the temperature is higher than the temperature on the right wall ($\Gamma_i=\Gamma_3$), while at the top of Γ_c the temperature is lower than the temperature on Γ_3.

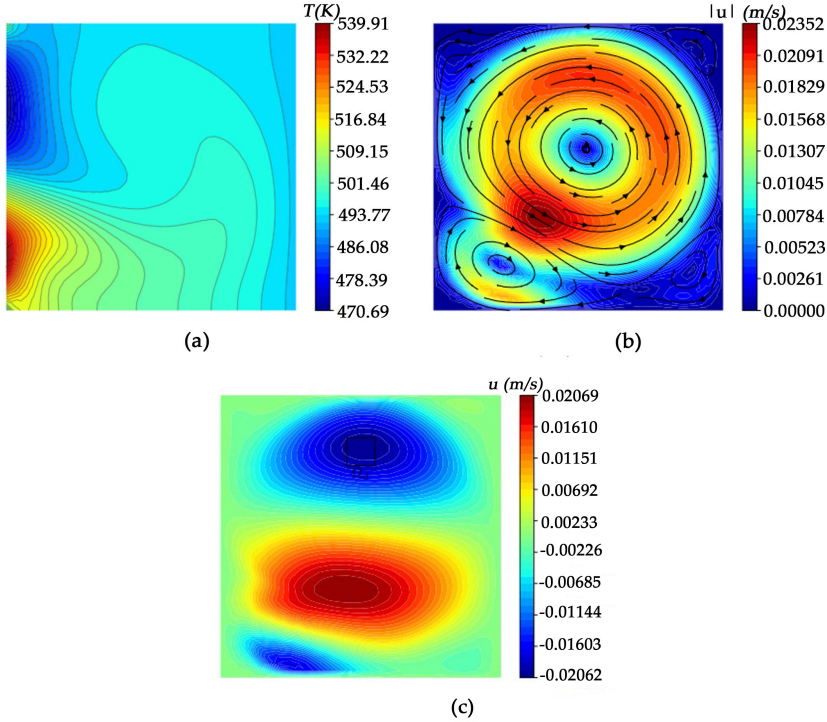


Figure 8: Velocity matching case with Dirichlet boundary control—Case 2: optimal solution for $\lambda=10^{-12}$. Contours of the temperature field T (a); contours and streamlines of the velocity field u (b); contours of the x-component of the velocity field u (c). The velocity magnitude is indicated by |u|, and Ω_d is the region where the objective is set.

This profile induces buoyancy forces which cause two vortexes; a smaller clockwise vortex behind the bottom-left corner and a bigger counterclockwise vortex in the center of the cavity, as shown in Figure 8b. The contours of the x-component of velocity are shown in Figure 8c, where the region Ω_d is in evidence. There, the x-component of velocity is quite uniform and close to the target value u_d.

In Table 4, we report the objective functional values $J^{(n)}$, the percentage reduction, and the number of iterations n of the optimization algorithm. For the highest value of λ (10^{-10}), the control is poor, and the functional value is quite similar to the reference value. For the other values of λ, the control is more effective. As observed in the previous test cases, with the lowest value of λ, we have the lowest functional value and the greatest percentage reduction.

Table 4: Velocity matching case with Dirichlet boundary control—Case 2: objective functional, percentage reduction and number of iterations of the optimization algorithm for different λ values

λ	10^{-10}	10^{-11}	10^{-12}	Reference
$\mathcal{J}^{(n)} \times 10^{13}$	246.6	36.04	1.677	5423
% Reduction	54.53	93.35	99.69	0
Iterations n	4	10	9	0

In Figure 9a, the temperature profiles along the boundary Γ_c are shown for different values of the regularization parameter λ (10^{-10}, 10^{-11}, 10^{-12}). For $\lambda=10^{-10}$ and $\lambda=10^{-11}$, the profiles present a minimum point at $0.4<y/L<0.7$. The temperature on Γ_c is lower than the temperature on the opposite wall Γ_i, namely, T=493K, to obtain a counterclockwise flow. For $\lambda=10^{-12}$, the optimal solution is unexpected, as previously noted. There is a variation of concavity in the profile and an inflection point at $y/L\approx0.5$. For $y/L<0.5$, the temperature on Γ_c is higher than the temperature on Γ_3, while at the top of the controlled wall, for $y/L>0.5$, the temperature on Γ_c is lower than the temperature on Γ_3. In Figure 9b, the *x*-component of the velocity is plotted along a line at y/L=0.8 for $0.45<x/L<0.55$ in the region Ω_d. The velocity profiles are shown for all values of λ, together with the target velocity profile u_d. We can observe that in all cases, the flow changes from clockwise to counterclockwise with a negative *x*-component of velocity at the top of the cavity. We note that in this test, the lower the value of λ, the closer the velocity profile is to the target profile.

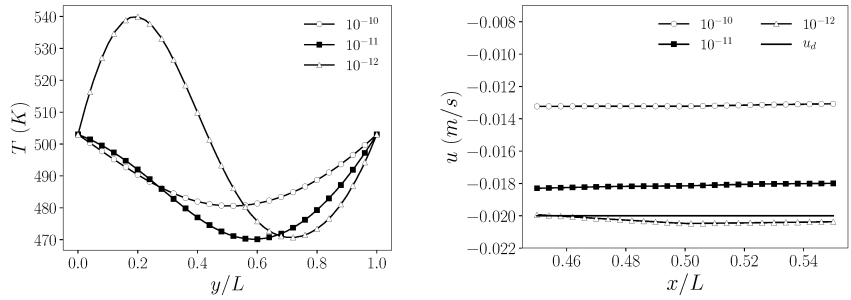


Figure 9: Velocity matching case with Dirichlet boundary control—Case 2: temperature profiles *T* plotted against y/L on the controlled boundary Γ_c (a); *x*-component of the velocity *u* profiles plotted against y/L on the region Ω_d along the line x/L=0.2 (b). Numerical results for $\lambda=10^{-10}$,10^{-11}, and 10^{-12}. The target value u_d is shown as a dotted line.

Neumann Boundary Control

For the Neumann control problem, we consider the geometry shown in Figure 1. The boundary conditions are reported in (38), where $\Gamma_d=\Gamma_3$, $\Gamma_n=\Gamma_1\cup\Gamma_2\cup\Gamma_4$, $\Gamma_i=\Gamma_2\cup\Gamma_4$, $\Gamma_c=\Gamma_1$. We set $g_{t,n}=0$, $g_t=493$K, and $g_u=0$ in (38) and $f=0$, $Q=0$ in (40). The wall-normal heat flux h acting on Γc is the control for the problem. To compute the reference case, we set $h^{(0)}=0$. Thus, the uncontrolled problem consists of three thermally-insulated walls, i.e., the left (Γ_1), bottom (Γ_2), and top ($\Gamma 4$) walls, and a wall with a fixed temperature, which is the right wall (Γ_3). The reference case is a trivial problem, characterized by a uniform and constant temperature, no buoyancy forces, and still fluid.

We performed several tests, varying the objective. We report the numerical results obtained considering the same objective on the x-component of velocity also studied with the Dirichlet control. We recall the main simulation parameters. Let $\Omega_d=[0.45L;0.55L]\times[0.75L;0.85L]$ be the region where we aim to achieve the objective, and let $u_d=-0.02$m/s be the target velocity profile. In the reference case, the fluid is still. Then, $u=0$m/s in Ω_d. The simulations were performed considering different values of λ, namely, 10^{-4}, 10^{-5},10^{-6}, and 10^{-7}. The reference objective functional is $J^{(0)}=2.061\times10^{-10}$.

In Table 5, the objective functional values $J^{(n)}$ and the number of iterations n of the optimization algorithm are reported for all the values of λ. The percentage reductions are also reported. In all tests, we observe large functional reductions. In particular, for lower values of λ, the control is more effective.

Table 5: Velocity matching case with Neumann boundary control: objective functional, percentage of reduction, and number of iterations of the optimization algorithm for the reference case and different λ values

λ	10^{-4}	10^{-5}	10^{-6}	10^{-7}	Reference
$\mathcal{J}^{(n)}\times10^{12}$	30.58	30.14	8.454	1.536	206.1
% Reduction	85.16	85.8	95.90	99.25	0
Iterations n	4	14	9	7	0

The optimal solution obtained with $\lambda=10^{-6}$ is reported in Figure 10. The contours of the temperature field T over the domain can be seen in Figure 10a. The heat flux imposed on the left wall is outgoing, and the wall is cooler than in the reference case, with a minimum value of around 473 K. In Figure 10b, the velocity streamlines and the contours of the velocity magnitude

are shown. The formation of a counterclockwise vortex is shown in this figure. The contours of the x-component of the velocity field u are reported in Figure 10, and the region Ω_d is highlighted.

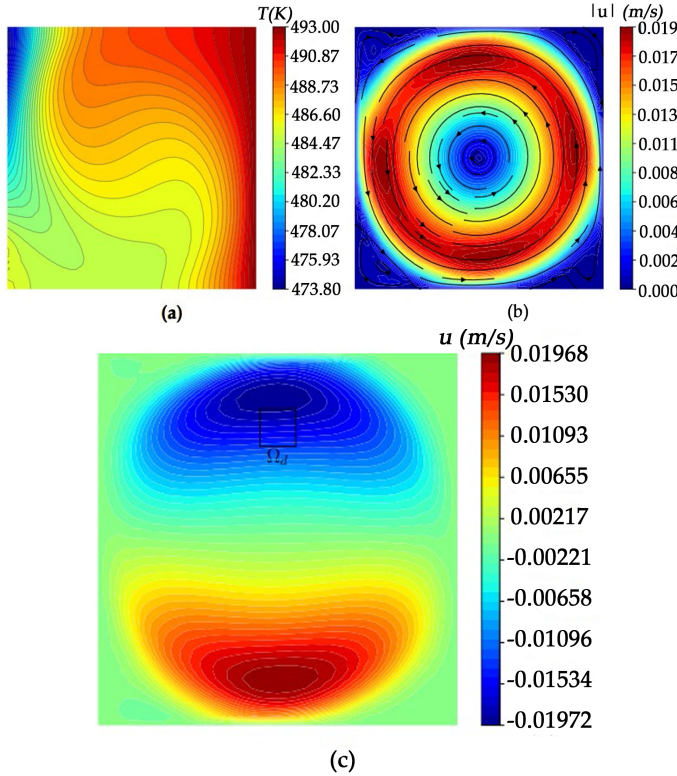
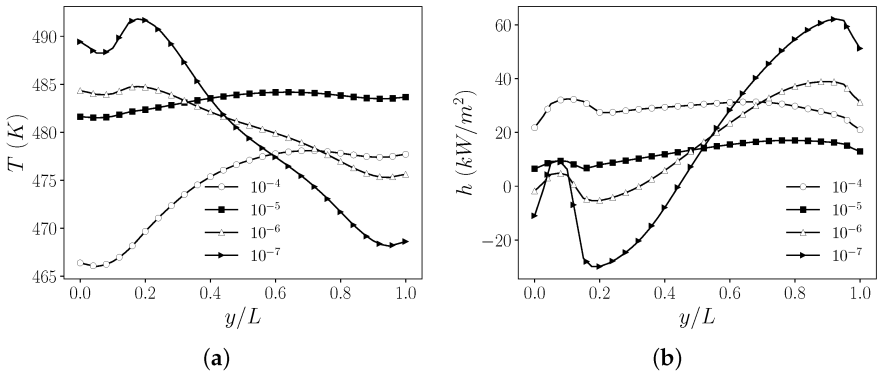


Figure 10: Velocity matching case with Neumann boundary control: optimal solution for $\lambda=10^{-6}$. Contours of the temperature field T (a); contours and streamlines of the velocity field u (b); contours of the x-component of the velocity field u (c). The velocity magnitude is indicated by |u|, and Ω_d is the region where the objective is set.

In Figure 11a, the temperature profiles along the boundary Γ_c are shown for different values of the regularization parameter λ (10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}). Comparing these profiles with the temperature profiles of Figure 9a obtained for a Dirichlet control, we observe very different trends. With a Dirichlet control, the temperature on Γ_c belongs to the Hilbert space $H^1(\Gamma_c)$, and the control T_c is nullified at the extremities of the boundary, i.e., $T_c=0K$ on $\partial\Gamma_c$. For this reason, with a Dirichlet control, $T=g_t=503K$ at $y/L=0$ and $y/L=1$. With Neumann controls, we do not have constraints on the temperature value on $\partial\Gamma_c$, and we obtain different shapes for the profiles. In Figure 11b,

the control parameter h expressed in kW/m^2 is reported along Γ_c. With the highest values of λ (10^{-4}, 10^{-5}), the control is quite uniform and regular, but it is less effective with respect to the functional reduction. With the lowest values of λ (10^{-6}, 10^{-7}), the profiles of the control h are sharp and present changes of sign.



(a) (b)

Figure 11: Velocity matching case with Neumann boundary control: temperature T (a) and wall-normal heat flux h (b) plotted against y/L on the controlled boundary Γ^c. Numerical results for λ=10^{-4},10^{-5},10^{-6}, and 10^{-7}.

Distributed Control

For the distributed control problem, we consider the geometry reported in Figure 1. The boundary conditions are reported in (9), where Γ_d=Γ_1∪Γ_3, Γ_n=Γ_2∪Γ_4. We set f=0, g_u=0 in (62), while in (9) we have $g_{t,n}$=0, g_t=493K on Γ_3, and g_t=503K on Γ1. The volumetric heat source Q is the control acting on the domain Ω. For the reference case, we consider $Q^{(0)}$=0. Thus, the reference case is the one considered for the Dirichlet boundary control. The buoyancy forces put the fluid in motion, and a clockwise vortex is formed. The contours and streamlines for the temperature and velocity are shown in Figure 2.

We performed several tests, varying the objectives and the values of the regularization parameter λ. We show the results for a velocity matching case. Let us consider Ω_d=[0.15L;0.25L]×[0.45L;0.55L]. We aim to control the y-component of the velocity, and therefore we set v_d=0.05m/s, as in the first velocity matching case presented for the Dirichlet boundary control. In the reference case, the mean value of v on Ω_d is equal to 0.0159m/s, as we aim to accelerate the fluid near the controlled boundary Γ_c. We consider

several values of the regularization coefficient, namely, 10^{-10}, 10^{-11}, and 10^{-12}.

In Table 6, the objective functional values $J^{(n)}$, the percentage reductions, and the number of iterations n of the optimization algorithm are reported for all values of λ. Thus, in all the tests, the functional is strongly reduced by a factor of 103. This is an expected result since the optimal control Q can act on the whole domain, and its influence is strong on the distribution of the temperature field and buoyancy forces. We remark that with boundary control problems, the control can act only on a portion of the boundary, and its impact is less effective on the solution.

Table 6: Velocity matching case with distributed control: objective functional $\mathcal{J}^{(n)}$, percentage reduction, and number of iterations n of the optimization algorithm for different values of λ.

λ	10^{-10}	10^{-11}	10^{-12}	Reference
$\mathcal{J}^{(n)} \times 10^{13}$	2.792	2.229	2.159	2061
% Reduction	99.96	99.97	99.97	0
Iterations n	3	13	35	0

In Figure 12, the contours of the optimal solution for $\lambda=10^{-11}$ are shown. The optimal control Q expressed in MW/m^3 is reported in Figure 12a. The heat source is not uniform over the domain, being positive in the proximity of the hottest wall (T=503K on Γ_1) and negative near the coolest wall (T=493K on Γ_3). This heat source distribution influences the temperature solution reported in Figure 12b. The isotherms are more stretched than in the reference case, and the fluid is locally hotter than 503K and cooler than 493K, due to the volumetric heat source. The streamlines and contours of the velocity field are reported in Figure 12c. Figure 12d shows the region Ω_d and the contours of the y-component of velocity. The solution is almost uniform in Ω_d and is close to the target value of v_d=0.02m/s. Comparing Figure 5c and Figure 12d, we can observe that the distributed control is the most effective in achieving the objective. The great effectiveness of the distributed control can be also seen by comparing Table 6 and the first three columns of Table 3. With the same λ coefficients (10^{-10}, 10^{-11}, and 10^{-12}), the distributed control leads to much greater reductions in the functional $J^{(n)}$ than the Dirichlet control. Moreover, by observing Figure 5a and Figure 12b, we see that with a distributed control, the optimal temperature solution is more uniform and regular than with a Dirichlet optimal control, which can lead to temperature variations that may not be acceptable in a practical context.

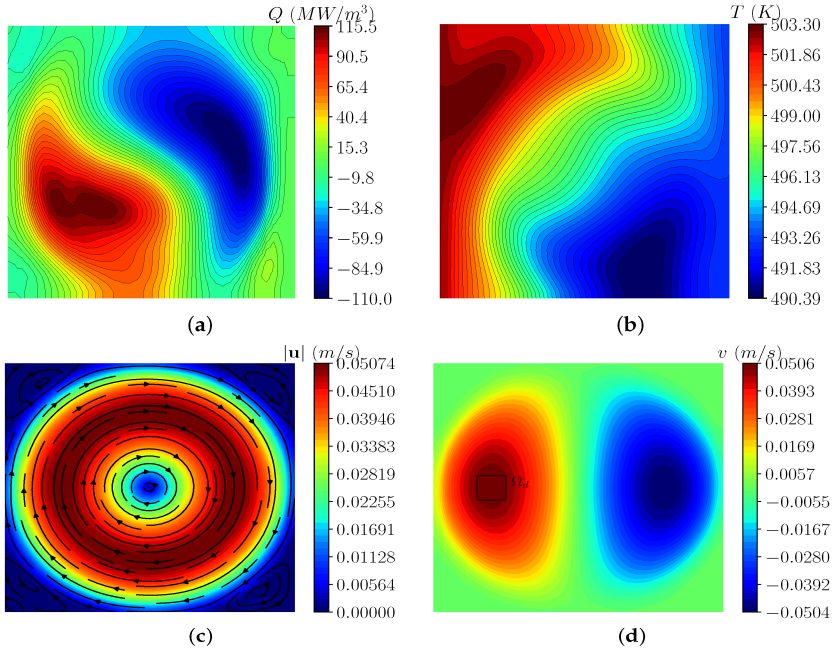


Figure 12: Velocity matching case with distributed control: optimal solution for $\lambda=10^{-11}$. Contours of the control Q (a); contours of the temperature field T (b); contours and streamlines of the velocity field u (c); contours of the y-component of velocity v (d). The velocity magnitude is indicated by |u|, and Ω_d is the region where the objective is set.

CONCLUSIONS

In this work, optimal control problems for incompressible Newtonian buoyant flows were presented and discussed. Starting from some important results already presented in previous studies on the existence of an optimal solution and the existence of the Lagrange multipliers, we analyzed Dirichlet, Neumann, and distributed optimal control problems. For each case, we obtained the optimality system, which consists of state, adjoint, and control equations. To solve this numerically, a gradient method was introduced, and an efficient numerical algorithm was proposed for each case. We observed that the three control mechanisms differed only in the control equation, which is an algebraic equation in the case of distributed and Neumann control and a differential equation in the case of Dirichlet control. In all the mechanisms, the controls depended on the adjoint temperature field Ta and on the regularization parameter λ. Numerical simulations were performed

to test the robustness of the algorithm and the feasibility of the method. The developed numerical simulations included velocity matching cases and temperature matching cases, both evaluated with various values of the regularization parameter λ. We observed that the temperature matching case is easier to achieve, since in this case the distance from the target temperature appears directly as source term in the adjoint temperature equation. The choice of the value of the regularization parameters proved to be a key issue: too much regularization leads to smoother but less effective controls, while a lack of regularization causes numerical issues and singular solutions. We observed that the appropriate choice of λ should be made on a case-by-case basis. A comparison among the three thermal control mechanisms allowed us to draw some conclusions as follows. The strongest control is the distributed control, followed by the Neumann and Dirichlet boundary controls. Of course, all these three different controls can be feasible at different costs, depending on the engineering applications. In general, the developed numerical algorithm showed good convergence properties and thus can be considered a useful tool for the numerical resolution of optimal control problems for Boussinesq equations.

AUTHOR CONTRIBUTIONS

Formal analysis, A.C., V.G. and S.M.; Investigation, A.C., V.G. and S.M.; Software, A.C., V.G. and S.M.; Validation, A.C., V.G. and S.M.; Writing—original draft, A.C., V.G. and S.M.; Writing—review & editing, A.C., V.G. and S.M. All authors have read and agreed to the published version of the manuscript.

REFERENCES

1. Chirco, L.; Chierici, A.; Da Vià, R.; Giovacchini, V.; Manservisi, S. Optimal Control of the Wilcox turbulence model with lifting functions for flow injection and boundary control. *J. Phys. Conf. Ser.* 2019, *1224*, 012006.
2. Gunzburger, M.D.; Hou, L.S.; Svobodny, T.P. The approximation of boundary control problems for fluid flows with an application to control by heating and cooling. *Comput. Fluids* 1993, *22*, 239–251.
3. Lee, H.C. Analysis and computational methods of Dirichlet boundary optimal control problems for 2D Boussinesq equations. *Adv. Comput. Math.* 2003, *19*, 255–275.
4. Aulisa, E.; Bornia, G.; Manservisi, S. Boundary control problems in convective heat transfer with lifting function approach and multigrid vanka-type solvers. *Commun. Comput. Phys.* 2015, *18*, 621–649.
5. Lee, H.C.; Shin, B.C. Piecewise optimal distributed controls for 2D Boussinesq equations. *Math. Methods Appl. Sci.* 2000, *23*, 227–254.
6. Gunzburger, M.D.; Kim, H.; Manservisi, S. On a shape control problem for the stationary Navier-Stokes equations. *ESAIM Math. Model. Numer. Anal.* 2000, *34*, 1233–1258.
7. Gunzburger, M.D. *Perspectives in Flow Control and Optimization*; SIAM: Philadelphia, PA, USA, 2003; Volume 5.
8. Smith, C.F.; Cinotti, L. Lead-cooled fast reactor. In *Handbook of Generation IV Nuclear Reactors*; Elsevier: Amsterdam, The Netherlands, 2016; pp. 119–155.
9. Lee, H.C.; Imanuvilov, O.Y. Analysis of optimal control problems for the 2-D stationary Boussinesq equations. *J. Math. Anal. Appl.* 2000, *242*, 191–211.
10. Lee, H.C. Optimal control problems for the two dimensional Rayleigh—Bénard type convection by a gradient method. *Jpn. J. Ind. Appl. Math.* 2009, *26*, 93–121.
11. Lee, H.C.; Kim, S.H. Finite element approximation and computations of optimal Dirichlet boundary control problems for the Boussinesq equations. *J. Korean Math. Soc.* 2004, *41*, 681–715.
12. Lee, H.C. Analysis and computations of Neumann boundary optimal control problems for the stationary Boussinesq equations. In Proceedings of the 40th IEEE Conference on Decision and Control

(Cat. No. 01CH37228), Orlando, FL, USA, 4–7 December 2001; Volume 5, pp. 4503–4508.

13. Alekseev, G. Solvability of stationary boundary control problems for heat convection equations. *Sib. Math. J.* 1998, *39*, 844–858.
14. Alekseev, G.; Tereshko, D. Boundary control problems for stationary equations of heat convection. In *New Directions in Mathematical Fluid Mechanics*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–21.
15. Baranovskii, E.S.; Domnich, A.A.; Artemov, M.A. Optimal boundary control of non-isothermal viscous fluid flow. *Fluids* 2019, *4*, 133.
16. Baranovskii, E.S. Optimal boundary control of the Boussinesq approximation for polymeric fluids. *J. Optim. Theory Appl.* 2021, *189*, 623–645.
17. Baranovskii, E.S. The optimal start control problem for two-dimensional Boussinesq equations. *Izv. Math.* 2022, *86*, 221–242.
18. Ciarlet, P.G. *The Finite Element Method for Elliptic Problems*; SIAM: Philadelphia, PA, USA, 2002.
19. Droniou, J. Non-coercive linear elliptic problems. *Potential Anal.* 2002, *17*, 181–203.
20. Chierici, A.; Giovacchini, V.; Manservisi, S. Analysis and numerical results for boundary optimal control problems applied to turbulent buoyant flows. *Int. J. Numer. Anal. Model.* 2022, *19*, 347–368.
21. Giovacchini, V. Development of a numerical platform for the modeling and optimal control of liquid metal flows. Ph.D. Thesis, University of Bologna, Bologna, Italy, 2022.
22. Chierici, A.; Barbi, G.; Bornia, G.; Cerroni, D.; Chirco, L.; Da Vià, R.; Giovacchini, V.; Manservisi, S.; Scardovelli, R.; Cervone, A. FEMuS-Platform: A numerical platform for multiscale and multiphysics code coupling. In Proceedings of the 9th edition of the International Conference on Computational Methods for Coupled Problems in Science and Engineering (COUPLED PROBLEMS 2021), Virtual, 13–16 June 2021.
23. Barakos, G.; Mitsoulis, E.; Assimacopoulos, D. Natural convection flow in a square cavity revisited: Laminar and turbulent models with wall functions. *Int. J. Numer. Methods Fluids* 1994, *18*, 695–719.

Chapter

FRACTALS: AN ECLECTIC SURVEY, PART II

15

Akhlaq Husain [1], Manikyala Navaneeth Nanda [2], Movva Sitaram Chowdary [2], and Mohammad Sajid [3]

[1]Department of Applied Sciences, BML Munjal University, Gurgaon 122413, India
[2]School of Engineering & Technology, BML Munjal University, Gurgaon 122413, India
[3]Department of Mechanical Engineering, College of Engineering, Qassim University, Buraydah 51452, Saudi Arabia

ABSTRACT

Fractals are geometric shapes and patterns that can describe the roughness (or irregularity) present in almost every object in nature. Many fractals may repeat their geometry at smaller or larger scales. This paper is the second (and last) part of a series of two papers dedicated to an eclectic survey of fractals describing the infinite complexity and amazing beauty of fractals

Citation: (APA): Husain, A., Nanda, M. N., Chowdary, M. S., & Sajid, M. (2022). Fractals: An Eclectic Survey, Part II. Fractal and Fractional, 6(7), 379.(38 pages).

from historical, theoretical, mathematical, aesthetical and technological aspects, including their diverse applications in various fields. In this article, our focus is on engineering, industrial, commercial and futuristic applications of fractals, whereas in the first part, we discussed the basics of fractals, mathematical description, fractal dimension and artistic applications. Among many different applications of fractals, fractal landscape generation (fractal landscapes that can simulate and describe natural terrains and landscapes more precisely by mathematical models of fractal geometry), fractal antennas (fractal-shaped antennas that are designed and used in devices which operate on multiple and wider frequency bands) and fractal image compression (a fractal-based lossy compression method for digital and natural images which uses inherent self-similarity present in an image) are the most creative, engineering-driven, industry-oriented, commercial and emerging applications. We consider each of these applications in detail along with some innovative and future ready applications.

Keywords: fractals; iterated function system; fractal landscapes; fractal antenna; fractal image compression; fractal batteries; fractal capacitors; fractal solar panels

INTRODUCTION

Mandelbrot conceived the term ‘Fractal’ (in 1975) from the Latin word *fractus*, which means “broken” or “fractured” to describe irregular geometries in mathematics and in nature. Fractals are geometric objets that may repeat their geometry at smaller (or larger) scales due to the inherent self-similarity present in the object. Among several examples of well-known fractals, some classical examples are the Cantor set, the Sierpinski triangle, the Koch curve, the Mandelbrot set and Julia sets.

Many natural and man-made objects can be characterized using the classical Euclidean geometry and have integer dimension. However, the random geometry of natural objects such as a fern leaf, branching in human lungs, flowering head of broccoli, lightening during a storm, turbulence in a terrestrial body, coastlines, etc. can only be described more precisely using fractal geometry, and they have a non-integer fractal dimension.

Several hundred research articles are available in the literature covering various aspects of fractals including their mathematical development, scientific importance, engineering and industry applications. However, only a few references exists that cover a broader spectrum of fractals in one place,

and most of these are in the form of monographs. Our prime objective of this survey is to provide a unified review of the work completed (over the past 5 decades) in the ever-growing field of fractal geometry covering length and breadth at once that will assist readers from various fields of academic and industry.

This comprehensive survey is written with the intent of providing a collative review of recent research, developments, and applications of fractals in a series of two papers. In Part-I [1], we covered a brief mathematical description of fractals, fractal dimension (which is usually a non-integer characteristic number attached to every fractal in contrast with Euclidean dimension) and applications of fractals in arts, tessellations, fashion designing, and other emerging fields such as econophysics, etc. This article is the second and last part of this survey with the aim of exploring engineering, industrial and commercial applications (including recent developments) of fractals in fractal landscapes, fractal antennas, fractal image compression, fracture mechanics and other evolving future applications of scientific and engineering research. We will see several fractal innovations which are making a great impact in modern technologies and will remain open for explorations in the future as well.

The article starts with an introduction to one of the most amazing discoveries in mathematics, namely the Mandelbrot set in Section 2. The space of fractals (the mathematical set where fractals live) and other elementary concepts are introduced in brief to give a flavor of the mathematics behind fractals to the reader, although the article is easy to follow by the majority of the scientific community without deeper understanding of mathematics.

Fractals are widely used for rendering landscapes in the computer graphics industry. The advent of fractal landscapes in computer graphics goes to Mandelbrot, who was the first to identify the similarity between the trace of fractional Brownian motion over time and the skyline of jagged mountain peaks [2] and explained the connection between visual approximation of natural mountains with the two-dimensional Brownian surfaces. This approach was implemented by Mandelbrot in [3] with the earliest computer graphics images of such surfaces and for the creation of fractal coastlines. Natural landscapes contain fractal characteristics and statistically self-similarity or self-affinity. In Section 3, we consider fractal landscapes, and standard algorithms for generating fractal landscapes are discussed.

In today's technology-driven world, antennas form an indispensable part of our life. They are used in cell phones, TV, radio, radars, WI-FI, IOT, bluetooth devices, and so on. There has been an incredible demand for the design of antennas that are compact and multiband or broadband. Properties of fractals can be exploited to achieve these multiple characteristics in a single antenna. Traditional antenna designs are based on Euclidean geometries; however, innovative antenna designs have emerged by exploiting the inherent self-similarity and space-filling properties of fractals. A fractal antenna is a revolutionary innovation in telecommunications. Fractal-shaped antennas have a large effective length, small size, and reduced weight with performance parameters, owing to the special geometry and compact structure of fractal shapes. Section 4 gives a detailed survey of different types of existing fractal antenna introduced over the last 2–3 decades along with historical developments and their applications in various communication systems.

Another important application of fractals is found in compressing data (e.g., images, music, and videos). Images are stored as a collection of bits representing pixels on a computer, and storing a collection of images requires large memory. This problem can be addressed using various image compression techniques that exist. Fractal Image Compression (FIC) is a powerful and evolving image compression technique, which is based on fractal coding that exploits the self-similarity property of an image. FIC is simple to implement, provides high compression ratios and fast decompression with the only drawback of slow compression. Barnsley introduced the fractal image compression in 1987, who founded Iterated Systems Inc. (a pioneer company in fractal image compression technology). In Section 5, we discuss various aspects, algorithms and applications of fractal compression.

Fracture mechanics is the study of propagation of cracks or failures of the structures in materials, and it is an important tool to improve the performance and quality of mechanical components. Mandelbrot was the first to interrelate the crack propagation and other fracture properties with the fractal geometry. He introduced the method of slit island analysis on the fracture surface to find fracture dimensions. Characteristics of the fractal geometry such as self-similarity (or self-affinity), scale invariance and fractal dimension have offered great help to analyze irregular or fractional shapes of fracture mechanics. Section 6 discusses these aspects in more details.

Finally, in Section 7, biological applications of fractals are discussed with particular emphasis on ophthalmology. Other emerging applications of fractals such as fractal batteries, fractal electromagnets, fractal cooling chips, fractal PCBs, fractal solar panels, fractal capacitors, and fractals in biometric applications are also given here.

The two-part survey is organized is such a way that a reader will enjoy reading both parts independently without losing continuity and it will delight the readers with the applications of fractals in emerging and innovative fields of current and future technologies.

MATHEMATICS OF FRACTALS

Figure 1 shows Benoît Mandelbrot's eponymous set, which is popularly known as the Mandelbrot set, which is a mathematical fractal. The Mandelbrot set is among the most complex sets in mathematics and the best-known examples of mathematical visualization, self-similarities, and delightful patterns that are visible when we zoom on the set.

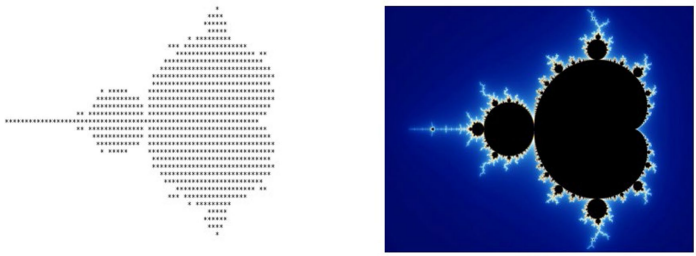


Figure 1: The Mandelbrot set: (a) first image (1978) and (b) image generated by Mandelbrot (1980).

R. Brooks and P. Matelski published the first image (Figure 1a) of the Mandelbrot set in the year 1978. Later, Mandelbrot plotted the true image of the Mandelbrot set on 1 March 1980 (Figure 1b). This set is obtained by plotting the complex numbers c in the simple (quadratic) polynomial

$$f_c(z) = z^2 + c,$$

whose orbits remain bounded. Generalized Mandelbrot sets can also be plotted by considering the higher degree polynomials

$$f_c(z) = z^n + c, \quad n > 1.$$

In Figure 2a–d, generalized Mandelbrot sets are displayed for n=3,4,5 and 10. We refer to [4] for an interesting work on generalized Mandelbrot sets with chaotic features obtained by replacing z2 with Möbius transformations, transcendental functions, etc. Some properties of these generalized sets are also discussed in contrast with the original Mandelbrot set [4].

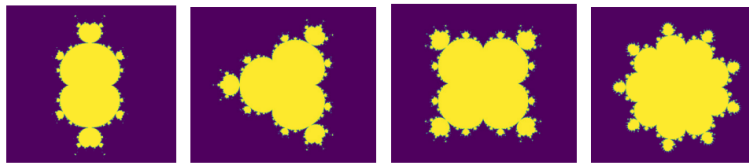


Figure 2: Generalized Mandelbrot Sets for (a) n=3 (b) n=4, (c) n=5 and (d) n=10.

The Mandelbrot set has become so popular that this set and its details (the Julia sets which live on the boundary of the Mandelbrot set) can be seen on cloths, ceramic products, tiles, hot air balloons, calenders, art prints, postcards, posters, commercials and so on. For an incredible zoom on the Mandelbrot set, we refer to [5].

Space of Fractals

Let X be a non-empty set, a function d $d : X \times X \to \mathbb{R}^+$ is called a metric or a distance function on X if it satisfies

(i) $d(x,y) \geq 0 \quad \text{and} \quad d(x,y) = 0 \Leftrightarrow x = y, \quad \forall \quad x,y \in X,$

(ii) $d(x,y) = d(y,x), \quad \forall \quad x,y \in X,$

(iii) $d(x,y) \leq d(x,z) + d(z,y), \quad \forall \quad x,y,z \in X.$

The set X together with the function d is called a metric space, and it is denoted by (X,d).

A metric space X is said to be *complete* if every Cauchy sequence is convergent in X and a subset S⊆X is said to be *compact* if every infinite sequence of points in S has a convergence subsequence. A complete metric space and its compact subsets are fundamental tools to describe and understand fractal geometry, which is essentially the classification, description, analysis and observations of subsets of metric spaces.

Definition 1.

Let (X,d) *be a complete metric space and* H(X) *be the set of non-empty compact subsets of X. For any* A,B∈H(X)*, define the distance between A and B by*

$h(A,B)=\max\{d(A,B),d(B,A)\}$,

where $d(A,B)=\sup_{x\in A}\inf_{y\in B}\{d(x,y)\}$.

Then, it is easy to verify that *h* is a metric on H(X). This metric *h* is called the Hausdorff metric on H(X), and the set H(X) is called the space of fractals equipped with the Hausdorff metric *h*.

Theorem 1.

The space (H(X),h) *is a complete metric space.*

Proof.

See Barnsley [6] (Chapter 2).

Any subset of (H(X),h) is a mathematical fractal, although the Euclidean objects such as rectangles, parallelograms, spheres and cylinders are not considered as fractals, since they do not possesses self-similarity, but they are elements of (H(X),h) and can be considered as (mathematical) fractals if there no confusion is likely to occur.

Iterated Function Systems and Attractors

Definition 2.

A mapping or a transformation w:X→X *on a metric space* (X,d) *is called a contraction mapping if*

$$d(w(x),w(y)) \le \alpha\, d(x,y) \quad \forall x,y \in X. \tag{1}$$

for some constant 0≤α<1. *The constant α is called contractivity factor of w.*

Definition 3.

A finite set of contraction mappings w_i:X→X*, where X is a metric space equipped with the metric d having contractivity factors* α_i*, for* i=1,2,…,m *is called an iterated function system (IFS). The number*

$$\alpha = \max_{1 \le i \le m} \alpha_i,$$

is called a contractivity factor of the IFS.

Theorem 2

(Hutchinson [7]). *Let* $\{X, w_i : i=1,2,\ldots,m\}$ *be an IFS with contractivity factor* α. *Then, the transformation* $W:H(X)\to H(X)$ *defined by*

$$W(B) = \bigcup_{i=1}^{m} w_i(B), \tag{2}$$

for all $B \in H(X)$ *is a contraction mapping on* $H(X,h(d))$ *with contractivity factor* α.

Therefore, by the contraction mapping theorem, the mapping W has a unique fixed point $A \in H(X)$ given by

$$A = \lim_{n\to\infty} W^{\circ n}(B), \quad B \in \mathcal{H}(X).$$

Here, $W^{\circ m}(B)$ denotes the m-fold forward iterate of W.

Definition 4.

The unique fixed point A described in Theorem 2 is called the attractor of the IFS. Moreover, since $A \in H(X)$, *therefore, it is a (mathematical) fractal.*

The examples of mathematical and natural fractals to be presented in the ensuing sections of this article are geometrically intricate subsets of Euclidean spaces $\mathbb{R}^2$ or $\mathbb{R}^3$, which are elements of H(X) with $X = \mathbb{R}^d$,d=2,3.

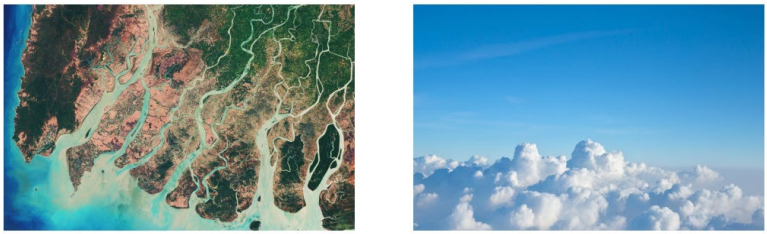
FRACTALS IN NATURAL AND ARTIFICIAL LANDSCAPES

A fractal landscape is typically a surface that displays fractal behavior obtained by an algorithm and mimics the appearance of a natural terrain. Midpoint displacement methods by Fournier et al. [8], Miller [9], Musgrave [10] and others were introduced as fast landscape and terrain generation techniques and are standard in fractal geometry. Ken Musgrave (a student of Mandelbrot) discovered new processes of fractal landscape generation [10]. He worked on Bryce landscape software, which made use of many algorithms (midpoint displacement algorithm was one of those). The midpoint

displacement methods were modified and improved in [11,12] for natural terrain simulations and to construct self-affine geometrical objects which are similar to rock fractures.

Examples of natural fractal landscapes are found in geography, mountains, rivers, and terrains. A natural fractal mountain is shown in Figure 3, and a natural delta formed by a flowing river and a fractal shape profile of clouds is displayed in Figure 4.

Figure 3: A fractal mountain.



(**a**) (**b**)

Figure 4: (a) A fractal river detla, (b) a fractal sky cloud.

F. Kenton Musgrave was the first to generate computer-based realistic landscapes. He was referred to as "*the first true fractal-based artist*" by Mandelbrot for his Ph.D. thesis work on *Methods for Realistic Landscape Imaging* [10]. Musgrave's thesis work turned out to be a comprehensive road map for rendering modern fractal landscapes using computer programs even today. Musgrave founded the Pandromeda, Inc. and developed the innovative MojoWorld software (obsolete now), a commercial and fractal-based modeling program for the creating digital landscapes, space art and science fiction scenes. The MojoWorld was applied in creating background mattes and terrains on big-budget movies such as *Titanic*, *The Day After Tomorrow*, etc. Figure 5 shows realistic examples of computer-generated fractal landscapes. Notice the true similarities between Figure 3 and Figure 5.

Figure 5: Computer-generated examples of (a) a fractal terrain, (b) a fractal woodhill, and (c) a fractal landscape. (Image source: https://en.wikipedia.org/wiki/Fractal_landscape, accessed on 22 June 2022).

Generation of Fractal Landscapes

There is a large variety of commercial and academic purpose software that can generate and allow for editing of fractal landscapes. The list includes Bryce (a feature-packed 3D modeling and animation package specializing in fractal landscapes), midpoint displacement algorithm (landscapes generation in many dimensions), diamond-square algorithm [8] (slightly better algorithm than midpoint displacement algorithm), Terragen (designed and developed by the Planetside Software for Microsoft Windows and Mac OS X and capable of generating captivating sceneries and animations of fractal landscapes), L3DT (similar as the Terragen program with a 2048×2048 limit) and World Creator (can create terrain, fully GPU powered), etc. Figure 6a displays a Julia island, and an example of a *Mandel River* generated by the software Terragen is shown in Figure 6b, which depicts the details of the Mandelbrot set.

Figure 6: (a) Julia island (Image source: https://en.wikipedia.org/wiki/Terragen, accessed on 22 June 2022) and (b) Mandel river (details of the Mandelbrot set) rendered by Terragen Classic. (Image source: https://en.wikipedia.org/wiki/Fractal-generating_software, accessed on 22 June 2022).

We now describe some of the above-mentioned fractal rendering algorithms to allow the reader deeper insight and better understanding on how the fractal landscape generation algorithms work.

Midpoint Displacement Algorithm in 1d (1DMD)

The midpoint displacement algorithm is based on the von Koch curve construction. The credit for its applicability and popularity in computer graphics goes to Fournier, Fussell, and Carpenter for rendering fractal landscapes and clouds. The algorithm is very simple and proceeds as follows:

Start with a straight line segment and mark its midpoint. Now, select a random (bounded) value and displace the midpoint of the line segment by this random value in the direction perpendicular to the line segment or displace only the y coordinate of the midpoint (see Figure 7a).

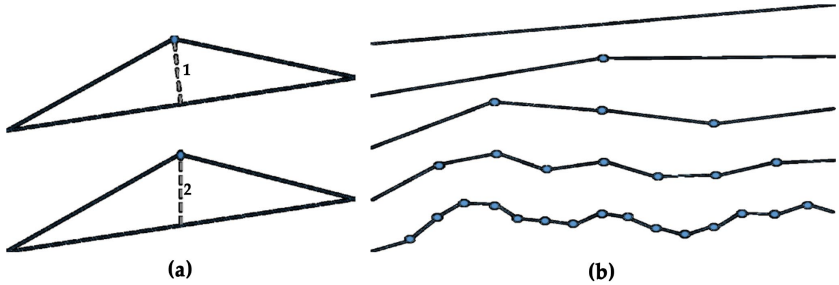


Figure 7: (a) Strategies to displace the midpoint and (b) Successive iterations of the algorithm (from left to right). (Image source: https://bitesofcode.wordpress.com/2016/12/23/landscape-generation-using-midpoint-displacement/, accessed on 22 June 2022).

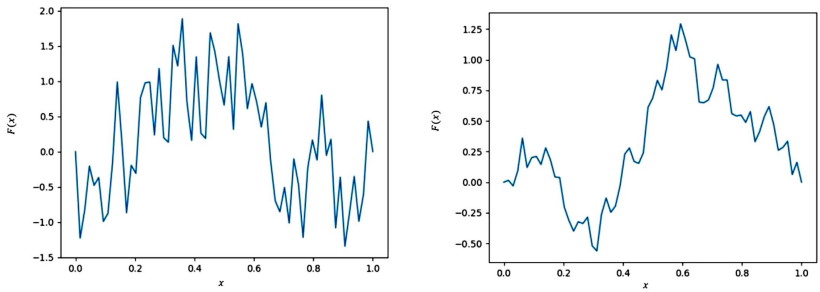
This will result in two smaller line segments. In the second iteration, repeat this process to mark and displace the midpoints of each line segment obtained in the first iteration by a random amount, and this will result in four straight line segments. The process is continued until the desired level of detail is achieved by reducing the random displacement in every iteration. For example, if the displacement was reduced by half in the first iteration and the random displacement value is chosen from the interval $[-1,1]$, then the range for the second iteration with two midpoints would be in the interval $[-0.5,0.5]$, in $[-0.25,0.25]$ for the third iteration, and so on. The equation for the midpoint value is given by

$$F(x) = \frac{\left(F\left(x + \frac{dx}{2}\right) + F\left(x - \frac{dx}{2}\right)\right)}{2} + Kr \cdot 2^{-nH}, \tag{3}$$

where r∈[−1,1] is a random number and K is a constant which controls the amplitude of the variation. H is the roughness parameter (the factor by which the perturbations are reduced on each iteration), and n denotes the iteration number. Increasing the value of H will produce smoother landscapes. Figure 7b displays successive iterations of the algorithm. The pseudocode for the algorithm is given in Algorithm 1.

Algorithm 1: Pseudocode for midpoint displacement algorithm.
Pseudocode:
initialize *line segment*
initialize *max_iter*, *min_len*
while *iteration* <*max_iter* and *segment_length*> *min_len*:
for each segment:
choose random displacement
compute midpoint
displace midpoint
update segments
reduce displacement
iteration+1

By suitably choosing the displacement bounds and the reduction factor H, one can control the geometry and the roughness of the landscape. Higher values of H result in smoother landscapes, and lower values result in spiky (sharp) landscapes. Figure 8 depicts several landscapes with varying H values. Observe the change in the smoothness of the landscape with the change in H values.



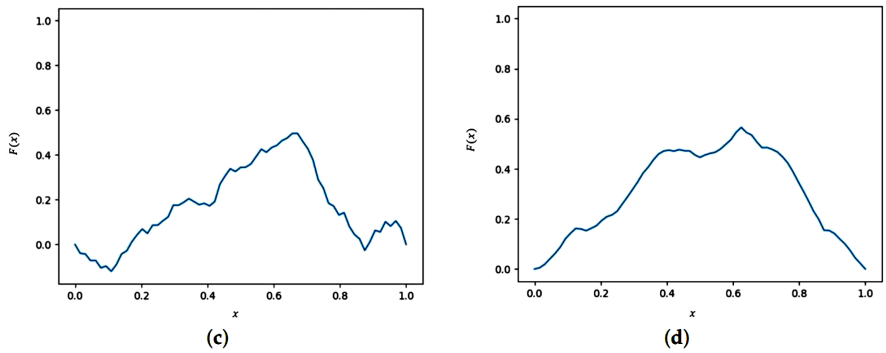


Figure 8. One-dimensional (1D) landscapes for (a) H=0.0, (b) H=0.25, (c) H=0.50, and (d) H=0.75.

In each iteration, the displacement bounds can be reduced by different approaches (e.g., linear, exponential, logarithmic, etc.) depending upon the choice of landscape being generated. The two extremes possibilities are no displacement reduction and exponential displacement reduction (in each iteration) shown in Figure 9 below.

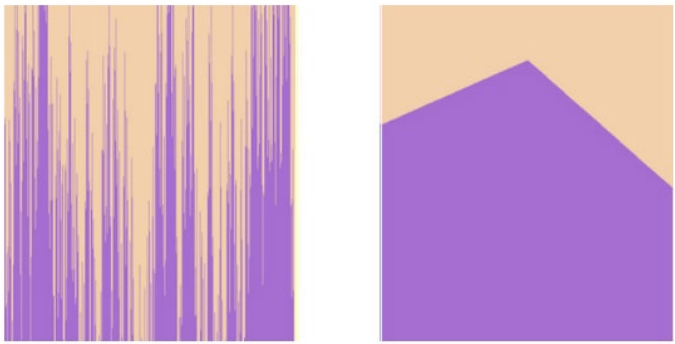


Figure 9: No displacement reduction (left image), Exponential displacement reduction in one iteration (right image). (Image source: https://bitesofcode.wordpress.com/2016/12/23/landscape-generation-using-midpoint-displacement/, accessed on 22 June 202).

Some colored pictures of landscapes generated from the 1D midpoint displacement algorithm are presented in Figure 10.

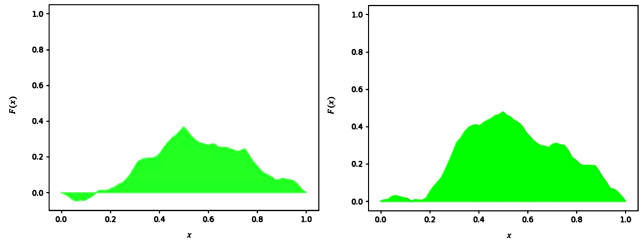


Figure 10: Colored landscapes generated from 1DMD.

Midpoint Displacement Algorithm in 2D (2DMD)

The 2D midpoint displacement algorithm is similar to the 1D algorithm described above, with the only difference that now, the displacement (height) in the *z*-direction is determined over the xy-plane. In most cases, a positive displacement results in the formation of a mountain, and a negative displacement results in the formation of a valley. The advantage of using this algorithm is that the landscapes are dynamically generated, and they will never be the same, as the elevations chosen are random every time.

The roughness of the landscape is controlled in the same way as in a one-dimensional landscape. Changes in H values show drastic changes in the landscape generated: for instance, if the value of H is 0, then the landscape is more spiky, and when it is 1, we obtain smooth landscapes, as seen in surface landscapes generated using 2DMD in Figure 11.

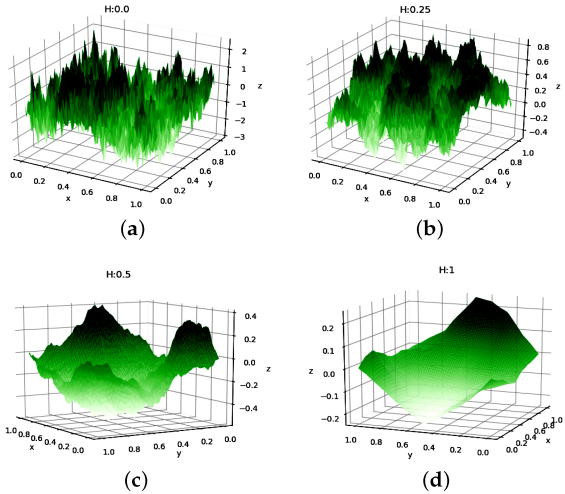


Figure 11: Surface landscapes generated by 2DMD for (a) H=0.0, (b) H=0.25, (c) H=0.50, and (d) H=1.0.

An extension of the 2D midpoint displacement algorithm to three-dimensions was presented in [13] for generating three-dimensional fractal porous media geometries whose surface area can also be controlled by adjusting the random component of the midpoint displacement. They also considered statistical properties for the geometries obtained using 3DMD and showed that the structures generated by 3DMD are more realistic.

The grids used with the midpoint displacement algorithm are uniform in all directions, and typically, they have a size of 2n on all sides, where *n* is an integer. The variable *n* (number of iterations) which is given as an input (by the user) has its own significance. Increasing the value of *n* leads to an increase in the resolution of the landscape, as minute details of fractals will be captured. However, generating fractals with high values of *n* is a time-consuming process and requires high computational powers, so it is important to select an optimal value of *n* by taking into consideration the time, computational power and required resolution.

Diamond Square Algorithm

The diamond square algorithm is a modification of the midpoint displacement method proposed by Fournier et al. [8] (1982), and its name is borrowed from the 2D midpoint displacement algorithm. The midpoint displacement method sometimes leaves square-shaped artifacts in generated terrains. The diamond square algorithm attempts to alleviate this by alternating calculated values to square and diamond patterned midpoints. The algorithm starts with a 2D square grid of boxes having 2^n squares containing 2^{n+1} grid points. The four corner points of the grid are first set to initial values. The diamond and square steps are then executed one after the other until all grid points have been assigned as follows:

- *The diamond step*: For each square in the array, set the midpoint of that square to be the average of the four corner points plus a random value.
- *The square step*: For each diamond in the array, set the midpoint of that diamond to be the average of the four corner points plus a random value.

Figure 12 shows the algorithmic steps of the algorithm. The magnitude of the random value should be reduced in each iteration.

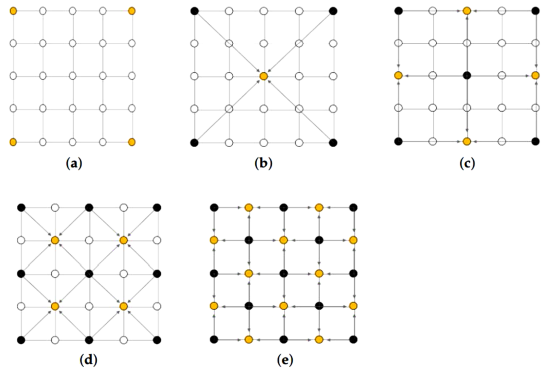


Figure 12: Diamond square algorithm on a 5×5 array: (a) Initialize corner grid values, (b) execute diamond step, (c) execute square step, (d) execute diamond step, (e) execute square step. (Image source: https://en.wikipedia.org/wiki/Diamond-square_algorithm, accessed on 22 June 2022).

Miller [9] analyzed the diamond square algorithm in 1986 and described it as flawed due to possible perturbations in the rectangular grid. The grid artifacts were resolved by J.P. Lewis in a generalized algorithm [14]. Some landscapes images generated by the diamond square algorithm at different *H*-values are shown in Figure 13.

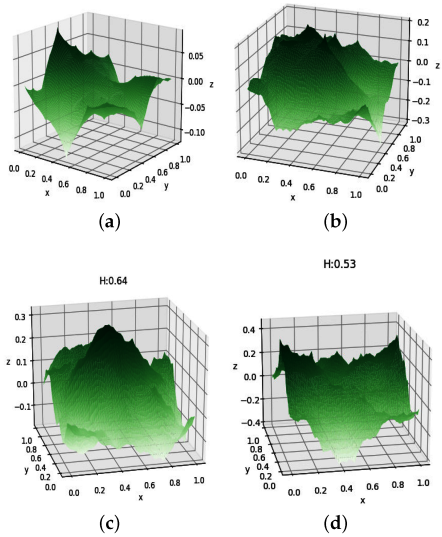


Figure 13. Surface landscapes from diamond square algorithm at different values of *H*: (a) H=1.0, (b) H=0.70, (c) H=0.64, and (d) H=0.53.

SUMMARY

According to Musgrave [10], the generation of realistic fractal landscapes or creating *fractal forgeries of nature* consists of geometric models, designing efficient algorithms, atmospheric effects (for sense of scale), surface textures, and a global context for embedding the scenes. In this brief essay on fractal landscapes, we briefed the pioneering work by several authors including the work of Musgrave on analysis and algorithms that are available for creating fractal landscapes. The review highlights the potential of fractal geometry to understand and design fractal landscapes. Fractal landscape generation is evolving rapidly, and the design of new and fast algorithms is still under development.

FRACTAL ANTENNAS

Antennas are an integral part of any communication system, and they are widely used in electromagentic devices such as cell phones, TV, radio, radars, electronic devices, and so on. With the advancement of technology, the world is becoming more dependent on compact, bluetooth, WI-FI and IOT smart devices. Therefore, the need is to design antennas for commercial and defence sectors that are compact, light weight, and multiband or broadband. A natural choice to obtain these antenna characteristics is to exploit the properties of fractals. Today, many novel and powerful antenna designs have emerged from modern (fractal) geometry, which are replacing the traditional antenna designs based on Euclidean geometries.

A fractal antenna is a revolutionary invention in the field of telecommunication. Using a fractal-shaped antenna as a replacement of a circuit with discrete components has helped in increasing the effective length and reducing the size and weight of the antenna. At the same time, the performance parameters have improved, owing to the self-similar geometry (which maximizes the effective length of an antenna for a given surface area) and compact structure of fractal shapes. A large number of fractal antenna designs have been proposed combining fractal geometry with electromagnetic theory, and this has led to an area called fractal antenna engineering [15].

In this section, we review standard fractal-shaped antennas proposed and simulated by many researchers in the past two decades, since the pioneering works of Cohen and Puente [16,17,18,19]. The work by Werner et al. [15] summarizes various techniques for compact (i.e., miniature) fractal antennas

designs. We also refer to the recent survey papers [20,21] for an extensive study of the literature and state of the art summary of fractal antenna research. The reader may also consider exploring the articles [19,20,22,23] for more detailed analysis, various types and applications of fractal antennas available in the literature.

Brief History

Nathan Cohen was the first to built a wire fractal antenna using von Koch curves in 1988 (at Boston University) by setting up a 'ham' radio station, and he also designed the planar fractal arrays using Sierpinski triangles. Cohen co-founded Fractal Antenna Systems Inc. in 1995 as the first fractal-based commercial antenna solutions, and he also designed fractal cellular antennas for Motorola phones, which were proven to be 25% more efficient than the conventional helical antenna. Another company founded by C. Puente and R. Bonet, namely Fractus S.A. in Barcelona (Spain), is involved in fractal antenna research, patents and commercialization.

In August 1995, Cohen published the first article on fractal antenna [16], and Puente carried out early work on fractals as multiband antennas [24]. Therefore, the credit for demonstrating the potential of fractal antennas as a replacement for traditional antennas is jointly shared by Cohen and Puente.

Because of their special geometry, fractal antennas are self-loading and often do not need matching circuitry for multiband or broadband characteristics. This lowers the fabrication cost and increases the reliability. Exploiting the self-similar fractal designs, one can fabricate fractal antennas that are compact and wideband. The fractal-shaped antennas can have multiple resonances (self-similar design works as a virtual network of capacitors and inductors), making a single antenna operate on multiple electromagnetic frequencies. Due to space-filling properties, fractal antennas make better use of the available volume inside the radian sphere. Therefore, they may radiate more effectively than the one-dimensional straight wire [18].

Antenna Parameters

While designing an antenna, one must consider different combinations of antenna parameters based on the type of application for which the antenna is being fabricated. For instance, antennas used for television must have higher bandwidths to support higher data transmission rates. For radio, the antennas' range and capability to work at multiple bands is considered

more important, and for modern antennas, the size of the antenna matters a lot, since nanotechnology is the direction in which the world is moving. Thus, antenna parameters play a vital role in the design, fabrications and applications. Before we look at some examples of fractal antennas, let us briefly describe some of the key antenna parameters.

Impedance

Transmission lines are used to feed antennas, and to transmit the maximum available power or to receive the transmitted power, it is necessary to know the impedance at the input where the transmission line is to be connected. For optimal power transfer from the antenna to the receiver or from the transmitter to the antenna, the input impedance of the transmission line must be same as the input impedance of the antenna. In case of impedance mismatch, an impedance matching circuit is required.

Return Loss

The return loss compares the power reflected by the antenna to the power that is fed into the antenna from the transmission line. It is measured in dB, and the relationship between SWR (Standing Wave Ratio, a measure of impedance matching) and return loss is given by

$$\text{Return loss}(dB) = 20 \log_{10} \frac{SWR}{SWR - 1}.$$

Bandwidth

Bandwidth refers to the range of frequencies over which the antenna can properly radiate or receive energy. The desired bandwidth is one of the key parameters for an antenna design. The antenna's bandwidth is the number of Hz for which the antenna will exhibit an SWR less than 2:1. The bandwitdh of an antenna is defined by

$$B = f_h - f_l,$$

where B= Bandwidth, f_h= Higher cut-off frequency, f_l= Lower cut-off frequency.

The bandwidth can also be described in terms of percentage of the center frequency of the band

$$B = \frac{f_h - f_l}{f_c} \times 100,$$

where f_c is the center frequency in the band. Bandwidth is typically quoted in terms of VSWR. The bandwidth of an antenna varies according to its type and application.

Directivity

Directivity is the ability to focus the concentration of an antenna's radiation pattern in a particular direction when transmitting or to receive energy from a particular direction. Directivity is denoted by *D* (expressed in dB) and defined by

$$D = \frac{F\text{max}}{F_{\text{iso}}},$$

where F_{max}= maximum signal strength radiated by the antenna, F_{iso}= maximum signal strength radiated by the isotropic antenna (an antenna that radiates power equally in all directions).

Antenna Efficiency

The efficiency of an antenna is the ratio of the power radiated by the antenna to the power radiated from the antenna.

$$\eta = \frac{P_{radiated}}{P_{input}},$$

where η= antenna efficiency, $P_{radiated}$= power radiated, and P_{input}= input power to the antenna.

Antenna Gain

The term antenna gain describes how much power is transmitted in the direction of peak radiation to that of an isotropic source. An antenna's gain (*G*) is a key parameter that combines an antenna's radiation efficiency (η) and directivity (*D*) by the relation:

$G=\eta \times D$.

The antenna gain is expressed in decibels (dB) by:

$$G_{dB} = 10 \cdot \log_{10}(G)$$

In principle, a high-gain antenna will radiate most of its power in one direction, and a low-gain antenna will radiate its power equally in all directions.

Radiation Pattern

The radiation pattern displays the variation of the power radiated by an antenna as a function of the direction away from the antenna. That is, the antenna's pattern describes how the antenna radiates energy out into space (or how it receives energy). A radiation pattern is "isotropic" if the radiation pattern is the same in all directions. Antennas with isotropic radiation patterns do not exist in practice, but they are used for benchmarking with real antennas.

Polarization

The polarization of an antenna is defined as the direction of the electromagnetic fields produced by the antenna as energy radiates away from it, with respect to the surface of the earth, and it is determined by the structure of the antenna and its orientation. These directional fields determine the direction in which the energy moves away from or is received by an antenna.

There are several categories of polarization, and within each type, there are several sub categories such as linear polarization (horizontal, vertical and slant), circular polarization (right-hand circular and left-hand circular), elliptical polarization, omnidirectional polarization, etc.

Types of Antennas

Antennas are classified in many categories based on their physical structure, functionality and types of applications. Well-known examples of antennas including their types and application areas are

- Wire antennas (e.g., dipole antenna, monopole antenna, helix antenna, loop antenna), used in personal applications, buildings, ships, automobiles, space crafts, etc.
- Aperture antennas (e.g., waveguide (opening), Horn antenna), used in flush-mounted applications, aircrafts, spacecrafts, etc.
- Reflector antennas (e.g., parabolic reflectors, corner reflectors) used in microwave communication, satellite tracking, radio astronomy.

- Lens antennas (e.g., convex–plane, concave–plane, convex–convex, concave–concave lenses), used for very high-frequency applications.
- Microstrip antennas (e.g., circular-shaped, rectangular-shaped metallic patch above the ground plane), used in aircrafts, spacecrafts, satellites, missiles, cars, mobile phones, etc.
- Array antennas (e.g., Yagi-Uda antenna, microstrip patch array, aperture array, slotted wave guide array), used for very high-gain applications.

Substrate

Low-profile antennas are needed for high-performance aircrafts, spacecrafts, satellites, missile applications, GSM, GPS and remote sensing applications where size, performance, weight, cost, ease of installation, and aerodynamic profile are constraints. All these requirements may be met using a microstrip antenna (MSA). An MSA (also called patch antenna) is a two-dimensional flat structure consisting of a very thin metallic strip placed on a ground plane with a dielectric material in between; this dielectric material is called the *substrate*.

The performance and radiation properties of an antenna can be improved by properly selecting the thickness (h) and permittivity (ϵr) of the substrate. In patch antennas, the smaller permittivity of the substrate yields better radiation. Several dielectric substrates are proposed in the literature for fabricating microstrip patch antennas. Table 1 lists some commonly used substrate materials in the design of fractal antennas along with their dielectric constants.

Table 1: Commonly used substrate materials in fractal antennas

S. No.	Name of the Substrate	Dielectric Constant (ϵ_r)
1.	Bakelite	4.8
2.	Duroid 6010	10.7
3.	Nylon fabric	3.6
4.	Roger 4350	3.48
5.	RT-Duroid	2.2
6.	Foam	1.05
7.	Taconic TLC	3.2
8.	FR-4	4.4

Standard Fractal Antennas

The first application of fractal antennas was in the form of wire antennas proposed by Cohen in a series of papers [16,17] based on fractalization of the geometry of a standard dipole or loop antenna. Almost at the same time, Puente and his collaborators [18,19,24] proposed Koch fractal monopole antennas with improved electrical performance over conventional linear monopole antennas.

Cohen observed that fractal Minkowski loops exhibit low resonant frequency relative to their electric size. Puente found that in Koch fractal antennas, the resonant frequency goes low or toward larger wavelengths with increase in iteration number. Thus, fractal-shaped antennas at higher iterations resonate at low frequencies due to increased length as compared to the antennas of the lower iterations (having smaller length).

Today, most of the wireless devices operate in multiple bands of frequencies. Thus, the design of a multiband antenna is a natural choice for present and future devices. We now provide a brief overview of some popular fractal-shaped antennas, which are proven to be very useful in developing novel, innovative designs for multiband fractal antennas. To keep the presentation shorter, we provide plots for multiband behavior only for the Sierpinski gasket antenna, and we encourage the reader to consider references mentioned here for details on the design, performance and applications of fractal antennas.

Sierpinski Gasket

Figure 14 shows the first five stages in the construction of the Sierpinski gasket antenna (named after the Polish mathematician Sierpinski.

Figure 14: Sierpinski gasket antenna through five stages of growth.

The Sierpinski gasket is obtained by continuing the iterations to infinity. From an antenna engineering perspective, the colored (filled) triangular regions represent a metallic conductor, whereas the white (hollow) triangular regions represent areas where the metal has been removed. The self-similar geometry of Sierpinski gaskets allows for fabricating multiband fractal antenna elements.

The Sierpinski gasket antennas resemble a bow-tie antenna, and one antenna can perform similar to multiple bow-tie antennas, since the iterated Sierpinski gasket consists of many Sierpinski gaskets at different scales, which can be seen by looking at Figure 15a. A fabricated Sierpinski gasket antenna is shown in Figure 15b, and the lengths of the largest side of the antenna are shown in Figure 15c at various scales.

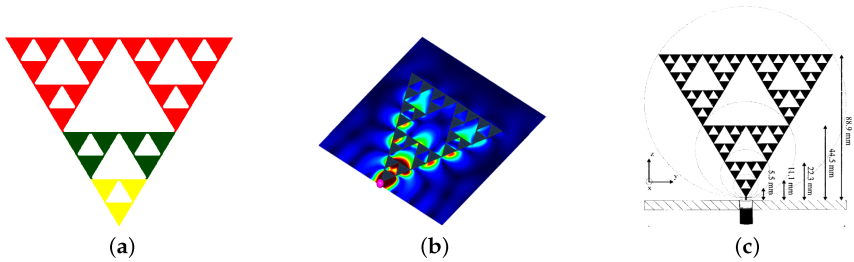


Figure 15: Resemblance of Sierpinski gasket antenna to bow-tie antenna. (a) Three stages of Sierpinski antenna, (b) Fabricated Sierpinski antenna, (c) Length scales of Sierpinski antenna. (Image source: https://www.emcos.com/wp-content/uploads/2014/01/Application_Note_Fractal_Antennas_Simulation_Sierpinski_Gasket.pdf, accessed on 22 June 2022).

The multiband performance of this Sierpinski antenna is visible in Figure 16, where some plots are given between the S_{11} parameter (which gives the amount of power reflected from the antenna) and the frequency. S_{11}=0 signifies that all the power is reflected, so the negative sharp down peaks are considered as the resonating frequencies.

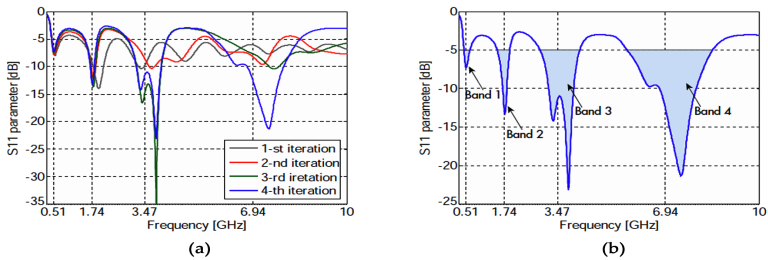
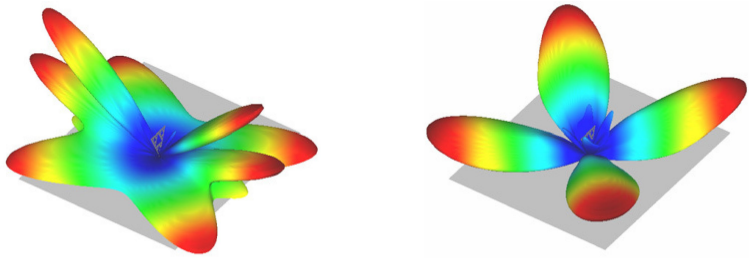


Figure 16: S_{11} plots for Sierpinski gasket antenna (a) all iterations (b) 4th iteration. (Image source: https://www.emcos.com/wp-content/uploads/2014/01/Application_Note_Fractal_Antennas_Simulation_Sierpinski_Gasket.pdf, accessed on 22 June 2022).

The simulated characteristics of the Sierpinski gasket monopole antenna were shown to be matching with the analytical results in [25]. Moreover, the antenna resonates at multiple frequencies, making the Sierpinski gasket antenna a multiband antenna. Vertical and horizontal polarization plots for the 4th frequency band are shown in Figure 17.



(a) (b)

Figure 17: Fourth frequency band: (a) Vertical polarization and (b) Horizontal polarization. (Image source: https://www.emcos.com/wp-content/uploads/2014/01/Application_Note_Fractal_Antennas_Simulation_Sierpinski_Gasket.pdf, accessed on 22 June 2022).

The simulations and plots in Figure 16 and Figure 17 are drawn using the EMCoS Antenna VLab environment, which is a software for electromagnetics, data visualization and simulation.

Simulations for other type of fractal antennas can be completed in a similar way using any EM simulation software (e.g., HFSS, CST Studio, EMCoS, COMSOL, etc.), and the details are available in many references cited throughout this section; therefore, we shall omit simulation details for other antennas to keep the presentation short.

Koch Curve

Figure 18 shows the first four iterations in the construction of the Koch curve monopole antenna, which became the first small size fractal antenna that improved bandwidth, resonance frequency, and radiation patterns of classical antennas in 1998. The von Koch curve is obtained by subdividing a line segment into three parts.

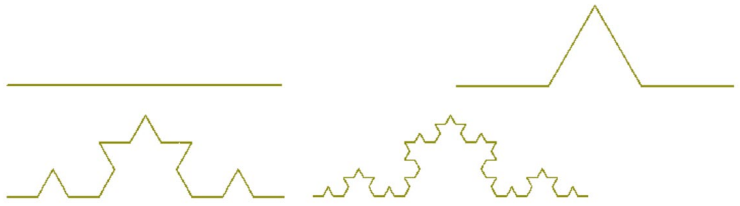


Figure 18: Four stages of Koch fractal.

The middle part is then replaced by adding two sides of an equilateral triangle having the length equal to the length of the segment being removed. This results in four line segments. Repeating this process for each of the four segments and taking the limit constitutes the Koch curve.

Puente et al. [19] studied the von Koch fractal as a monopole wire antenna. They considered five different iterations of the von Koch antenna, having an overall height h=6 cm, and a total length of $L = h \times \left(\frac{4}{3}\right)^5 = 25.3 \text{ cm}$ (see [19] for complete analysis and simulation reults). In general, the length of the Koch curve can be determined by formula $L_n = \left(\frac{4}{3}\right)^n$ (L_n is the length of the Koch curve at the *n*th iteration). Since $\frac{4}{3} > 1$, therefore, as $n \to \infty$, the length of the Koch curve will tend to infinity. So, theoretically, we can design an antenna of desired length in a given area using the Koch curve.

Koch Snowflake

Another popular fractal-shaped antenna is the Koch snowflake. To construct a Koch snowflake, start with a filled equilateral triangle and construct a von Koch curve on each side of the triangle to obtain the geometry (iteratively), as shown in Figure 19, where the first three stages in the construction of a Koch snowflake are shown.

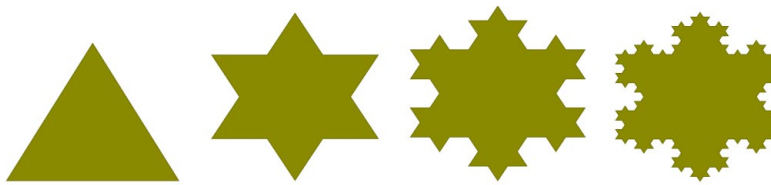


Figure 19: Four stages of the Snowflake fractal.

Minkowski Island Fractal Antenna

The construction of a Minkowski island fractal antenna is shown in Figure 20. Start with a filled square (called initiator). Then, replace each of the four sides of the initiator with the generator (shown at the bottom of Figure 20) and replace the four sides of the square with the generator and keep iterating. The result of this process is the Minkowski island fractal with intricate fundamental structure, which is nowhere differentiable.

Figure 20: Four stages of the Minkowski fractal.

The Koch snowflake and Minkowski island fractal antennas have been extensively used to create new designs for miniaturized loops as well as microstrip patch antennas.

Hilbert Curve Antenna

The Hilbert fractal antenna is another type of wire antenna made from a space-filling curve and falls into the broad category of space-filling fractal antennas. The first four iterations in the construction of the Hilbert curve are shown in Figure 21. The Hilbert curve has properties such as self-avoidance (no intersection points), self-similarity, space filling and simplicity.

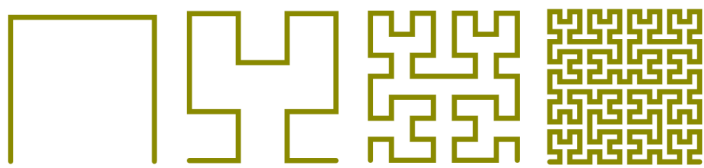


Figure 21: Four stages of the Hilbert fractal.

The space-filling properties of the Hilbert curve and related curves (e.g., Peano curves) make them suitable candidates for the design of fractal antennas.

Sierpinski Carpet or Fractal Pifa

An inverted-F antenna is another type of antenna first proposed by Ronold King at Harvard in 1958 for use in wireless communications. King's antenna was also a wire antenna and was designed for military use. It consists of a monopole antenna running parallel to a ground plane and grounded at one end.

Today, many cell phones comes with a Planar Inverted-F Antenna (in short PIFA), which are small, low profile, and sensitive to both horizontal and vertical polarized radio waves (see Baliarda et al. [19]), but the drawback is that PIFA are narrowband.

To overcome this difficulty, the fractal-shaped PIFA shown in Figure 22a has been designed, and the results are promising. A Fractal PIFA works similar to a traditional PIFA except that its design is a fractal based on a 2D Cantor array. A perfect fractal PIFA would be obtained by iterating the Cantor array an infinite number of times, but for practical design, two to three iterations are enough. A fractal PIFA mounted on a candy bar phone is also shown in Figure 22b, and a double PIFA is presented in Figure 22c.

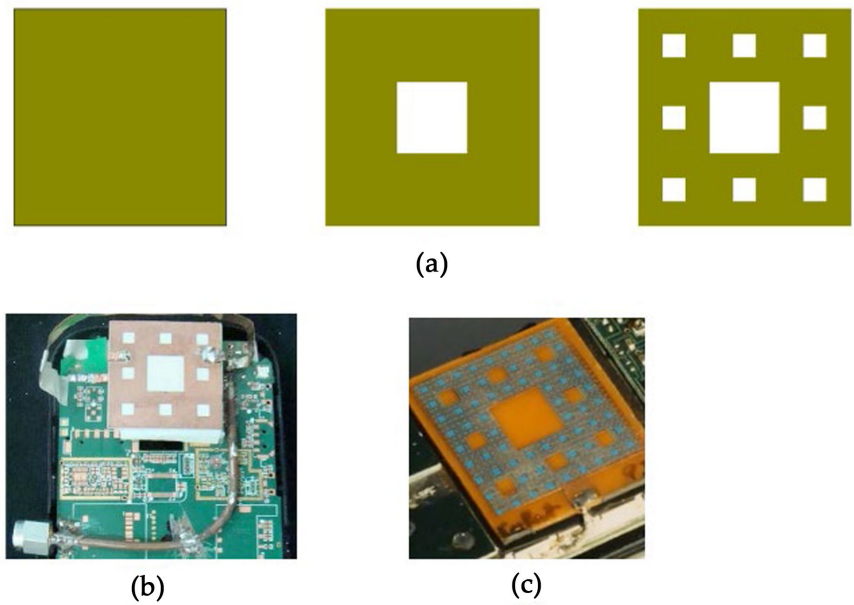


Figure 22: PIFA Antennas, (a) Three stages of Cantor fractal PIFA, (b) F-PIFA mounted on the candy bar phone, (c) Double-PIFA antenna.

Fractal Tree Antenna

Fractal trees antenna are used to fabricate miniaturized dipole antennas, and a number of new design of fractal tree antennas have evolved. An example of a ternary (three-branch) fractal tree is shown in Figure 23b, which looks like an analogue of the Sierpinski gasket of Figure 14. In fact, the ternary fractal tree shown in Figure 23a can be interpreted as a wire equivalent model of the Stage 4 Sierpinski gasket of Figure 14.

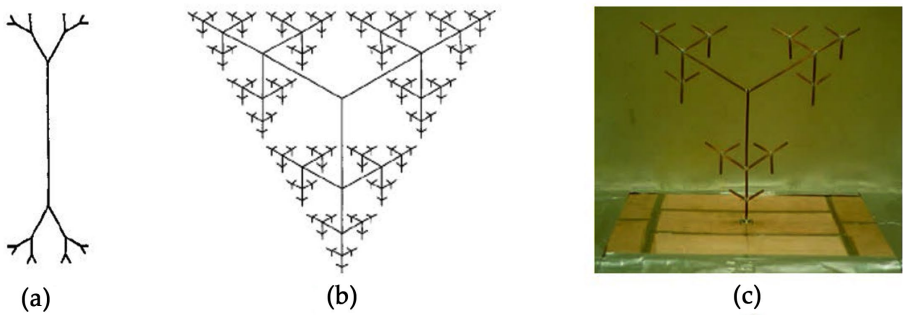


Figure 23: Fractal Tree Antennas: (a) Fractal tree, (b) A Stage 4 ternary fractal tree (Image source: Werner and Ganguly [15]), and (c) A prototype Tri-band fractal ternary tree monopole antenna used in miniaturized dipole antennas (Image source: http://cearl.ee.psu.edu/projects/project2-1-1.html, accessed on 22 June 2022).

We refer to the early papers by Werner [26] and Petko and Werner [27] for new designs and a variety of 2D and 3D multiband fractal tree antennas based on Koch curve and fractal trees, which are also reconfigurable (i.e., tunable) and exploit the self-similar branching structure of 3D fractal trees.

Other Innovative Fractal Antenna Designs

A multiband Cantor fractal monopole antenna covering GSM, DCS, PCS, UMTS, and WLAN applications was presented in [28].

A complementary stacked patch antenna based on Sierpinski fractal was introduced in [29], which enhanced antenna performances, retaining the basic characteristics of the Sierpinski antenna. A design procedure for custom made fractal antennas using artificial neural networks and the particle swarm optimization (PSO) was presented in [30]. A compact

multiband E-shape fractal patch antenna was proposed in [31] multiband applications to achieve size reduction and increase the operating bands. This antenna operates on LTE/WWAN (GSM850/900/1800/1900/UMTS/LTE2300/2500) bands.

At present, many fractals are being used as antennas, and several patents are also registered on new discoveries. Some of the fractal antennas used in mobile phones are shown below in Figure 24. A microstrip patch antenna with edges in the shape of a Minkowski island fractal is shown in Figure 24a, which is used in iphones. The Sierpinski fractal carpet shown in Figure 24b was designed by the Spanish company FRACTUS as a built-in antenna for a GSM 900/1800 mobile handset.

Figure 24: Some commercial antennas used in mobile phones and other applications, (a) Microstrip patch antenna, (b) Sierpinski triangle antenna.

Table 2 gives a summary of the literature for some standard fractal antennas and their modifications where antenna size(s), band utility, gain and applications are shown. It is clear from the table that the focus of designs is on multibandness with higher gain and effective bandwidth utility. Notice that reducing the dimensions of the designed antenna helps in miniaturization.

Table 2: Summary of the performance of some fractal antennas

Antenna Type (Ref. No.)	Dimension (mm^2)	No. of Bands	Bands (GHz)	Gain (dB)	Applications
Modified Sierpinski Gasket [32]	27 × 29	1	3.16–9	9.00	WLAN, WiMAX, public safety band, point to point high-speed applications for high data rates
Modified Sierpinski Gasket [33]	30 × 34.64	2	12.2–13.4 21–30	21.20 8.00	Broadband satellite receivers, mobile space research activities, active sensors, passive sensors
Modified Sierpinski Carpet [34]	29.44 × 38.04	6	4.285 5.455 6.265 6.805 8.02 9.145	–	Radio telecommunication in *C*-band, space communications in *X*-band and satellite communication
Modified Sierpinski Carpet [35]	30 × 30	6	2.23 4.75 5.23 6.61 6.79 9.58	15.27 (max)	*S* (2–4 GHz) band, *C* (4–8 GHz) band Weather radar and satellite applications, etc.
Koch Snowflake [36]	28.8 mm (diameter)	1	3.34–4.52 2.2–3.4 1.45–4.1	3.30 (max)	Wideband applications
Koch Snowflake [37]	60 mm (length of equal sides) 70 mm (base)	5	11.44 13.178 15.482 19.902 23.529	–	*X*-band, *Ku*-band and *K*-band
Minkowski Fractal [38]	27.5 × 25	1	1.575	0.369	Satellite Receiver
Hilbert Curve [39]	49 × 52	4	0.876 1.225 1.850 2.400	–	WSN Europe GPS-L1 GSM1800 Wi-Fi
Hilbert Curve [40]	56 × 39.4	2	12.5–37.5 0.4–1.4	3.35	HF/UHF dual band operation
Koch Curve Fractal Defected Ground Structure [41]	1994.02	1	1.492–1.518	5.41	*L*-band
Dual-Reverse-Arrow Fractal [42]	46.4 mm (side length of triangle)	1	2.4	2.5	ISM Applications
Sierpinski Carpet and Minkowski Hybrid [43]	40 × 40	2	3.5 5.8	4.50	WiMAX LTE
Hetero Triangle Linked Hybrid Web Fractal [44]	12 mm (diameter)	1	1.945–20	7.17 (max)	3G, LTE, ISM, Bluetooth, Wi-Fi, WLAN, WiMAX, Satellites (Ku-Band), etc.

Fractal Metamaterials

Metamaterials are synthetic electromagentic materials having properties not found in standard conducting materials. These artificial composites inherit their properties from internal micro and nanostructures rather than the chemical composition as compared with natural materials.

Figure 25 shows the first manufactured fractal metamaterial invented by fractenna.com (which also holds a patent on this discovery). Fractal metamaterials can achieve wideband and multiband performance in the fields

of cloaking, shielding, absorption and conveyance, whereas conventional metamaterial technology is limited to narrow passbands. This wideband/multiband performance is the key to employ fractal metamaterials in commercial and government applications. The field of fractal metamaterials is in the developmental stages, and their applications are still emerging.

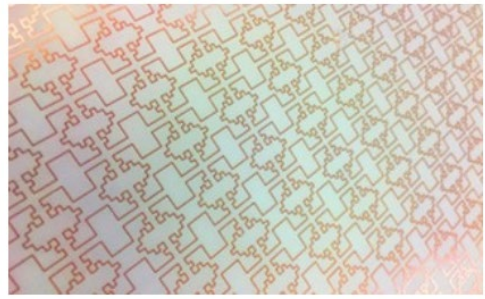


Figure 25: The first manufactured fractal metamaterial. (Image source: https://www.fractenna.com/, accessed on 22 June 2022).

Commercialization of Fractal Antennas

(1) www.fractenna.com, accessed on 22 June 2022

Dr. Cohen co-founded Fractal Antenna Systems, Inc. in the year 1995 to deliver the world's first fractal-based commercial antenna solutions (see Figure 26). Over the last 25 years, the company has deepened the theory of fractal antennas and deployed fractal antennas in a vast range of commercial and government applications. The company is also working on the capabilities and benefits of fractal metamaterials.

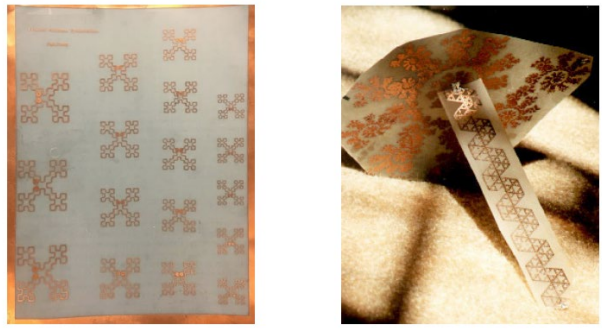


Figure 26. The first manufactured fractal antenna sheets (1995). Image source: https://www.fractenna.com/, accessed on 22 June 2022.

(2) www.fractus.com, accessed on 22 June 2022

Fractus is an early pioneer in the design and development of fractal antennas for smartphones, tablets, wireless and IOT devices. It was founded by fractal antenna pioneers Dr. Carles Puente and Ruben Bonet in 1999 and is leading the world market for its research, innovations and commercialization of multiband and miniature fractal antennas. The company holds the recognition of the world's first application for a patent on fractal and MultiFractal antennas.

(3) Fractus Antennas S.L. (www.fractusantennas.com), accessed on 22 June 2022

Founded in 2015, Fractus Antennas SL is actively involved in designing, manufacturing and commercializing miniature chip antennas for smartphones, short-range wireless and connected IoT devices. The company has received many patents for novel antenna designs. The recently developed Virtual Antenna™ Technology (2019) by Fractus SL is so unique that each antenna can be used for any application such as GSM (2G, 3G, 4G, 5G), GPRS, GPS, Bluetooth, WI-FI, RFID, NB-IOT, NBLTE and many more.

SUMMARY

Fractal antennas are a replacement for traditional wideband/multiband antennas that are smaller and lighter, require less circuitry, have fewer radiative elements to resonate at multiple frequencies and provide higher gains. Antennas with fractal shapes have many possible applications ranging from dual-mode phones to location services such as GPS, satellites, etc. Fractal-shaped antennas can lower the radar cross-section (RCS), which can be exploited in military applications where the RCS is an extremely important design parameter.

In the future, fractal antennas will play a much bigger role in the developing technologies for wireless communications which require compact, wideband and multiband antennas. Examples include wireless devices such as cell phones, tablets, wearable devices, smart homes, smart cities, airplanes, and IoT devices. The design of a high-performance wideband antenna is critical to IoT and wireless connectivity, and the fractal antenna engineering is enabling the changes that are required.

FRACTALS IN IMAGE COMPRESSION

The need for mass information storage and retrieval is growing rapidly with the advancement of the data and information age. On a computer, images are stored as a collection of bits representing pixels. Storing a single image or a collection of images on a computer may require large memory. This problem can be addressed using various image compression techniques. Storing images in less memory leads to a direct reduction in cost. This is where image compression plays an important role. Another useful feature of image compression is rapid data transfer, since less data need less time to transfer.

The Discrete Cosine Transform Algorithm is one of the most popular image compression methods, which is used in JPEG (still images), MPEG (motion video images), H.26x digital audio (such as Dolby Digital, MP3, AAC), and television (SDTV, HDTV) compression algorithms.

Fractal image compression is a fractal-based compression technique that makes use of the self-similarity present in an image for fractal coding. It is simple to implement, easy to execute and yields high compression ratios and quick decompression. Fractal image compression (FIC) was introduced by M. Barnsley, who started a company based on FIC technology. However, it was Arnaud Jacquin (a doctoral student of Barnsley) who published a fractal image compression algorithm for the first time.

History of Fractal Image Compression

After Mandelbrot's pioneering work [2], John Hutchinson introduced the iterated function theory in 1981 as an answer to the search of an underlying mathematical framework for fractal geometry. Later, M. Barnsley, another leading researcher in developing a mathematical framework for fractal geometry, wrote the famous book *Fractals Everywhere*. In this book, Barnsley described Iterated Functions Systems (IFS) and a very useful result known as the *Collage Theorem*, which became a fundamental result for fractal image compression. For example, the Pythagorean tree in the Figure 27 can be generated using the two-dimensional IFS

$$f_1\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{bmatrix}1/2 & -1/2\\1/2 & 1/2\end{bmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix}, \tag{4}$$

$$f_2\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{bmatrix}1/2 & 1/2\\-1/2 & 1/2\end{bmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix} + \begin{pmatrix}1/2\\3/2\end{pmatrix}, \tag{5}$$

$$f_3\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \tag{6}$$

Iterated function systems produce attractors (fractals), which are fixed points of a contraction mapping defined using the IFS, and the collage theorem does the reverse; i.e., for a given initial image, find an IFS whose attractor is as close as possible to the given image.

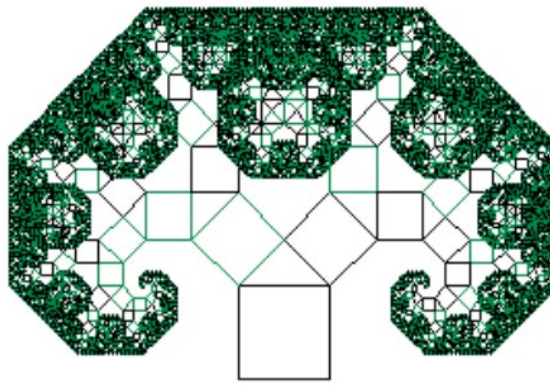


Figure 27: Pythagorean tree constructed using IFS in Equations (4)–(6).

Michael Barnsley suggested that storing images (for instance, the fractal tree shown in Figure 27) as a set of transformations given in Equations (4)–(6) may lead to image compression. IFS is a set of transformations from which the image of an attractor can be obtained. Barnsley did it in reverse by generating an IFS of the image which maps onto itself by making use of the collage theorem [6]. This leads to the compression of images. Barnsley observed many affine redundancies in real-life images and noticed that memory can be saved if we store suitable IFS. He was granted a patent and co-founded Iterated Systems Incorporation along with Alan Sloan. Barnsley published his results in the January 1988 issue of the BYTE magazine. This article exhibit several images compressed in excess of 10,000:1. The images were named as “Black Forest”, “Monterey Coast” and “Bolivian Girl”, but they were all manually constructed. Barnsley’s patent is referred to as the “*graduate student algorithm*.”

In March 1988, Arnaud Jacquin found a modified scheme for representing images called *Partitioned Iterated Function Systems* (PIFS) that made the graduate student algorithm obsolete. In 1991, Barnsley gave

another algorithm that can automatically convert an image into a PIFS, compressing the image in the process, and he received another patent for this. All contemporary fractal image compression algorithms are based on Jacquin's algorithm, and attempts to improve it have continued to date.

Mathematics of Images

Mathematically, an image is expressed as a function z=f(x,y), where *z* is the grayscale. We define the distance between two images f(x,y) and g(x,y) by the metric

$$d_{\max}(f,g) = \max_{(x,y)\in P} |f(x,y) - g(x,y)|, \tag{7}$$

where *f* and *g* are values of the level of gray pixel (for grayscale image), *P* is the space of images, and *x*, *y* are the coordinates of any pixel. It is clear from (**7**) that the d_{max} metric searches for the point (x,y) at which the two images *f* and *g* differ the most and assigns this as the distance between *f* and *g*. Another useful metric used in image compression is the root mean square (rms) metric (more useful for practical calculations) defined by

$$d_{\mathrm{rms}}(f,g) = \sqrt{\int_P (f(x,y) - g(x,y))^2 dxdy}. \tag{8}$$

Grayscale images are representations of subsets of the plane. An image is represented as a collection of pixels, and an image containing m·n pixels can be regarded as a vector in r=m×n dimensional space. Typically, the space is $\mathbb{R}^2$, and the usual norm on $\mathbb{R}^2$ is the 2-norm (also called the Euclidean norm or the L_2 norm), which is defined by

$$||x||_2 = \sqrt{|x_1|^2 + |x_2|^2}, \tag{9}$$

which induces the rms metric,

$$d_{\mathrm{rms}}(x,y) = ||x - y||_2.$$

Thus, if $x=(x_1,\ldots,x_r)$ and $y=(y_1,\cdots,y_r)$ are images, then the L_2 norm or rms distance (gap) between them is given by

$$d_{\mathrm{rms}}(x,y) = ||x - y||_2 = \sqrt{\sum_{i=1}^{r} (x_i - y_i)^2}. \tag{10}$$

Fidelity (a measure of the correctness of the reconstructed image) of an image is computed using the root mean square error (erms), the signal to noise ratio (SNR) and the peak signal to noise ratio (PSNR) of the image.

Let I(x,y) and A(x,y), respectively, denote the gray levels on the original and the reconstructed image (attractor), respectively; then,

$$e_{\text{rms}} = \sqrt{\sum_{x=1}^{m}\sum_{y=1}^{n}(e(x,y))^2},\ e(x,y) = (I(x,y) - A(x,y)), \tag{11}$$

$$SNR = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n}(A(x,y))^2}{\sum_{x=1}^{m}\sum_{y=1}^{n}(e(x,y))^2},\quad PSNR_{\text{rms}} = 20\log_{10}\left(\frac{2^p - 1}{e_{\text{rms}}}\right), \tag{12}$$

where p is the number of bits per pixel used for definition of the gray level.

Self-Similarity in Target Images

In general, a typical image does not show exact self-similarity, which is seen in mathematical fractals. However, it still contains a type of self-similarity in the sense that the entire image may not be self-similar, but parts of the image are self-similar with properly transformed parts of itself. For example, Figure 28 shows some parts of the Lena image that are self-similar at different scales.

Figure 28: Self-similarity in the Lena image.

A portion of the reflection of the hat in the mirror is similar to a smaller part of her hat, and a part of her shoulder overlaps a smaller region that is almost identical. Studies [2,45,46] suggest that most of the natural images contain this kind of self-similarity. The search for the resemblance (self-similarity) is the base of fractal compression algorithms.

Classical Approach

Imagine a Multiple Reduction Copying Machine (MRCM) shown in Figure 29. A MRCM (with multiple lens arrangements) is just like a regular copying machine except that it will scale the original image (to be copied) by half and print it three times on the copy.

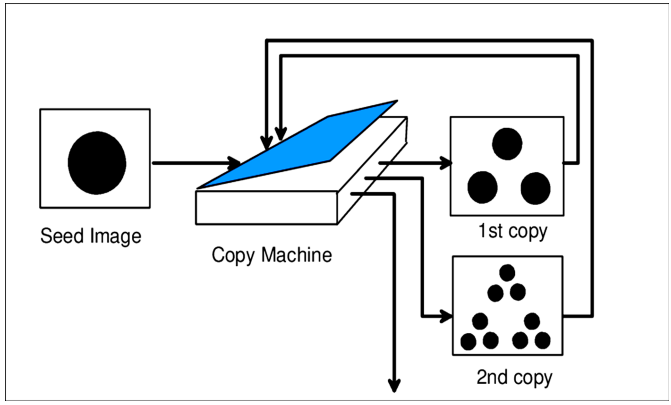


Figure 29: A Multiple Reduction Copying Machine (MRCM) with sample outputs. Reprinted with permission from [47]. Copyright 1997 Springer.

Figure 30 shows a few iterations of feeding an input (a Mandelbrot image) to the machine, and on repeated back feeding the output as input, the final image (attractor) is the Sierpinksi triangle.

Figure 30: The first 4 copies of an input image generated by the MRCM of Figure 29.

Clearly, any initial image will shrink to a point on repeated iterations due to size reduction in every iteration on the photocopying machine. Therefore, the shape of the final image (attractor) is determined by the position and the orientation of the image and not by its initial size.

In fractal image compression, to encode an image f, we need to find the transformations $w_1, w_2, \ldots, w_n$ such that f is the attractor of the map $W = \bigcup_{i=1}^{n} w_i$. Thus, we partition the image into pieces, find the transformations w_i, and acquire the original image f again by applying the transformations w_i.

The final output from the photocopying machine is determined by the way in which an input image is transformed by the transformations w_i when running the machine in a feedback loop. Theretofore, the transformations must be contractive; that is, each of these transformations must bring any two points of the input image closer in the output. In practice, it is sufficient to choose affine transformations of the form

$$w_i\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}, \quad i = 1, 2, \cdots, n. \tag{13}$$

Each transformation can rotate, scale (shrink) and translate an input image. Each w_i is a contraction mapping as long as the determinant of the transformation is strictly less than one, and the IFS will converge to the attractor A starting with any image A_0. Indeed, we have

$$A = \lim_{n\to\infty} W^n(A_0), \quad \text{with} \quad W(A) = \bigcup_{i=1}^{N} w_i(A). \tag{14}$$

In Figure 30, the final image obtained on repeated application of the transformation W possesses geometric self-similarity, and that is why IFSs are always expected to generate fractal images.

Contemporary Approach

The basic idea of partitioned iterated function system (PIFS) is as follows: if finding self-similarity between an image as a whole and its parts is impractical, then finding self-similarity between larger and smaller parts of the image is more reasonable. Using Jacquin's approach, this can be done by partitioning the original image at different scales into larger parts called *domain blocks* and small parts called *range blocks*. The idea of the PIFS is illustrated in Figure 31, where some mappings from domain blocks to

range blocks are shown. The range blocks are disjoint and partition the image uniformly. The domain blocks may overlap and need not contain every pixel of the original image. The goal of the compression process is to find a closely matching pre-image (i.e., a domain block for every range block). The size of the *domain pool* (determined by the number of domain blocks) is important for the encoding purpose. In general, a larger domain pool implies better fidelity of the mappings between the domain blocks and the range blocks. However, this also leads to more comparisons, which slow down the encoding. A scheme for classifying the domain and range blocks can be found in [46,48].

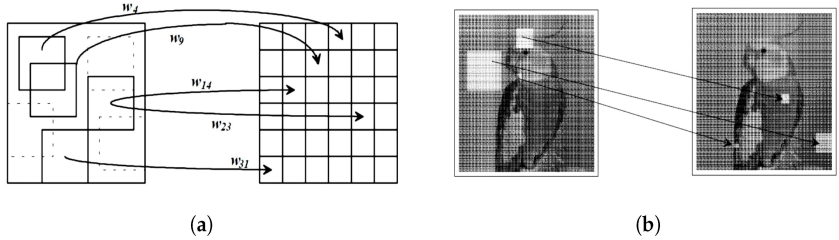


Figure 31: Self-similarity in Partitioned Iterated Function System, (a) Domain (left) and Range (right) blocks, (b) Domain–range pair self-similarity at three scales. Reprinted with permission from [47]. Copyright 1997 Springer.

Partitioned Iterated Function System

Jacquin extended the definition of an IFS to Partitioned Iterated Function Systems (PIFS) [48] in an attempt to ease the IFS computations. Theoretically, each image has a unique fixed point, but it is not feasible to find a unique fixed point for a whole image in practice. Thus, as an alternative, the image should be partitioned into several parts, and the fixed points for each part should be obtained through different transformations. We will use only affine transformations to illustrate a PIFS for simplicity, although the PIFS is independent of the type of transformations used. There are two spatial dimensions x and y, and the gray level adds a third dimension to the IFS so that the modified affine transformation w_i for PIFS becomes

$$w_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ o_i \end{pmatrix}. \tag{15}$$

To achieve convergence, the intensity value of a pixel must be scaled and offset, i.e.,

$$z' = s_i z + o_i. \tag{16}$$

Here, x and y are the spatial locations of a pixel, while z is the gray-level intensity of the pixel at location (x,y). Coefficients a_i,b_i,c_i,d_i,e_i and f_i control skewing, stretching, rotation, scaling, and translation, while the coefficients s_i and o_i determine the contrast and brightness of the transformation, respectively, which allow the affine transformation to map grayscale domain blocks to grayscale range blocks accurately (see Figure 31b for three examples). To speed up the compression and bring it under control, Jacquin constrained Equation (15) so that the domain blocks are always squares and equal to two times the size of range blocks. For instance, if the range blocks are (say) 8×8 pixels in size, then the domain blocks are chosen to be of the size of 16×16 pixels, which reduces the number of domain blocks to a large extent, and the search time is reduced during compression.

Thus, the image can be represented as a union of maps $w_1,w_2,\ldots,w_N$, such that $w_i:D_i\rightarrow\hat{R}_i$. That is, the application of w_i to a region of the image D_i produces $\hat{R}_i$, which is a result that approximates another region of the image, R_i. Minimizing the error between R^i and R_i will minimize the error between the original image and the approximation. In practice, the RMS metric is used to find the "best" transform to map D_i to R_i.

The Encoding

To encode a given image f, our aim is to find transformations $w_1,w_2,\ldots,w_n$ such that f is the fixed point of the map W. In other words, we decompose f into parts, apply the transformations w_i, and recover the original image f.

Fractal coding can produce a high compression ratio, which makes it one of the main advantages in compressing images. In Jacquin's algorithm, the aim is to minimize the Hausdorff distance (i.e., greatest pixel-to-pixel difference) between a candidate domain block and a specific range block.

The optimal scaling parameters can be computed algebraically if the root mean square error measure is used. To see this, assume that the domain block D_{xy} has been reduced to the size of the range block R_{xy}. Then, the mean square error between the blocks is

$$e_{\text{rms}} = \frac{1}{n^2}\sum_{x=1}^{n}\sum_{y=1}^{n}\left(s_i D_{xy} - R_{xy}\right)^2. \tag{17}$$

setting the derivative equal to zero

$$\frac{\partial e_{\text{rms}}}{\partial s_i} = \frac{2}{n^2} \sum_{x=1}^{n} \sum_{y=1}^{n} \left(s_i D_{xy} - R_{xy}\right) D_{xy} = 0, \tag{18}$$

we obtain

$$s_i = \frac{\sum_{x=1}^{n} \sum_{y=1}^{n} R_{xy} D_{xy}}{\sum_{x=1}^{n} \sum_{y=1}^{n} (D_{xy})^2}. \tag{19}$$

Figure 32 displays the flowchart of the encoding process.

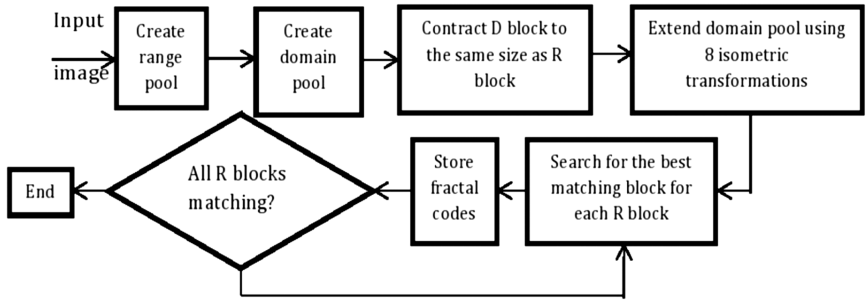


Figure 32: Encoding process.

Consider, for example, an image of size 128×128 pixels such that each pixel is of 256 gray levels. The image is partitioned into 8×8 blocks of non-overlapping range blocks and 16×16 overlapping domains blocks. For each range block R_i, a search is done through the entire set of domain blocks D to find the domain block which matches best with R_i. The position of the range, the best matching domain block, and transformation w_i, which minimizes the distance between domain and range blocks, are stored. This process is repeated until we have found the best matching block for the domain–range pair. This method of partition is a fixed range size partition method.

Table 3 shows the results of this process on the compression and reconstruction of 13 images using the classical approach [45,49].

Table 3: Performance of the Barnsley's algorithm [45] on various images

Image Name	$\frac{\text{Time}}{\text{Time Average}}$	e_{rms}	SNR_{rms}	$PSNR$ (dB)
Lena	1.000200294	7.61672	13.3478	30.4954
Peppers	1.000266348	7.5512	11.9571	30.5705
Mandril	1.000238648	13.168	1 8.919	25.7403
LAX	1.00021521	17.4734	4.9517	23.2832
Cameraman	1.000095885	14.0104	8.1885	25.2018
Columbia	1.000159809	16.3936	5.3475	23.8373
Goldhill	1.000138501	6.79771	13.4355	31.4836
Couple	1.000006392	13.6817	8.3402	25.408
Plane	0.999230786	13.2835	7.5457	25.6646
Women	1.000091624	11.0847	10.2303	27.2363
Milk	1.000093755	9.6735	8.4824	28.4191
Man	1.000025569	11.7124	9.3384	26.7579
Lake	0.999232916	15.8357	2.9538	24.138
Average	1.000000	12.1756	8.69527	26.7874

All images are of size 128×128 pixels (=16384 pixels) and 256 gray level. The range blocks are 4×4 pixels, and the domain blocks are 8×8 pixels. Therefore, the number of blocks to be encoded is $\left(\frac{128}{4}\right)^2 = 1024$. For the purpose of comparing image quality on the reconstruction of these 13 pictures, we refer to [49].

Decompression Process

The decoding process involves repeatedly applying the transform until it converges to an image, which closely approximates the original image. The decompression starts by setting the image buffer to a uniform mid-gray value, which is used as the seed image, and the pixels of each range block in the transform list are evaluated during the iteration. The result of the first iteration is used as input for the second stage of iteration. Usually, the original image is recognizable in just two iterations, and typically, the decompression process will converge in four or five iterations (when 8-bit precision is used per pixel). The decompression process for two encoded grayscale images of a 'Bird' and a 'Cameraman' is shown in Figure 33.

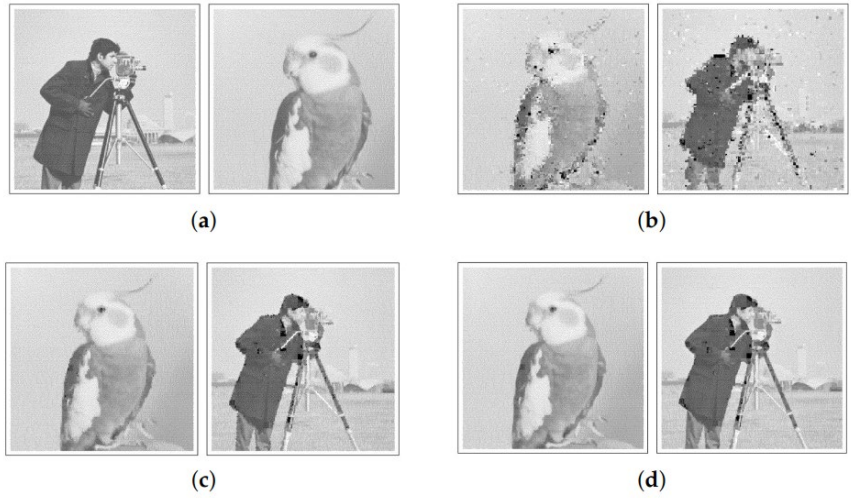


Figure 33: Decompression process for Bird and Cameraman (Reprinted with permission from [47]. Copyright 1997 Springer), (a) Seed for Bird (left) and seed for Cameraman (right), (b) 2 iterations of Bird IFS (left), 2 iterations of Cameraman IFS (right), (c) 4 iterations of Bird IFS (left), 4 iterations of Cameraman IFS (right), (d) 6 iterations of Bird IFS (left), 6 iterations of Cameraman IFS (right).

The choice of seed image has no impact on the outcome, since the IFS in Equation (14) describes the same attractor regardless of the starting image. This fact is well observed in Figure 33, where the Cameraman image is used as the seed image for Bird, and the Bird image is used as the seed image for Cameraman (see Figure 33a). One can notice the defects in Figure 33b which result from choosing a 'wrong' initial image that ultimately disappear with increasing iterations. The choice of seed can affect the decompression time, though, and it can be verified by starting with an all-black seed or an all-white seed image. However, for practical purposes, a mid-gray or a low-resolution version of the original image is preferred as the seed. See Figure 34 for a comparison of convergence using various seed images.

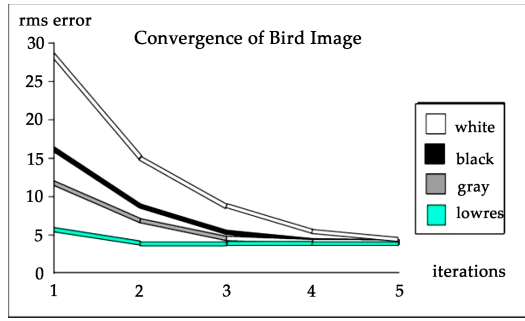
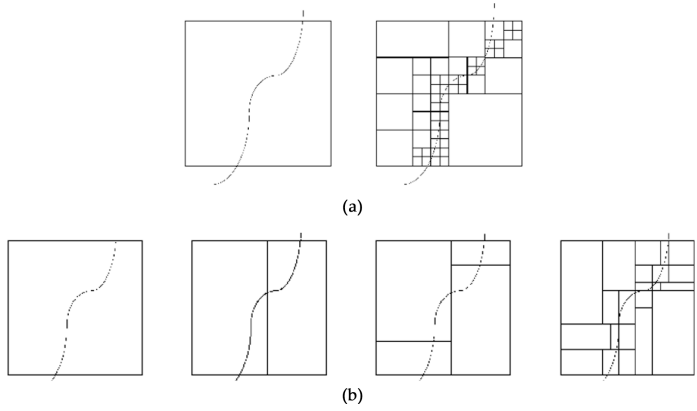


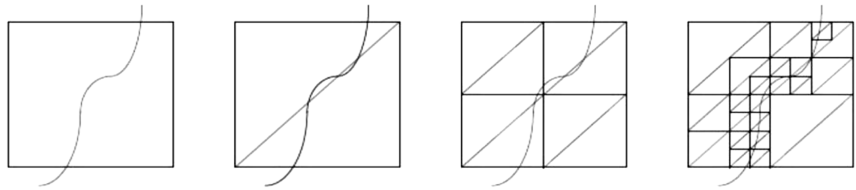
Figure 34: Convergence speed for various seed images. Reprinted with permission from [47]. Copyright 1997 Springer.

Partitioning Schemes

Partitioning of an input image is an important aspect of fractal image compression. Image partitioning refers to dividing the image into sections that are more appropriate for the application to work on.

In the classical approach of Jacquin [48], the image is partitioned into a fixed size square range blocks and domain blocks in which the size of domain blocks is twice the size of the range blocks. Several other flexible partitioning methods have evolved over the years, which allow for a higher compression ratio and shorter encoding times. Fisher [46] introduced the quadtree, HV Partitioning and Triangular partitioning schemes shown in Figure 35. We also refer to the review paper by Wohlberg and Jager [50] for the details on various partitioning schemes studied in the literature. Among all partitioning schemes, the quadtree partitioning is the most widely used technique.



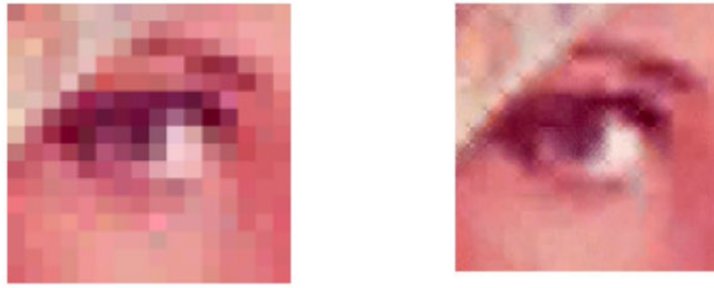


(c)

Figure 35: Some popular partitioning schemes, (a) Quadtree partitioning, (b) HV partitioning, (c) Triangular partitioning.

Summary of Fractal Image Compression

Fractal image compression is a promising, block-based, lossy and asymmetrical compression method. The images generated by fractal coding are *resolution/scale independent*, i.e., the image can be decoded at any resolution. Magnifying an image reveals additional detail, and after every iteration, details on the decoded image are sharper than before. This feature of fractal image compression is unique. Figure 36 shows magnification of the original image of Lena's eye (on the left). On the right is the same part of the fractal image rendered at the same scale. Sometimes, magnified fractal encoded images often look better than magnified original images due to reasonable interpolation.



(a) (b)

Figure 36: Resolution independence: (a) Original image enlarged 4 times, (b) Decoded image enlarged 4 times.

Another main advantage of FIC is that it is easy to automate. Decompression is quick, and fractal compression can achieve high compression ratios while maintaining image quality, and at higher compression, it is relatively superior to JPEG and wavelet compressions.

Fractal compression is also useful in multimedia applications. Fractal compression methods are probably best suitable for archival applications, such as digital encyclopedias, where encoding is done only once. The greatest challenge for the coding community is how to precisely measure and quantify signal-to-noise ratio, root mean square error, etc.

Fractal image compression is still under development. Many research groups worldwide are developing new algorithms to shorten the encoding time. We refer the reader to [45,46,48,49,51] for more detailed literature on the theoretical concepts, existing methods, algorithms and experimental results on fractal image compression.

FRACTALS IN FRACTURE MECHANICS

Fracture mechanics is the study of the propagation of cracks in materials, and it is an important tool to improve the performance of mechanical components. The phenomenon of fracture is to divide an object or material into two or different pieces on applying physical stress (see Figure 37 for different types of fracture modes). Thus, there exists a crack on the surface irregularly which penetrates into the body, too. All these physical appearances such as crack length, area, etc. cannot be described easily using Euclidean geometry. The fractal geometry equipped with self-similarity (or self-affinity), scale invariance and fractal dimension offers great help to analyze irregular or fractional shapes of fracture mechanics.

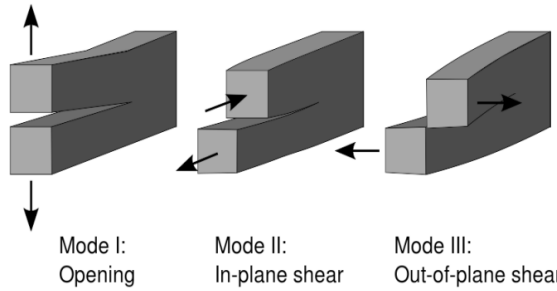


Figure 37: The three fracture modes. (Image source: https://en.wikipedia.org/wiki/Fracture_mechanics, accessed on 22 June 2022).

Mandelbrot was the first to interrelate the crack propagation and other fracture properties of materials with the fractal geometry [52]. He introduced a method called *slit island analysis* on the fracture surface to find fracture dimensions, which is shown to be a measure of toughness in metals.

Mandelbrot characterized the structure of a surface by the fractal dimension, D, as a scaling factor. As D increases from 0 to 1, the irregularities of the surface become more significant, and shape becomes predominantly less meaningful. He experimented through fractured steel specimen plated with electroless nickel and proposed the "slit island analysis" method to calculate the fractal dimension.

The quantitative analysis of fracture surfaces in brittle alumina and glass ceramic materials using fractal geometry was considered by Mecholsky et al. [53] by calculating the fractal dimension of crack surfaces using slit island analysis (SIA) and fracture profile analysis (FPA) methods. They proved that the fractal dimension increases with increase in fracture toughness, in general.

Fractal geometries are often characterized by a scaling (power) law:

$$Nr^D = 1. \tag{20}$$

where N is the number of segments, r is the similarity ratio (or reduction factor), and D is the fractal dimension.

Equation (20) describes how many new features will appear by a magnification factor r for a given fractal dimnesion. For example, if $r = \frac{1}{4}$ and D=1.5, then the number of features will be N=8. The number of features would increase to N≈11 at the same scale with D=1.75. Thus, the higher fractal dimension leads to more features or structures.

The toughness of a fracture surface is measured in terms of difficulty in the crack growth, and researchers have attempted to relate the fracture toughness and surface energy with the fractal dimension. In this connection, Mecholsky et al. [54] discovered the following formula relating fractal dimension with the fracture toughness

$$K_{IC} = E(a_0 D^*)^{\frac{1}{2}}. \tag{21}$$

Here, E is the modulus of elasticity of the material, a_0 is the lattice parameter, $D^*=D-d$ with d as the Euclidean dimension in the projection of fracture. Mu and Lung [55] proposed an alternate equation which is a power law relation connecting the fractal dimension with surface energy.

Zhang [56] studied the fracture of rocks under the effect of high temperature considering the fractal dimension as a crucial factor. Fractal dimension and the rockburst tendency index can predict the failure of the rocks, and variations in rockburst tendency laws were been obtained. The relation between fractal dimension and rockburst tendency can be explained by a quadratic expression

$$K_{\text{eff}} = A(d_f - \bar{d}_f)^2 + B.$$

Here, K_{eff} is the effective burst energy index, A and B are rock material constants, d_f is the fractal dimension of the fracture surface and $\bar{d}_f$ is the fractal dimension threshold, and there is a directly proportional correlativity between rockburst index and the fractal dimension when $d_f \leq \bar{d}_f$ and inverse proportionality correlation when $d_f > \bar{d}_f$. This is how mechanical properties such as energy dissipation energy release rates related with the fractal dimension of the fractured surface during the rock failure mechanism, and that will reflect in the degree of rockburst tendency.

After the pioneering work of Mandelbrot et al. [52], fractal geometry has been applied to the fractality of cracked surfaces, fracture mechanics and material science problems by several authors, and we refer to the papers [54,56,57,58] for further details, analysis and determination of the fractal dimension of microcrack structures and fracture surfaces.

OTHER FRACTAL APPLICATIONS AND INNOVATIONS

Fractals in ophthalmology: The human retina shown in Figure 38 exhibits fractal structure properties in its vascular network, so fractal geometry is the right tool for modeling such a complex structure [59]. The damage of the blood vessels of the retina in diabetic people is known as *diabetic retinopathy*.

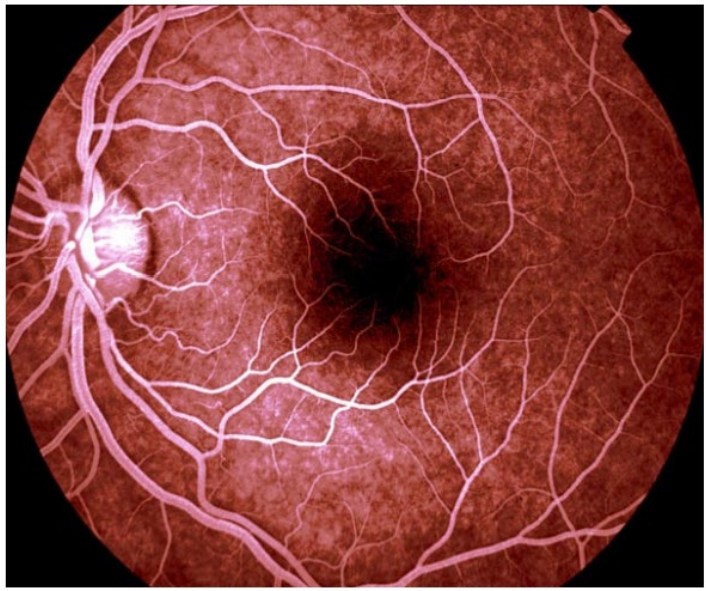


Figure 38: Human retina. Image by: Paul van der Meer. (Image source: https://fractalfoundation.org/OFC/OFC-1-3.html, accessed on 22 June 2022).

The examination of fundus of the eye is a classical old technique for screening diabetic retinopathy and takes more time. In recent times, the technique of taking digital photographs of the fundus is used, which are transmitted to a central database for testing. Fractal analysis is the best method in processing this data with more accurate results as compared to other methods where the fractal dimension is the prominent tool for analysis.

Fractals are also important in other life science studies and biological fields. They are now used to predict or analyze the growth patterns of bacteria, the pattern of nerve dendrites, pathology, study of cancer, wildlife and landscape ecology, etc. The expository article by G.A. Losa [60] is a rich source of information on the extension of fractal geometry for the life sciences to understand complex functional properties, morphological, and structural features characterizing cells and tissues. The reader may also refer to [61] and references therein for further study. In most of these studies, fractal dimension is a key tool for analysis.

***Fractal Capacitors*:** Wearable and implantable electronic devices are common nowadays and are expected to dominate the future soon. However, these devices suffer the problem of inadequate power supply limited by the size of these gadgets. Microsupercapacitors (MSCs) are emerging

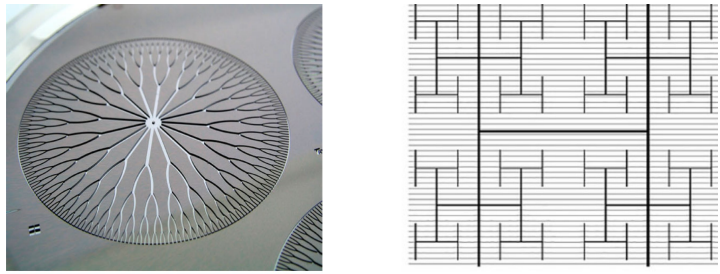
miniaturized high-power microelectrochemical energy-storage devices that can circumvent this difficulty, as they are capable of delivering high power density, fast charge and discharge, and a superior lifetime (millions of cycles). In a recent study, Hota et al. [62] fabricated integrated MSCs using three different fractal designs—namely, Hilbert, Peano, and Moore (they used anhydrous RuO2 thin-film electrodes as prototypes)—and proved that fractal-shaped electrode designs is a viable solution to improve the performance of MSCs. It is shown that among the three proposed designs, the Moore design shows the best performance. Many more MSCs may be fabricated by exploiting the self-similarity and scale invariance of fractals.

***Fractal Batteries*:** Fractal structures have proven to be advantageous in electrochemical energy conversion systems, since fractals maximize the electrochemically active surface area while minimizing the energy loss in the network. Fractals can be used in the "fractalization" of battery electrodes to increase power density and reduce dendrite formation. The fractalization technique can be applied to any electrode material (e.g., C, Si, MgX, etc.). In this connection, we refer to [63], wherein the theoretical analysis of fractal type electrodes for lithium-ion batteries is presented along with simulation results. More recently, Thekkekara and Gu [64] proposed bio-inspired fractal electrode designs for solar energy storage using space-filling properties of fractal curves from the Peano family.

***Fractal Electromagnets*:** The techniques of fractal geometries can be used to fabricate fractal electromagnets to increase the magnetic flux for a given size, or, alternatively, shrink the size for a given flux. This size reduction permits embedding electromagnets and solenoids in places where it was almost impossible until now.

***Fractal PCBs*:** Fractals are being applied on printed circuit boards (PCBs) to reduce corrosion possibilities by fabricating fractal-shaped PCBs. Fractal PCBs can be applied to any trace or joints of contact with a high-voltage differential to reduce the risk of corrosion. Less corrosion delivers high reliability in electrical components, resulting in reduced overall cost.

***Fractal in Cooling Devices*:** Fractal-shaped smart cooling devices such as cooling chips, PC coolers, fractal microchannel heat sink, etc. are now becoming popular, which are based on fractal geometry. A cooling circuit for a computer chip printed in the form of a fractal branching pattern is shown in Figure 39a. The liquid nitrogen passes across the surface through this device to keep the chip cool.



(**a**) (**b**)

Figure 39: (a) Computer chip cooling circuit, (b) A fractal solar panel [65].

***Fractal Solar Panels*:** A group of researchers from the University of Oregon [65] have recently proposed a new electrode design based on the H-tree fractal tree structure (see Figure 39b) for fractal-patterned rooftop solar panels that combines the aesthetic advantages of the technology with the efficiency of busbar design. These modern electrode patterns are expected to emerge into the mainstream electrodes that would be adopted for a wider range of applications, especially engineering and design.

***Fractals in Biometric Applications*:** Fingerprints are the simplest and most reliable biometric features that are widely used for identification purposes. Fingerprints exhibit self-similarity at multiple scales, and a fingerprint database can be classified using fractal dimension, but a fingerprint cannot be identified with fractal dimension uniquely. In [66], a novel Fingerprint Fractal Identification System (FFIS) was presented for identifying a fingerprint uniquely using fractal geometry and game theoretic techniques.

CONCLUSIONS

This article presents a comprehensive survey of fractals with focus on their applications in innovative and emerging research fields. With this extensive survey, we have tried to demonstrate the importance of fractals in engineering, industry and commercial applications by considering fractals in the design of fractal antennas, image processing, landscape generation, and fracture mechanics. Some future-ready applications of fractals are also discussed toward the end. In Part I of this survey of fractals [1], we considered the mathematics of fractals using iterated function systems, attractors, fractal dimensions, etc. and their appearance in fractal arts, ceramic products, fractal clothing and in fractal tilings.

Fractals have been studied in mathematics, computer science, engineering, physics, chemistry, biology, geology, social science, economics, technology, art, architecture and many other areas. Fractals have deep relevance in chaos theory because the graphs of most chaotic processes are fractals. The field of fractals has enormous potential to expand and take hold into many evolving areas of research, and even a voluminous book would be inadequate to discuss all of these in one place.

In summary, fractal geometry is the language of nature, and Benoît Mandelbrot has given us a new science which is applicable almost everywhere with an mind-opening effect on everyone who has come across it. This new language is changing our scientific world rapidly with sustainable solutions.

We close with a remark by Mandelbrot from the book *The Fractalist. Memoirs of a Scientific Maverick*, which is an inspirational collection of his own reflections and thoughts.

"*Within the sciences of nature, I was a pioneer in the study of familiar shapes, like mountains, coastlines, clouds, turbulent eddies, galaxy clusters, trees, the weather, and others beyond counting*".

Benoît B. Mandelbrot, (2010)

AUTHOR CONTRIBUTIONS

Conceptualization, A.H. and M.S.; methodology, A.H., M.S. and M.N.N.; software, M.N.N. and M.S.C.; validation, A.H., M.S. and M.N.N.; formal analysis, A.H.; investigation, M.N.N., M.S.C. and M.S.; resources, M.N.N. and M.S.C.; writing—original draft preparation, A.H., M.N.N. and M.S.C.; writing—review and editing, A.H., M.N.N., M.S.C. and M.S.; supervision, A.H. and M.S. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENTS

The authors are thankful to the referees for their careful reading of the manuscript and for giving valuable suggestions to improve it.

REFERENCES

1. Husain, A.; Nanda, M.N.; Chowdary, M.S.; Sajid, M. Fractals: An Eclectic Survey, Part-I. *Fractal Fract.* 2022, *6*, 89.
2. Mandelbrot, B.B. *The Fractal Geometry of Nature*; W. H. Freeman and Company: New York, NY, USA, 1982.
3. Mandelbrot, B.B. *Fractals: Form Chance and Dimension*; W. H. Freeman and Company: New York, NY, USA, 1977.
4. Schlenker, A. Generalized Mandelbrot Sets, Undergraduate Honors Thesis Collection, 229. 2014. Available online: https://digitalcommons.butler.edu/ugtheses/229 (accessed on 22 March 2022).
5. Wikipedia. Mandelbrot Set. Available online: https://en.wikipedia.org/wiki/Mandelbrot_set (accessed on 22 March 2022).
6. Barnsley, M.F. *Fractals Everywhere*, 2nd ed.; Academic Press: Cambridge, MA, USA; Elsevier: New York, NY, USA, 1993.
7. Hutchinson, J. Fractals and Self-Similarity. *Indiana Univ. Math. J.* 1981, *30*, 713–747.
8. Fournier, A.; Fussell, D.; Carpenter, L. Computer rendering of stochastic models. *Commun. ACM* 1982, *25*, 371–384.
9. Miller, G. The definition and rendering of terrain maps. *ACM Siggraph Comput. Graph.* 1986, *20*, 39–48.
10. Musgrave, F.K. Methods for Realistic Landscape Imaging. Ph.D. Thesis, Yale University, New Haven, CT, USA, 1993.
11. Gothall, R.; Eriksson, M.; Tille, H. A Modification of the Random Midpoint Displacement Method for Generating Rock Fracture Similar Surfaces. In Proceedings of the ICFXI–11th International Conferenceon Fracture, Turin, Italy, 20–25 March 2005.
12. Huang, S. Xiang-Xin Li Improved Random Midpoint-Displacement Method for Natural Terrain Simulation. In Proceedings of the Third International Conference on Information and Computing, Delhi, India, 29–30 July 2020; Volume 1, pp. 255–258.
13. Jilsen, J.; Kuo, J.; Lien, F.S. Three-dimensional midpoint displacement algorithm for the generation of fractal porous media. *Comput. Geosci.* 2012, *46*, 164–173.
14. Lewis, J.P. Generalized stochastic subdivision. *ACM Trans. Graph.* 1987, *6*, 167–190.

15. Werner, D.H.; Ganguly, S. An overview of fractal antenna engineering research. *IEEE Antennas Propag. Mag.* 2003, *45*, 38–57.
16. Cohen, N. Fractal antennas: Part 1. *Commun. Q. Summer* 1995, 7–22.
17. Cohen, N. Fractal antennas: Part 2. *Commun. Q. Summer* 1996, 53–66.
18. Puente, C.; Romeu, J.; Pous, R.; Ramis, J.; Hijazo, A. Small but long Koch fractal monopole. *IEEE Electron. Lett.* 1998, *34*, 9–10.
19. Puente, C.; Romeu, J.; Cardama, A. The Koch monopole: A small fractal antenna. *IEEE Trans. Antennas Propag.* 2000, *48*, 1773–1781.
20. Anguera, J.; Andújar, A.; Jayasinghe, J.; Chakravarthy, V.V.S.S.S.; Chowdary, P.S.R.; Pijoan, J.L.; Ali, T.; Cattani, C. Fractal antennas: An historical perspective. *Fractal Fract.* 2020, *4*, 3.
21. Karmakar, A. Fractal antennas and arrays: A review and recent developments. *Int. J. Microw. Wirel. Technol.* 2020, *13*, 173–197.
22. Krzysztofik, W.J. Modified Sierpinski fractal monopole for ISM-bands handset applications. *IEEE Trans. Antennas Propag.* 2009, *57*, 606–615.
23. Werner, D.H.; Haup, R.L.; Werner, P.L. Fractal Antenna Engineering: The Theory and Design of Fractal Antenna Arrays. *IEEE Antennas Propag. Mag.* 1999, *41*, 5.
24. Puente, C.; Romeu, J.; Pous, R.; Garcia, X.; Benítez, F. Fractal multiband antenna based on the Sierpinski gasket. *Electron. Lett.* 1996, *32*, 1–2.
25. Mishra, R.K.; Ghatak, R.; Poddar, D.R. Design formula for Sierpinski gasket pre-fractal planar-monopole antennas. *IEEE Antennas Propag. Mag.* 2008, *50*, 104–107.
26. Werner, D.H.; Bretones, A.R.; Long, B.R. Radiation Characteristics of Thin-Wire Ternary Fractal Trees. *Electron. Lett.* 1999, *35*, 609–610.
27. Petko, J.S.; Werner, D.H. Miniature Reconfigurable Three Dimensional Fractal Tree Antennas. *IEEE Trans. Antennas Propag.* 2004, *52*, 1945–1956.
28. Manimegalai, B.; Raju, S.; Abhaikumar, V. A multifractal cantor antenna for multiband wireless applications. *IEEE Antennas Wirel. Propag. Lett.* 2009, *8*, 359–362.
29. Ghatak, R.; Mishra, R.K.; Poddar, D.R. Stacked dual layer complementing Sierpinski gasket planar antenna. *Microw. Opt. Technol. Lett.* 2007, *49*, 2831–2833.

30. Patnaik, A.A.; Sinha, S.N. Design of custom-made fractal multi-band antennas using ANN-PSO. *IEEE Antennas Propag. Mag.* 2011, *53*, 94–101.
31. Bayatmaku, N.; Lotfi, P.; Azarmanesh, M.; Soltani, S. Design of simple multiband patch antenna for mobile communication applications using new E-shape fractal. *IEEE Antennas Wirel. Propag. Lett.* 2011, *10*, 873–875.
32. Devesh; Ansari, J.A.; Siddiqui, M.G.; Saroj, A.K. Analysis of modified square Sierpinski gasket fractal microstrip antenna for wireless communications. *Int. J. Electron. Commun.* 2018, *94*, 377–385.
33. Singh, A.; Singh, S. A modified coaxial probe-fed Sierpinski fractal wideband and high gain antenna. *Int. J. Electron. Commun.* 2015, *69*, 884–889.
34. Sivia, J.S.; Kaur, G.; Sarao, A.K. A modified Sierpinski carpet fractal antenna for multiband applications. *Wirel. Pers. Commun.* 2017, *95*, 4269–4279.
35. Raghavendra, C.; Saritha, V.; Alekhya, B. Design of modified sierpinski carpet fractal patch antenna for multiband applications. In Proceedings of the 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), Chennai, India, 21–22 September 2017; pp. 868–871.
36. Choukiker, Y.K.; Behera, S.K. Wideband frequency reconfigurable Koch snowflake fractal antenna. *IET Microw. Antennas Propag.* 2017, *11*, 203–208.
37. Siddiqui, M.G.; Saroj, A.K.; Devesh; Ansari, J.A. Multiband fractaled triangular microstrip antenna for wireless applications. *Prog. Electromagn. Res.* 2018, *65*, 51–60.
38. Joy, S.; Natarajamani, S.; Vaitheeswaran, S.M. Minkowski fractal circularly polarized planar antenna for GPS application, (ICACC-2018). *Procedia Comput. Sci.* 2018, *143*, 66–73.
39. Azaro, R.; Viani, F.; Lizzi, L.; Zeni, E.; Massa, A. A Monopolar quad-band antenna based on a Hilbert self-affine prefractal geometry. *IEEE Antenna Wirel. Propag. Lett.* 2016, *8*, 177–180.
40. Alibakhshi-Kenari, M.; Naser-Moghadasi, M.; Sadeghzadeh, R.A.; Virdee, B.S.; Limiti, E. Dual-band RFID tag antenna based on the Hilbert curve fractal for HF and UHF applications. *IET Circuits Devices Syst.* 2016, *10*, 140–146.

41. Prajapati, P.R.; Murthy, G.G.K.; Patnaik, A.; Kartikeyan, M.V. Design and testing of a compact circularly polarised microstrip antenna with fractal defected ground structure for L-band applications. *IET Microwaves Antennas Propag.* 2015, *9*, 1179–1185.
42. Orazi, H.; Soleimani, H. Miniaturisation of the triangular patch antenna by the novel dual-reverse-arrow fractal. *IET Microwaves Antennas Propag.* 2015, *9*, 627–633.
43. Varaminia, G.; Keshtkar, A.; Daryasafar, N.; Moghadasi, M.N. Microstrip Sierpinski fractal carpet for slot antenna with metamaterial loads for dual-band wireless application. *Int. J. Electron. Commun.* 2018, *84*, 93–99.
44. Deepak, B.S.; Madhav, B.T.P.; Prabhakar, V.S.V.; Lakshman, P.; Anilkumar, T. Venkateswara Rao, M. Design and analysis of hetero triangle linked hybrid web fractal antenna for wide band applications. *Prog. Electromagn. Res.* 2018, *83*, 147–159.
45. Barnsley, M.; Hurd, L.P. *Fractal Image Compression*; AK Peters Ltd.: Natick, MA, USA, 1993.
46. Fisher, Y. *Fractal Image Compression-Theory and Application*; Springer: Berlin/Heidelberg, Germany, 1995.
47. Kominek, J. Advances in fractal compression for multimedia applications. *Multimed. Syst.* 1997, *5*, 255–270.
48. Jacquin, A. Image Coding Based on a Fractal theory of Iterated Contractive Image Transformations. *IEEE Trans. Image Process.* 1992, *1*, 18–30.
49. Conci, A.; Aquino, F.R. Fractal coding based on image local fractal dimension. *Comput. Appl. Math.* 2005, *24*, 83–98.
50. Wohlberg, B.; de Jager, G. A review of the fractal image coding literature. *IEEE Trans. Image Process.* 1999, *8*, 1716–1729.
51. Wang, J.; Zheng, N. A novel fractal image compression scheme with block classification and sorting based on Pearson's correlation coefficient. *IEEE Trans. Image Process.* 2013, *22*, 3690–3702.
52. Mandelbrot, B.B.; Passoja, D.; Paullay, A.J. Fractal Character of Fracture Surfaces of Metals. *Nature* 1984, *308*, 721–722.
53. Mecholsky, J.J.; Passoja, D.E.; Feinberg-Ringel, K.S. Quantitative Analysis of Brittle Fracture Surfaces Using Fractal Geometry. *J. Am. Ceram. Soc.* 2005, *72*, 60–65.

54. Mecholsky, J.J.; Mackin, T.; Passoja, D.E. Self-Similar Crack Propagation In Brittle Materials. In *Advances in Ceramics, Fractography of Glasses and Ceramics*; Varner, J., Frechette, V.D., Eds.; The American Ceramic Society, Inc.: Westerville, OH, USA, 1988; Volume 22, pp. 127–134.
55. Mu, Z.Q.; Lung, C.W. Studies on the fractal dimension and fracture toughness of steel. *J. Phys. Appl. Phys.* 1988, *21*, 848.
56. Zhang, Z.Z. Fractal Dimension of Fracture Surface in Rock Material after High Temperature. *Adv. Mater. Sci. Eng.* 2015, *2015*, 468370.
57. Alves, L.M.; de Lacerda, L.A. Fractal Fracture Mechanics Applied to Materials Engineering. In *Applied Fracture Mechanics*; BoD—Books on Demand: Norderstedt, Germany, 2012; pp. 67–106.
58. Yavari, A.; Sarkani, S.; Moyer, E.T. The mechanics of self-similar and self-affine fractal cracks. *Int. J. Fract.* 2002, *114*, 1–27.
59. Uahabi, K.L.; Atounti, M. Applications of fractals in medicine, Annals of the University of Craiova. *Math. Comput. Sci. Ser.* 2015, *42*, 167–174.
60. Losa, G.A. The Fractal Geometry of Life. *Riv. Biol. Biol. Forum* 2009, *102*, 29–60.
61. Losa, G.A.; Castelli, C. Nuclear patterns of human breast cancer cells during apoptosis: Characterization by fractal dimension and (GLCM) co-occurrence matrix statistics. *Cell Tissue Res.* 2005, *322*, 257–267.
62. Hota, M.K.; Jiang, Q.; Mashraei, Y.; Salama, K.N.; Alshareef, H.N. Fractal Electrochemical Microsupercapacitors. *Adv. Electron. Mater.* 2017, *3*, 1700185.
63. Teixidor, G.T.; Park, B.Y.; Mukherjee, P.P.; Kang, Q.; Madou, M.J. Modeling fractal electrodes for Li-ion batteries. *Electrochim. Acta* 2009, *54*, 5928–5936.
64. Thekkekara, L.; Gu, M. Bioinspired fractal electrodes for solar energy storages. *Sci. Rep.* 2017, *7*, 45585.
65. Roe, E.T.; Bies, A.J.; Montgomery, R.D.; Watterson, W.J.; Parris, B.; Boydston, C.R.; Sereno, M.E.; Taylor, R.P. Fractal solar panels: Optimizing aesthetic and electrical performances. *PLoS ONE* 2020, *15*, e0229945.
66. Jampour, M.; Yaghoobi, M.; Ashourzadeh, M.; Soleimani, A. A New Fast Technique for Fingerprint Identification with Fractal and Chaos Game Theory. *Fractals* 2010, *18*, 293–300.

INDEX

S

T

Applied Mathematics in Engineering

"Applied Mathematics in Engineering" is an edited book comprising 15 contemporaneous open-access articles focused on mathematical modelling, mathematical optimization, and fractal mathematics and their applications in modern engineering and industry. The first part covers a wide range of mathematical models for solving real-world problems in mechanical engineering, biomedicine, fluid dynamics, and other applications. The second part of the book presents various mathematical optimization methods and algorithms (e.g. topology optimization, hybrid arithmetic optimization, and isogeometric topology optimization) for solving engineering problems. The third part of the book is dedicated to the mathematics of fractals and its applications in the engineering of antennas, image compression technology, fracture mechanics, ophthalmology and more.
The intended audience of this book is undergraduate and graduate students, as well as junior researchers.

Olga Moreira is a Ph.D. and M.Sc. in Astrophysics and B.Sc. in Physics/Applied Mathematics (Astronomy). She is an experienced technical writer and data analyst. As a graduate student, she held two research grants to carry out her work in Astrophysics at two of the most renowned European institutions in the fields of Astrophysics and Space Science (the European Space Agency, and the European Southern Observatory). She is currently an independent scientist, peer-reviewer and editor. Her research interest is solar physics, machine learning and artificial neural networks.

ARCLER PRESS

ISBN 978-1-77469-559-3