ICIAM2023 Springer Series 2

Xiangfeng Wang · Xingju Cai
Deren Han *Editors*

Splitting Optimization

Theory, Methodology, and Applications

10th International Congress on Industrial and Applied Mathematics
ICIAM 2023 TOKYO

Springer

ICIAM2023 Springer Series

Volume 2

This series aims to publish some of the most relevant results presented at the ICIAM 2023 held in Tokyo in August 2023.

The series is managed by an independent Editorial Board, and will include peer-reviewed content only, including the Invited Speakers volume as well as books resulting from mini-symposia and collateral workshops.

The series is aimed at providing useful reference material to academic and researchers at an international level.

Xiangfeng Wang · Xingju Cai · Deren Han
Editors

Splitting Optimization

Theory, Methodology, and Applications

Springer

Editors
Xiangfeng Wang
School of Computer Science
and Technology
East China Normal University
Shanghai, China

Xingju Cai
School of Mathematical Science
Nanjing Normal University
Nanjing, China

Deren Han
School of Mathematical Sciences
Beihang University
Beijing, China

ISSN 3091-3098 ISSN 3091-3101 (electronic)
ICIAM2023 Springer Series
ISBN 978-981-96-8573-8 ISBN 978-981-96-8574-5 (eBook)
https://doi.org/10.1007/978-981-96-8574-5

Mathematics Subject Classification: 65K05, 65K15, 49M27, 90-08

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

We are delighted to present *Splitting Optimization: Theory, Methodology, and Applications*, a comprehensive volume dedicated to the forefront of research in splitting methods within the field of optimization. As optimization problems become increasingly complex and high-dimensional across various scientific and engineering disciplines, the need for efficient and scalable algorithms is more critical than ever. Splitting optimization has emerged as a powerful paradigm, offering robust theoretical frameworks and practical algorithms for decomposing and solving large-scale optimization problems.

This book brings together a collection of pioneering research papers authored by leading experts and scholars. Each contribution explores innovative aspects of splitting optimization, encompassing theoretical advancements, methodological developments, and a diverse array of applications. Our aim is to provide readers with a deep and cohesive understanding of the current state of splitting optimization, as well as to inspire future research and innovations in this dynamic field.

Splitting methods have gained significant attention due to their ability to handle complex optimization problems by breaking them down into simpler subproblems. This decomposition not only facilitates parallel and distributed computing but also allows for the exploitation of problem structures, leading to more efficient algorithms. The versatility of splitting methods makes them applicable to a wide range of problems, including those in machine learning, signal processing, image analysis, and game theory.

“A Trust-Region-Based Splitting Method for Optimization Problems with Linear Constraints” with authors Leyu Hu, Yannan Chen, Xingju Cai, and Deren Han is presented in Chap. 1. This chapter addresses a fundamental challenge in augmented Lagrangian-based splitting methods: the selection of penalty parameters. Fixed parameters can lead to slow convergence or inadequate progress in either the primal or dual variables. The authors introduce a novel trust-region-based approach that adaptively adjusts the trust-region radius during iterations. This method smartly balances the trade-off between primal and dual advancements, ensuring robust global convergence under mild conditions. An $O(1/\epsilon)$ convergence rate is achieved in an ergodic sense. Numerical experiments on medical image recovery and logistic regression problems demonstrate the efficiency and potential of the proposed method, highlighting its applicability in real-world scenarios.

The following chapter contains “Forward-Reflected-Backward Method with Extrapolation and Linesearch for Monotone Inclusion Problems” with authors Tanxing Wang, Heng Zhang, and Xingju Cai. Monotone inclusion problems are central to various applications in optimization and variational analysis. The authors present an enhanced forward-reflected-backward algorithm that incorporates a new extrapolation direction and a novel line-search procedure using locally Lipschitz constants. Unlike existing methods that rely on global Lipschitz constants, this approach allows for larger step sizes, improving convergence speed and computational efficiency. Weak convergence is established under standard assumptions. Extensive numerical experiments on the lasso problem and ℓ_1-regularized logistic regression illustrate the method’s superiority over classical algorithms, making it a valuable tool for solving large-scale optimization problems.

“Fast Adaptive ADMM with Gaussian Back Substitution for Multiple Block Linear Constrained Separable Problems” with the author Xiangfeng Wang is presented in the following chapter. The Alternating Direction Method of Multipliers (ADMM) is a popular algorithm for solving separable optimization problems with linear constraints. This chapter introduces the Fast Adaptive ADMM with Gaussian Back Substitution (ADMM-G-V), an innovative framework designed for multiple-block settings. By integrating an adaptive penalty parameter that dynamically adjusts during the iterative process and utilizing Gaussian back substitution, the proposed method enhances both convergence properties and computational efficiency. Theoretical analyses establish global convergence and optimal convergence rates in both ergodic and non-ergodic senses. Numerical experiments on consensus problems over networked agents and distributed logistic regression tasks showcase the algorithm’s effectiveness, underscoring its potential in distributed optimization and large-scale machine learning applications.

Chapter 4 contains Inertial “Inertial Alternating Direction Method of Multipliers with Logarithmic-Quadratic Proximal Regularization” with the author Zhongming Wu. Acceleration techniques are crucial for improving the performance of iterative optimization algorithms. This chapter explores the inertial proximal point method applied to the ADMM and symmetric ADMM with logarithmic-quadratic proximal (LQP) regularization. By incorporating inertial terms and appropriate step sizes, the proposed methods accelerate convergence while leveraging the separable structure of the problem. The utilization of LQP regularization transforms constrained subproblems into more manageable unconstrained ones during iterations. Under mild conditions, global convergence is rigorously established, enriching the theoretical understanding of inertial methods in the context of splitting optimization.

“A Class of Augmented-Lagrangian-Type Algorithms for Solving Generalized Nash Equilibrium Problems” with authors Xiaoxi Jia, Shiwei Wang, and Lingling Xu is presented in the following chapter. Generalized Nash Equilibrium Problems (GNEPs) with shared constraints present significant analytical and computational challenges. The authors propose a class of regularized augmented Lagrangian methods that penalize shared linear constraints within each player's augmented Lagrangian function, transforming the original GNEP into a convex Nash Equilibrium subproblem (NEP). Under strong monotonicity and Lipschitz continuity assumptions of the pseudo-gradient, Fejér monotonicity of iterative points is proved with respect to the solution set. By adding a correction step, the authors relax the cocoercivity requirement of the pseudo-gradient, broadening the applicability of their method. Numerical examples validate the effectiveness of the proposed algorithms, demonstrating their potential in economic modeling and resource allocation problems.

The chapters in this volume collectively advance the field of splitting optimization through:—Adaptive Strategies: Introducing adaptive mechanisms for parameter selection, such as trust-region adjustments and adaptive penalty parameters, to enhance convergence and stability without relying on fixed or global constants.—Acceleration Techniques: Employing inertial terms, extrapolation directions, and line-search procedures to accelerate the convergence of iterative methods, making them more practical for large-scale and real-time applications.—Theoretical Advancements: Providing rigorous convergence analyses under less restrictive assumptions, thereby strengthening the theoretical foundations and broadening the applicability of splitting methods.—Diverse Applications: Showcasing the versatility of splitting optimization through practical applications in medical imaging, logistic regression, consensus networks, and game theory, highlighting the impact of these methods across different domains.

Splitting Optimization: Theory, Methodology, and Applications represents a significant milestone in the ongoing evolution of optimization research. As the demand for efficient and scalable algorithms continues to grow, splitting methods will undoubtedly play an increasingly vital role. By presenting these innovative contributions, we hope to foster a deeper understanding of splitting optimization and stimulate further advancements in both theoretical and practical realms. We believe that the insights and methodologies presented in this book will not only enrich the knowledge of readers but also inspire new ideas and collaborations. The challenges addressed and solutions proposed here reflect the dynamic nature of optimization, and we are confident that this volume will serve as a valuable foundation for future exploration and innovation.

We extend our sincere gratitude to all the contributing authors for their outstanding work and dedication. Their commitment to excellence has made this volume possible. We also thank the reviewers for their insightful comments and suggestions, which have greatly enhanced the quality of the chapters. Finally, we express our appreciation to the broader optimization community for their ongoing support and engagement.

Shanghai, China Xiangfeng Wang
Nanjing, China Xingju Cai
Beijing, China Deren Han

Contents

Chapter 1
A Trust-Region-Based Splitting Method for Optimization Problems with Linear Constraints

Leyu Hu, Yannan Chen, Xingju Cai, and Deren Han

Abstract The augmented Lagrangian-based splitting methods have found more and more applications in scientific and engineering computation, such as compressive sensing, covariance selection, image processing, and transportation research. One of the basic difficulties in such algorithms is the selection of the parameter in the augmented Lagrangian function; a larger one may make the primal progress too small, while a smaller one may slow down the dual progress. To overcome this difficulty, in this paper, we propose to solve the splitting subproblems in a trust region manner, and the radius can be adjusted smartly. Under the same mild conditions as those for classical augmented Lagrangian-based splitting methods, we prove the global convergence of the proposed algorithm. Moreover, the $\mathcal{O}(1/\epsilon)$ convergence rate is also analyzed in an ergodic sense. We present some preliminary numerical experiments on medical image recovery and logistic regression, which show that the trust region-based splitting method is efficient and promising.

Keywords Trust region · Splitting method · Separable convex optimization · Convergence rate

L. Hu
LMIB, School of Mathematical Sciences, Beihang University, Beijing, China
e-mail: huleyu@buaa.edu.cn

Y. Chen
School of Mathematical Sciences, South China Normal University, Guangzhou, China
e-mail: ynchen@scnu.edu.cn

X. Cai
School of Mathematical Sciences, Ministry of Education Key Laboratory of NSLSCS, Nanjing Normal University, Nanjing, China
e-mail: caixingju@njnu.edu.cn

D. Han (✉)
LMIB, School of Mathematical Sciences, Beihang University, Beijing, China
e-mail: handr@buaa.edu.cn

X. Wang et al. (eds.), *Splitting Optimization*, ICIAM2023 Springer Series 2,
https://doi.org/10.1007/978-981-96-8574-5_1

1.1 Introduction

We consider the following separable convex optimization problem with linear constraints:

$$\begin{cases} \min & f(x) + g(y) \\ \text{s.t.} & Ax + By = c, \\ & x \in \mathbb{R}^n, \ y \in \mathbb{R}^m, \end{cases} \tag{1.1}$$

where $A \in \mathbb{R}^{\ell \times n}$ and $B \in \mathbb{R}^{\ell \times m}$ are two fixed matrices, the vector $c \in \mathbb{R}^\ell$, and f and g are convex and smooth functions with Lipschitz continuous Hessian matrices. Hence, the solution set of (1.1) is convex, and we assume it is nonempty.

Numerous applications from science and engineering fall into the well-structured model (1.1). For example, the sparse solution recovery problem in compressive sensing (basis pursuit) [4, 25]; the matrix completion problem with or without noise [3]; the nonnegative tensor factorization [2]; the constrained total-variation image restoration and reconstruction problems [18]; the estimation of the higher-order generalized diffusion tensor in nuclear magnetic resonance imaging of medical engineering [5]; the ℓ_1-norm penalized log-likelihood covariance selection models [28]; the route based traffic assignment problems in transportation [14]; etc. Boyd et al. [1] show that several problems from statistical and machine learning can be modeled naturally as the separable convex optimization problem (1.1).

One of the efficient numerical algorithms for solving (1.1) is the classical augmented Lagrangian function method [19, 21]. At iteration k with an estimator of multiplier $\lambda^k \in \mathbb{R}^\ell$, it minimizes variables x and y simultaneously to get the next iterate

$$(x^{k+1}, y^{k+1}) := \text{argmin}_{x,y}\ \mathcal{L}_H(x, y, \lambda^k), \tag{1.2}$$

and then update the multiplier

$$\lambda^{k+1} := \lambda^k - H(Ax^{k+1} + By^{k+1} - c). \tag{1.3}$$

Here, the augmented Lagrangian function is defined as

$$\mathcal{L}_H(x, y, \lambda) := f(x) + g(y) - \lambda^\top(Ax + By - c) + \frac{1}{2}\|Ax + By - c\|_H^2$$
$$\text{for } (x, y, \lambda) \in \mathbb{R}^{n \times m \times \ell}, \tag{1.4}$$

and $H \in \mathbb{R}^{\ell \times \ell}$ is a penalized symmetric positive definite matrix. In many classical literatures, the matrix H is a scalar matrix $H = \beta I$, where β is a positive scalar and I is the identity matrix.

To exploit the inherent separable structure, the alternating direction method of multiplier (ADMM) [10, 12] minimizes the variables x and y separately. Given current iterates y^k and λ^k, it generates the next iterate via the following three steps successively:

$$\begin{cases} x^{k+1} := \operatorname{argmin}_{x\in\mathbb{R}^n} \mathcal{L}_H(x, y^k, \lambda^k), \\ y^{k+1} := \operatorname{argmin}_{y\in\mathbb{R}^m} \mathcal{L}_H(x^{k+1}, y, \lambda^k), \\ \lambda^{k+1} := \lambda^k - H(Ax^{k+1} + By^{k+1} - c). \end{cases} \tag{1.5}$$

That is to say, the large optimization problem (1.2) is divided into two small optimization problems in (1.5). The global convergence of ADMM is extensively studied by many researchers including Gabay [9], Tseng [22], and Eckstein and Bertsekas [7]. Recently, He and Yuan [17] show that ADMM admits the $O(1/\epsilon)$ convergence rate in an ergodic sense.

Both in (1.2) and (1.5), the matrix H plays an important role by penalizing the violation of the linear constraint. Although theoretically the algorithms converge for any positive definite matrix H, the numerical performance varies significantly for different choices. Generally, a larger one may lead to larger progress for the dual variable (it can be viewed as the stepsize for the dual variable λ), while it may cause smaller progress for the primal variables, and vice versa. Hence, how to choose a reasonable one is a difficult task. He et al. [15] propose to add proximal point terms to the subproblems,

$$\begin{cases} x^{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^n} \mathcal{L}_H(x, y^k, \lambda^k) + \dfrac{1}{2}\|x - x^k\|^2_{R_k}, \\ y^{k+1} = \operatorname{argmin}_{y\in\mathbb{R}^m} \mathcal{L}_H(x^{k+1}, y, \lambda^k) + \dfrac{1}{2}\|y - y^k\|^2_{S_k}, \\ \lambda^{k+1} = \lambda^k - H(Ax^{k+1} + By^{k+1} - c), \end{cases} \tag{1.6}$$

where $\{R_k\}$ and $\{S_k\}$ are symmetric positive definite matrices. So the resulting subproblems in (1.6) are strongly convex. Alternatively, Xu [24] and Yuan [27] prefer to use fixed matrices R and S in the proximal point terms. It is worthwhile to note that a special choice of matrices R and S reduces to the split Bregman method [13], which is powerful for problems arising from compressive sensing and image denoising.

In this paper, we propose to solve the subproblems in a trust region framework, since the trust region method [6] is robust and efficient in nonlinear programming. Different from the soft regularization, trust region methods impose an explicit trust region constraint to force the new iterate being in a neighborhood of the current iterate. Formally, we represent the trust region-based splitting method as follows:

$$\begin{cases} x^{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^n} \left\{\widetilde{\mathcal{L}}_H(x, y^k, \lambda^k | x^k, y^k, \lambda^k) \mid \|x - x^k\| \le \Delta_k\right\}, \\ y^{k+1} = \operatorname{argmin}_{y\in\mathbb{R}^m} \left\{\widetilde{\mathcal{L}}_H(x^{k+1}, y, \lambda^k | x^{k+1}, y^k, \lambda^k) \mid \|y - y^k\| \le \Gamma_k\right\}, \\ \lambda^{k+1} = \lambda^k - H(Ax^{k+1} + By^{k+1} - c), \end{cases}$$

where Δ_k and Γ_k are trust region radii for x-subproblem and y-subproblem, respectively, and $\widetilde{\mathcal{L}}_H$ is the quadratic approximation of the augmented Lagrangian function, which is defined as

$$\begin{aligned}
&\widetilde{\mathcal{L}}_H(x, y, \lambda | x^k, y^k, \lambda^k) \qquad (1.7)\\
&= f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top \widetilde{\nabla}^2 f(x^k)(x - x^k)\\
&\quad + g(y^k) + \nabla g(y^k)^\top (y - y^k) + \frac{1}{2}(y - y^k)^\top \widetilde{\nabla}^2 g(y^k)(y - y^k)\\
&\quad - \lambda^k(Ax + By - c) + \frac{1}{2}\|Ax + By - c\|_H^2,
\end{aligned}$$

where $\widetilde{\nabla}^2 f(x^k)$ and $\widetilde{\nabla}^2 g(y^k)$ are the Hessian-like matrices of f and g at x^k and y^k respectively obtained by certain mechanism such as BFGS. Theoretically, finding a solution to an optimization problem in a trust region is equivalent to solve a quadratic proximal regularized optimization problem with a suitable regularization parameter. However, for the regularization-based method, the selection of the regularization parameter in each iteration is a challenging problem. It is noteworthy that in a trust region framework, we do not need to directly choose the regularization parameter. This is because the concept of the trust region is geometric and more intuitive. According to nonlinear programming [6, 19, 21], dozens of algorithms and codes have been established to solve the trust region subproblem fast and accurately.

The proposed trust region-based splitting method is built in a prediction and correction framework. In the prediction step, we solve the x-subproblem and y-subproblem in an explicit trust region constraint and construct an efficient descent direction. In the correction step, a suitable step length is customized for the descent direction to accelerate the convergence of the proposed algorithm. Combining these two steps, we show that the sequence of iterates generated by the trust region-based splitting method is globally convergent to the optimal solution set. Moreover, an ϵ-approximate optimal solution of the equivalent variational inequality is obtained within the $\mathcal{O}(1/\epsilon)$ number of iterations in the worst case. Numerical experiments on the constrained TV-ℓ_2 image deblurring and denoising problem and the ℓ_1 regularized logistic regression problem are performed. The results indicate that the novel trust region-based splitting method is efficient and promising. Significantly, in the image processing example, the involved trust region subproblem is solved efficiently and accurately when the classical fast Fourier transformation is explored.

The outline of this paper is as follows. In Sect. 1.2, we give some useful notations and the variational characterization of the separable convex optimization with linear constraints. The prediction and correction steps of the trust region-based splitting method are described in Sect. 1.3. The global convergence of the proposed algorithm is given in Sect. 1.4.1. And in Sect. 1.4.2, the $\mathcal{O}(1/\epsilon)$ convergence rate is analyzed in an ergodic sense. Numerical validation on some real problems is reported in Sect. 1.5. Finally, we complete the paper by drawing some conclusions in Sect. 1.6.

1.2 Variational Characterization

In this section, we give some notations and preliminary results that are useful in the rest of this paper. Let $\lambda \in \mathbb{R}^{\ell}$ be a multiplier corresponding to the equality constraint $Ax + By - c = 0$ in (1.1). We define

$$u := \begin{pmatrix} x \\ y \end{pmatrix}, w := \begin{pmatrix} u \\ \lambda \end{pmatrix}, \qquad \text{then } w \in \Omega := \mathbb{R}^{n \times m \times \ell}.$$

The presentation of the new splitting method is in the framework of variational inequalities. Now, we give the variational characterization of the separable convex optimization (1.1).

The Lagrangian function of the original optimization problem (1.1) is

$$L(x, y, \lambda) := f(x) + g(y) - \lambda^{\top}(Ax + By - c), \tag{1.8}$$

where $(x, y, \lambda) \in \Omega$. Suppose $(x^*, y^*, \lambda^*) \in \Omega$ is a primal-dual solution of the convex optimization (1.1). Then, it must be a saddle point of the Lagrangian function, i.e.,

$$L(x^*, y^*, \lambda) \le L(x^*, y^*, \lambda^*) \le L(x, y, \lambda^*).$$

From the viewpoint of variational inequalities, $(x^*, y^*, \lambda^*) \in \Omega$ is a saddle point of the Lagrangian function (1.8) if and only if it satisfies

$$\begin{cases} (x - x^*)^{\top}(\nabla f(x) - A^{\top}\lambda^*) \ge 0, \\ (y - y^*)^{\top}(\nabla g(y) - B^{\top}\lambda^*) \ge 0, \\ (\lambda - \lambda^*)^{\top}(Ax^* + By^* - c) \ge 0, \end{cases} \qquad \forall\, (x, y, \lambda) \in \Omega. \tag{1.9}$$

For convenience, we define a mapping F:

$$F(w) := \begin{pmatrix} \nabla f(x) - A^{\top}\lambda \\ \nabla g(y) - B^{\top}\lambda \\ Ax + By - c \end{pmatrix} = \begin{pmatrix} 0 & 0 & -A^{\top} \\ 0 & 0 & -B^{\top} \\ A & B & 0 \end{pmatrix} w + \begin{pmatrix} \nabla f(x) \\ \nabla g(y) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ c \end{pmatrix}. \tag{1.10}$$

Obviously, the linear mapping F is a monotone operator since the coefficient matrix is skew-symmetric and the convexity of f and g. Then, the variational inequality (VI) (1.9) has a compact form: find $w^* \in \Omega$ such that

$$(w - w^*)^{\top} F(w^*) \ge 0, \qquad \forall\, w \in \Omega. \tag{1.11}$$

Since we suppose the solution set of the convex optimization (1.1) is nonempty, the solution set Ω^* of the VI (1.11), which is convex, is also nonempty. We remark that the VI (1.11) plays a critical role in the following analysis.

1.3 The Trust-Region-Based Splitting Method

The trust region method is a powerful and robust method in nonlinear programming [6, 19, 21]. At each iteration, the trust region method finds the solution of a subproblem in a special neighborhood of the current iterate, which is called a trust region. In the process of iteration, the size of the trust region could be enlarged or contracted according to some rules.

Our trust region-based splitting method for solving the separable optimization problem (1.1) contains two steps: prediction and correction. We now describe the prediction step in detail.

1.3.1 The Prediction Step

Algorithm 1 (*Prediction Step*) In iteration k, (x^k, y^k, λ^k) is given. we perform the following three steps successively.

$$\begin{aligned} \widetilde{x}^k &:= \operatorname{argmin}_{x\in\mathbb{R}^n} \widetilde{\mathcal{L}}_H(x, y^k, \lambda^k | x^k, y^k, \lambda^k) \\ &\quad \text{s.t. } \|x - x^k\| \le \Delta_k, \end{aligned} \tag{1.12}$$

$$\begin{aligned} \widetilde{y}^k &:= \operatorname{argmin}_{y\in\mathbb{R}^m} \widetilde{\mathcal{L}}_H(\widetilde{x}^k, y, \lambda^k | \widetilde{x}^k, y^k, \lambda^k) \\ &\quad \text{s.t. } \|y - y^k\| \le \Gamma_k, \end{aligned} \tag{1.13}$$

and

$$\widetilde{\lambda}^k := \lambda^k - H(A\widetilde{x}^k + B\widetilde{y}^k - c). \tag{1.14}$$

The quadratic approximation of the augmented Lagrangian function $\widetilde{\mathcal{L}}_H$ is defined in (1.7). The optimal condition for the trust region subproblem (1.12) in the variable x. For the convenience of the following discussion, we rewrite (1.12) as follows, by removing the irrelevant terms and rearranging the terms:

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} \nabla f(x^k)^\top (x - x^k) + \dfrac{1}{2}(x - x^k)^\top \widetilde{\nabla}^2 f(x^k)(x - x^k) \\ \qquad + \dfrac{1}{2}\|Ax + By^k - c - H^{-1}\lambda^k\|_H^2 \\ \text{s.t. } \|x - x^k\| \le \Delta_k. \end{cases}$$

According to nonlinear programming, the first-order necessary condition for the trust region subproblem (1.12′) is

$$0 \in \nabla f(x^k) + \widetilde{\nabla}^2 f(x^k)(x - x^k) + A^\top H(Ax + By^k - c - H^{-1}\lambda^k) + \mathcal{N}_{\odot(x^k,\Delta_k)}(x)$$

where $\mathcal{N}_{\odot(x^k,\Delta_k)}(x)$ is the normal cone at x of the trust region constraint $\|x - x^k\| \leq \Delta_k$ at x^k was

$$\mathcal{N}_{\odot(x^k,\Delta_k)}(x) := \left\{ v \in \mathbb{R}^n \middle| v = \begin{cases} \delta(x - x^k), & \|x - x^k\| = \Delta_k, \\ 0, & \|x - x^k\| < \Delta_k, \end{cases}, \delta \geq 0 \right\}.$$

Then we have the KKT conditions of (1.12′) as, for $\|\widetilde{x}^k - x^k\| \leq \Delta_k$, $\delta_k \geq 0$,

$$\begin{cases} \nabla f(x^k) + \widetilde{\nabla}^2 f(x^k)(\widetilde{x}^k - x^k) + A^\top H(A\widetilde{x}^k + By^k - c - H^{-1}\lambda^k) + \delta_k(\widetilde{x}^k - x^k) = 0, \\ \nabla g(y^k) + \widetilde{\nabla}^2 g(y^k)(\widetilde{y}^k - y^k) + B^\top H(A\widetilde{x}^k + B\widetilde{y}^k - c - H^{-1}\lambda^k) + \gamma_k(\widetilde{y}^k - y^k) = 0. \end{cases} \tag{1.15}$$

where δ_k here is the optimal Lagrange multiplier for the trust region constraint. Through this, we could get the optimal δ_k when we solve the trust region subproblem (1.12):

$$\delta_k = \frac{\|\nabla f(x^k) + \widetilde{\nabla}^2 f(x^k)(\widetilde{x}^k - x^k) + A^\top H(A\widetilde{x}^k + By^k - c - H^{-1}\lambda^k)\|}{\|\widetilde{x}^k - x^k\|},$$

when $\widetilde{x}^k \neq x^k$. And we define $\delta_k = 0$ when $\widetilde{x}^k = x^k$. For convenience, we denote

$$\begin{aligned} \nabla_x^k \widetilde{\mathcal{L}} &:= \nabla f(x^k) + A^\top H(Ax^k + By^k - c - H^{-1}\lambda^k), \\ \nabla_y^k \widetilde{\mathcal{L}} &:= \nabla g(y^k) + B^T H(A\widetilde{x}^k + By^k - c - H^{-1}\lambda^k). \end{aligned} \tag{1.16}$$

Then we have

$$\delta_k = \frac{\left\|\nabla_x^k \widetilde{\mathcal{L}} + \left(\widetilde{\nabla}^2 f(x^k) + A^\top HA\right)(\widetilde{x}^k - x^k)\right\|}{\|\widetilde{x}^k - x^k\|}, \tag{1.17}$$

and same for the trust region subproblem (1.13) in the variable y as

$$\gamma_k = \frac{\left\|\nabla_y^k \widetilde{\mathcal{L}} + \left(\widetilde{\nabla}^2 g(y^k) + B^\top HB\right)(\widetilde{y}^k - y^k)\right\|}{\|\widetilde{y}^k - y^k\|}. \tag{1.18}$$

Similarly to (1.15), we could get the KKT conditions of the trust region subproblem (1.13) as, for $\|y - y^k\| \leq \Gamma_k$, $\gamma_k \geq 0$.

Combining (1.15) and substituting (1.14), we get the following equations:

$$\begin{cases} \nabla f(x^k) + \widetilde{\nabla}^2 f(x^k)(\widetilde{x}^k - x^k) - A^\top \widetilde{\lambda}^k + \delta_k(\widetilde{x}^k - x^k) - HA^\top B(\widetilde{y}^k - y^k) = 0, \\ \nabla g(y^k) + \widetilde{\nabla}^2 g(y^k)(\widetilde{y}^k - y^k) - B^\top \widetilde{\lambda}^k + \gamma_k(\widetilde{y}^k - y^k) = 0, \\ (A\widetilde{x}^k + B\widetilde{y}^k - c) + H^{-1}(\widetilde{\lambda}^k - \lambda^k) = 0. \end{cases} \tag{1.19}$$

Note that by the mean value theorem for integrals, we have for some $\zeta_x^k = x^k + \theta_x^k(\widetilde{x}^k - x^k)$, $\zeta_y^k = y^k + \theta_y^k(\widetilde{y}^k - y^k)$, where $\theta_x^k, \theta_y^k \in (0, 1)$,

$$\begin{aligned} \nabla f(\widetilde{x}^k) - \nabla f(x^k) &= \nabla^2 f(\zeta_x^k)(\widetilde{x}^k - x^k), \\ \nabla g(\widetilde{y}^k) - \nabla g(y^k) &= \nabla^2 g(\zeta_y^k)(\widetilde{y}^k - y^k), \end{aligned} \tag{1.20}$$

Denote

$$\begin{aligned} R_x^k &:= \begin{cases} \widetilde{\nabla}^2 f(x^k) - \nabla^2 f(\zeta_x^k), & \text{if } \widetilde{x}^k \neq x^k, \\ 0, & \text{if } \widetilde{x}^k = x^k, \end{cases} \\ R_y^k &:= \begin{cases} \widetilde{\nabla}^2 g(y^k) - \nabla^2 g(\zeta_y^k), & \text{if } \widetilde{y}^k \neq y^k, \\ 0, & \text{if } \widetilde{y}^k = y^k, \end{cases} \end{aligned} \tag{1.21}$$

$$M_k := \begin{pmatrix} \delta_k I_n + R_x^k & & \\ & B^\top HB + \gamma_k I_m + R_y^k & \\ & & H^{-1} \end{pmatrix}, \tag{1.22}$$

and

$$N := \begin{pmatrix} 0 & -A^\top HB & \\ & -B^\top HB & \\ & & 0 \end{pmatrix}. \tag{1.23}$$

Then, the equations (1.19) can be rewritten as

$$F(\widetilde{w}^k) + (M_k + N)(\widetilde{w}^k - w^k) = 0. \tag{1.24}$$

Lemma 1.1 *Suppose $\widetilde{w}^k$ is generated by Algorithm 1 from an iterate w^k. And suppose $M_k \succeq 0$, then,*

$$\begin{aligned} (w^* - w^k)^\top M_k(\widetilde{w}^k - w^k) &\geq \|\widetilde{w}^k - w^k\|_{M_k}^2 + (\widetilde{\lambda}^k - \lambda^k)B(\widetilde{y}^k - y^k) \\ &\geq \frac{1}{2}\|\widetilde{w}^k - w^k\|_{M_k}^2 \end{aligned} \tag{1.25}$$

where matrices M_k is defined in (1.22).

Proof Since $w^* \in \Omega^*$ and the mapping F (1.10) is monotone, we have

$$(\widetilde{w}^k - w^*)^\top F(\widetilde{w}^k) \geq (\widetilde{w}^k - w^*)^\top F(w^*) = 0. \tag{1.26}$$

Thus, we have

$$\begin{aligned} &(w^* - w^k)^\top M_k(\widetilde{w}^k - w^k) \\ &= \|\widetilde{w}^k - w^k\|_{M_k}^2 + (w^* - \widetilde{w}^k)^\top M_k(\widetilde{w}^k - w^k) \end{aligned}$$

$$\overset{(1.24)}{=} \|\widetilde{w}^k - w^k\|_{M_k}^2 + (\widetilde{w}^k - w^*)^\top F(\widetilde{w}^k) + (\widetilde{w}^k - w^*)^\top N(\widetilde{w}^k - w^k)$$
$$\overset{(1.26)}{\geq} \|\widetilde{w}^k - w^k\|_{M_k}^2 + (\widetilde{w}^k - w^*)^\top N(\widetilde{w}^k - w^k).$$

According to the definition (1.22) of N, we have

$$\begin{aligned}(\widetilde{w}^k - w^*)^\top N(\widetilde{w}^k - w^k) &= \begin{pmatrix}\widetilde{x}^k - x^* \\ \widetilde{y}^k - y^*\end{pmatrix}^\top \begin{pmatrix}-A^\top \\ -B^\top\end{pmatrix} HB(\widetilde{y}^k - y^k)\\ &= -(A\widetilde{x}^k + B\widetilde{y}^k - Ax^* - By^*)^\top HB(\widetilde{y}^k - y^k)\\ &= \left(-H(A\widetilde{x}^k + B\widetilde{y}^k - c)\right)^\top B(\widetilde{y}^k - y^k)\\ &\overset{(1.14)}{=} (\widetilde{\lambda}^k - \lambda^k)B(\widetilde{y}^k - y^k)\\ &\geq -\frac{1}{2}\|\widetilde{\lambda}^k - \lambda^k\|_{H^{-1}}^2 - \frac{1}{2}\|B(\widetilde{y}^k - y^k)\|_H^2\\ &\geq -\frac{1}{2}\|\widetilde{w}^k - w^k\|_{M_k}^2.\end{aligned}$$

Here, $Ax^* + By^* = c$ is used in the second equality. And we prove the inequality (1.25). □

Remark 1.1 Note that if M_k is positive semi-definite, then we could use M_k to construct a merit function to characterize the decline after the prediction step, which is

$$\begin{aligned}&\|w^k - w^*\|_{M_k}^2 - \|\widetilde{w}^k - w^*\|_{M_k}^2\\ &= \|w^k - w^*\|_{M_k}^2 - \|\widetilde{w}^k - w^k + w^k - w^*\|_{M_k}^2\\ &= 2(w^* - w^k)^\top M_k(\widetilde{w}^k - w^k) - \|\widetilde{w}^k - w^k\|_{M_k}^2 \geq 0.\end{aligned} \tag{1.27}$$

1.3.2 *Prediction Step with Trust Region Radii Update*

Lemma 1.1 shows that the prediction step is a descent direction of the merit function $\|w - w^*\|_{M_k}^2$. But we need to ensure $(w^* - w^k)^\top M_k(\widetilde{w}^k - w^k) \geq \|\widetilde{w}^k - w^k\|_{M_k}^2 > 0$ to guarantee the descent property, and we need $M_k \succeq 0$ to ensure the norm is valid.

By definition,

$$\|\widetilde{w}^k - w^k\|_{M_k}^2 = \|\widetilde{x}^k - x^k\|_{\delta_k I_n + R_x^k}^2 + \|\widetilde{y}^k - y^k\|_{\gamma_k I_m + R_y^k + B^\top HB}^2 + \|A\widetilde{x}^k + B\widetilde{y}^k - c\|_H^2.$$

To ensure $\|\widetilde{w}^k - w^k\|_{M_k}^2 > 0$, it suffices to demonstrate under the following conditions:

(i) Either $\widetilde{x}^k = x^k$ or $\delta_k I_n + R_x^k$ is positive definite;
(ii) Either $\widetilde{y}^k = y^k$ or $\gamma_k I_m + R_y^k + B^\top HB$ is positive definite;

(iii) Either $A\widetilde{x}^k + B\widetilde{y}^k - c = 0$ or H is positive definite;
(iv) Either $\widetilde{x}^k \neq x^k$, $\widetilde{y}^k \neq y^k$, or $A\widetilde{x}^k + B\widetilde{y}^k - c \neq 0$.

Since H is positive definite, the condition (iii) is always satisfied. We first study when the matrix $\delta_k I_n + R_x^k$ and $\gamma_k I_m + R_y^k + B^\top HB$ are positive definite.

Assumption 1.1 The Hessian matrix $\nabla^2 f(x)$ and $\nabla^2 g(y)$ are Lipschitz continuous, which means there exist positive constants L_f and L_g such that

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L_f \|x - y\|, \quad \|\nabla^2 g(x) - \nabla^2 g(y)\| \leq L_g \|x - y\|. \tag{1.28}$$

Assumption 1.2 The BFGS-like matrix $\widetilde{\nabla}^2 f(x^k)$ and $\widetilde{\nabla}^2 g(y^k)$ are close to the Hessian matrix $\nabla^2 f(x)$ and $\nabla^2 g(y)$, respectively. Specifically, there exists a positive constant ε such that

$$\|\widetilde{\nabla}^2 f(x^k) - \nabla^2 f(x^k)\| \leq \varepsilon, \quad \|\widetilde{\nabla}^2 g(y^k) - \nabla^2 g(y^k)\| \leq \varepsilon. \tag{1.29}$$

Note that, by (1.20), and Lipschitz continuity of the Hessian matrix, and the error bound of the BFGS-like matrix, we have

$$\begin{aligned}
\|R_x^k\| &= \|\nabla^2 f(\zeta_x^k) - \widetilde{\nabla}^2 f(x^k)\| \leq L_f \|\zeta_x^k - x^k\| + \varepsilon \leq L_f \|\widetilde{x}^k - x^k\| + \varepsilon, \\
\|R_y^k\| &= \|\nabla^2 g(\zeta_y^k) - \widetilde{\nabla}^2 g(y^k)\| \leq L_g \|\zeta_y^k - y^k\| + \varepsilon \leq L_g \|\widetilde{y}^k - y^k\| + \varepsilon.
\end{aligned} \tag{1.30}$$

Now we can see if δ_k and γ_k are large enough, we could have the positive definiteness. And this can always be satisfied as long as the trust region radii Δ_k and Γ_k are small enough. We can see this through (1.16). By using the triangle inequality and induced norm, we have

$$\begin{aligned}
\delta_k &\geq \frac{\left| \|\nabla_x^k \widetilde{\mathcal{L}}\| - \left\| \left(\widetilde{\nabla}^2 f(x^k) + A^\top HA\right)(\widetilde{x}^k - x^k)\right\| \right|}{\|\widetilde{x}^k - x^k\|} \\
&\geq \frac{\|\nabla_x^k \widetilde{\mathcal{L}}\| - \left\| \left(\widetilde{\nabla}^2 f(x^k) + A^\top HA\right)(\widetilde{x}^k - x^k)\right\|}{\|\widetilde{x}^k - x^k\|} \\
&\geq \frac{\|\nabla_x^k \widetilde{\mathcal{L}}\|}{\|\widetilde{x}^k - x^k\|} - \left\|\widetilde{\nabla}^2 f(x^k) + A^\top HA\right\| \\
&\geq \frac{\|\nabla_x^k \widetilde{\mathcal{L}}\|}{\Delta_k} - \left\|\widetilde{\nabla}^2 f(x^k) + A^\top HA\right\|,
\end{aligned} \tag{1.31}$$

where the last inequality is due to the trust region constraint $\|\widetilde{x}^k - x^k\| \leq \Delta_k$. This lower bound of δ_k is not tight, but it gives us an approximate lower bound before we solve the trust region subproblem (1.12). Similarly, we also have a lower bound of γ_k, combine the two lower bounds, we have

$$\begin{aligned} &\|\nabla_x^k \widetilde{\mathcal{L}}\|/\Delta_k - \|\widetilde{\nabla}^2 f(x^k) + A^\top H A\| \leq \delta_k, \\ &\|\nabla_y^k \widetilde{\mathcal{L}}\|/\Gamma_k - \|\widetilde{\nabla}^2 g(y^k) + B^\top H B\| \leq \gamma_k. \end{aligned} \tag{1.32}$$

So, it suffices to have

$$\delta_k \geq \|\nabla_x^k \widetilde{\mathcal{L}}\|/\Delta_k - \|\widetilde{\nabla}^2 f(x^k) + A^\top H A\| \geq L_f \|\widetilde{x}^k - x^k\| + \varepsilon > 0,$$

to ensure $\delta_k I_n + R_x^k$ is positive definite. And since then $\delta_k > 0$, the constraint is active, $\|\widetilde{x}^k - x^k\| = \Delta_k$. We can then solve the above inequality to get the necessary condition of Δ_k. When

$$\Delta_k \leq \frac{\sqrt{(\|\widetilde{\nabla}^2 f(x^k) + A^\top H A\| + \varepsilon)^2 + 4L_f \|\nabla_x^k \widetilde{\mathcal{L}}\|} - (\|\widetilde{\nabla}^2 f(x^k) + A^\top H A\| + \varepsilon)}{2L_f}, \tag{1.33}$$

$\delta_k I_n + R_x^k$ is positive definite.

Note that, when $\|\nabla_x^k \widetilde{\mathcal{L}}\|$ approaches to zero, the valid range of Δ_k vanishes. And also note that, $\|\nabla_x^k \widetilde{\mathcal{L}}\| = 0$ means the subproblem (1.12) attains its minimum at x^k. So we could set a tolerance TOL_x to determine whether the subproblem (1.12) is solved to optimality. When $\|\nabla_x^k \widetilde{\mathcal{L}}\| \leq \mathrm{TOL}_x$, we set $\widetilde{x}^k = x^k$ and solve other subproblems or update λ. And if $\|\nabla_x^k \widetilde{\mathcal{L}}$, $\|\nabla_y^k \widetilde{\mathcal{L}}$, and $\|A\widetilde{x}^k + B\widetilde{y}^k - c\|$ are all small enough, then we find the saddle point of the Lagrangian function, and we stop the algorithm. Following this logic, we could have Algorithm 2 below.

Algorithm 2 (*Prediction Step with Trust Region Radii Update*)

1. Set $1 < \eta_1 < \eta_2$, tolerance $\mathrm{TOL}_x, \mathrm{TOL}_y, \mathrm{TOL}_\lambda > 0$. Calculate estimated Hessians $\widetilde{\nabla}^2 f(x^k)$ and $\widetilde{\nabla}^2 g(y^k)$ by a BFGS-like scheme satisfying (1.29) and calculate $\nabla_x^k \widetilde{\mathcal{L}}$ and $\nabla_y^k \widetilde{\mathcal{L}}$ by (1.16);
2. If $\|\nabla_x^k \widetilde{\mathcal{L}}\| \leq \mathrm{TOL}_x$, then set $\widetilde{x}^k = x^k$, $\Delta_{k+1} = \Delta_k$; otherwise, obtain $\widetilde{x}^k$ and update the trust region radii Δ_{k+1} through Step $2'$;
3. If $\|\nabla_y^k \widetilde{\mathcal{L}}\| \leq \mathrm{TOL}_y$, then set $\widetilde{y}^k = y^k$, $\Gamma_{k+1} = \Gamma_k$; otherwise, obtain $\widetilde{y}^k$ and update the trust region radii Γ_{k+1} through Step $3'$;
4. If $\widetilde{u}^k = u^k$ and $\|A\widetilde{x}^k + B\widetilde{y}^k - c\| \leq \mathrm{TOL}_\lambda$, then stop;

For the x-subproblem, if $\|\nabla_x^k \widetilde{\mathcal{L}}\| > \mathrm{TOL}_x$, we update Δ_{k+1} as follows:

Step 2′(x-subproblem). Set $t = 0$, and $\Delta_{k,0} := \Delta_k$.

(a) For given $\Delta_{k,t}$, complete a prediction step of the x-subproblem (1.12) to get $\widetilde{x}^{k,t}$ and evaluate $\delta_{k,t}$ by (1.17);

(b) If

$$\delta_{k,t} \geq L_f\|\widetilde{x}^{k,t} - x^k\| + \varepsilon := r_x^{k,t}, \tag{1.34}$$

then go to step (c), otherwise set $\Delta_{k,t+1} = \|\widetilde{x}^{k,t} - x^k\|/4$, $t := t + 1$ and then go back to step (a);

(c) Set $\widetilde{x}^k = \widetilde{x}^{k,t}$, and

$$\Delta_{k+1} = \begin{cases} \max\left\{\Delta_{k,t}, 1.5\|\widetilde{x}^{k,t} - x^k\|\right\}, & \delta_{k,t} \in [\eta_2 r_x^{k,t}, +\infty), \\ \Delta_{k,t}, & \delta_{k,t} \in [\eta_1 r_x^{k,t}, \eta_2 r_x^{k,t}], \\ \max\left\{0.5\Delta_{k,t}, 0.75\|\widetilde{x}^{k,t} - x^k\|\right\}, & \delta_{k,t} \in [r_x^{k,t}, \eta_1 r_x^{k,t}]. \end{cases} \tag{1.35}$$

Similarly, for the y-subproblem, if $\|\nabla_y^k\widetilde{\mathcal{L}}\| > \text{TOL}_y$, we update Γ_{k+1} as follows:

Step 3′(y-subproblem). Set $t = 0$, and $\Gamma_{k,0} := \Gamma_k$.

(a) For given $\Gamma_{k,t}$, complete a prediction step of the y-subproblem (1.13) to get $\widetilde{y}^{k,t}$ and evaluate $\gamma_{k,t}$ by (1.18);

(b) If

$$\gamma_{k,t} \geq L_g\|\widetilde{y}^{k,t} - y^k\| + \varepsilon := r_y^{k,t}, \tag{1.36}$$

then go to step (c), otherwise set $\Gamma_{k,t+1} = \|\widetilde{y}^{k,t} - y^k\|/4$, $t := t + 1$ and then go back to step (a);

(c) Set $\widetilde{y}^k = \widetilde{y}^{k,t}$, and

$$\Gamma_{k+1} = \begin{cases} \max\left\{\Gamma_{k,t}, 1.5\|\widetilde{y}^{k,t} - y^k\|\right\}, & \gamma_{k,t} \in [\eta_2 r_y^{k,t}, +\infty), \\ \Gamma_{k,t}, & \gamma_{k,t} \in [\eta_1 r_y^{k,t}, \eta_2 r_y^{k,t}], \\ \max\left\{0.5\Gamma_{k,t}, 0.75\|\widetilde{y}^{k,t} - y^k\|\right\}, & \gamma_{k,t} \in [r_y^{k,t}, \eta_1 r_y^{k,t}]. \end{cases} \tag{1.37}$$

Following the analysis above, we the lemma below to illustrate how we get $\|\widetilde{w}^k - w^k\|_{M_k}^2 > 0$ and $M_k \succeq 0$.

Lemma 1.2 *Suppose $\widetilde{w}^k$ is generated by Algorithm 2, which means for the x-subproblem, either $\|\nabla_x^k\widetilde{\mathcal{L}}\| \leq TOL_x$ or (1.34) holds; for the y-subproblem, either $\|\nabla_y^k\widetilde{\mathcal{L}}\| \leq TOL_y$ or (1.36) holds. Then one of the following conditions holds:*

(1) $\|\nabla_x^k\widetilde{\mathcal{L}}\| \le TOL_x$, $\|\nabla_y^k\widetilde{\mathcal{L}}\| \le TOL_y$, *and* $\|A\widetilde{x}^k + B\widetilde{y}^k - c\| \le TOL_\lambda$*;*
(2) *We have*

$$\|\widetilde{w}^k - w^k\|_{M_k}^2 > 0, \quad M_k \succeq 0. \tag{1.38}$$

Proof If $\|\nabla_x^k\widetilde{\mathcal{L}}\| \le \mathrm{TOL}_x$ and $\|\nabla_y^k\widetilde{\mathcal{L}}\| \le \mathrm{TOL}_y$, then we have $\widetilde{x}^k = x^k$ and $\widetilde{y}^k = y^k$. Then by the definition of $\|\widetilde{w}^k - w^k\|_{M_k}^2$, we have $\|\widetilde{w}^k - w^k\|_{M_k}^2 = \|\widetilde{\lambda} - \lambda^k\|_{H^{-1}}^2 = \|Ax^k + By^k - c\|^2$.

Thus, either $\|A\widetilde{x}^k + B\widetilde{y}^k - c\| \le \mathrm{TOL}_\lambda$ which means statement (1) holds or $\|\widetilde{w}^k - w^k\|_{M_k}^2 \ge \mathrm{TOL}_\lambda^2 > 0$. Since $\widetilde{x}^k = x^k$ and $\widetilde{y}^k = y^k$, we have $\zeta_x^k = x^k$ so that $R_x^k = R_y^k = 0$, which means $M_k \succeq 0$.

If $\|\nabla_x^k\widetilde{\mathcal{L}}\| > \mathrm{TOL}_x$ then

$$\|\widetilde{w}^k - w^k\|_{M_k}^2 \ge \|\widetilde{x}^k - x^k\|_{M_k}^2 \ge (\delta_k - \|R_x^k\|)\|\widetilde{x}^k - x^k\|^2.$$

By (1.30) and (1.34), we have $\|\widetilde{w}^k - w^k\|_{M_k}^2 > 0$ and $\|\delta_k I_n - R_x^k\| \ge 0$. Similarly, if $\|\nabla_y^k\widetilde{\mathcal{L}}\| > \mathrm{TOL}_y$, we also have $\|\widetilde{w}^k - w^k\|_{M_k}^2 > 0$ and $\|\gamma_k I_m - R_y^k\| \ge 0$. Combine the above analysis, we have the statement (2) holds.

Remark 1.2 When the condition (1) in Lemma 1.2 hold, we have $\widetilde{x}^k = x^k$, and $\widetilde{y}^k = y^k$. Thus the condition is equivalent to $\|F(w^k)\| \le \mathrm{TOL}$ with $\mathrm{TOL} = \sqrt{\mathrm{TOL}_x^2 + \mathrm{TOL}_y^2 + \mathrm{TOL}_\lambda^2}$. This means the algorithm stops when the optimality condition (1.11) is satisfied with tolerance TOL.

Next, we will show that given the tolerance, $\|M_k\|$ has an upper bound globally. We could prove it by giving an upper bound of both δ_k and γ_k. And this could be done by showing the lower bound of the trust region radii Δ_k and Γ_k. Note that we could get an upper bound of δ_k by the same way as we get the lower bound in (1.31):

$$\delta_k \le \frac{\|\nabla_x^k\widetilde{\mathcal{L}}\|}{\|\widetilde{x}^k - x^k\|} + \left\|\widetilde{\nabla}^2 f(x^k) + A^\top HA\right\|.$$

If $\|\widetilde{x}^k - x^k\| < \Delta_k$, which means the trust region constraint is inactive, we have $\delta_k = 0$, so that $\delta_k \le \left\|\nabla_x^k\widetilde{\mathcal{L}}\right\|/\Delta_k + \left\|\widetilde{\nabla}^2 f(x^k) + A^\top HA\right\| \ge 0$. And when $\|\widetilde{x}^k - x^k\| = \Delta_k$, this upper bound is also valid. Thus we have

$$\begin{aligned}\delta_k &\le \|\nabla_x^k\widetilde{\mathcal{L}}\|/\Delta_k + \|\widetilde{\nabla}^2 f(x^k) + A^\top HA\|,\\ \gamma_k &\le \|\nabla_y^k\widetilde{\mathcal{L}}\|/\Gamma_k + \|\widetilde{\nabla}^2 g(y^k) + B^\top HB\|.\end{aligned} \tag{1.39}$$

By Assumption 1.1, if the iteration points are bounded, we would have a uniform upper bound of the Hessian matrix $\widetilde{\nabla}^2 f(x^k)$, $\widetilde{\nabla}^2 g(y^k)$, and the gradients $\nabla_x^k\widetilde{\mathcal{L}}$, $\nabla_y^k\widetilde{\mathcal{L}}$. Then we could have a uniform upper bound of δ_k and γ_k in the following lemma.

Lemma 1.3 *Suppose $\widetilde{w}^k$ is generated by the prediction step with trust region radii update above from an iterate w^k. Assume $\|\widetilde{\nabla}^2 f(x^k)\| \leq \Sigma_f$, and $\|\widetilde{\nabla}^2 g(y^k)\| \leq \Sigma_g$, then we have*

$$\begin{aligned}\Delta_k > \Delta_{min} &:= 0.5 \cdot \frac{1}{4} \cdot \frac{\sqrt{(\Sigma_f + \varepsilon)^2 + 4L_f TOL_x} - (\Sigma_f + \varepsilon)}{2L_f},\\ \Gamma_k > \Gamma_{min} &:= 0.5 \cdot \frac{1}{4} \cdot \frac{\sqrt{(\Sigma_g + \varepsilon)^2 + 4L_g TOL_y} - (\Sigma_g + \varepsilon)}{2L_g}.\end{aligned} \tag{1.40}$$

And further assume $\|\nabla_x^k \widetilde{\mathcal{L}}\| \leq \nabla_x^{\max}$, $\|\nabla_y^k \widetilde{\mathcal{L}}\| \leq \nabla_y^{\max}$, then there exists a constant $\kappa_1 > 0$ such that for all $k \geq 0$, we have

$$\|M_k\| \leq \kappa_1. \tag{1.41}$$

Proof According to (1.32), before we conduct a prediction step, in order to ensure (1.34) holds, it is sufficient to let Δ_k satisfy

$$L_f \|\widetilde{x}^k - x^k\| + \varepsilon = L_f \Delta_k + \varepsilon \leq \|\nabla_x^k \widetilde{\mathcal{L}}\| / \Delta_k - \|\widetilde{\nabla}^2 f(x^k) + A^\top HA\| \leq \delta_k.$$

So, as long as Δ_k satisfies

$$\Delta_k \geq \frac{\sqrt{(\|\widetilde{\nabla}^2 f(x^k) + A^\top HA\| + \varepsilon)^2 + 4L_f \|\nabla_x^k \widetilde{\mathcal{L}}\|} - (\|\widetilde{\nabla}^2 f(x^k) + A^\top HA\| + \varepsilon)}{2L_f},$$

Since $\|\widetilde{\nabla}^2 f(x^k)\| \leq \Sigma_f$, and Δ_k only shrink when we update the trust region radii, which means $\|\nabla_x^k \widetilde{\mathcal{L}}\| > \text{TOL}_x$, we have for all $k \geq 0$,

$$\Delta_k > \Delta_{\min} = 0.5 \cdot \frac{1}{4} \cdot \frac{\sqrt{(\Sigma_f + \varepsilon)^2 + 4L_f \text{TOL}_x} - (\Sigma_f + \varepsilon)}{2L_f}.$$

The other inequality for Γ_k is similar.

Having the uniform lower bound of the trust region radii, the uniform upper bound of the Hessian matrix $\widetilde{\nabla}^2 f(x^k)$, $\widetilde{\nabla}^2 g(y^k)$, and the gradients $\nabla_x^k \widetilde{\mathcal{L}}$, $\nabla_y^k \widetilde{\mathcal{L}}$, we could obtain a uniform upper bound of δ_k and γ_k through (1.39) as

$$\begin{aligned}\delta_k &\leq \nabla_x^{\max} / \Delta_{\min} + \Sigma_f + \varepsilon + \|A^\top HA\|,\\ \gamma_k &\leq \nabla_y^{\max} / \Gamma_{\min} + \Sigma_g + \varepsilon + \|B^\top HB\|.\end{aligned} \tag{1.42}$$

Thus there exists $\kappa_1 > 0$ such that for all $k \geq 0$, we have (1.41).

Remark 1.3 The x-subproblem and y-subproblem are solved by trust region subproblems. The reason we use trust region subproblems is usually that the actual subproblems are hard to solve exactly. If one of the subproblems is easy to solve or even has a closed-form solution, we could use the exact solution to update the corresponding variable. In this case, we see the variable to be solved exactly as y-subproblem. Since the y-subproblem is solved exactly, we have $R_y^k = 0$ and $\gamma_k = 0$. Then we always have the y-part of M_k is $B^\top HB \succeq 0$.

As a result, this version of the Algorithm still satisfies all the Lemmas above. Specifically, (1.25), (1.38), and (1.41) still hold for this version of Algorithm 2 with Step 3$''$.

Step 3$''$(exact y-subproblem). Obtain $\widetilde{y}$ by solving the exact y-subproblem:

$$\widetilde{y}^k = \operatorname{argmin}_{y\in\mathbb{R}^m} \mathcal{L}_H(\widetilde{x}^k, y, \lambda^k), \tag{1.43}$$

1.3.3 The Correction Step

Through Remark 1.1 and Lemma 1.2, we know that the direction achieved by the prediction step is a descent direction of the merit function $\|w - w^*\|_{M_k}^2$. However, M_k is not a constant matrix, so we need a correction step to handle this issue.

Let W be a symmetric positive definite matrix and $w^* \in \Omega^*$. Then, $\|w - w^*\|_{W^{-1}}^2$ could be viewed as a measurement for the iterate w being an optimal solution. Next, we show that $WM_k(\widetilde{w}^k - w^k)$ is a descent direction of the merit function $\|w - w^*\|_{W^{-1}}^2$ at an iterate w^k.

Before we proceed, it is worth noting that $WM_k(\widetilde{w}^k - w^k)$ is calculable although M_k is unknown. Specifically, we have

$$\begin{aligned}
&M_k(\widetilde{w}^k - w^k)\\
&= \begin{pmatrix} \left(\delta_k I_n + R_x^k\right)(\widetilde{x}^k - x^k) \\ \left(B^\top HB + \gamma_k I_m + R_y^k\right)(\widetilde{y}^k - y^k) \\ A\widetilde{x}^k + B\widetilde{y}^k - c \end{pmatrix}\\
&= \begin{pmatrix} \delta_k(\widetilde{x}^k - x^k) + \widetilde{\nabla}^2 f(x^k)(\widetilde{x}^k - x^k) - \nabla f(\widetilde{x}^k) + \nabla f(x^k) \\ \gamma_k(\widetilde{y}^k - y^k) + \left(\widetilde{\nabla}^2 g(y^k) + B^\top HB\right)(\widetilde{y}^k - y^k) - \nabla g(\widetilde{y}^k) + \nabla g(y^k) \\ A\widetilde{x}^k + B\widetilde{y}^k - c \end{pmatrix}.
\end{aligned} \tag{1.44}$$

So $WM_k(\widetilde{w}^k - w^k)$ is a well-defined descent direction. We also have terms like $\|\widetilde{w}^k - w^k\|_{M_k}^2$ be calculable. Next, we equip the descent direction with a suitable step length.

For convenience, we define

$$\varphi(\widetilde{w}^k, w^k) := \|\widetilde{w}^k - w^k\|_{M_k}^2 + (\widetilde{\lambda}^k - \lambda^k)^\top B(\widetilde{y}^k - y^k). \tag{1.45}$$

To obtain the maximal decrease of the merit function $\|w - w^*\|_{W^{-1}}^2$, we define

$$w^{k+1}(\alpha) := w^k + \alpha W M_k(\widetilde{w}^k - w^k)$$

and the maximal following difference

$$\begin{aligned}
&\|w^k - w^*\|^2_{W^{-1}} - \|w^{k+1}(\alpha) - w^*\|^2_{W^{-1}} \\
&= \|w^k - w^*\|^2_{W^{-1}} - \|w^k - w^* + \alpha W M_k(\widetilde{w}^k - w^k)\|^2_{W^{-1}} \\
&= -2\alpha(w^k - w^*)^\top M_k(\widetilde{w}^k - w^k) - \alpha^2\|W M_k(\widetilde{w}^k - w^k)\|^2_{W^{-1}} \\
&\geq 2\alpha\varphi(\widetilde{w}^k, w^k) - \alpha^2\|M_k(\widetilde{w}^k - w^k)\|^2_W \\
&=: \Psi(\alpha),
\end{aligned}$$

It is worthwhile to note that the quadratic function $\Psi(\alpha)$ attains its maximum at

$$\alpha_k = \frac{\varphi(\widetilde{w}^k, w^k)}{\|M_k(\widetilde{w}^k - w^k)\|^2_W}, \tag{1.46}$$

Since $\|w^k - w^*\|^2 - \|w^{k+1}(\alpha) - w^*\|^2 \geq \Psi(\alpha)$, we choose a larger step size $\rho\alpha_k$, where $\rho = [1, 2)$. The resulting correction step is drawn as follows.

Algorithm 3 (*Correction step*) Obtain a prediction step $\widetilde{w}^k$ by Algorithm 2. Compute a step size α_k by (1.46) and generate the new iterate:

$$w^{k+1} := w^k + \rho\alpha_k W M_k(\widetilde{w}^k - w^k). \tag{1.47}$$

From this correction step, we show the decrease of the merit function $\|w - w^*\|^2_{W^{-1}}$ in the next lemma.

Lemma 1.4 *Suppose $\widetilde{w}_k$ and w^{k+1} are generated by Algorithms 2 and 3, respectively. Suppose Lemma 1.2 (2) holds. Then,*

$$\|w^{k+1} - w^*\|^2_{W^{-1}} \leq \|w^k - w^*\|^2_{W^{-1}} - \frac{2-\rho}{\rho}\|w^{k+1} - w^k\|^2_{W^{-1}}. \tag{1.48}$$

Proof By Lemma 1.2 (2), we have Lemma 1.1. By Lemma 1.1, (1.45), the definition (1.46) of α_k and the update rule (1.47) for w^{k+1}, we have

$$\begin{aligned}
&\|w^{k+1} - w^*\|^2_{W^{-1}} - \|w^k - w^*\|^2_{W^{-1}} \\
&\overset{(1.47)}{=} \|w^k - w^* + \rho\alpha_k W M_k(\widetilde{w}^k - w^k)\|^2_{W^{-1}} - \|w^k - w^*\|^2_{W^{-1}} \\
&= \rho^2\alpha_k^2\|W M_k(\widetilde{w}^k - w^k)\|^2_{W^{-1}} + 2\rho\alpha_k(w^k - w^*)^\top M_k(\widetilde{w}^k - w^k) \\
&\overset{(1.25),(1.45)}{\leq} \rho^2\alpha_k^2\|M_k(\widetilde{w}^k - w^k)\|^2_W - 2\rho\alpha_k\varphi(\widetilde{w}^k, w^k) \\
&\overset{(1.46)}{=} -(2-\rho)\rho\alpha_k^2\|M_k(\widetilde{w}^k - w^k)\|^2_W
\end{aligned}$$

$$= -(2-\rho)\rho\alpha_k^2\|WM_k(\widetilde{w}^k - w^k)\|_{W^{-1}}^2$$
$$\overset{(1.47)}{=} -(2-\rho)\rho^{-1}\|w^{k+1} - w^k\|_{W^{-1}}^2.$$

1.4 Convergence Analysis

To prove the global convergence of the trust region-based splitting method, we need an additional assumption.

According to (1.41), there exists a positive constant κ such that

$$\|WM_k\| \le \kappa_1\|W\| := \kappa, \qquad \forall\, k. \tag{1.49}$$

Along with Lemma 1.1, the step size α_k defined in (1.46) is obviously bounded away from zero:

$$\alpha_k = \frac{\varphi\left(\widetilde{w}^k, w^k\right)}{\left\|M_k\left(\widetilde{w}^k - w^k\right)\right\|_W^2} \ge \frac{\frac{1}{2}\left\|\widetilde{w}^k - w^k\right\|_{M_k}^2}{\left\|\widetilde{w}^k - w^k\right\|_{M_k^\top WM_k}^2} \ge \frac{1}{2\kappa} \quad \forall k. \tag{1.50}$$

1.4.1 Global Convergence

Now, we present the main convergence result.

Theorem 1.1 *Given an arbitrary initial iterate w^0 and let sequences $\{w^{k+1}\}$ be generated by Algorithm 3. Then, either we have Lemma 1.2 (1) holds some k or*

- $\lim_{k\to\infty}\|w^{k+1} - w^k\|_{W^{-1}} = 0.$
- $\lim_{k\to\infty}\|M_k(\widetilde{w}^k - w^k)\| = 0.$
- *Any accumulation point of $\{\widetilde{w}^k\}$ is a solution of VI (1.11).*
- *$\{w^k\}$ converges to a solution of VI (1.11).*

Proof Once the initial iterate w^0 is given, according to Lemma 1.2, either (1) holds or (2) holds for every k. If (1) holds for some k, then the theorem is proved. Otherwise, we have (2) holds for every k, then by Lemma 1.4, the sequence of merit function $\{\|w^k - w^*\|_{W^{-1}}^2\}$ is decrease. So $\{w^k\}$ is bounded above and $\|w^{k+1} - w^k\|_{W^{-1}}$ must be vanish as $k \to \infty$.

By the update rule (1.47) of w^{k+1} and the low bound (1.50) of the step size α_k, we get the second assertion.

Since the merit function $\{\|w^k - w^*\|_{W^{-1}}^2\}$ is decreasing, $\{w^k\}$ is bounded, so Lemma 1.3 holds. Thus we have (1.41) and then the lower bound (1.50) of α_k. Combining with Lemma 1.1, (1.47), and the definitions (1.22) of matrices M_k and N, we have

$$\begin{aligned}\left\| M_k\left(\widetilde{w}^k - w^k\right)\right\|_W^2 &= \alpha_k \cdot \varphi\left(\widetilde{w}^k, w^k\right)\\ &\geq \frac{1}{2\kappa}\cdot\frac{1}{2}\left\|\widetilde{w}^k - w^k\right\|_{M_k}^2\\ &\geq \frac{1}{4\kappa}\left\|B\left(\widetilde{y}^k - y^k\right)\right\|_H^2.\end{aligned}$$

Since $\|M_k(\widetilde{w}^k - w^k)\|_W \to 0$ as $k \to \infty$, we get $\|B(\widetilde{y}^k - y^k)\|_H \to 0$. So $\|N(\widetilde{w}^k - w^k)\| \to 0$. Recalling the important inequality (1.24), we obtain

$$\|F(\widetilde{w}^k)\| = \|\left(M_k + N\right)\left(\widetilde{w}^k - w^k\right)\| \to 0.$$

So, the third assertion is also valid.

Since $\{w^k\}$ is bounded, it has at least one accumulation point, which is also an accumulation point of $\{\widetilde{w}^k\}$; hence solution point of VI (1.11). The inequality (1.48) implies that $\{w^k\}$ has just one accumulation point, and the last assertion follows immediately. □

1.4.2 Convergence Rate

In this section, we show that the proposed trust region-based splitting method enjoys the $O(1/\epsilon)$ convergence rate in an ergodic sense.

Theorem 1.2 ([Theorem 2.1 in *[17]*) *The optimal solution set Ω^* of the VI (1.11) is convex and can be characterized as*

$$\Omega^* = \bigcap_{w\in\Omega}\left\{\widetilde{w}\in\Omega \mid (w-\widetilde{w})^\top F(w) \geq 0\right\}. \tag{1.51}$$

According to this theorem, we say that $\overline{w}_t$ is an ϵ-approximate optimal solution if it satisfies

$$(w - \overline{w}_t)^\top F(w) \geq -\epsilon \qquad \forall\, w\in\Omega, \tag{1.52}$$

where ϵ is a small positive number. To obtain an ϵ, e-approximate optimal solution, we reveal that the worst-case iterative number t satisfies $t = O(1/\epsilon)$.

Next, we introduce a new vector

$$\widehat{w}^k = \begin{pmatrix}\widehat{x}^k\\ \widehat{y}^k\\ \widehat{\lambda}^k\end{pmatrix} := \begin{pmatrix}\widetilde{x}^k\\ \widetilde{y}^k\\ \lambda^k - H(A\widetilde{x}^k + By^k - c)\end{pmatrix}, \tag{1.53}$$

and a new matrix

$$\widehat{M}_k := \begin{pmatrix} \delta_k I_n + R_x^k & & \\ & B^\top HB + \gamma_k I_m + R_y^k & \\ & -B & H^{-1} \end{pmatrix}, \tag{1.54}$$

that will be used in this section. By some calculation, we obtain the following Lemma.

Lemma 1.5 *According to the new notations (1.53) and (1.54), the prediction step produces* $\widehat{w}^k \in \Omega$ *satisfying*

$$(w - \widehat{w}^k)^\top [F(\widehat{w}^k) + \widehat{M}_k(\widehat{w}^k - w^k)] \geq 0, \qquad \forall\, w \in \Omega. \tag{1.55}$$

Moreover, the following equalities hold

$$M_k(\widetilde{w}^k - w^k) = \widehat{M}_k(\widehat{w}^k - w^k), \tag{1.56}$$

and

$$\varphi(\widetilde{w}^k, w^k) = (\widehat{w}^k - w^k)^\top \widehat{M}_k^\top (\widehat{w}^k - w^k). \tag{1.57}$$

Proof Based on the new vector $\widehat{w}^k$, (1.19) could be rewritten as

$$\begin{cases} \nabla f(\widehat{x}^k) + \widetilde{\nabla}^2 f(x^k)(\widehat{x}^k - x^k) - A^\top \widehat{\lambda}^k + \delta_k(\widehat{x}^k - x^k) = 0, \\ \nabla g(\widehat{y}^k) + \widetilde{\nabla}^2 g(y^k)(\widehat{y}^k - y^k) - B^\top \widehat{\lambda}^k + (B^\top HB + \gamma_k I_m)(\widehat{y}^k - y^k) = 0, \\ (A\widehat{x}^k + B\widehat{y}^k - c) - B(\widehat{y}^k - y^k) + H^{-1}(\widehat{\lambda}^k - \lambda^k) = 0. \end{cases}$$

In a compact form, the above VI equals to (1.55).

Since $H^{-1}(\widetilde{\lambda}^k - \lambda^k) = -(A\widetilde{x}^k + B\widetilde{y}^k - c) = H^{-1}(\widehat{\lambda}^k - \lambda^k) - B(\widehat{y}^k - y^k)$, the equation (1.56) hold.

From the definition of $\varphi(\widetilde{w}^k, w^k)$ in (1.45), the equality (1.56) and the definitions (1.22) and (1.54) of matrices M_k and $\widehat{M}_k$, we have

$$\begin{aligned} \varphi(\widetilde{w}^k, w^k) &= \|\widetilde{w}^k - w^k\|_{M_k}^2 + (\widetilde{\lambda}^k - \lambda^k)^\top B(\widetilde{y}^k - y^k) \\ &= (\widetilde{w}^k - w^k)^\top \widehat{M}_k(\widehat{w}^k - w^k) + (\widetilde{\lambda}^k - \lambda^k)^\top B(\widetilde{y}^k - y^k) \\ &= (\widetilde{w}^k - w^k)^\top M_k(\widehat{w}^k - w^k) \\ &= (\widehat{w}^k - w^k)^\top M_k(\widetilde{w}^k - w^k) \\ &= (\widehat{w}^k - w^k)^\top \widehat{M}_k(\widehat{w}^k - w^k). \end{aligned}$$

The proof is complete.

Lemma 1.6 *Suppose the sequence* $\{\widehat{w}^k\}$ *is generated by the trust region-based splitting method. Then, we get*

$$(w - \widehat{w}^k)^\top F(w) \geq (\widehat{w}^k - w)^\top \widehat{M}(\widehat{w}^k - w^k), \tag{1.58}$$

Proof Since the mapping F (1.10) is monotone, we have

$$(w-\widehat{w}^k)^\top F(w)-(w-\widehat{w}^k)^\top F(\widehat{w}^k)\geq 0.$$

From the VI (1.55) in Lemma 1.5, we get

$$(w-\widehat{w}^k)^\top F(\widehat{w}^k)\geq(\widehat{w}^k-w)^\top \widehat{M}_k(\widehat{w}^k-w^k)$$

Adding the above two inequalities, we obtain the validation of this lemma.

Lemma 1.7 *Suppose the sequences* $\{\widehat{w}^k\}$ *and* $\{w^{k+1}\}$ *are generated by the prediction and correction steps, respectively. Then, we have*

$$(\widehat{w}^k-w)^\top W^{-1}(w^{k+1}-w^k)\geq\frac{1}{2}\left(\|w-w^{k+1}\|^2_{W^{-1}}-\|w-w^k\|^2_{W^{-1}}\right), \tag{1.59}$$

Proof By some calculation, we have

$$\begin{aligned}&(\widehat{w}^k-w)^\top W^{-1}(w^{k+1}-w^k)\\&=\frac{1}{2}\left(\|w-w^{k+1}\|^2_{W^{-1}}-\|w-w^k\|^2_{W^{-1}}\right)+\frac{1}{2}\left(\|w^k-\widehat{w}^k\|^2_{W^{-1}}-\|w^{k+1}-\widehat{w}^k\|^2_{W^{-1}}\right).\end{aligned}$$

Hence, the rest of the proof is to show that the second term is nonnegative. By the update rule (1.47) of w^{k+1}, the definition (1.46) of α_k, and Lemma 1.5, we get the following equality.

$$\begin{aligned}\left(w^{k+1}-w^k\right)^\top W^{-1}\left(w^k-\widehat{w}^k\right)&=\left[\rho\alpha_k WM_k\left(\widetilde{w}^k-w^k\right)\right]^\top W^{-1}\left(w^k-\widehat{w}^k\right)\\&=-\rho\alpha_k\left(\widehat{w}^k-w^k\right)^\top\widehat{M}_k^\top\left(\widehat{w}^k-w^k\right)\\&=-\rho\alpha_k\varphi\left(\widetilde{w}^k,w^k\right)\\&=-\rho\alpha_k^2\left\|M_k\left(\widetilde{w}^k-w^k\right)\right\|^2_W\\&=-\rho^{-1}\left\|w^{k+1}-w^k\right\|^2_{W^{-1}}.\end{aligned}$$

So, the second term is

$$\begin{aligned}&\|w^{k+1}-\widehat{w}^k\|^2_{W^{-1}}-\|w^k-\widehat{w}^k\|^2_{W^{-1}}\\&=\|w^{k+1}-w^k\|^2_{W^{-1}}+2(w^{k+1}-w^k)^\top W^{-1}(w^k-\widehat{w}^k)\\&=\|w^{k+1}-w^k\|^2_{W^{-1}}-2\rho^{-1}\left\|w^{k+1}-w^k\right\|^2_{W^{-1}}\\&=\frac{\rho-2}{\rho}\|w^{k+1}-w^k\|^2_{W^{-1}}\geq 0.\end{aligned}$$

Therefore, this lemma is proved.

Finally, we give the main theorem.

Theorem 1.3 *For any positive integer t, we define*

$$\overline{w}_t = \frac{\sum_{k=0}^{t} \alpha_k \widehat{w}^k}{\sum_{k=0}^{t} \alpha_k}. \tag{1.60}$$

Then, $\overline{w}_t \in \Omega$ and

$$(\overline{w}_t - w)^\top F(w) \le \frac{\kappa}{\rho(t+1)} \|w - w^0\|_{W^{-1}}^2, \tag{1.61}$$

which means that the proposed trust region-based splitting method enjoys the $O(1/\epsilon)$ convergence rate in an ergodic sense.

Proof Since $\overline{w}_t$ is a convex combination of iterates $\{\widehat{w}^0, \widehat{w}^1, \dots, \widehat{w}^t\}$ and the set Ω is convex, we get $\overline{w}_t \in \Omega$.

From Lemmas 1.5, 1.6 and 1.7, we get for all $w \in \Omega$,

$$\begin{aligned}\rho\alpha_k \left((w - \widehat{w}^k)^\top F(w)\right) &\ge (\widehat{w}^k - w)^\top \left(\rho\alpha_k \widehat{M}_k(\widehat{w}^k - w^k)\right) \\ &= (\widehat{w}^k - w)^\top W^{-1}(w^{k+1} - w^k) \\ &\ge \frac{1}{2}\left(\|w - w^{k+1}\|_{W^{-1}}^2 - \|w - w^k\|_{W^{-1}}^2\right)\end{aligned}$$

Then, we sum over the above inequalities for $k = 0, 1, \dots, t$ and divide $\rho \sum_{k=0}^{t} \alpha_k$,

$$\begin{aligned}\left(w - \frac{\sum_{k=0}^{t} \alpha_k \widehat{w}^k}{\sum_{k=0}^{t} \alpha_k}\right)^\top F(w) &\ge \frac{1}{2\rho \sum_{k=0}^{t} \alpha_k}\left(\|w - w^{t+1}\|_{W^{-1}}^2 - \|w - w^0\|_{W^{-1}}^2\right) \\ &\ge -\frac{1}{2\rho \sum_{k=0}^{t} \alpha_k} \|w - w^0\|_{W^{-1}}^2.\end{aligned}$$

Furthermore, we note that $\alpha_k \ge \frac{1}{2\kappa}$ by (1.50), so we have $2\sum_{k=0}^{t} \alpha_k \ge \frac{1}{\kappa}(t+1)$. Hence, we conclude that

$$(w - \overline{w}_t)^\top F(w) \ge -\frac{\kappa}{\rho(t+1)} \|w - w^0\|_{W^{-1}}^2, \qquad \forall\, w \in \Omega.$$

The proof is complete.

Therefore, for any given initial iterate $w^0 \in \Omega$ and any nonempty compact set $\mathcal{D} \in \Omega$, we denote $d := \max\{\|w - w^0\|_{W^{-1}} \mid w \in \mathcal{D}\}$. Then, for any small positive number ϵ, after

$$t = \left\lfloor \frac{\kappa d^2}{\rho \epsilon} \right\rfloor$$

iterations, the trust region-based splitting method produces a point $\overline{w}_t$ (1.60) such that

$$\sup_{w\in\mathcal{D}} \left\{(\overline{w}_t - w)^\top F(w)\right\} \le \epsilon^{-1}.$$

In this viewpoint, we approve that the trust region-based splitting method enjoys the $\mathcal{O}(1/\epsilon)$ convergence rate in an ergodic sense.

1.5 Numerical Experiments

In this section, we apply the proposed trust region-based splitting method to image recovery problems and logistic regression problems, and compare it with some existing splitting methods.

1.5.1 Constrained TV-ℓ^2 Image Recovery Problems

The image recovery problem is a sort of inverse problem in image analysis. Our interest is in finding the true image z from its degraded observation z_0. The degradation comes from two facts. The first one is blur which is created by an improper lens adjustment or a movement. The other one is random noise b that is assumed to be Gaussian, white, and additive. The linear model of the degradation process is

$$z_0 = \Phi z + b,$$

where $z_0, z, b \in \mathbb{R}^n$, $\Phi \in \mathbb{R}^{n\times n}$ is a linear blur map and n is the number of pixels.

Next, we give the definition of the total variation (TV) of a two-dimensional (2D) image [20]. For a given image z, its discrete derivative ∇z in horizontal and vertical directions are denoted by $\partial_1 z$ and $\partial_2 z$, respectively. Then,

$$\nabla z := \begin{pmatrix} \partial_1 z \\ \partial_2 z \end{pmatrix} \in \mathbb{R}^{2n}. \tag{1.62}$$

When the circular boundary conditions are assumed, these two partial derivatives could be efficiently computed using the 2D discrete Fourier transformation $\mathcal{F}$:

$$\partial_1 z = \mathcal{F}^{-1} D_1 \mathcal{F} z, \quad \text{and} \quad \partial_2 z = \mathcal{F}^{-1} D_2 \mathcal{F} z,$$

where $D_1, D_2 \in \mathbb{C}^{n\times n}$ are two diagonal matrices. Moreover, the blur map is also diagonalized as $\Phi = \mathcal{F}^{-1} K \mathcal{F}$, where $K \in \mathbb{C}^{n\times n}$ is the blur kernel. Then, the TV of z is defined as:

$$\||\nabla z|\|_1 := \sum_{i=1}^{n} \sqrt{[\partial_1 z]_i^2 + [\partial_2 z]_i^2} \in \mathbb{R}^n. \tag{1.63}$$

In this experiment, we consider the constrained TV-ℓ^2 model for image recovery which is studied in [18]:

$$\begin{cases} \min & \||\nabla z|\|_1 \\ \text{s.t.} & \|\Phi z - z_0\|^2 \le \sigma, \end{cases} \tag{1.64}$$

where σ denotes a noise level which is known in advance. To apply the proposed trust region-based splitting algorithm, we introduce two auxiliary variables $s = \nabla z$, $t = \Phi z - z_0$ and an indicator function of a ball $\odot = \{t \mid \|t\|^2 \le \sigma\}$

$$\pi_\odot(t) = \begin{cases} 0 & \text{if } \|t\|^2 \le \sigma, \\ +\infty & \text{otherwise.} \end{cases}$$

Then, the constrained TV-ℓ^2 model (1.64) is equivalent to

$$\begin{cases} \min & \||s|\|_1 + \pi_\odot(t) \\ \text{s.t.} & \nabla z - s = 0, \ s \in \mathbb{R}^{2n}, \\ & \Phi z - t = z_0, \ t \in \mathbb{R}^n. \end{cases}$$

Let

$$x = z, \qquad y = \begin{pmatrix} s \\ t \end{pmatrix}, \qquad \lambda = \begin{pmatrix} \mu \\ \nu \end{pmatrix} \in \mathbb{R}^{2n} \times \mathbb{R}^n,$$
$$f(x) = 0, \qquad g(y) = \||s|\|_1 + \pi_\odot(t),$$
$$A = \begin{pmatrix} \nabla \\ \Phi \end{pmatrix}, \quad B = \begin{pmatrix} -I_{2n} & 0 \\ 0 & -I_n \end{pmatrix}, \quad c = \begin{pmatrix} 0 \\ z_0 \end{pmatrix}, \quad H = \begin{pmatrix} \xi I_{2n} & 0 \\ 0 & \eta I_n \end{pmatrix}.$$

The augmented Lagrangian function is represented as follows:

$$\begin{aligned} \mathcal{L}_{(\xi,\eta)}(s,t,z,\mu,\nu) = {} & \||s|\|_1 + \pi_\odot(t) - \mu^\top(\nabla z - s) - \nu^\top(\Phi z - t - z_0) \\ & + \frac{\xi}{2}\|\nabla z - s\|^2 + \frac{\eta}{2}\|\Phi z - t - z_0\|^2. \end{aligned}$$

whereafter, we give the initial image $z^0 = z_0$, initial auxiliary variables $s = 0$ and $t = 0$, initial multiplier $\mu^0 = 0$ and $\nu^0 = 0$ and begin the following loops with $k \leftarrow 0$.

- Update the variable $x = z$ in a trust region $\|z - z^k\| \le \Delta_k$.

$$\begin{aligned} \tilde{z}^k = {} & \text{argmin}_z \left\{ -(\mu^k)^\top \nabla z - (\nu^k)^\top \Phi z + \frac{\xi}{2}\|\nabla z - s^k\|^2 + \frac{\eta}{2}\|\Phi z - t^k - z_0\|^2 \right\} \\ & \text{s.t. } \|z - z^k\| \le \Delta_k \\ = {} & \text{argmin}_z \left\{ \frac{\xi}{2}\|\nabla z - s^k - \xi^{-1}\mu^k\|^2 + \frac{\eta}{2}\|\Phi z - t^k - z_0 - \eta^{-1}\nu^k\|^2 \right\} \\ & \text{s.t. } \|z - z^k\| \le \Delta_k. \end{aligned}$$

Let $d := z - z^k$. Then we turn to minimize d

$$\begin{cases} \text{argmin}_d \; \frac{\xi}{2}\|\nabla d + \nabla z^k - s^k - \xi^{-1}\mu^k\|^2 + \frac{\eta}{2}\|\Phi d + \Phi z^k - t^k - z_0 - \eta^{-1}\nu^k\|^2 \\ \quad\;\; \text{s.t.} \;\; \|d\| \leq \Delta_k. \end{cases} \tag{1.65}$$

When the 2D discrete Fourier transformation is employed, we define $\widehat{d} := \mathcal{F}d$. Then, the trust region constraint $\|d\| \leq \Delta_k$ is equivalent to $\|\widehat{d}\| \leq \sqrt{n}\Delta_k$. Furthermore, let

$$\begin{pmatrix} p \\ q \end{pmatrix} := \nabla z^k - s^k - \xi^{-1}\mu^k, \qquad \text{and} \qquad r := \Phi z^k - t^k - z_0 - \eta^{-1}\nu^k,$$

where $p, q, r \in \mathbb{R}^n$. And we define $\widehat{p} := \mathcal{F}(p)$, $\widehat{q} := \mathcal{F}(q)$, $\widehat{r} := \mathcal{F}(r)$. Then, solving the optimization problem (1.65) is equivalent to minimizing $\widehat{d}$

$$\begin{cases} \text{argmin}_{\widehat{d}} \; \frac{\xi}{2}\left(\|D_1\widehat{d} + \widehat{p}\|^2 + \|D_2\widehat{d} + \widehat{q}\|^2\right) + \frac{\eta}{2}\|H\widehat{d} + \widehat{r}\|^2 \\ \quad\;\; \text{s.t.} \;\; \|\widehat{d}\| \leq \sqrt{n}\Delta_k. \end{cases} \tag{1.66}$$

This trust region subproblem, whose Hessian is diagonal, could be solved by a dozen of classical algorithm, and the corresponding multiplier δ_k is estimated simultaneously. Here, we prefer to use a nearly exact solver which is introduced in the famous book [19, Algorithm 4.4].
After the solution of the trust region subproblem (1.66) is obtained, we compute

$$\widetilde{z}^k = z^k + \mathcal{F}^{-1}\widehat{d}.$$

- Update $y = (s, t)$. Since variables s and t are not coupled. we could update them separately. For the simplicity of the computation, we set the trust region radius Γ_k of this subproblem is large enough. So, the trust region constraints for variables s and t are inactive and the corresponding multiplier $\gamma_k = 0$.
First, we consider the variable s.

$$\begin{aligned} \widetilde{s}^k &= \text{argmin}_s \left\{ |||s|||_1 + (\mu^k)^\top s + \frac{\xi}{2}\|\nabla\widetilde{z}^k - s\|^2 \right\} \\ &= \text{argmin}_s \left\{ |||s|||_1 + \frac{\xi}{2}\|s - (\nabla\widetilde{z}^k - \xi^{-1}\mu^k)\|^2 \right\}. \end{aligned} \tag{1.67}$$

According to the definition (1.63) of TV, the above optimization problem is separable. Recalling that the auxiliary variable $s(= \nabla z) \in \mathbb{R}^{2n}$ and the special structure (1.62) of ∇z, we correspondingly rewrite

$$s =: \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \qquad \text{and} \qquad \nabla\widetilde{z}^k - \xi^{-1}\mu^k =: \begin{pmatrix} \upsilon_1 \\ \upsilon_2 \end{pmatrix},$$

where $s_1, s_2, \upsilon_1, \upsilon_2 \in \mathbb{R}^n$. Let

$$\check{s}^i := \begin{pmatrix} [s_1]_i \\ [s_2]_i \end{pmatrix} \in \mathbb{R}^2 \qquad \text{and} \qquad \check{\upsilon}^i := \begin{pmatrix} [\upsilon_1]_i \\ [\upsilon_2]_i \end{pmatrix} \in \mathbb{R}^2.$$

Then, the separable optimization problem (1.67) is equivalent to the following small problems:

$$\operatorname{argmin}_{\check{s}^i} \left\{ \|\check{s}^i\| + \frac{\xi}{2}\|\check{s}^i - \check{\upsilon}^i\|^2 \right\}, \qquad \text{for } i = 1, \ldots, n.$$

The closed-form solution of this 2D optimization is given by the 2D shrinkage formula [23]:

$$\check{s}^i = \max\left\{ \|\check{\upsilon}^i\| - \frac{1}{\xi}, 0 \right\} \frac{\check{\upsilon}^i}{\|\check{\upsilon}^i\|}, \qquad \text{for } i = 1, \ldots, n,$$

where $0 \cdot (0/0) = 0$ is followed. So $\widetilde{s}^k$ is obtained cheaply.

Second, we turn to the variable t.

$$\begin{aligned} \widetilde{t}^k &= \operatorname{argmin}_t \left\{ \pi_{\odot}(t) + (\nu^k)^\top t + \frac{\eta}{2}\|\Phi\widetilde{z}^k - t - z_0\|^2 \right\} \\ &= \operatorname{argmin}_t \left\{ \pi_{\odot}(t) + \frac{\eta}{2}\|t - (\Phi\widetilde{z}^k - z_0 - \eta^{-1}\nu^k)\|^2 \right\} \\ &= \mathcal{P}_{\odot}(\Phi\widetilde{z}^k - z_0 - \eta^{-1}\nu^k), \end{aligned}$$

where $\mathcal{P}_{\odot}(\cdot)$ is the projective operator onto the ball $\odot$:

$$\mathcal{P}_{\odot}(a) = \begin{cases} a & \text{if } \|a\|^2 \le \sigma, \\ \dfrac{\sqrt{\sigma}}{\|a\|} a & \text{otherwise.} \end{cases}$$

- Update the multiplier $\lambda = (\mu, \nu)$.

$$\begin{aligned} \widetilde{\mu}^k &= \mu^k - \xi(\nabla\widetilde{z}^k - \widetilde{s}^k), \\ \widetilde{\nu}^k &= \nu^k - \eta(\Phi\widetilde{z}^k - \widetilde{t}^k - z_0). \end{aligned}$$

- The correction step is straightforward.

Then, we apply the novel trust region-based splitting method (TRSP) to a T1-weighted magnetic resonance image. Figure 1.1a, which is the true image, shows a horizontal slice of a human brain and Fig. 1.1b illustrates its gradient map

$$\left(\sqrt{[\partial_1 z]_i^2 + [\partial_2 z]_i^2} \right)_{i=1,2,\ldots,n}.$$

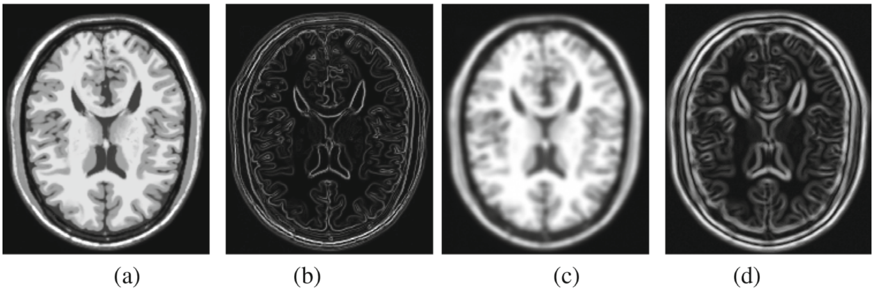


Fig. 1.1 A T1-weighted magnetic resonance image on a horizontal slice of a human brain. The true image (**a**) and its gradient map (**b**). The blurry and noisy image (**c**) and its gradient map (**d**)

A blurry and noisy image and its gradient are shown in Fig. 1.1c and d. Obviously, the degraded image has thick and blurry edges. The TV-based model is an efficient way to impose sharp edges.

Here, we compare the novel TRSP with the basic ADMM (bADM) and the descent ADMM (dADM) [26]. In the first 50 iterations of TRSP, we allow parameters ξ and η in the penalty matrix adjust adaptively [8, 16]. All these algorithms start from the same initial points, and stop when the relative improvement of the objective TV of the restored image is small enough:

$$\frac{\left| \| |\widetilde{z}^k| \| - \| |\widetilde{z}^{k-1}| \| \right|}{\| |\widetilde{z}^k| \|} \leq 10^{-5}.$$

To measure the quality of restored images, we employ the scalar measurement named the signal-to-noise ratio (SNR):

$$\mathrm{SNR} = 10 \log_{10} \frac{\|z_* - \overline{z}_*\|^2}{\|z_* - z\|^2},$$

where z_* is the true image and $\overline{z}_*$ is its mean intensity value. We draw the SNRs versus iterations for bADM, dADM, and TRSP in Fig. 1.2. While the SNR of images obtained by bADM and dADM are respectively 26.5 and 27.2, TRSP returns an image with largest SNR= 27.5 which means that a higher quality image is reconstructed. We illustrate restored images and corresponding gradient maps in Fig. 1.3. However, all the restored images have little difference in visualization.

Finally, we turn to compare the convergence rate of bADM, dADM, and TRSP. The detailed results on the number of iterations and CPU times are shown in Table 1.1. Compared with the basic ADM, the proposed TRSP saves about 59% iterations and 44% CPU times. These results indicate that the trust region-based splitting algorithm is efficient and promising.

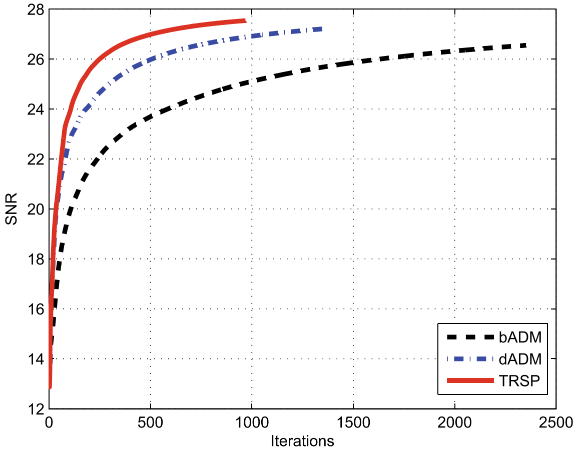


Fig. 1.2 Comparison of SNRs for bADM, dADM and TRSP

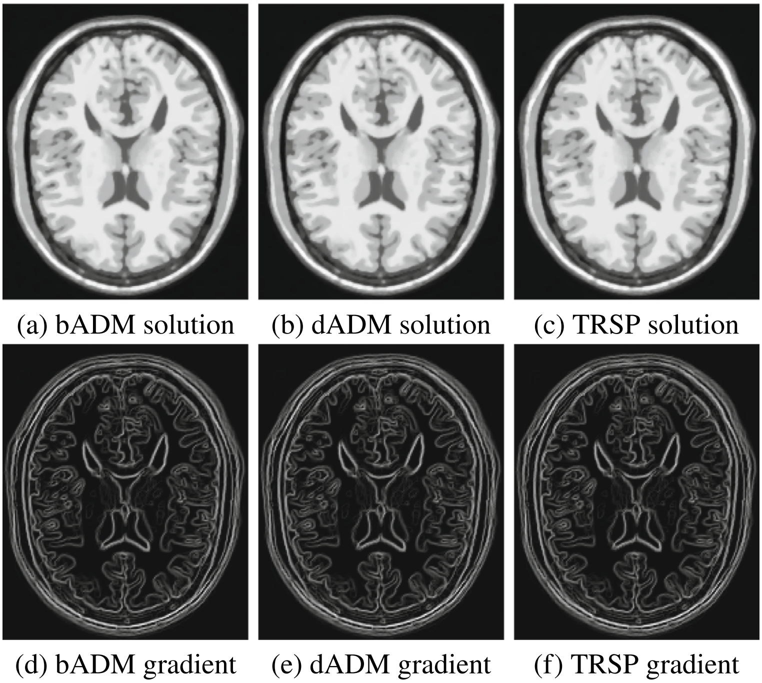


Fig. 1.3 Restored images and their gradient maps

Table 1.1 Iterations and CPU times (second) for the image recovery problem

	bADM	dADM	TRSP
Iterations	2352	1351	969
CPU times	44.7	31.8	24.9

1.5.2 ℓ_1 Regularized Logistic Regression

The image recovery problem has a quadratic objective function, which lacks analysis between TRSP, linearized ADMM (lADM), and majorized ADMM (mADM). In this section, we consider a ℓ_1 regularized logistic regression problem for binary classification in [1, p. 92, (11.1)]. The problem is

$$\min_{w\in\mathbb{R}^n, v\in\mathbb{R}} \sum_{i=1}^{m} \log\left(1+\exp(-b_i(a_i^T w+v))\right)+\mu\|w\|_1, \tag{1.68}$$

with the training set consists of m pairs (a_i, b_i), where $a_i \in \mathbb{R}^n$ is the feature vector and $b_i \in \{-1, 1\}$ is the label. The parameter μ is the regularization parameter.

We generated a problem instance with $m = 2000$ training examples and $n = 200$ features. Each feature vector a_i has a random number of nonzero features which were drawn from a standard normal distribution. The number of nonzero features was generated by a Poisson distribution with mean 40, and the positions of the nonzero features were chosen uniformly at random. We pick a random vector w^* and a random scalar v^* to be the true solution. $w^* \in \mathbb{R}^n$ was generated to have approximately 40 nonzero entries each row, and v^* was sampled from a standard normal distribution. The number of nonzero entries in w^* was generated by a binomial distribution with probability 0.2 with a uniformly random position. The labels b_i were generated by

$$b_i = \operatorname{sign}(a_i^T w^* + v^* + \epsilon_i),$$

where ϵ_i is a random variable drawn from $\mathcal{N}(0, 0.1)$.

As in [1, p. 92, (11.1)], we set $\mu = 0.1\mu_{\max}$, where $\mu_{\max}$ is obtained by $\mu_{\max} = \|A^T\widetilde{b}\|_\infty$, where

$$\widetilde{b}_i = \begin{cases} \theta^{\text{pos}}, & \text{if } b_i = 1, \\ \theta^{\text{neg}}, & \text{if } b_i = -1, \end{cases}$$

where θ^{pos} and θ^{neg} are the fraction of positive and negative labels, respectively.

We fit the ℓ_1 regularized logistic regression model (1.68) into a global consensus problem as

$$\begin{aligned} &\min_{x_i,\, z} \sum_{i=1}^{m} l_i(x_i) + \mu\|z\|_1, \\ &\text{s.t.} \quad x_i = z, \quad i = 1, 2, \ldots, m, \end{aligned} \tag{1.69}$$

where $x_i = (v_i, w_i^T)^T$, $z = (v, w^T)^T$, $v, v_i \in \mathbb{R}$, $w, w_i \in \mathbb{R}^n$, $l_i(x_i) = \log(1 + \exp(-b_i(a_i^T w_i + v_i)))$.

Let

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad y = z, \quad \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{pmatrix},$$

$$f(x) = \textstyle\sum_{i=1}^m l_i(x_i), \quad g(y) = \mu \|y\|_1,$$

$$A = I_{mn}, \quad B = \begin{pmatrix} -I_n \\ -I_n \\ \vdots \\ -I_n \end{pmatrix} \in \mathbb{R}^{mn \times n}, \quad c = 0, \quad H = \xi I_{mn},$$

For TRSP, we give initial points $x^0 = 0$, $y^0 = 0$, $\lambda^0 = 0$ and begin the following loops. Before the iteration, we set proper TOL_x, TOL_y and TOL_λ, and give a guess of L_f and approximate error of Hessian matrix ε.

- Update the variable x in a trust region $\|x - x^k\| \leq \Delta_k$, due to the separability of the objective function, we could update x_i separately.

  $$\widetilde{x}_i^k = \mathrm{argmin}_{\|x_i - x_i^k\| \leq \Delta_{k,i}} \left\{ \frac{\xi}{2} \|x_i - y^k - \xi^{-1}\lambda_i^k\|^2 + \widetilde{l}_i^k(x_i) \right\}.$$

  where $\widetilde{l}_i^k(x_i)$ is the quadratic approximation of $l_i(x_i)$ at x_i^k. According to Sect. 1.3.1, we have the subproblem is equivalent to (1.15). Then we could solve the subproblem by

  $$\widetilde{x}_i^k = (\widetilde{\nabla}^2 l_i(x_i) + (\xi + \delta_{k,i})I_n)^{-1}(\nabla l_i(x_i) + \xi(x_i^k - y^k - \xi^{-1}\lambda_i^k)). \qquad (1.70)$$

  This inverse matrix could be computed under an L-BFGS framework, such as in [11]. If $\|\nabla l_i(x_i) + \xi(x_i^k - y^k - \xi^{-1}\lambda_i^k)\| \leq \mathrm{TOL}_x$, we skip the subproblem and set $\widetilde{x}_i^k = x_i^k$. Otherwise, we need to ensure $\delta_{k,i} \geq L_f \|\widetilde{x}_i^k - x_i^k\| + \varepsilon$ to get $\widetilde{x}_i^k$. Here we let $\delta_k \leq \delta_{k,i}$ for all i such that $\widetilde{x}_i^k \neq x_i^k$. After we get $\widetilde{x}^k$, we update δ_k in a similar way in Algorithm 2, Step 2′. For given boundaries $1 < \eta_1 < \eta_2$, we recall the notation $r_{x,i}^k = L_f \|\widetilde{x}^k - x^k\| + \varepsilon$ and update δ_k by

  $$\delta_{k+1,i} = \begin{cases} 0.5 \max(\delta_{k,i}, (1 + \eta_1) r_{x,i}^k), & \delta_{k,i} \in [\eta_2 r_{x,i}^k, +\infty), \\ \delta_{k,i}, & \delta_{k,i} \in [\eta_1 r_{x,i}^k, \eta_2 r_{x,i}^k), \\ .5\delta_{k,i}, & \delta_{k,i} \in [r_{x,i}^k, \eta_1 r_{x,i}^k). \end{cases}$$

- Update the variable y in an exact solution as the soft-thresholding operator

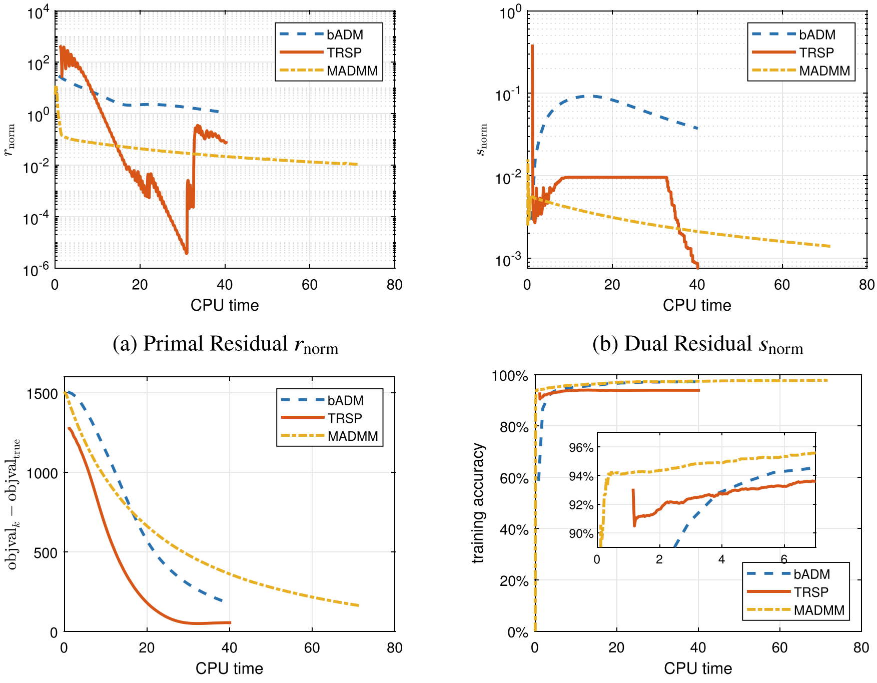
  $$\widetilde{y}^k = \operatorname*{shrinkage}_{\mu\xi/m} \left(\frac{1}{m} \sum_{i=1}^{m} (\widetilde{x}_i^k + \xi^{-1}\lambda^k) \right).$$

Solve the y-subproblem exactly is valid in the TRSP scheme as discussed in Remark 1.3.

- Update the multiplier λ by

$$\widetilde{\lambda}^k = \lambda^k - \xi(\widetilde{x}^k - B\widetilde{y}^k).$$

Here we compare the TRSP with the basic ADMM, with the subproblem solved by the L-BFGS method. All the algorithms start from the same initial points, $x = 0$, $y = 0$, $\lambda = 0$. Through Fig. 1.4a, we observe the convergence rate of the residual norm, which is the residual of the consensus constraint, defined as $r^k_{\text{norm}} := \sqrt{\sum i = 1^m \|x_i^k - z^k\|^2}$, for the three algorithms. The TRSP method demonstrates a significantly faster convergence compared to both the basic ADMM and Majorized iPADMM (MADMM), with the residual norm decreasing more rapidly and smoothly. Figure 1.4b shows the dual residual norm, defined as $s^k_{\text{norm}} := \|z^k - z^{k-1}\|_2$. The TRSP method again exhibits superior performance, converging faster and achieving lower dual residual norms compared to the basic ADMM and MADMM. Figure 1.4c presents the comparison of the objective values among the methods. We calculate



(a) Primal Residual r_{norm}

(b) Dual Residual s_{norm}

(c) Objective Value $\sum_{i=1}^m l_i(x_i) + \mu\|z\|_1$

(d) Training Accuracy

Fig. 1.4 Comparison among TRSP, basic ADMM and majorized iPADMM for ℓ_1 regularized logistic regression

the objective value of the k-th iteration and true solution with $x_i^* = z^* = (v^*, w^{*T})^T$ as

$$\text{objval}_k = \sum_{i=1}^{m} l_i(x_i^k) + \mu\|z^k\|_1, \quad \text{objval}_{\text{true}} = \sum_{i=1}^{m} l_i(x_i^k) + \mu\|z^k\|_1$$

and compare $\text{objval}_k - \text{objval}_{\text{true}}$ for all methods. The TRSP method consistently achieves lower objective values faster than both the basic ADMM and MADMM. Lastly, Fig. 1.4d illustrates the training accuracy for all methods. The training accuracy is defined as the percentage of correctly classified training examples, and is calculated as

$$\text{acc}_{\text{train}}^k = \frac{1}{2m}\|b_{\text{pred}}^k - b\|_1,$$

where $b_{\text{pred}}^k = \text{sign}\left[(1 + \exp(-a_i^T w^k - v^k))^{-1} - 0.5\right]$. All algorithms achieve an accuracy over 90%.

In summary, the numerical experiments clearly demonstrate that TRSP outperforms both the basic ADMM and Majorized iPADMM (MADMM) in terms of convergence rate, dual residual norm reduction, objective value optimization, and training accuracy. The use of L-BFGS for solving subproblems in TRSP contributes to its enhanced performance.

1.6 Conclusion

In this paper, we propose to solve the subproblems of a splitting method in an explicit trust region constraint. As a result, we established an efficient trust region-based splitting method for solving a class of separable convex optimization and variational inequalities. We analyzed the global convergence of the new algorithm under some mild assumptions. Moreover, we proved that the new algorithm enjoys the $O(1/\epsilon)$ convergence rate in an ergodic sense. Numerical experiments showed that the novel trust region-based splitting method is competitive with many existing algorithms for image recovery and logistic regression problems.

References

1. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2010). https://doi.org/10.1561/2200000016
2. Cai, X., Chen, Y., Han, D.: Nonnegative tensor factorizations using an alternating direction method. Front. Math. China **8**(1), 3–18 (2013). https://doi.org/10.1007/s11464-012-0264-8
3. Chen, C., He, B., Yuan, X.: Matrix completion via an alternating direction method. IMA J. Numer. Anal. **32**(1), 227–245 (2012). https://doi.org/10.1093/imanum/drq039

4. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**(1), 33–61 (1998). https://doi.org/10.1137/S1064827596304010
5. Chen, Y., Dai, Y., Han, D., Sun, W.: Positive semidefinite generalized diffusion tensor imaging via quadratic semidefinite programming. SIAM J. Imaging Sci. **6**(3), 1531–1552 (2013). https://doi.org/10.1137/110843526
6. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. SIAM (2000)
7. Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**(1–3), 293–318 (1992). https://doi.org/10.1007/BF01581204
8. Fu, X., He, B.: Self-adaptive projection-based prediction-correction method for constrained variational inequalities. Front. Math. China **5**(1), 3–21 (2010). https://doi.org/10.1007/s11464-009-0045-1
9. Gabay, D.: Applications of the method of multipliers to variational inequalities. In: Fortin, M., Glowinski, R. (eds.) Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems. Studies in Mathematics and its Applications, vol. 15, chap. IX, pp. 299–331. Elsevier (1983)
10. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput. Math. Appl. **2**(1), 17–40 (1976). https://doi.org/10.1016/0898-1221(76)90003-1
11. Gao, G., Florez, H., Vink, J.C., Wells, T.J., Saaf, F., Blom, C.P.: Performance analysis of trust region subproblem solvers for limited-memory distributed BFGS optimization method. Front. Appl. Math. Stat. **7**, 673412 (2021). https://doi.org/10.3389/fams.2021.673412
12. Glowinski, R., Marroco, A.: Sur l'approximation, par éléments finis et la résolution par pénalisation-dualité d'une classe de problémes de Dirichlet non linéaires. Rev. Fr. Autom. Inform. Rech. Oper. **9**(R-2), 41–76 (1975). https://doi.org/10.1051/m2an/197509R200411
13. Goldstein, T., Osher, S.: The split Bregman method for L1-regularized problems. SIAM J. Imaging Sci. **2**(2), 323–343 (2009). https://doi.org/10.1137/080725891
14. Han, D., Lo, H.K.: Solving non-additive traffic assignment problems: A descent method for co-coercive variational inequalities. Eur. J. Oper. Res. **159**(3), 529–544 (2004). https://doi.org/10.1016/S0377-2217(03)00423-5
15. He, B., Liao, L.Z., Han, D., Yang, H.: A new inexact alternating directions method for monotone variational inequalities. Math. Program. **92**(1), 103–118 (2002). https://doi.org/10.1007/s101070100280
16. He, B., Yang, H., Wang, S.: Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. J. Optim. Theory Appl. **106**(2), 337–356 (2000). https://doi.org/10.1023/A:1004603514434
17. He, B., Yuan, X.: On the O(1/n) convergence rate of the Douglas Rachford alternating direction method. SIAM J. Numer. Anal. **52**(2), 700–709 (2012). https://doi.org/10.1137/110836936
18. Ng, M.K., Weiss, P., Yuan, X.: Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. SIAM J. Sci. Comput. **32**(5), 2710–2736 (2010). https://doi.org/10.1137/090774823
19. Nocedal, J., Wright, S.J.: Numerical Optimization. Science Press, Beijing (2006)
20. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D Nonlinear Phenom. **60**(1–4), 259–268 (1992). https://doi.org/10.1016/0167-2789(92)90242-F
21. Sun, W., Yuan, Y.X.: Optimization Theory and Methods: Nonlinear Programming. Springer, New York (2006)
22. Tseng, P.: Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. SIAM J. Control Optim. **29**(1), 119–138 (1991). https://doi.org/10.1137/0329006
23. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. SIAM J. Imaging Sci. **1**(3), 248–272 (2008). https://doi.org/10.1137/080724265

24. Xu, M.: Proximal alternating directions method for structured variational inequalities. J. Optim. Theory Appl. **134**(1), 107–117 (2007). https://doi.org/10.1007/s10957-007-9192-2
25. Yang, J., Zhang, Y.: Alternating direction algorithms for ℓ_1-problems in compressive sensing. SIAM J. Sci. Comput. **31**(1), 250–278 (2011). https://doi.org/10.1137/090777761
26. Ye, C., Yuan, X.: A descent method for structured monotone variational inequalities. Optim. Methods Softw. **22**(2), 329–338 (2007). https://doi.org/10.1080/10556780600552693
27. Yuan, X.: An improved proximal alternating direction method for monotone variational inequalities with separable structure. Comput. Optim. Appl. **49**(1), 17–29 (2011). https://doi.org/10.1007/s10589-009-9293-y
28. Yuan, X.: Alternating direction method for covariance selection models. J. Sci. Comput. **51**(2), 261–273 (2012). https://doi.org/10.1007/s10915-011-9507-1

Chapter 2
Forward-Reflected-Backward Method with Extrapolation and Linesearch for Monotone Inclusion Problems

Tanxing Wang, Heng Zhang, and Xingju Cai

Abstract The extrapolation technique has been widely used to accelerate the forward-reflected-backward method for monotone inclusion problems. This paper considers a new forward-reflected-backward method with extrapolation ($\mathrm{FRB}_{\mathrm{e}}$), which adapts a new extrapolation direction different from the existing acceleration method and uses the latest extrapolation point for the Lipschitz operator. Further, to improve the numerical performance, we propose the linesearch procedure based on the $\mathrm{FRB}_{\mathrm{e}}$ by using only the locally Lipschitz constant. Compared to existing methods, our proposed methods not only cover some classical methods, but can also offer a larger stepsize that does not depend on the global Lipschitz constant. We establish the weak convergence of the proposed methods under mild and standard assumptions. In addition, we conduct some numerical experiments on the lasso problem and the ℓ_1 regularized logistic regression problem to demonstrate the advantage of the proposed methods.

Keywords Forward-reflected-backward method · Monotone inclusion · Extrapolation · Linesearch · Weak convergence

2.1 Introduction

In this paper, we propose two methods for finding a zero in the sum of two monotone operators in a real Hilbert space $\mathcal{H}$. Specifically, we consider the monotone inclusion problem:

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in (A + B)x, \tag{2.1}$$

T. Wang · H. Zhang · X. Cai (✉)
School of Mathematical Sciences, Ministry of Education Key Laboratory of NSLSCS, Nanjing Normal University, Nanjing, China
e-mail: caixingju@njnu.edu.cn

T. Wang
e-mail: wangtanxing2021@163.com

H. Zhang
e-mail: 1174848718@qq.com

X. Wang et al. (eds.), *Splitting Optimization*, ICIAM2023 Springer Series 2,
https://doi.org/10.1007/978-981-96-8574-5_2

where $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$ and $B\colon \mathcal{H} \to \mathcal{H}$ are two (maximally) monotone operators with B (locally) Lipschitz continuous such that $(A+B)^{-1}(0) \neq \emptyset$. The monotone inclusion problem (2.1) has drawn much attention because it provides a broad unifying framework for variational inequalities, convex minimization problems, split feasibility problems and equilibrium problems, and has been applied to solve various real-world problems from machine learning, signal processing and image restoration, see [1, 2, 5, 10, 13, 27, 32].

A popular model, which can be formulated under the monotone inclusion, is the following optimization problem of the sum of two functions:

$$\min_{x\in\mathcal{H}} f(x) + g(x), \tag{2.2}$$

where $f\colon \mathcal{H} \to (-\infty, \infty]$ is proper, closed, convex, and $g\colon \mathcal{H} \to \mathcal{R}$ is convex with (locally) Lipschitz continuous gradient denoted as ∇g. The solutions to this minimization problem are precisely the points $x \in \mathcal{H}$ which satisfy the first-order optimality condition:

$$0 \in \partial f(x) + \nabla g(x),$$

where ∂f is the subdifferential of f. In this case, the optimization problem is equivalent to the monotone inclusion problem (2.1) with $A = \partial f$ and $B = \nabla g$.

In recent years, there is a growing interest in the design and analysis of splitting algorithms for solving the monotone inclusion problem (2.1). In 1979, Passty [24] and Lions et al. [16] proposed the following forward-backward (FB) method, which combines one forward evaluation of B and one backward evaluation of A in each iteration. More precisely, the method generates a sequence $\{x^k\}_{k\in N}$ according to

$$x^{k+1} = J_{\lambda A}(x^k - \lambda B x^k),$$

where $\lambda \in (0, 2/L)$, L is the Lipschitz continuity modulus of B and $J_{\lambda A} = (I + \lambda A)^{-1}$. Under the assumption that the operator $B\colon \mathcal{H} \to \mathcal{H}$ is $1/L$-cocoercive, the authors proved that each bounded sequence generated by FB converges weakly to a solution.

It is worthing that coercivity of an operator is a stronger property than Lipschitz continuity and hence can be difficult to satisfy for a general monotone inclusion problem. In 2000, Tseng [33] proposed a modification of the forward-backward method to relax the coercivity assumption, which is known as Tseng's method or the forward-backward-forward (FBF) method. It only requires the Lipschitzness of B at the expense of an additional forward evaluation. Applied to (2.1), Tseng's method generates sequences $\{x^k\}_{k\in N}$ according to

$$\begin{cases} y^k = J_{\lambda A}(x^k - \lambda B x^k), \\ x^{k+1} = y^k - \lambda B y^k + \lambda B x^k, \end{cases}$$

where $\lambda \in (0, 1/L)$ and converges weakly to a solution.

Under the same assumptions as Tseng's method [33], Malitsky and Tam [18] proposed the forward-reflected-backward (FRB) method for solving (2.1), which requires only one forward evaluation per iteration instead of two. The corresponding update scheme reads as

$$x^{k+1} = J_{\lambda A}(x^k - \lambda Bx^k - \lambda(Bx^k - Bx^{k-1})),$$

and converges weakly if B is L-Lipschitz and the stepsize is chosen to satisfy $\lambda < 1/2L$.

As FB, FBF, and FRB are the first-order splitting algorithms, acceleration techniques are of great practical interest to improve the performance of these algorithms. One simple and efficient strategy is to incorporate the inertial technique, which was first introduced by Polyak [25] in 1964. Recently, there are increasing interests in studying inertial type algorithms, such as inertial forward-backward splitting methods for separable optimization problems under the nonconvex setting [20] and strongly convex setting [19], inertial versions of the Douglas-Rachford operator splitting method [4], inertial forward-backward-forward method [3] based on Tseng's approach [33] and general inertial proximal point method for the mixed variational inequality problem [9]. Specially, motivated by the idea of the heavy-ball method [25], Malitsky and Tam [18] proposed an inertial forward-reflected-backward splitting algorithm (iFRB) for solving problem (2.1). The corresponding update scheme reads as

$$\begin{cases} y^k = x^k + \alpha(x^k - x^{k-1}), \\ x^{k+1} = J_{\lambda A}(y^k - \lambda Bx^k - \lambda(Bx^k - Bx^{k-1})), \end{cases} \tag{2.3}$$

where $\alpha \in [0, 1/3)$, $0 < \lambda L < (1 - 3\alpha)/2$. It is worth noting that the iteration scheme (2.3) does not use the latest extrapolation point y^k for the calculation of operator B. In view of this, in this paper, we propose a new forward-reflected-backward method with extrapolation (FRB$_\mathrm{e}$) for solving the monotone inclusion problem (2.1). Further, taking into account that the global Lipschitz constant is not easily obtained and it can often lead to overconservative stepsize, we propose a forward-reflected-backward method with extrapolation and linesearch (FRB$_\mathrm{el}$). Under mild and standard assumptions, we establish the weak convergence of the sequences generated by the proposed methods. Numerical experiments on the lasso problem and the ℓ_1 regularized logistic regression problem to demonstrate the advantage of the proposed methods.

The rest of this paper is organized as follows. In Sect. 2.2, we recall some definitions and results for further analysis. In Sect. 2.3, we introduce FRB$_\mathrm{e}$ and FRB$_\mathrm{el}$ and investigate their convergence. Some numerical results are reported in Sect. 2.4. Finally, we draw some conclusions in Sect. 2.5.

2.2 Preliminaries

In this section, we summarize some necessary notations and results for further analysis.

Let $\mathcal{H}$ be a real Hilbert space with the inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\| \cdot \|$, and use $\| \cdot \|_1$ to denote the ℓ_1 norm. Let $\{x^k\}_{k\in N}$ be a sequence in $\mathcal{H}$. We write $x^k \rightharpoonup x$ to stand for the weak convergence of the sequence $\{x^k\}_{k\in N}$ to $x \in \mathcal{H}$ as $k \to +\infty$. Let $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$ be a set-valued operator. The graph of $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$, denoted by gph$(A)$ is defined by

$$\mathrm{gph}(A) := \{(x, u) \in \mathcal{H} \times \mathcal{H} \mid \S \in \mathcal{H}, \sqcap \in \mathscr{A}\S\}.$$

Let us review some basic definitions and concepts.

Definition 2.1 ([2]) We say that the operator A is

(i) monotone if

$$\langle x - y, u - v \rangle \geqslant 0, \ \forall\, (x, u),\ (y, v) \in \mathrm{gph}(A);$$

(ii) maximal monotone if it is monotone and gph$(A)\supset$ gph(B) where B is any other monotone operator.

Remark 2.1 An important property of a maximal monotone operator $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$ is that, if a pair $(x, u) \in \mathcal{H} \times \mathcal{H}$ and $\langle x - y, u - v \rangle \geqslant 0$ for all $(y, v) \in \mathrm{gph}(A)$, then $u \in A(x)$.

For a given maximal monotone operator A, the resolvent

$$J_{\lambda A}(x) = (I + \lambda A)^{-1}(x), \tag{2.4}$$

for $x \in \mathcal{H}$ and $\lambda > 0$ is a single-valued mapping, where I is the identity operator on $\mathcal{H}$. Furthermore, $\| J_{\lambda A}(x) - J_{\lambda A}(y)\| \leqslant \|x - y\|$ for all $x, y \in \mathcal{H}$. A single-valued operator $B\colon \mathcal{H} \to \mathcal{H}$ is L-Lipschitz continuous if there exists $L > 0$ such that

$$\|Bx - By\| \leqslant L\|x - y\|, \ \forall\, x,\ y \in \mathcal{H},$$

and B is β-cocoercive if there exists $\beta > 0$ such that

$$\langle Bx - By, x - y \rangle \geqslant \beta\|Bx - By\|^2, \ \forall\, x,\ y \in \mathcal{H}.$$

Let $C(x, \delta)$ denote the open ball in $\mathcal{H}$ with centra x and radius $\delta > 0$. Recall that $B\colon \mathcal{H} \to \mathcal{H}$ is called locally Lipschitz provided for each $x \in \mathcal{H}$, there exists $\delta_x > 0$ such that the restriction of B to $C(x, \delta_x)$ is Lipschitz. The following result gives the maximal monotonicity of the sum of two monotone operators.

Lemma 2.1 ([23]) *Suppose $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$ is a maximal monotone operator and $B\colon \mathcal{H} \to \mathcal{H}$ is a Lipschitz continuous and monotone operator. Then $A + B$ is a maximal monotone operator.*

We use the following identities in our convergence analysis.

Lemma 2.2 ([2, 14]) *For any vectors a, b, c and $d \in \mathcal{H}$, the following identity holds*

$$\langle a - b, c - d\rangle = \frac{1}{2}(\|a - d\|^2 - \|a - c\|^2) + \frac{1}{2}(\|b - c\|^2 - \|b - d\|^2).$$

Lemma 2.3 ([2]) *For any vectors x and $y \in \mathcal{H}$, $\alpha \in \mathcal{R}$, then*

$$\|\alpha x + (1 - \alpha)y\|^2 = \alpha\|x\|^2 + (1 - \alpha)\|y\|^2 - \alpha(1 - \alpha)\|x - y\|^2.$$

The lemma below is quite well known and plays an important role in proving weak convergence of sequences in a Hilbert space.

Lemma 2.4 ([21]) *Let K be a nonempty subset of a Hilbert space $\mathcal{H}$ and let $\{x^k\}_{k\in N}$ be bounded sequence in $\mathcal{H}$. Assume the following two conditions are satisfied:*

(i) $\lim_{k\to\infty}\|x - x^k\|$ *exists for each $x \in K$;*
(ii) every weak cluster point of $\{x^k\}_{k\in N}$ belongs to K.

Then $\{x^k\}_{k\in N}$ is weakly converges to a point in K.

Lemma 2.5 ([2]) *Let $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone and let $x \in \mathcal{H}$. Set*

$$(\forall\, \lambda \in \mathcal{R}_{++})\; x_\lambda = J_{\lambda A}x.$$

Then $x_\lambda \to P_{\overline{\mathrm{dom} A}}(x)$ as $\lambda \downarrow 0$, and exactly one of the following holds:

(i) $\mathrm{zer}A \neq 0$ *and $x_\lambda \to P_{\mathrm{zer}A}$ as $\lambda \uparrow +\infty$;*
(ii) $\mathrm{zer}A = 0$ *and $||x_\lambda|| \to \infty$ as $\lambda \uparrow +\infty$.*

2.3 Algorithms and Convergence Analysis

In this section, we first introduce a forward-reflected-backward method with extrapolation (FRB$_\mathrm{e}$) for solving the monotone inclusion problem (2.1), described in Algorithm 1 in Sect. 2.3.1 This method adapts a new extrapolation direction different from the existing acceleration method [5, 7, 8, 11, 12, 22, 29–31] and uses the latest extrapolation point for operator B. Then, we establish the convergence results of FRB$_\mathrm{e}$ in Sect. 2.3.2 Furthermore, we incorporate a linesearch procedure into the FRB$_\mathrm{e}$ method, named forward-reflected-backward method with extrapolation and linesearch (FRB$_\mathrm{el}$) in Sect. 2.3.3 To this end, we make some necessary assumptions on problem (2.1) below, which will be used throughout this section.

Assumption 2.1 (i) $A : \mathcal{H} \rightrightarrows \mathcal{H}$ is a set-valued maximal monotone operator;
(ii) $B : \mathcal{H} \to \mathcal{H}$ is a single-valued monotone operator and L-Lipschitz continuous;
(iii) The solution set $(A + B)^{-1}(0)$ of inclusion problem (2.1) is nonempty.

Note that points (i), (ii), and (iii) in Assumption 2.1 are the common assumptions for monotone inclusion problem, which are consistent with the forward-backward-forward method in [33], the forward-reflected-backward method in [18] and the double inertial forward-backward-forward method in [34].

2.3.1 *Proposed* FRB$_\mathrm{e}$ *for Solving Inclusion Problem*

Motivated by the idea of forward-reflected-backward method in [18] and the inertial algorithms in [5, 7, 8, 11, 12, 22, 29–31], we propose a forward-reflected-backward method with extrapolation, denoted as FRB$_\mathrm{e}$, in Algorithm 1. Compared with existing algorithms, our algorithm provides more flexibility in choosing parameters, which would potentially enhance the algorithm's performance.

Algorithm 1 Forward-reflected-backward method with extrapolation (FRB$_\mathrm{e}$)

```
1: Choose $\lambda_0 > 0$, $\epsilon > 0$ which are close to 0, $\alpha \in [0, 1)$. Set $x^0 = y^{-1} \in \mathcal{H}$.
2: For $k = 0, 1 \cdots$, compute
```

$$\begin{cases} y^k = x^k + \alpha(x^k - y^{k-1}), \\ x^{k+1} = J_{\lambda_k A}(y^k - \lambda_k By^k - \lambda_{k-1}(Bx^k - By^{k-1})), \end{cases} \tag{2.5}$$

where the stepsize sequence $\{\lambda_k\}_{k \in N}$ is nondecreasing and satisfies

$$\lambda_0 \leqslant \lambda_k \leqslant \frac{1 - \alpha - \epsilon}{L(\alpha^2 + 2\alpha + 2)}. \tag{2.6}$$

Remark 2.2 FRB$_\mathrm{e}$ can reduce to some classical algorithms when the operators and the parameters take specific values, such as proximal point algorithm [28], Popov's method [26], projected reflected gradient method [17], forward-reflected-backward method [18], and so on. For simplicity, we only consider the fixed stepsize case, i.e., there exists $\lambda \in (0, (1 - \alpha)/L(\alpha^2 + 2\alpha + 2))$ such that $\lambda_k = \lambda$ for all k. In this case, FRB$_\mathrm{e}$ can be expressed compactly as

$$\begin{cases} y^k = x^k + \alpha(x^k - y^{k-1}), \\ x^{k+1} = J_{\lambda A}(y^k - \lambda By^k - \lambda(Bx^k - By^{k-1})). \end{cases} \tag{2.7}$$

(i) If $\alpha = 0$, FRB$_\mathrm{e}$ reduce to the forward-reflected-backward method [18], that is,

$$x^{k+1} = J_{\lambda A}(x^k - \lambda Bx^k - \lambda(Bx^k - Bx^{k-1})).$$

(ii) If $\alpha = 0$ and $B = 0$, then FRB$_\mathrm{e}$ simplifies to the proximal point algorithm [14, 28]. That is,

$$x^{k+1} = J_{\lambda A}(x^k).$$

(iii) If $\alpha = 0$, $A = N_C$ is the normal cone to a set C, and B is an affine operator, then FRB$_\mathrm{e}$ can be expressed as

$$x^{k+1} = P_C(x^k - \lambda B(2x^k - x^{k-1})),$$

which coincides with the projected reflected gradient method [17] for VIs.

(iv) If $\alpha = 0$ and $A = N_{\mathcal{H}} = 0$, then FRB$_\mathrm{e}$ method (2.7) becomes

$$x^{k+1} = x^k - 2\lambda Bx^k + \lambda Bx^{k-1}. \tag{2.8}$$

It is worth noting that the iteration scheme (2.8) can be expressed as the two-step recursion

$$\begin{cases} y^{k+1} = y^k - \lambda Bx^k, \\ x^{k+1} = y^{k+1} - \lambda Bx^k. \end{cases}$$

This is exactly Popov's algorithm [26] for unconstrained VIs. Furthermore, in the GANs literature, (2.8) is also known to be equivalent to the optimistic gradient method. For details, see the discussion in [15].

2.3.2 *Convergence Analysis of* FRB$_\mathrm{e}$

In this section, we investigate the weak convergence of the proposed FRB$_\mathrm{e}$ with the help of Assumption 2.1. To this aim, we consider the metric function

$$\begin{aligned} E_k := {} & \frac{1}{1+\alpha}\|x^\star - y^k\|^2 + 2\lambda_{k-1}\langle x^\star - x^k, Bx^k - By^{k-1}\rangle \\ & + \lambda_{k-1}L(\alpha^2 + \alpha + 1)\|x^k - y^{k-1}\|^2, \end{aligned} \tag{2.9}$$

where $x^\star \in (A+B)^{-1}(0)$, i.e., the element of the solution set. In the following, we show that the sequence $\{E_k\}_{k\in N}$ is monotonically nonincreasing.

Lemma 2.6 *Suppose that Assumption 2.1 holds. Let $\{x^k\}_{k\in N}$ be the sequence generated by Algorithm 1. Then, the sequence $\{E_k\}_{k\in N}$ is monotonically nonincreasing. In particular, for any $k \geqslant 0$, it holds that*

$$E_{k+1} \leqslant E_k - C\|x^{k+1} - y^k\|^2, \tag{2.10}$$

where $C = (1-\alpha) - \lambda_k L(\alpha^2 + 2\alpha + 2) > 0$.

Proof Using the definition of the resolvent in (2.4) and the x^{k+1} in (2.5), we obtain

$$x^{k+1} - y^k + \lambda_k By^k + \lambda_{k-1}(Bx^k - By^{k-1}) \in -\lambda_k Ax^{k+1}. \tag{2.11}$$

By the monotonicity of A, we have

$$\langle x^\star - x^{k+1}, \lambda_k Ax^\star - \lambda_k Ax^{k+1}\rangle \geqslant 0. \tag{2.12}$$

Pick $x^\star \in (A+B)^{-1}(0)$, then

$$-\lambda_k Bx^\star \in \lambda_k Ax^\star. \tag{2.13}$$

Plugging (2.11) and (2.13) into (2.12), we have

$$\langle x^\star - x^{k+1}, x^{k+1} - y^k + \lambda_k By^k + \lambda_{k-1}(Bx^k - By^{k-1}) - \lambda_k Bx^\star\rangle \geqslant 0.$$

Rearranging the above inequality gives

$$\begin{aligned} 0 \leqslant & 2\langle x^\star - x^{k+1}, x^{k+1} - y^k\rangle + 2\lambda_k\langle x^\star - x^{k+1}, By^k - Bx^\star\rangle \\ & + 2\lambda_{k-1}\langle x^\star - x^{k+1}, Bx^k - By^{k-1}\rangle \\ = & \|x^\star - y^k\|^2 - \|x^\star - x^{k+1}\|^2 - \|x^{k+1} - y^k\|^2 \\ & + 2\lambda_k\langle x^\star - x^{k+1}, By^k - Bx^{k+1}\rangle + 2\lambda_k\langle x^\star - x^{k+1}, Bx^{k+1} - Bx^\star\rangle \\ & + 2\lambda_{k-1}\langle x^\star - x^k, Bx^k - By^{k-1}\rangle + 2\lambda_{k-1}\langle x^k - x^{k+1}, Bx^k - By^{k-1}\rangle \\ \leqslant & \|x^\star - y^k\|^2 - \|x^\star - x^{k+1}\|^2 - \|x^{k+1} - y^k\|^2 \\ & + 2\lambda_k\langle x^\star - x^{k+1}, By^k - Bx^{k+1}\rangle + 2\lambda_{k-1}\langle x^\star - x^k, Bx^k - By^{k-1}\rangle \\ & + \lambda_{k-1}L\|x^{k+1} - x^k\|^2 + \lambda_{k-1}L\|x^k - y^{k-1}\|^2, \end{aligned} \tag{2.14}$$

where the equality follows from Lemma 2.2 and the second inequality holds by using the monotonicity of B, the Lipschitz continuity of B, and the Cauchy inequality $2ab \leqslant a^2 + b^2$ with $a = \|x^k - x^{k+1}\|$ and $b = \|x^k - y^{k-1}\|$.
Therefore, we have

$$\begin{aligned} & \|x^\star - x^{k+1}\|^2 + 2\lambda_k\langle x^\star - x^{k+1}, Bx^{k+1} - By^k\rangle + \|x^{k+1} - y^k\|^2 \\ \leqslant & \|x^\star - y^k\|^2 + 2\lambda_{k-1}\langle x^\star - x^k, Bx^k - By^{k-1}\rangle + \lambda_{k-1}L\|x^{k+1} - x^k\|^2 \\ & + \lambda_{k-1}L\|x^k - y^{k-1}\|^2. \end{aligned} \tag{2.15}$$

From (2.5), we obtain

$$x^{k+1} = \frac{1}{1+\alpha}y^{k+1} + \frac{\alpha}{1+\alpha}y^k, \tag{2.16}$$

$$y^{k+1} - y^k = (1+\alpha)(x^{k+1} - y^k), \tag{2.17}$$

$$y^k - x^k = \alpha(x^k - y^{k-1}). \tag{2.18}$$

Thus, it follows from (2.16), (2.17), (2.18) and Lemma 2.3, we have

$$\begin{aligned}
&\|x^\star - x^{k+1}\|^2\\
=&\left\|x^\star - \left(\frac{1}{1+\alpha}y^{k+1} + \frac{\alpha}{1+\alpha}y^k\right)\right\|^2\\
=&\left\|\frac{1}{1+\alpha}(x^\star - y^{k+1}) + \frac{\alpha}{1+\alpha}(x^\star - y^k)\right\|^2\\
=&\frac{1}{1+\alpha}\left\|x^\star - y^{k+1}\right\|^2 + \frac{\alpha}{1+\alpha}\left\|x^\star - y^k\right\|^2 - \frac{\alpha}{(1+\alpha)^2}\left\|y^{k+1} - y^k\right\|^2\\
=&\frac{1}{1+\alpha}\left\|x^\star - y^{k+1}\right\|^2 + \frac{\alpha}{1+\alpha}\left\|x^\star - y^k\right\|^2 - \alpha\left\|x^{k+1} - y^k\right\|^2,
\end{aligned} \tag{2.19}$$

and

$$\begin{aligned}
&\|x^{k+1} - x^k\|^2\\
=&\|x^{k+1} - y^k + y^k - x^k\|^2 = \|x^{k+1} - y^k + \alpha(x^k - y^{k-1})\|^2\\
=&\|x^{k+1} - y^k\|^2 + \alpha^2\|x^k - y^{k-1}\|^2 + 2\alpha\langle x^{k+1} - y^k, x^k - y^{k-1}\rangle\\
\leqslant&(1+\alpha)\|x^{k+1} - y^k\|^2 + (\alpha^2+\alpha)\|x^k - y^{k-1}\|^2,
\end{aligned} \tag{2.20}$$

where the last inequality follows from Cauchy-Schwarz and Young inequalities.

Substituting (2.19) and (2.20) into (2.15), we get

$$\begin{aligned}
&\frac{1}{1+\alpha}\|x^\star - y^{k+1}\|^2 + 2\lambda_k\langle x^\star - x^{k+1}, Bx^{k+1} - By^k\rangle + (1-\alpha)\|x^{k+1} - y^k\|^2\\
\leqslant&\frac{1}{1+\alpha}\|x^\star - y^k\|^2 + 2\lambda_{k-1}\langle x^\star - x^k, Bx^k - By^{k-1}\rangle + \lambda_{k-1}L\|x^k - y^{k-1}\|^2\\
&+ \lambda_{k-1}L\|x^{k+1} - x^k\|^2\\
\leqslant&\frac{1}{1+\alpha}\|x^\star - y^k\|^2 + 2\lambda_{k-1}\langle x^\star - x^k, Bx^k - By^{k-1}\rangle + \lambda_{k-1}L\|x^k - y^{k-1}\|^2\\
&+ \lambda_{k-1}L(1+\alpha)\|x^{k+1} - y^k\|^2 + \lambda_{k-1}L(\alpha^2+\alpha)\|x^k - y^{k-1}\|^2.
\end{aligned} \tag{2.21}$$

Combining the definition of $\{E_k\}_{k\in N}$ in (2.9) and (2.21), we obtain

$$\begin{aligned}
&E_{k+1}\\
\leqslant&E_k - ((1-\alpha) - \lambda_k L(\alpha^2+\alpha+1) - \lambda_{k-1}L(\alpha+1))\|x^{k+1} - y^k\|^2\\
\leqslant&E_k - ((1-\alpha) - \lambda_k L(\alpha^2+2\alpha+2))\|x^{k+1} - y^k\|^2\\
=&E_k - C\|x^{k+1} - y^k\|^2,
\end{aligned}$$

where the second inequality holds follows from the stepsize sequence $\{\lambda_k\}_{k\in N}$ is nondecreasing. Therefore, we get assertion (2.10). In addition, it follows from (2.6) that

$$C = (1-\alpha) - \lambda_k L(\alpha^2 + 2\alpha + 2) > 0.$$

Therefore, the sequence $\{E_k\}_{k\in N}$ is monotonically nonincreasing. □

Theorem 2.1 *The sequence $\{x^k\}_{k\in N}$ generated by Algorithm 1 converges weakly to a point in $(A+B)^{-1}(0)$ when Assumption 2.1 is satisfied.*

Proof We first show that sequence $\{E_k\}_{k\in N}$ is bounded. It follows from (2.10) that

$$\begin{aligned} E_{k+1} \leqslant & E_k - C\|x^{k+1} - y^k\|^2 \\ \leqslant & E_k \leqslant \ldots \leqslant E_0 = \frac{1}{1+\alpha}\left\|x^\star - y^0\right\|^2 < +\infty. \end{aligned} \tag{2.22}$$

Therefore, the sequence $\{E_k\}_{k\in N}$ has an upper bound. And we will next prove that it has a lower bound. Indeed, we can find that

$$\begin{aligned}
&2\lambda_k\langle x^\star - x^{k+1}, Bx^{k+1} - By^k\rangle \\
\geqslant & -2\lambda_k L\|x^\star - x^{k+1}\|\|x^{k+1} - y^k\| \\
\geqslant & -\lambda_k L(\|x^\star - x^{k+1}\|^2 + \|x^{k+1} - y^k\|^2) \\
= & -\lambda_k L\left(\frac{1}{1+\alpha}\left\|x^\star - y^{k+1}\right\|^2 + \frac{\alpha}{1+\alpha}\left\|x^\star - y^k\right\|^2 + (1-\alpha)\left\|x^{k+1} - y^k\right\|^2\right) \\
= & -\lambda_k L\left(\frac{1}{1+\alpha}\|x^\star - y^{k+1}\|^2 + \frac{\alpha}{1+\alpha}\|x^\star - y^{k+1} + y^{k+1} - y^k\|^2\right) \\
& -\lambda_k L(1-\alpha)\|x^{k+1} - y^k\|^2 \\
\geqslant & -\lambda_k L\left(\frac{1}{1+\alpha}\|x^\star - y^{k+1}\|^2 + \frac{3\alpha}{1+\alpha}\|x^\star - y^{k+1}\|^2 + \frac{3\alpha}{2(1+\alpha)}\|y^{k+1} - y^k\|^2\right) \\
& -\lambda_k L(1-\alpha)\|x^{k+1} - y^k\|^2 \\
= & -\lambda_k L\left(\frac{1}{1+\alpha}\|x^\star - y^{k+1}\|^2 + \frac{3\alpha}{1+\alpha}\|x^\star - y^{k+1}\|^2\right) \\
& -\lambda_k L\left(\frac{3\alpha(1+\alpha)}{2}\|x^{k+1} - y^k\|^2 + (1-\alpha)\|x^{k+1} - y^k\|^2\right) \\
= & -\lambda_k L\frac{1+3\alpha}{1+\alpha}\|x^\star - y^{k+1}\|^2 - \lambda_k L\left(\frac{3\alpha^2}{2} + \frac{\alpha}{2} + 1\right)\|x^{k+1} - y^k\|^2,
\end{aligned} \tag{2.23}$$

where the first inequality follows from the Cauchy-Schwarz inequality and the Lipschitz continuity of B, the second inequality follows from the Young inequality, the third inequality follows from the Cauchy-Schwarz inequality $ab \leqslant \frac{\beta}{2}a^2 + \frac{1}{2\beta}b^2$, $\beta > 0$ with $a = \|x^\star - y^{k+1}\|$, $b = \|y^{k+1} - y^k\|$ and $\beta = 2$, and the first equality and the third equality follows from (2.19) and (2.17), respectively.

We then obtain from (2.6) and (2.23) that

$$\begin{aligned} & E_{k+1} \\ =& \frac{1}{1+\alpha}\|x^\star - y^{k+1}\|^2 + 2\lambda_k\langle x^\star - x^{k+1}, Bx^{k+1} - By^k\rangle \\ & + \lambda_k L(\alpha^2+\alpha+1)\|x^{k+1}-y^k\|^2 \\ \geqslant & \left(\frac{1}{1+\alpha} - \lambda_k L\frac{1+3\alpha}{1+\alpha}\right)\|x^\star - y^{k+1}\|^2 - \lambda_k L\frac{\alpha(\alpha-1)}{2}\|x^{k+1}-y^k\|^2 \geqslant 0, \end{aligned} \tag{2.24}$$

we know that the sequence $\{E_k\}_{k\in N}$ is lower bounded. So we can get $\{E_k\}_{k\in N}$ is bounded. Then, summing up (2.10) from $k = 0, \cdots, N$, it yields

$$\sum_{k=0}^{N} C\|x^{k+1} - y^k\|^2 \leqslant \sum_{k=0}^{N}(E_k - E_{k+1}) = E_0 - E_{N+1} \leqslant E_0 < +\infty,$$

Therefore, we can deduce that

$$\lim_{k\to+\infty}\|x^{k+1} - y^k\| = 0. \tag{2.25}$$

One can then obtain from (2.22) and (2.24) that $\{y^k\}_{k\in N}$ is bounded. Similarly, by the definition of x^{k+1} in (2.16), we have that $\{x^k\}_{k\in N}$ is also bounded. Then $\{x^k\}_{k\in N}$ has a weakly convergent subsequence $\{x^{k_j}\}_{j\in N}$ such that $\{x^{k_j}\}_{j\in N}$ converges weakly to $x^\infty \in \mathcal{H}$.

Multiplying both sides by $-1/\lambda_k$ and adding Bx^{k+1} to both sides in (2.11) and choosing $k = k_j - 1$, we have

$$-\frac{1}{\lambda_{k_j-1}}(x^{k_j} - y^{k_j-1}) + Bx^{k_j} - By^{k_j-1} - \frac{\lambda_{k_j-2}}{\lambda_{k_j-1}}(Bx^{k_j-1} - By^{k_j-2}) \in (A+B)x^{k_j}. \tag{2.26}$$

Since $\lim_{k\to+\infty}\|x^{k+1} - y^k\| = 0$ and B is Lipschitz continuous, we have $Bx^{k_j} - By^{k_j-1} \to 0$, as $j \to +\infty$. By Lemma 2.1, we have that $A + B$ is maximal monotone. Therefore, the graph of $A + B$ is demiclosed. From (2.6), we obtain $\{\lambda_k\}_{k\in N}$ is bounded and far away from zero. Further, passing to the limit in (2.26), we have $0 \in (A + B)x^\infty$. To show that $\{x^k\}_{k\in N}$ is weakly convergent, first note that, by the boundedness of $\{E_k\}_{k\in N}$ and (2.10), we deduce the existence of the limit

$$\begin{aligned} \lim_{k\to+\infty} & \frac{1}{1+\alpha}\|x^\star - y^{k+1}\|^2 + 2\lambda_k\langle x^\star - x^{k+1}, Bx^{k+1} - By^k\rangle \\ & + \lambda_k(\alpha^2+\alpha+1)\|x^{k+1} - y^k\|^2. \end{aligned}$$

Further, it follows from (2.5), we know

$$\lim_{k\to+\infty} \frac{1}{1+\alpha}\|x^\star - x^{k+1}\|^2 + \frac{\alpha^2}{1+\alpha}\|x^{k+1} - y^k\|^2 \\ - \frac{2\alpha}{1+\alpha}\langle x^\star - x^{k+1}, x^{k+1} - y^k\rangle + 2\lambda_k\langle x^\star - x^{k+1}, Bx^{k+1} - By^k\rangle \\ + \lambda_k(\alpha^2 + \alpha + 1)\|x^{k+1} - y^k\|^2$$

exists. Let

$$F_k = -2\lambda_k\langle x^\star - x^{k+1}, Bx^{k+1} - By^k\rangle - \lambda_k(\alpha^2 + \alpha + 1)\|x^{k+1} - y^k\|^2,$$

and

$$G_k = \frac{\alpha^2}{1+\alpha}\|x^{k+1} - y^k\|^2 - \frac{2\alpha}{1+\alpha}\langle x^\star - x^{k+1}, Bx^{k+1} - By^k\rangle.$$

Since $Bx^{k+1} - By^k \to 0$, as $k \to +\infty$, $x^{k+1} - y^k \to 0$, as $k \to +\infty$ and $\{x^k\}_{k\in N}$ and $\{\lambda_k\}_{k\in N}$ are bounded, we have that $F_k \to 0, k \to +\infty$ and $G_k \to 0, k \to +\infty$. Then

$$\frac{1}{1+\alpha}\|x^\star - x^{k+1}\|^2 = E_k - G_k + F_k.$$

Since $\lim_{k\to+\infty} E_k$ exists and $\frac{1}{1+\alpha} > 0$, we then have that $\lim_{k\to+\infty}\|x^\star - x^k\|$ exists. By Lemma 2.4, we conclude that $\{x^k\}_{k\in N}$ converges weakly to a point in $(A+B)^{-1}(0)$. The proof is complete. □

Theorem 2.1 has an immediate result when the steps of the sequence $\{\lambda_k\}_{k\in N}$ are constant, and we obtain the following corollary.

Corollary 2.1 *Let $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone, let $B\colon \mathcal{H} \to \mathcal{H}$ be monotone and L-*Lipschitz *and suppose that $(A+B)^{-1}(0) \neq \emptyset$. Choose $\lambda \in \left(0, \frac{1-\alpha}{(\alpha^2+\alpha+2)L}\right)$. Givens $x^0, y^{-1} \in \mathcal{H}$, the sequence $\{x^k\}_{k\in N}$ generated by*

$$\begin{cases} y^k = x^k + \alpha(x^k - y^{k-1}), \\ x^{k+1} = J_{\lambda A}(y^k - \lambda By^k - \lambda(Bx^k - By^{k-1})). \end{cases}$$

Then $\{x^k\}_{k\in N}$ converges weakly to a point contained in $(A+B)^{-1}(0)$.

2.3.3 Forward-Reflected-Backward Method with Extrapolation and Linesearch

The algorithm presented in Sect. 2.3.1 requires information about the single-valued operator's Lipschitz constant in order to choose an appropriate stepsize. But for many practical problems, this requirement is difficult to meet. On the one hand, obtaining the global Lipschitz constant of a single-valued operator requires a significant cost in many cases and leading to poor numerical performance. On the other hand, we can only obtain a locally Lipschitz constant in practical problems. In view of these, we propose a forward-reflected-backward method with a linesearch procedure based on FRB$_{\mathrm{e}}$ (FRB$_{\mathrm{el}}$), which converges whenever the single-valued operator is locally Lipschitz.

Algorithm 2 Forward-reflected-backward method with extrapolation and linesearch (FRB$_{\mathrm{el}}$)

```
Choose $x^0 = y^{-1} \in \mathcal{H}$, $\lambda_0, \lambda_{-1} > 0$, $\delta \in \left(0, \frac{2(1-\alpha)}{\alpha^2+2\alpha+2}\right)$, $\alpha \in [0, 1)$, $\sigma \in (0, 1)$, $\rho \in \{1, \sigma^{-1}\}$.
For $k = 0, 1 \cdots$ , compute
    $\begin{cases} y^k = x^k + \alpha(x^k - y^{k-1}), \\ x^{k+1} = J_{\lambda_k A}(y^k - \lambda_k B y^k - \lambda_{k-1}(Bx^k - By^{k-1})), \end{cases}$
    where $\lambda_k = \rho\lambda_{k-1}\sigma^i$, with $i$ being the smallest nonnegative integer satisfying
    $\lambda_k ||Bx^{k+1} - By^k|| \leqslant \frac{\delta}{2} ||x^{k+1} - y^k||. \quad (2.27)$
```

Remark 2.3 As discussed at the beginning of this section, FRB$_{\mathrm{el}}$ combines an inertial step with a linesearch step and a resolvent step. It is worth noting that it is not necessary to estimate the Lipschitz constant of the Lipschitz continuous monotone operator B in our proposed Algorithm 2.

The following lemma shows that the linesearch procedure described in Algorithm 2 is well-defined when operator B is locally Lipschitz continuous.

Lemma 2.7 *Suppose that operator $B : \mathcal{H} \to \mathcal{H}$ is locally $Lipschitz$. Then the linesearch criterion (2.27) is well defined, which means that it will be satisfied after a finite number of iterations.*

Proof Let $x^{k+1}(\lambda) := J_{\lambda A}(y^k - \lambda B y^k - \lambda_{k-1}(Bx^k - By^{k-1}))$. According to the Lemma 2.5, we obtain that $J_{\lambda A}(x^{k+1}(0)) \to P_{\overline{\mathrm{dom} A}}(x^{k+1}(0))$ when $\lambda \downarrow 0$. Since operator A is maximal monotone, we know that $J_{\lambda A}$ is nonexpansive. Therefore, we have

$$\begin{aligned} &\|x^{k+1}(\lambda) - P_{\overline{\mathrm{dom} A}}(x^{k+1}(0))\| \\ \leqslant &\|x^{k+1}(\lambda) - J_{\lambda A}(x^{k+1}(0))\| + \|J_{\lambda A}(x^{k+1}(0)) - P_{\overline{\mathrm{dom} A}}(x^{k+1}(0))\| \\ \leqslant &\|\lambda B y^k\| + \|J_{\lambda A}(x^{k+1}(0)) - P_{\overline{\mathrm{dom} A}}(x^{k+1}(0))\|. \end{aligned}$$

By taking the limit as $\lambda \downarrow 0$,

$$x^{k+1}(\lambda) \to P_{\overline{\mathrm{dom} A}}(x^{k+1}(0)).$$

Assuming that linesearch finds λ at the kth iteration failed, then for all $\lambda = \rho\lambda_{k-1}\sigma^i$, $i = 0, 1 \ldots$, it implies

$$\rho\lambda_{k-1}\sigma^i \|Bx^{k+1}(\lambda) - By^k\| > \frac{\delta}{2}\|x^{k+1}(\lambda) - y^k\|.$$

Since operator B is locally Lipschitz, there exists $L > 0$ when i is large enough. And we have

$$\rho\lambda_{k-1}\sigma^i \|Bx^{k+1}(\lambda) - By^k\| > \frac{\delta}{2}\|x^{k+1}(\lambda) - y^k\| \geqslant \frac{\delta}{2L}\|Bx^{k+1}(\lambda) - By^k\|.$$

Therefore, it implies

$$\rho\lambda_{k-1}\sigma^i > \frac{\delta}{2L}.$$

Since $\sigma^i \to 0$ as $i \to \infty$, this inequality gives a contradiction, which completes the proof. □

The next Lemma 2.8 is a direct extension of Lemma 2.6.

Lemma 2.8 *Let $\{x^k\}_{k\in N}$ be the sequence generated by Algorithm 2. Then, the sequence $\{E_k\}_{k\in N}$ is monotonically nonincreasing. In particular, for any $k \geqslant 0$, it holds that*

$$E_{k+1} \leqslant E_k - C\|x^{k+1} - y^k\|^2,$$

where $C = (1-\alpha) - \frac{\delta}{2}(\alpha^2 + 2\alpha + 2) > 0$.

Proof The proof is similar to Lemma 2.6. We use inequality (2.27), which is well-defined due to Lemma 2.7, instead of the Lipschitzness of B to get the inequalities (2.14) and (2.23). □

Theorem 2.2 *Let $\mathcal{H}$ be finite dimensional, $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone, and $B\colon \mathcal{H} \to \mathcal{H}$ be monotone and locally Lipschitz continuous, and suppose that $(A+B)^{-1}(0)$ is nonempty. Then the sequence $\{x^k\}_{k\in N}$ generated by Algorithm 2 converges to a point contained in $(A+B)^{-1}(0)$.*

Proof It is similar to Theorem 2.1 but using Lemma 2.8 instead of Lemma 2.6, and (2.27) instead of the Lipschitzness of B. We take $\lambda_k L = \delta/2$ for all $k \in N$ in (2.22) and (2.24), so we can deduce $\{y^k\}_{k\in N}$ is bounded. Since (2.16) and (2.25), we know that $\{x^k\}_{k\in N}$ is bounded and $\|x^{k+1} - y^k\| \to 0$. As a locally Lipschitz operator on a finite-dimensional space, B is Lipschitz on bounded sets. Thus, since $\{x^k\}_{k\in N}$ is bounded, there exists a constant $L > 0$ such that

$$\|Bx^{k+1} - By^k\| \leqslant L\|x^{k+1} - y^k\|. \tag{2.28}$$

Combining (2.27) and (2.28), we see that $\{\lambda_k\}_{k\in N}$ is bounded away from zero. The remainder of the proof is similar to Theorem 2.1. □

2.4 Numerical Experiment

In this section, to demonstrate the effectiveness of the proposed FRB$_\mathrm{e}$ and FRB$_\mathrm{el}$, we apply them to solve the lasso problem and the ℓ_1 regularized logistic regression problem. All experiments are performed in MATLAB R2021a on a PC with Intel Core i7-13700H and 16.0 GB of RAM.

2.4.1 Lasso Problem

In this subsection, we consider the lasso problem:

$$\min_{x \in R^n} F(x) := \frac{1}{2}\|Dx - b\|^2 + \mu\|x\|_1, \tag{2.29}$$

where $D \in R^{m \times n}$, $b \in R^m$ and $\mu > 0$. We observe that (2.29) is in the form of (2.2) with $f(x) = \mu\|x\|_1$ and $g(x) = \frac{1}{2}\|Dx - b\|^2$. Therefore, the minimization problem (2.29) is equivalent to the following monotone inclusion problem

$$\text{find } x \in R^n \text{ such that } 0 \in (A + B)x,$$

where $A = \partial(\mu\|x\|_1)$ and $B = D^\mathrm{T}(Dx - b)$.

It is clear that g has a Lipschitz continuous gradient and $f + g$ has compact lower level sets. Thus, Assumption 2.1 is satisfied for (2.29), we can apply Algorithms 1 and 2 to solve (2.29). In addition, it is not hard to show that ∇g has a Lipschitz continuity modulus of $\lambda_{\max}(D^\mathrm{T}D)$. In view of this, in the experiment below, we take $L = \lambda_{\max}(D^\mathrm{T}D)$ for iFRB and FRB$_\mathrm{e}$.

Now we perform numerical experiments to study the performance of FRB$_\mathrm{e}$ and FRB$_\mathrm{el}$. We choose $\mu = 1$ and $\mu = 3$ in (2.29), initialize all algorithms at the origin, and use the duality gap of the primal problem (2.29) and its dual problem to terminate the algorithms as in [35]. Specifically, we define

$$u^k = \min\left\{1, \frac{\mu}{\|D^\mathrm{T}h(Ax^k)\|_\infty}\right\}\nabla h(Ax^k),$$

and terminate the algorithms when the duality gap is small, i.e.,

$$\frac{|g(x^k) + f(x^k) - d_{ls}(u^k)|}{\max\{f(x^k) + g(x^k), 1\}} \le 10^{-6},$$

where $g(x) = h(Ax) = \frac{1}{2}\|Ax - b\|^2$, and $d_{ls}(u)$ is the optimal value of dual problem.

Table 2.1 Numerical comparisons of different algorithms for solving the lasso problem. In this paper, we use the notation (m, n) to denote the corresponding choices of m and n $(\mu = 1)$

	(100, 200)		(200, 1000)		(400, 2000)		(600, 3000)	
	Iter	Time	Iter	Time	Iter	Time	Iter	Time
iFRB	1900	0.058	11924	0.677	19513	2.148	28368	8.285
FRB$_{\mathrm{e}}$	1210	0.036	7579	0.620	12401	1.967	18028	7.475
FRB$_{\mathrm{l}}$	399	0.017	1762	0.138	3040	0.429	4572	1.578
FRB$_{\mathrm{el}}$	**341**	**0.013**	**1390**	**0.120**	**2325**	**0.375**	**3389**	**1.521**

Table 2.2 Numerical comparisons of different algorithms for solving the lasso problem. In this paper, we use the notation (m, n) to denote the corresponding choices of m and n $(\mu = 3)$

	(100, 200)		(200, 1000)		(400, 2000)		(600, 3000)	
	Iter	Time	Iter	Time	Iter	Time	Iter	Time
iFRB	1122	0.028	4800	0.313	7343	0.897	10272	3.100
FRB$_{\mathrm{e}}$	715	0.025	3053	0.297	4668	0.839	6530	2.987
FRB$_{\mathrm{l}}$	204	0.011	661	0.052	1463	0.159	1470	0.542
FRB$_{\mathrm{el}}$	**183**	**0.009**	**573**	**0.045**	**1172**	**0.151**	**1175**	**0.507**

The problems used in our experiments are generated as follows. For each m and n, we generate an $m \times n$ matrix D with i.i.d. standard Gaussian entries. We then choose a support set T of size s uniformly at random, and generate an s-sparse vector $\hat{x}$ supported on T with i.i.d. standard Gaussian entries. The vector b is then generated as $b = D\hat{x} + 0.01\tilde{e}$, where $\tilde{e}$ has standard i.i.d. Gaussian entries.

To illustrate the effectiveness of FRB$_{\mathrm{e}}$ and FRB$_{\mathrm{el}}$, we compare them with iFRB [18] and FRB$_{\mathrm{l}}$ [18]. In the experiment, we choose the values of stepsize and inertial parameter to achieve the optimal performance for each algorithm. As outlined in [18], we specify the parameters as follows: $\alpha = 0.2, \lambda = 0.99 \times \frac{1}{5L}$ for iFRB; $\alpha = 0$, $\delta = 0.99$, $\sigma = 0.7$, $\rho = \sigma^{-1}$, and the stepsize sequence $\{\lambda_k\}_{k \in N}$ satisfy (2.27) with $y^k = x^k$ for FRB$_{\mathrm{l}}$. As for FRB$_{\mathrm{e}}$ and FRB$_{\mathrm{el}}$, we choose: $\alpha = 0.2$, $\lambda = 0.99 \times \frac{2}{13L}$ for FRB$_{\mathrm{e}}$; $\alpha = 0.3$, $\delta = 0.99 \times \dfrac{2(1-\alpha)}{\alpha^2 + 2\alpha + 2}$, $\sigma = 0.7$, $\rho = \sigma^{-1}$ and the stepsize sequence $\{\lambda_k\}_{k \in N}$ satisfy (2.27) for FRB$_{\mathrm{el}}$.

The computational results with $\mu = 1$ and $\mu = 3$ averaged over 50 instances for a range of choices of m and n are presented in Tables 2.1 and 2.2. From these two tables, we can see that FRB$_{\mathrm{e}}$ outperforms the iFRB, and FRB$_{\mathrm{el}}$ outperforms the FRB$_{\mathrm{l}}$ in terms of the number of iterations and CPU time for all scenarios. To further observe the convergence of the tested algorithms, we plot the evolutions of $F(x^k) - F^{\star}$ with respect to the iteration numbers for each algorithm in Figs. 2.1 and 2.2, where $F^{\star}$ represents the minimum of the objective function values obtained by all tested algorithms. The results show that the FRB$_{\mathrm{e}}$ converges faster than the iFRB,

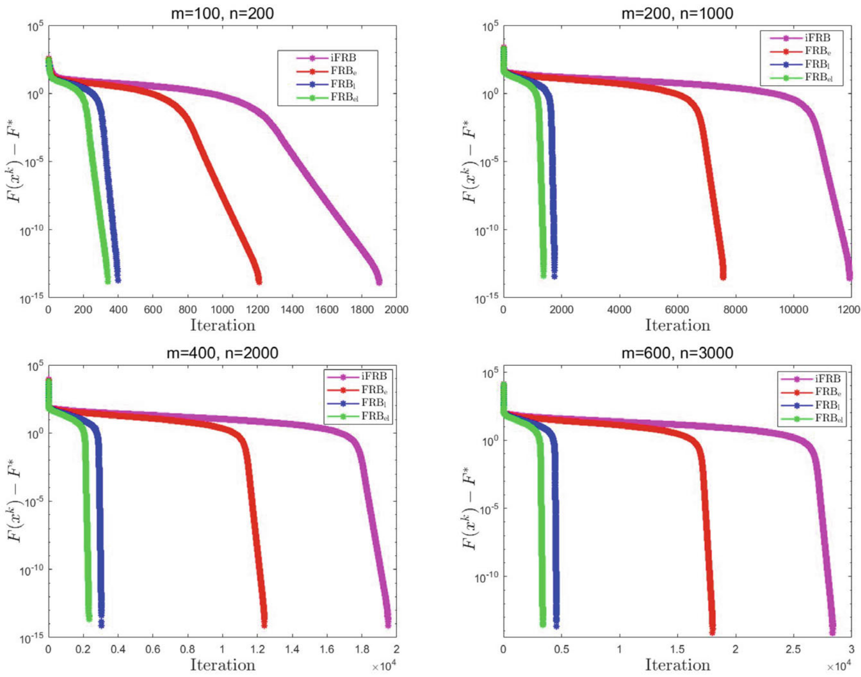


Fig. 2.1 Evolutions of $F(x^k) - F^\star$ with respect to the number of iterations for solving the lasso problem with $\mu = 1$

and the FRB$_{\mathrm{el}}$ converges faster than the FRB$_{\mathrm{l}}$, demonstrating the advantage of the FRB$_{\mathrm{e}}$ and FRB$_{\mathrm{el}}$.

2.4.2 ℓ_1 Regularized Logistic Regression Problem

In this subsection, we consider the ℓ_1 regularized logistic regression problem,

$$\min_{\widetilde{x}\in R^n,\, x^0\in R} F(x) := \sum_{i=1}^{m} \log(1 + e^{-b_i(a_i^{\mathrm{T}}\widetilde{x}+x^0)}) + \gamma\|\widetilde{x}\|_1, \tag{2.30}$$

where $a_i \in R^n$, $b_i \in \{-1, 1\}$, $i = 1, \cdots, m$, with b_i not all the same, $m < n$, and $\gamma > 0$ is the regularization parameter. So we can apply our methods for solving the problem (2.2) in case $f(\tilde{x}) = \gamma\|\tilde{x}\|_1$, $g(\tilde{x}) = \sum_{i=1}^{m} \log(1 + e^{-b_i(P\tilde{x})_i})$, where $x := (\widetilde{x}, x^0) \in R^{n+1}$, and P is the matrix whose ith row is given by $(a_i^{\mathrm{T}}\ 1)$. Moreover, one can show that ∇g is Lipschitz continuous with modulus $0.25\lambda_{\max}(P^{\mathrm{T}}P)$ [35]. Thus, in our experiments below we take $L = 0.25\lambda_{\max}(P^{\mathrm{T}}P)$.

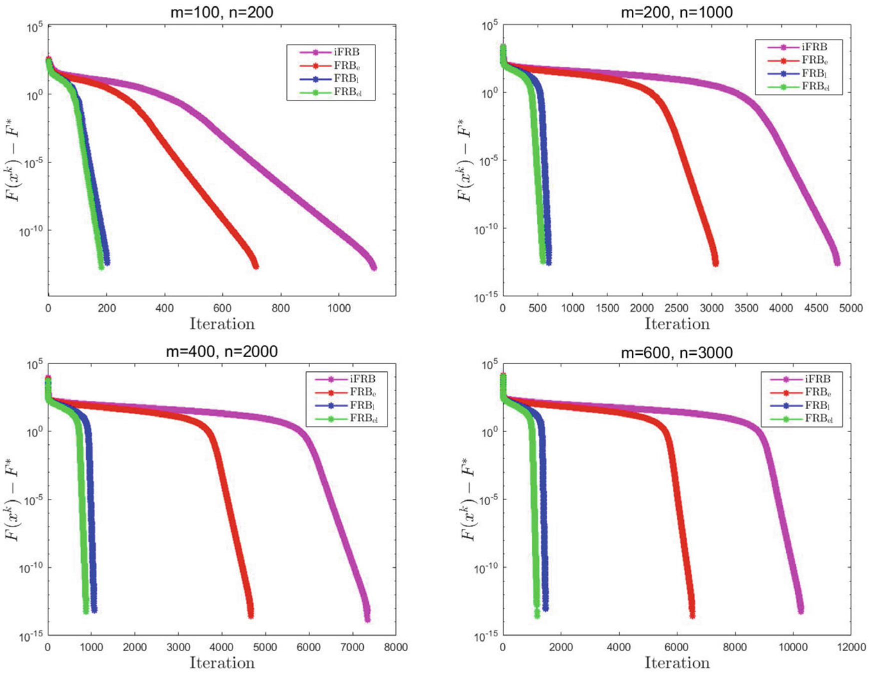


Fig. 2.2 Evolutions of $F(x^k) - F^\star$ with respect to the number of iterations for solving the lasso problem with $\mu = 3$

As that in [35], Assumption 2.1 is satisfied for (2.30). Thus, Algorithm 1 and Algorithm 2 are applicable. In the experiments below, we choose $\gamma = 1$ and $\gamma = 3$ in (2.30). We initialize all algorithms at the origin, and terminate the algorithms as in [35].

We consider random instances for our experiments. For each m and n, we generate an $m \times n$ matrix A with i.i.d. standard Gaussian entries, where A is the matrix whose ith row is a_i^{T}. We then choose a support set T of size s uniformly at random and generate an s-sparse vector $\hat{x}$ supported on T with i.i.d. standard Gaussian entries. The vector b is then generated as $b = \operatorname{sign}(A\hat{x} + ce)$, where c is chosen uniformly at random from $[0, 1]$.

We now perform numerical experiments to verify the efficiency of FRB$_\mathrm{e}$ and FRB$_\mathrm{el}$. In the experiment, we choose the values of stepsize and inertial parameter to achieve the optimal performance for each algorithm. As outlined in [18], we specify the parameters as follows: $\alpha = 0.2$, $\lambda = 0.99 \times \frac{1}{5L}$ for iFRB; $\alpha = 0$, $\delta = 0.99$, $\sigma = 0.8$, $\rho = 1$ and the stepsize sequence $\{\lambda_k\}_{k \in N}$ satisfy (2.27) with $y^k = x^k$ for FRB$_\mathrm{l}$. As for FRB$_\mathrm{e}$ and FRB$_\mathrm{el}$, we choose: $\alpha = 0.2$, $\lambda = 0.99 \times \frac{2}{13L}$ for FRB$_\mathrm{e}$; $\alpha = 0.3$, $\delta = 0.99 \times \dfrac{2(1-\alpha)}{\alpha^2 + 2\alpha + 2}$, $\sigma = 0.8$, $\rho = 1$ and the stepsize sequence $\{\lambda_k\}_{k \in N}$ satisfy (2.27) for FRB$_\mathrm{el}$.

Table 2.3 Numerical comparisons of different algorithms for solving the logistic regression problem. In this paper, we use the notation (m, n) to denote the corresponding choices of m and n $(\gamma = 1)$

	(200, 400)		(400, 1000)		(600, 2000)		(1000, 3000)	
	Iter	Time	Iter	Time	Iter	Time	Iter	Time
iFRB	21861	12.108	67358	257.697	119334	1097.831	166041	3727.979
FRB$_{\mathrm{e}}$	13890	7.616	42803	163.784	75829	670.698	105511	2353.380
FRB$_{\mathrm{l}}$	650	0.696	1253	10.355	1545	30.246	1918	99.612
FRB$_{\mathrm{el}}$	**478**	**0.496**	**914**	**7.923**	**1126**	**23.103**	**1204**	**61.330**

Table 2.4 Numerical comparisons of different algorithms for solving the logistic regression problem. In this paper, we use the notation (m, n) to denote the corresponding choices of m and n $(\gamma = 3)$

	(200, 400)		(400, 1000)		(600, 2000)		(1000, 3000)	
	Iter	Time	Iter	Time	Iter	Time	Iter	Time
iFRB	6178	3.348	13008	50.094	26691	243.296	37269	863.537
FRB$_{\mathrm{e}}$	3928	2.160	8269	33.829	16963	157.555	23683	564.829
FRB$_{\mathrm{l}}$	378	0.409	541	4.786	845	16.440	1032	53.490
FRB$_{\mathrm{el}}$	**319**	**0.342**	**412**	**3.599**	**668**	**13.675**	**791**	**41.608**

The computational results averaged over 50 instances for a range of choices of m and n are presented in Tables 2.3 and 2.4. From these two tables, we can see that FRB$_{\mathrm{e}}$ outperforms the iFRB, and FRB$_{\mathrm{el}}$ outperforms the FRB$_{\mathrm{l}}$ in terms of the number of iterations and CPU time for all scenarios. The results show that the FRB$_{\mathrm{e}}$ converges faster than the iFRB, and the FRB$_{\mathrm{el}}$ converges faster than the FRB$_{\mathrm{l}}$, demonstrating the advantage of the FRB$_{\mathrm{e}}$ and FRB$_{\mathrm{el}}$.

2.5 Conclusion

In this paper, we propose two forward-reflected-backward type methods, named FRB$_{\mathrm{e}}$ and FRB$_{\mathrm{el}}$, for monotone inclusion problems. The proposed methods not only unify some well-known operator splitting methods based on the inertial and linesearch techniques, but can also leads to improved numerical performance. We establish the weak convergence of the proposed methods under mild and standard assumptions. Further, we apply FRB$_{\mathrm{e}}$ and FRB$_{\mathrm{el}}$ to solve the lasso problem and the ℓ_1 regularized logistic regression problem. The numerical results on these two important problems verify the efficiency of our proposed methods. In the future, we shall investigate how to combine the relaxation technique and two different extrapolation steps in order to have better performance.

Acknowledgements The work of Xingju Cai was supported by the National Natural Science Foundation of China (NSFC) under Grant 12131004. The work of Tanxing Wang was supported by Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant 181200003024152.

References

1. Abubakar, J., Kumam, P., Hassan Ibrahim, A., Padcharoen, A.: Relaxed inertial Tseng's type method for solving the inclusion problem with application to image restoration. Mathematics **8**, 1–19 (2020)
2. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, Berln (2011)
3. Ḅot, R.I., Csetnek, E.R.: An inertial Tseng's type proximal algorithm for nonsmooth and nonconvex optimization problems. J. Optim. Theory Appl. **171**, 600–616 (2016)
4. Ḅot, R.I., Csetnek, E.R., Hendrich, C.: Inertial Douglas-Rachford splitting for monotone inclusion problems. Appl. Math. Comput. **256**, 472–487 (2015)
5. Cai, G., Dong, Q.L., Peng, Y.: Strong convergence theorems for inertial Tseng's extragradient method for solving variational inequality problems and fixed point problems. Optim. Lett. **15**, 1457–1474 (2021)
6. Cai, X.J., Guo, K., Jiang, F., Wang, K., Wu, Z.M., Han, D.R.: The developments of proximal point algorithms. J. Oper. Res. Soc. China. **10**, 197–239 (2022)
7. Cholamjiak, P., Hieu, D.V., Cho, Y.J.: Relaxed forward-backward splitting methods for solving variational inclusions and applications. J. Sci. Comput. **88**, 1–23 (2021)
8. Cholamjiak, P., Hieu, D.V., Muu, L.D.: Inertial splitting methods without prior constants for solving variational inclusions of two operators. Bull. Iran. Math. Soc. **48**, 3019–3045 (2022)
9. Chen, C.H., Ma, S.Q., Yang, J.F.: A general inertial proximal point algorithm for mixed variational inequality problem. SIAM J. Optim. **25**, 2120–2142 (2015)
10. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. **4**, 1168–1200 (2005)
11. Çopur, A.K., Hacioğlu, E., Gursoy, F., Ertürk, M.: An efficient inertial type iterative algorithm to approximate the solutions of quasi variational inequalities in real Hilbert spaces. J. Sci. Comput. **89**, 1–28 (2021)
12. Dong, Q., Jiang, D., Cholamjiak, P., Shehu, Y.: A strong convergence result involving an inertial forward-backward algorithm for monotone inclusions. J. Fixed Point Theory Appl. **19**, 3097–3118 (2017)
13. Duchi, J., Singer, Y.: Efficient online and batch learning using forward-backward splitting. J. Mach. Learn. Res. **10**, 2899–2934 (2009)
14. He, B.S., Yuan, X.M.: Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. SIAM J. Imaging Sci. **5**, 119–149 (2012)
15. Hsieh, Y.G., Iutzeler, F., Malick, J., Mertikopoulos, P.: On the convergence of single-call stochastic extra-gradient methods. Adv. Neural Inf. Process. Syst. **32**, (2019)
16. Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. **16**, 964–979 (1979)
17. Malitsky, Y.: Projected reflected gradient methods for monotone variational inequalities. SIAM J. Optim. **25**, 502–520 (2015)
18. Malitsky, Y., Tam, M.K.: A forward-backward splitting method for monotone inclusions without cocoercivity. SIAM J. Optim. **30**, 1451–1472 (2020)
19. Ochs, P., Brox, T., Pock, T.: iPiasco: inertial proximal algorithm for strongly convex optimization. J. Math. Imaging Vision. **53**, 171–181 (2015)

20. Ochs, P., Chen, Y.J., Brox, T., Pock, T.: iPiano: inertial proximal algorithm for nonconvex optimization. SIAM J. Imaging Sci. **7**, 1388–1419 (2014)
21. Opial, Z.: Weak convergence of the sequence of successive approximations for nonexpansive mappings. Bull. Am. Math. Soc. **73**, 591–597 (1967)
22. Padcharoen, A., Kitkuan, D., Kumam, W., Kumam, P.: Tseng methods with inertial for solving inclusion problems and application to image deblurring and image recovery problems. Comput. Math. Methods **3**, 1–14 (2020)
23. Pascall, D., Sburlan, S.: Nonlinear Mappings of Monotone Type. Springer, Dordrecht (1978)
24. Passty, G.B.: Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. J. Math. Anal. Appl. **72**, 383–390 (1979)
25. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. Comput. Math. Math. Phys. **4**, 1–17 (1964)
26. Popov, L.D.: A modification of the Arrow-Hurwicz method for finding saddle points. Math. Notes **28**, 845–848 (1980)
27. Raguet, H., Fadili, J., Peyré, G.: A generalized forward-backward splitting. SIAM J. Imaging Sci. **6**, 1199–1226 (2013)
28. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. **14**, 877–898 (1976)
29. Shehu, Y., Iyiola, O.S., Reich, S.: A modified inertial subgradient extragradient method for solving variational inequalities. Optim. Eng. **23**, 421–449 (2021)
30. Tan, B., Cho, S.Y.: Strong convergence of inertial forward-backward methods for solving monotone inclusions. Appl. Anal. **101**, 5386–5414 (2022)
31. Thong, D.V., Vinh, N.T., Cho, Y.J.: A strong convergence theorem for Tseng's extragradient method for solving variational inequality problems. Optim. Lett. **14**, 1157–1175 (2020)
32. Thong, D.V., Yang, J., Cho, Y.J., Rassias, T.M.: Explicit extragradient-like method with adaptive stepsizes for pseudomonotone variational inequalities. Optim. Lett. **15**, 2181–2199 (2021)
33. Tseng, P.: A modified forward-backward splitting method for maximal monotone mapping. SIAM J. Control. Optim. **38**, 431–446 (2000)
34. Wang, Z.B., Lei, Z.Y., Long, X., Chen, Z.Y.: A modified Tseng splitting method with double inertial steps for solving monotone inclusion problems. J. Sci. Comput. **96**, 1–29 (2023)
35. Wen, B., Chen, X.J., Pong, T.K.: Linear convergence of proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems. SIAM J. Optim. **27**, 124–145 (2017)

Chapter 3
Fast Adaptive ADMM with Gaussian Back Substitution for Multiple Block Linear Constrained Separable Problems

Xiangfeng Wang

Abstract This work presents a novel algorithmic framework, called *Fast Adaptive Alternating Direction Method of Multipliers with Gaussian Back Substitution* (ADMM-G-V), tailored for solving multiple block linear constrained separable problems. The proposed method extends the classical multi-block ADMM by incorporating an adaptive penalty parameter, which is dynamically adjusted during the iterative process to enhance convergence properties and computational efficiency. A comprehensive theoretical analysis is provided by establishing the global convergence and worst-case convergence rate of the algorithm in both ergodic and non-ergodic senses. We demonstrate the effectiveness of our method through numerical experiments on consensus problems over networked agents and distributed logistic regression tasks.

Keywords ADMM · Self-adaptive · Gaussian back substitution · Multiple-block

3.1 Introduction and Motivation

In this paper, we consider a general structured multiple block minimization model, i.e.,

$$\min_{\{x_i \in \mathcal{E}_i\}} \left\{ \sum_{i=1}^{m} \theta_i(x_i) \;\middle|\; \sum_{i=1}^{m} \mathcal{A}_i(x_i) = b \right\}, \tag{3.1}$$

where each $\theta_i : \mathcal{E}_i \to \mathbb{R}$ is a convex real-valued function, each $\mathcal{A}_i : \mathcal{E}_i \to \mathbb{R}^\ell$ denotes a linear operator with $b \in \mathbb{R}^\ell$ be a given vector. Throughout, the solution set $X^\star$ of (3.1) is assumed to be nonempty. This general formulation (3.1) includes the popular two-block problem as a special case, which has proved to be a reasonable and effective model in plenty of applications [1]. The motivation for considering this

X. Wang (✉)
School of Computer Science and Technology and Key Laboratory of Mathematics and Engineering Applications MoE, East China Normal University, Shanghai, China
e-mail: xfwang@cs.ecnu.edu.cn

X. Wang et al. (eds.), *Splitting Optimization*, ICIAM2023 Springer Series 2,
https://doi.org/10.1007/978-981-96-8574-5_3

Table 3.1 $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, C and D for RASL and TILT: $\mathcal{I}$ denotes the identity transform; $\mathcal{J}, \mathcal{J}_i, \epsilon_i, M$ and τ are all given matrix or vector; $\circ$ denotes a given linear operation

Model	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}(C)$	C	D
RASL	$\mathcal{I}$	$\mathcal{I}$	$-\sum_{i=1}^{n} \mathcal{J}_i \Delta\tau(\epsilon_i \epsilon_i^{\mathrm{T}})$	$\Delta\tau$	$M \circ \tau$
TILT	$\mathcal{I}$	$\mathcal{I}$	$-\mathcal{J}\Delta\tau$	$\Delta\tau$	$M \circ \tau$

particular, more general structured problem (3.1) is that for many interesting applications, the goal (objective) and environment (constraint) become more complicated, especially as more and more application-driven distinctive structures or information are introduced to guarantee better performance. In the following, we propose some practical applications to draw forth our work.

Motivation applications (Low-rank or sparse structure-based image processing problems): The low-rank structure is popularly used to model the invariant property or great relevance between images, while the sparse structure is a byproduct of modeling uncorrelated information. A general formulation can be summarized as

$$\begin{aligned} \min_{L,S,\Delta} \quad & \|L\|_\star + \gamma \|S\|_1 \\ \text{s.t.} \quad & \mathcal{A}(L) + \mathcal{B}(S) + \mathcal{C}(C) = D, \end{aligned} \tag{3.2}$$

where the matrix $L \in \mathbb{R}^{m\times n}$ denotes the low-rank part, the matrix $S \in \mathbb{R}^{m\times n}$ represents the sparse part, while Δ usually additional parameters driven by the application task. Concretely, we evoke two popular models, e.g., *robust alignment by sparse and low-rank decomposition* (RASL) [2] and *transform invariant low-rank textures* (TILT) [3]. In order to solve these two models, some subproblem has to be efficiently calculated, which can be included into (3.2) as in Table 3.1.

The unified formulation (3.2) can be contained in the general model (3.1); as a result, the requirement of efficiently computing impels us to design a structured algorithm framework for (3.1). For algorithmic-design purposes, we may need to consider using these two functions individually as well because they usually have different properties/structures. Define the augmented Lagrange function of (3.1) for the case $m = 2$ be

$$\mathcal{L}_\beta\left(x_1, x_2, \lambda\right) = \theta_1(x_1) + \theta_2(x_2) - \lambda^{\mathrm{T}}\left(A_1 x_1 + A_2 x_2 - b\right) + \frac{\beta}{2}\left\|A_1 x_1 + A_2 x_2 - b\right\|^2, \tag{3.3}$$

with $\lambda \in \mathbb{R}^m$ denotes the Lagrange multiplier and $\beta > 0$ the penalty parameter. The alternating direction method of multipliers (ADMM) was first proposed by Glowinski and Marrocco in [4] for solving specific nonlinear elliptic equations, and its iterative scheme for (3.1) reads as

$$x^{k+1} = \arg\min\left\{\mathcal{L}_\beta\left(x, y^k, \lambda^k\right) \mid x \in \mathcal{X}\right\}; \tag{3.4a}$$

$$y^{k+1} = \arg\min\left\{\mathcal{L}_\beta\left(x^{k+1}, y, \lambda^k\right) \mid y \in \mathcal{Y}\right\}; \tag{3.4b}$$

$$\lambda^{k+1} = \lambda^k - \beta\left(Ax^{k+1} + By^{k+1} - b\right). \tag{3.4c}$$

Recently, ADMM has found many applications arising in different areas such as image processing, statistical learning, computer vision, and wireless communication network. We refer to [1, 5, 6] for some review papers of ADMM. The convergence of ADMM has been well analyzed in some earlier references such as [7, 8]. For the convergence rate, according to the known explanation in [7, 9] that the ADMM is a special case of the proximal point algorithm (PPA) in [10] and the convergence results of PPA in [11], it is easy to perceive that the ADMM scheme (3.4), without further assumptions such as the strong convexity or special structure assumptions on one or both the functions in the objective, some restrictions on the penalty parameter β, or some error bounds or metric subregularity assumptions on the solution set, is sublinear. In [12, 13], the authors established the sublinear convergence rate for the ADMM scheme (3.4), in sense of both the egrodic and non-ergodic worst-case $\mathcal{O}\left(\frac{1}{n}\right)$ natures respectively measured by the iteration complexity where n is the iteration counter.

The classical ADMM (3.4) has been comparatively thoroughly analyzed and understood. In this paper, we mainly focus on the multiple-block case, which also has popular applications in image processing, machine learning, etc. In order to take advantage of each θ_i's properties individually, a natural idea for solving the general case of (3.1) with $m \geq 3$ is to extend the classical ADMM scheme (3.4) straightforwardly—yielding the following scheme for iteration k+1:

$$\begin{cases} x_1^{k+1} = \arg\min\left\{\mathcal{L}_\beta\left(x_1, x_2^k, \ldots, x_m^k, \lambda^k\right) \;\middle|\; x_1 \in \mathcal{X}_1\right\}, \\ \quad\cdots \\ x_i^{k+1} = \arg\min\left\{\mathcal{L}_\beta\left(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_m^k, \lambda^k\right) \;\middle|\; x_i \in \mathcal{X}_i\right\}, \\ \quad\cdots \\ x_m^{k+1} = \arg\min\left\{\mathcal{L}_\beta\left(x_1^{k+1}, \ldots, x_{m-1}^{k+1}, x_m, \lambda^k\right) \;\middle|\; x_m \in \mathcal{X}_m\right\}, \\ \lambda^{k+1} = \lambda^k - \beta\left(\sum\limits_{i=1}^m A_i x_i^{k+1} - b\right), \end{cases} \tag{3.5}$$

with the introduction of the augmented Lagrangian function as follows

$$\mathcal{L}_\beta\left(x_1, \ldots, x_m, \lambda\right) = \sum_{i=1}^m \theta_i(x_i) - \lambda^{\mathrm{T}}\left(\sum_{i=1}^m A_i x_i - b\right) + \frac{\beta}{2}\left\|\sum_{i=1}^m A_i x_i - b\right\|^2. \tag{3.6}$$

Just as the original ADMM scheme (3.4), the iterative scheme (3.5) can be easily derived by decomposing the augmented Lagrangian function of (3.1) in the Gauss-Seidel fashion. In (3.5), the variables x_i's are minimized in alternating order, and the decomposed subproblems are much easier than the original problem (3.1) since only one function θ_i is involved in its x_i-subproblem. Then, the step of updating the Lagrange multiplier coordinates all these solutions to local small subproblems to find a solution to a global large problem. Note that the direct extension of the

ADMM scheme (3.5) reduces to the augmented Lagrangian method (ALM) [14] and the standard ADMM scheme when $m = 1$ and $m = 2$ in (3.1), respectively.

The convergence of the scheme (3.5), however, had perplexed authors for a long time. On one hand, the scheme (3.5) empirically works well for some applications, see [2, 15]. On the other hand, in the literature, its convergence could be shown only under some further assumptions. For example, in [16–18], the convergence of (3.5) was shown under some additional strongly convex conditions, and the penalty parameter β should be chosen judiciously within a certain interval. Moreover, when each function θ_i in (3.1) is of particular structure and the update of λ^{k+1} in (3.5) is required to adopt a new step size rather than β, i.e.,

$$\lambda^{k+1} = \lambda^k - \tau\beta \left(\sum_{i=1}^{m} A_i x_i^{k+1} - b \right), \tag{3.7}$$

where $\tau > 0$ is sufficiently small to fulfill certain error bound conditions, the resulting scheme was proved to be convergent in [19], together with some linear convergent results. In fact, the scheme (3.5) but with the new update of λ in (3.7) can be regarded as an implementation of the dual ascent method to the dual of (3.1) with a shrunk step size. A counterexample was given in [20] showing that the scheme (3.5) is not necessarily convergent without further assumptions, and a sufficient condition ensuring the convergence of (3.5) was given therein.

In general, it is not easy to verify whether the step size τ in (3.7) is small enough to satisfy the desired error bound. Thus it should be stick to the direct extension of ADMM (3.5) where the step size for updating λ is taken as the same as the penalty parameter (thus it is not necessarily very small) and the function θ_i's in (3.5) are only assumed to be generic nonsmooth convex functions, and study in which way the convergence of (3.5) can be derived. In [21], the authors have shown that the resulting sequence is convergent if the output of (3.5) is further corrected by a Gaussian back substitution procedure. The numerical efficiency of the Gaussian back substitution procedure, together with its superiority to some other relevant work based on (3.5), has been illustrated numerically in [15, 22]. The detailed algorithm framework of ADMM with Gaussian back substitution (ADMM-G) is presented as follows, in which the direct extension of ADMM (3.5) can be considered as the first prediction step:

$$\begin{cases} \tilde{x}_1^k = \arg\min \left\{ \mathcal{L}_\beta \left(x_1, x_2^k, \ldots, x_m^k, \lambda^k \right) \;\middle|\; x_1 \in \mathcal{X}_1 \right\}, \\ \quad \cdots \\ \tilde{x}_i^k = \arg\min \left\{ \mathcal{L}_\beta \left(\tilde{x}_1^k, \ldots, \tilde{x}_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_m^k, \lambda^k \right) \;\middle|\; x_i \in \mathcal{X}_i \right\}, \\ \quad \cdots \\ \tilde{x}_m^k = \arg\min \left\{ \mathcal{L}_\beta \left(\tilde{x}_1^k, \ldots, \tilde{x}_{m-1}^k, x_m, \lambda^k \right) \;\middle|\; x_m \in \mathcal{X}_m \right\}, \\ \tilde{\lambda}^k = \lambda^k - \beta \left(\sum_{i=1}^{m} A_i \tilde{x}_i^k - b \right), \end{cases} \tag{3.8a}$$

$$\begin{cases} \lambda^{k+1} = \lambda^k + \alpha\left(\tilde{\lambda}^k - \lambda^k\right), \\ x_m^{k+1} = x_m^k + \alpha\left(\tilde{x}_m^k - x_m^k\right), \\ \quad \cdots \\ x_i^{k+1} = x_i^k + \alpha\left(\tilde{x}_i^k - x_i^k\right) - \sum\limits_{j=i+1}^{m} \left(A_i^{\mathrm{T}} A_i\right)^{-1} \left(A_i^{\mathrm{T}} A_j\right)\left(x_j^{k+1} - x_j^k\right), \\ \quad \cdots \\ x_1^{k+1} = \tilde{x}_1^k, \end{cases} \tag{3.8b}$$

where we need to assume that all A_is have full column rank. The above Gaussian back substitution procedure (3.8) still requires to compute the inverses of $A_i^{\mathrm{T}} A_i$ for $i = 2, \ldots, m-1$, which could be computationally expensive for generic A_i's arising in some image processing applications. The computing procedure of ADMM-G can be graphically explained through the following Fig. 3.1. If we choose $\alpha = 1$, it is obvious that the direct extension of ADMM with Gaussian back substitution can reduce to the classical original ADMM (3.4a)–(3.4c) for (3.1) with $m = 2$. In classical ADMM and ADMM-G, the penalty parameter β can be an arbitrary positive scalar to theoretically guarantee the convergence. This is a nice property, and it provides flexibility in choosing different values when the ADMM is implemented for various specific applications. Meanwhile, it is well known that the efficiency of ADMM highly depends on the appropriate choice of the penalty parameter, and the efficiency can vary dramatically with different values of the penalty parameter for the same problem. Moreover, choosing a good value of the penalty parameter is application-sensitive; a value working well for one application might be completely ineffective for another one. Indeed, so far, this is no general theory for how to choose the "optimal" value of β for the ADMM-type scheme (3.4) or (3.8). This concern forces practitioners to tune the penalty parameter a priori for the specific application under discussion when the ADMM-type scheme (3.4) or (3.8) is implemented. There is a large volume of literature mentioning this issue.

On the other hand, the penalty parameter β can be adjusted dynamically subject to certain rules and it results in the ADMM-type methods with variable penalty parameter. The prior searching for an appropriate value of β can thus be avoided. Indeed, the convergence of ADMM with variable penalty parameter can be theoretically carried over if the rules are meticulously chosen; see, e.g., [21, 23–25], for some relatively earlier literature for the convergence analysis. Some important properties for establishing the convergence of ADMM with variable penalty parameter can be

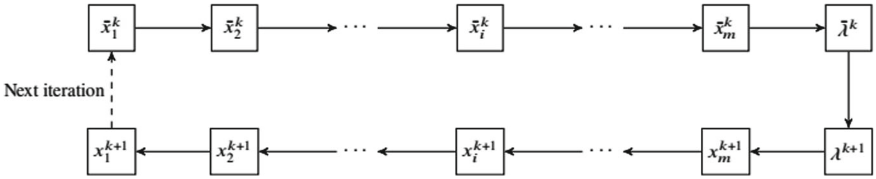


Fig. 3.1 Computing procedure of ADMM-G

found in [24]. Empirically, it is arguable to say which one is preferred: tuning the ADMM-type scheme a prior to find a suitable choice of β or iteratively adjusting the parameters in the iterative process. We believe there is no conclusive assertion in this regard, and both methodologies are equally important from either the mathematical or empirical perspective. Depending on different computation loads in different phases, each of the strategies can find supportive applications in the very large set of literature. Recently, [26] extends the self-adaptive penalty parameter scheme to the consensus problem, which is in a special form of (3.1). Another adaptive penalty scheme is proposed in [27], which also can be included in the framework of [21]. [27] introduces an adaptive relaxed ADMM, which is a popular practical variant of ADMM, and proves the convergence result for the adaptive penalty parameter case. In applications, the rule initiated in [21] for adjusting the penalty parameter has been well verified in many applications such as machine learning [28, 29], computer vision [30], computer graphics [31], smart grid [32], geophysical image processing [33, 34] and elucidated in [1].

Nevertheless, until now, there is no result about adaptive ADMM-G (3.8) with variable penalty parameters in which the penalty parameters are iteratively adjusted. In this paper, we will propose an adaptive ADMM-G algorithm framework, while further establishing the global convergence and worst-case convergence rate of the ADMM with variable penalty parameters in both the ergodic and non-ergodic senses. Before presenting the algorithm framework, we mention that we have to compute all inverses of $A_i^{\mathrm{T}}A_i$ $(i = 1, \dots, m-1)$ in ADMM-G, which maybe heavy computational complexity. In this paper, we introduce a new variable $u(w)$ which will help modify ADMM-G to avoid calculating the inverses, and some concrete related definitions are as follows

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad w = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ \lambda \end{pmatrix}, \quad u = \begin{pmatrix} x_2 \\ \vdots \\ x_m \\ \lambda \end{pmatrix}, \tag{3.9a}$$

$$v(w) = \begin{pmatrix} v(w)_1 \\ v(w)_2 \\ \vdots \\ v(w)_m \\ v(w)_{m+1} \end{pmatrix} := \begin{pmatrix} A_1x_1 \\ A_2x_2 \\ \vdots \\ A_mx_m \\ \lambda \end{pmatrix}, \quad u(w) := \begin{pmatrix} A_2x_2 \\ \vdots \\ A_mx_m \\ \lambda \end{pmatrix}, \tag{3.9b}$$

and we further rewrite the augmented Lagrangian function (3.6) into the following equivalent reformulation, e.g.,

$$\begin{aligned}\mathcal{L}_\beta\left(x_1,\ldots,x_m,\lambda\right) &= \sum_{i=1}^m \theta_i(x_i) - \lambda^{\mathrm{T}}\left(\sum_{i=1}^m A_i x_i - b\right) + \frac{\beta}{2}\left\|\sum_{i=1}^m A_i x_i - b\right\|^2 \\ &= \sum_{i=1}^m \theta_i(x_i) - \left[v\left(w\right)_{m+1}\right]^{\mathrm{T}}\left(\sum_{i=1}^m v\left(w\right)_i - b\right) + \frac{\beta}{2}\left\|\sum_{i=1}^m v\left(w\right)_i - b\right\|^2 .\end{aligned}$$

We re-record the above equation when only focusing on x_i as

$$\begin{aligned}&\hat{\mathcal{L}}_\beta\left(v\left(w\right)_1,\ldots,v\left(w\right)_{i-1},x_i,v\left(w\right)_{i+1},\ldots,v\left(w\right)_m,v\left(w\right)_{m+1}\right) \\ =&\theta_i(x_i) - \left[v\left(w\right)_{m+1}\right]^{\mathrm{T}}\left(A_i x_i + \sum_{j=1,j\neq i}^m v\left(w\right)_j - b\right) + \frac{\beta}{2}\left\|A_i x_i + \sum_{j=1,j\neq i}^m v\left(w\right)_j - b\right\|^2 .\end{aligned} \tag{3.10}$$

As a result, each subproblem with respect to x_i is determined only by

$$\begin{aligned}v\left(w\right)_{-i} &:= \left[v\left(w\right)_{[1,i-1]}, v\left(w\right)_{[i+1,m+1]}\right] \\ &= \left[v\left(w\right)_1,\ldots,v\left(w\right)_{i-1},v\left(w\right)_{i+1},\ldots,v\left(w\right)_m,v\left(w\right)_{m+1}\right],\end{aligned}$$

which means in iteration k+1 of (3.8), we have

$$\begin{aligned}\tilde{x}_i^k &= \mathrm{argmin}\left\{\mathcal{L}_\beta\left(\tilde{x}_1^k,\ldots,\tilde{x}_{i-1}^k,x_i,x_{i+1}^k,\ldots,x_m^k,\lambda^k\right)\ \middle|\ x_i\in\mathcal{X}_i\right\} \\ &= \arg\min_{x_i\in\mathcal{X}_i}\left\{\theta_i(x_i) + \frac{\beta}{2}\left\|A_i x_i + \sum_{j=1}^{i-1} v(\tilde{w}^k)_j + \sum_{j=i+1}^m v(w^k)_j - b - \frac{\lambda^k}{\beta}\right\|^2\right\} \\ &= \arg\min_{x_i\in\mathcal{X}_i}\left\{\hat{\mathcal{L}}_\beta\left(v(\tilde{w}^k)_1,\ldots,v(\tilde{w}^k)_{i-1},x_i,v(w^k)_{i+1},\ldots,v(w^k)_m,v(w^k)_{m+1}\right)\right\}.\end{aligned} \tag{3.11}$$

Thus, the recursion of ADMM-G with adaptive variable penalty parameter (ADMM-spsGspsV) for the structured convex optimization (3.1) can be written as follows in Algorithm 1. We give some further explanations of Algorithm 1 (ADMMspsGspsV) as follows:

- The matrix $\mathcal{M}_k$ in each iteration is defined as

$$\mathcal{M}_k = \begin{pmatrix} I_\ell & -I_\ell & 0 & \cdots & \cdots & 0 \\ 0 & I_\ell & -I_\ell & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -I_\ell & 0 \\ 0 & \cdots & \cdots & 0 & I_\ell & 0 \\ \ell & -\beta_k I_\ell & \cdots & \cdots & -\beta_k I_\ell & I_\ell \end{pmatrix} \in \mathbb{R}^{(m\ell)\times(m\ell)};$$

Algorithm 1 ADMM-G with adaptive variable penalty parameters (ADMM-G-V)

```
Require $u\left(w^0\right) \in \underbrace{\mathbb{R}^\ell \times \cdots \times \mathbb{R}^\ell}_{m}, \beta_0 > 0, \alpha \in (0,1)$;
Calculate a new adaptive variable penalty parameter $\beta_k$;
while not converged do
    For given $u\left(w^k\right)$, $\tilde{w}^k := \left(\tilde{x}_1^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k\right)$ are calculated according to (3.8a), e.g.,
    $$\begin{cases} \tilde{x}_1^k = \arg\min\left\{\hat{\mathcal{L}}_{\beta_k}\left(x_1, v\left(w^k\right)_2, \ldots, v\left(w^k\right)_m, v\left(w^k\right)_{m+1}\right) \mid x_1 \in \mathcal{X}_1\right\}, \\ \quad\vdots \\ \tilde{x}_i^k = \arg\min\left\{\hat{\mathcal{L}}_{\beta_k}\left(v\left(\tilde{w}^k\right)_1, \ldots, v\left(\tilde{w}^k\right)_{i-1}, x_i, v\left(w^k\right)_{i+1}, \ldots, v\left(w^k\right)_m, v\left(w^k\right)_{m+1}\right) \mid x_i \in \mathcal{X}_i\right\}, \\ \quad\vdots \\ \tilde{x}_m^k = \arg\min\left\{\hat{\mathcal{L}}_{\beta_k}\left(v\left(\tilde{w}^k\right)_1, \ldots, v\left(\tilde{w}^k\right)_m, v\left(w^k\right)_{m+1}\right) \mid x_i \in \mathcal{X}_i\right\}, \\ \tilde{\lambda}^k = \lambda^k - \beta_k\left(A_1\tilde{x}_1^k + \sum_{i=2}^m A_i x_i^k - b\right); \end{cases} \tag{3.12}$$
    Generate the new iterate $u^{k+1}$ and $x_1^{k+1}$ by correcting $\tilde{w}^k$ in the backward order, e.g.,
    $$u(w^{k+1}) = u(w^k) - \alpha\mathcal{M}_k\left[u(w^k) - u(\tilde{w}^k)\right]; \tag{3.13}$$
end while=0
```

- Two sequences are iteratively computed in ADMM-G, e.g., $\left\{\tilde{w}^k\right\}$ and $\left\{u\left(w^k\right)\right\}$. $\left\{u\left(w^k\right)\right\}$ denotes the truly iterative sequence with $\left\{\tilde{w}^k\right\}$ being an intermediate iterative sequence. As a result, we will consider the theoretical results on both sequences.

The computing procedure of ADMM-G can be graphically explained through the following Fig. 3.2.

This algorithm framework, ADMM-G-V, obviously includes ADMM-G if the penalty parameter is set to be a constant β. In the following, we will consider this new algorithm in a general prediction-contraction framework form. We would like to mention that for succinctness, we skip the rapidly increasing references appearing in the literature for discussing sharper convergence rate results of the ADMM-type methods from various perspectives under stronger assumptions. Instead, we stick

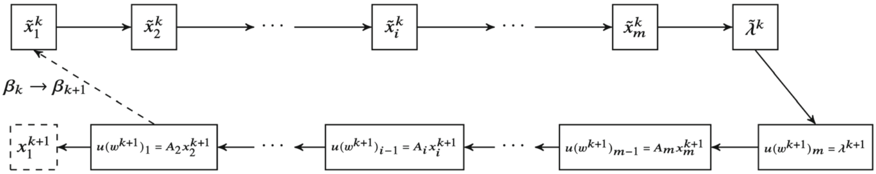

$\tilde{x}_1^k$
$\tilde{x}_2^k$
$\cdots$
$\tilde{x}_i^k$
$\cdots$
$\tilde{x}_m^k$
$\tilde{\lambda}^k$
$\beta_k \to \beta_{k+1}$
x_1^{k+1}
$u(w^{k+1})_1 = A_2 x_2^{k+1}$
$u(w^{k+1})_{i-1} = A_i x_i^{k+1}$
$u(w^{k+1})_{m-1} = A_m x_m^{k+1}$
$u(w^{k+1})_m = \lambda^{k+1}$

Fig. 3.2 Computing procedure of ADMM-G

to the most general model setting in (3.1) and the iterative scheme in Algorithm 1; and only discuss the global convergence and sublinear convergence rate. The results in this plain context can immediately boost further analysis for deriving sharper convergence rates under various stronger assumptions for the ADMM with variable penalty parameters.

The rest of the paper is organized as follows. In Sect. 3.2, we will first propose some preliminaries and notations. Discussions about the adaptive variable penalty parameter β_k will be given in Sect. 3.3. A prediction-correction reformulation framework will be given in Sect. 3.4. In Sect. 3.5, we will prove the global convergence result and will establish the convergence rate analysis in both ergodic and non-ergodic sense. Section 3.6 contains some numerical experiments to prove the efficiency of variable penalty parameters and the well-established theoretical results. Conclusions are presented in Sect. 3.7.

3.2 Preliminaries and Notations

In this section, we summarize some preliminaries that will be used later in our analysis. Recall we have defined λ as the Lagrange multiplier of the linear constraints in (3.1). Then, the Lagrangian function of this problem is

$$\mathcal{L}(x_1, \ldots, x_m, \lambda) = \sum_{i=1}^{m} \theta_i(x_i) - \lambda^{\mathrm{T}} \left(\sum_{i=1}^{m} A_i x_i - b \right),$$

which is defined on $\mathcal{W} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_m \times \mathbb{R}^{\ell}$. Let $\left(x_1^*, \ldots, x_m^*, \lambda^*\right) \in \mathcal{W}$ be an saddle point of the Lagrangian function, then as discussed in [35] that finding a saddle point of $\mathcal{L}(x_1, \ldots, x_m, \lambda)$ is equivalent to finding a vector

$$w^* = \begin{pmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_m^* \\ \lambda^* \end{pmatrix} \text{ such that } \begin{cases} \theta_1(x_1) - \theta_1(x_1^*) + \left(x_1 - x_1^*\right)^{\mathrm{T}} \left(-A_1^{\mathrm{T}} \lambda^*\right) \geq 0, \forall x_1 \in \mathcal{X}_1, \\ \theta_2(x_2) - \theta_2(x_2^*) + \left(x_2 - x_2^*\right)^{\mathrm{T}} \left(-A_2^{\mathrm{T}} \lambda^*\right) \geq 0, \forall x_2 \in \mathcal{X}_2, \\ \quad \vdots \\ \theta_m(x_m) - \theta_m(x_m^*) + \left(x_m - x_m^*\right)^{\mathrm{T}} \left(-A_m^{\mathrm{T}} \lambda^*\right) \geq 0, \forall x_m \in \mathcal{X}_m, \\ (\lambda - \lambda^*)^{\mathrm{T}} \left(\sum\limits_{i=1}^{m} A_i x_i^* - b \right) \geq 0, \forall \lambda \in \mathbb{R}^{\ell}. \end{cases} \tag{3.14}$$

More compactly, the inequalities in (3.14) can be equivalently rewritten as the following variational inequality (VI), e.g.,

$$\mathrm{VI}(\mathcal{W}, F, \theta): \; w^* \in \mathcal{W}, \; \theta(x) - \theta(x^*) + \left(w - w^*\right)^{\mathrm{T}} F\left(w^*\right) \geq 0, \; \forall\, w \in \mathcal{W}, \tag{3.15a}$$

where

$$\theta(x)=\sum_{i=1}^{m}\theta_i(x_i),\quad F(w)=\begin{pmatrix}-A_1^{\mathrm{T}}\lambda\\-A_2^{\mathrm{T}}\lambda\\\vdots\\-A_m^{\mathrm{T}}\lambda\\\sum_{i=1}^{m}A_ix_i-b\end{pmatrix}. \tag{3.15b}$$

Note that the operator $F(w)$ defined in (3.15b) is monotone because it is affine with a skew-symmetric matrix. Let $\mathcal{W}^*$ be the solution set of VI($\mathcal{W}$, F, θ). Since the solution set of (3.1) is assumed to be nonempty, so that $\mathcal{W}^*$ is also nonempty. In the convergence rate analysis, we shall use a characterization of the solution set $\mathcal{W}^*$ of VI (3.15a). We present it as the following theorem, and its proof can be found in [36, Theorem 2.3.5] or [12, Theorem 2.1].

Theorem 3.1 *The solution set of VI($\mathcal{W}$, F, θ) is convex and it can be characterized as*

$$\mathcal{W}^*=\bigcap_{w\in\mathcal{W}}\left\{\tilde{w}\in\mathcal{W}:\theta(x)-\theta(\tilde{x})+(w-\tilde{w})^{\mathrm{T}}F(w)\geq 0\right\}. \tag{3.16}$$

In the following, we summarize some notations which will be used in later analysis. These notations will make the presentation of our theoretical analysis in later sections more compact. Also, these notations are based on previous work [35, 37], and you can find more details in these references. Define $\mathcal{V}=\mathcal{X}_2\times\cdots\times\mathcal{X}_m\times\mathbb{R}^{\ell}$, and accordingly we also use the notation

$$\mathcal{V}^*=\left\{\left(x_2^*,\ldots,x_m^*,\lambda^*\right)\ \middle|\ (x_1^*,x_2^*,\ldots,x_m^*,\lambda^*)\in\mathcal{W}^*\right\}.$$

Further define a matrix sequence $\left\{H_k=\beta_kI_\ell\in\mathbb{R}^{\ell\times\ell}\right\}$ and a new variable sequence $\{u_k\}$ based on the variable w and the matrix H_k, e.g.,

$$u_k(w)=\begin{pmatrix}u_k(x_2)\\\vdots\\u_k(x_m)\\u_k(\lambda)\end{pmatrix}=\begin{pmatrix}\sqrt{H_k}A_2x_2\\\vdots\\\sqrt{H_k}A_mx_m\\\sqrt{H_k}^{-1}\lambda\end{pmatrix}\in\mathbb{R}^{m\ell}, \tag{3.17}$$

from which the notation $u_k(w^k)=\left\{u_k(x_2^k),\ldots,u_k(x_m^k),u_k(\lambda^k)\right\}$ is also clear. For some concrete applications of (3.1) such as those in [22, 37], the matrices $A_2,\cdots A_m$ are all identity matrices. For these cases, $u_k(w)$ reduces to v if all β_k equal to 1. With these notations, the scheme (1) can be summarized as generating the new iteration $u_k\left(w^{k+1}\right)$ with the input $u_k\left(w^k\right)$. Further based on some notations in [35], we define

$$\mathcal{A}_k = \operatorname{diag}\left(\sqrt{H_k}A_2, \sqrt{H_k}A_3, \dots, \sqrt{H_k}A_m, \sqrt{H_k}^{-1} I_\ell\right) \in \mathbb{R}^{m\ell \times \left(\sum\limits_{i=2}^{m} n_i + \ell\right)}, \quad (3.18)$$

which indicates that $u_k(w) = \mathcal{A}_k z$. Then, we introduce several matrices in block-wise form as follows:

$$Q = \begin{pmatrix} I_\ell & 0 & \cdots & \cdots & 0 \\ I_\ell & I_\ell & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ I_\ell & \cdots & I_\ell & I_\ell & 0 \\ -I_\ell & -I_\ell & \cdots & -I_\ell & I_\ell \end{pmatrix} \in \mathbb{R}^{m\ell \times m\ell}, \quad \mathcal{M} = \begin{pmatrix} I_\ell & -I_\ell & 0 & 0 & 0 \\ 0 & I_\ell & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -I_\ell & 0 \\ 0 & \cdots & 0 & I_\ell & 0 \\ -I_\ell & -I_\ell & \cdots & -I_\ell & I_\ell \end{pmatrix} \in \mathbb{R}^{m\ell \times m\ell}, \quad (3.19)$$

which will be used in the substitution procedures to be proposed. Note that Q and $\mathcal{M}$ are all well-structured block lower triangular matrices, and for (3.1) with $m = 2$, we have

$$Q = \mathcal{M} = \begin{pmatrix} I_\ell & 0 \\ -I_\ell & I_\ell \end{pmatrix} \in \mathbb{R}^{2\ell \times 2\ell}.$$

3.3 Adaptive Variable Penalty Parameter $\boldsymbol{\beta_k}$

In this section, we will discuss the adaptive updating scheme of the important penalty parameter β_k. The main concern is not only to efficiently accelerate our algorithm framework, but also to guarantee better theoretical results, which include global convergence and iteration complexity for general convex problems. At the beginning, we present the following generic property that the sequence $\{\beta_k\}$ needs to satisfy, and our theoretical analysis is based on the following property.

Summable Bounded Property: there exists another non-negative sequence $\{\eta_k\}$ such that $\{\beta_k, \eta_k\}$ satisfies

$$\frac{1}{1+\eta_k}\beta_k \le \beta_{k+1} \le (1+\eta_k)\beta_k, \qquad \sum_{k=0}^{+\infty} \eta_k < +\infty. \quad (3.20)$$

This summable bounded property indicates that although the penalty parameter β_k can be dynamically adjusted, each pair β_k and β_{k+1} should be “close.” They have at most $1 + \eta_k$ times the gap.

Motivated by [21], in which residual balancing scheme is applied to adaptively tune the parameter sequence $\{\beta_k\}$, we define the primal and dual “residuals” with respect to iterations $u\left(w^{k-1}\right)$ and $u\left(w^{k}\right)$ as follows

$$p_k := \mathcal{A}\left[u\left(w^k\right) - u\left(w^{k-1}\right)\right],$$

$$d_k := \sum_{i=1}^{m} A_i x_i^k - b,$$

where $\mathscr{A} := \operatorname{diag}(A_2, \cdots, A_m) \in \mathbb{R}^{(m-1)\ell \times \sum_{i=2}^{m} n_i}$. According to the convergence results in the following sections, we guarantee that both of these two residuals approach zero as the iterates become more accurate. We observe that increasing β_k strengthens the penalty term, yielding larger primal residual but smaller dual residual; conversely, decreasing β_k leads to smaller primal residual but larger dual residual. Both residuals must be small enough at convergence, it makes sense to "balance" them, i.e., tune β_k to keep both residuals of similar magnitude, i.e., $\|p_k\| \approx \|d_k\|$.

Given a constant $\mu \in (0, 1)$ and a non-negative sequence $\{\eta_k\}$ that satisfies

$$\sum_{k=0}^{+\infty} \eta_k < +\infty, \tag{3.21}$$

we consider the following three strategies, where $t \in \mathbb{N}_+$:

1. $\{\beta_k\}$ is monotonically nondecreasing:

$$\beta_{k+1} = \begin{cases} \beta_k (1 + \eta_k), & \text{if } \|p_k\| < \mu \|d_k\|, \\ \beta_k, & \text{otherwise,} \end{cases} \tag{3.22}$$

with $\eta_k = 0$ if $(k \bmod t) \neq 0$;

2. $\{\beta_k\}$ is monotonically nonincreasing:

$$\beta_{k+1} = \begin{cases} \frac{\beta_k}{1+\eta_k}, & \text{if } \|d_k\| < \mu \|p_k\|, \\ \beta_k, & \text{otherwise,} \end{cases} \tag{3.23}$$

with $\eta_k = 0$ if $(k \bmod t) \neq 0$;

3. $\{\beta_k\}$ is self-adaptive:

$$\beta_{k+1} = \begin{cases} \beta_k (1 + \eta_k), & \text{if } \|p_k\| < \mu \|d_k\|, \\ \frac{\beta_k}{1+\eta_k}, & \text{if } \|d_k\| < \mu \|p_k\|, \\ \beta_k, & \text{otherwise,} \end{cases} \tag{3.24}$$

with $\eta_k = 0$ if $(k \bmod t) \neq 0$.

All the above three cases indicate that we will adjust β every t iterations if possible and we always have

$$\beta_{k+1} = \beta_k (1 + \eta_k), \text{ or } \beta_{k+1} = \frac{\beta_k}{1 + \eta_k}, \text{ or } \beta_{k+1} = \beta_k,$$

which demonstrates that these strategies all guarantee the **summable bounded property**, i.e., (3.20). Furthermore, under condition (3.21), we can obtain that

$$\prod_{i=1}^{\infty}(1+\eta_k) < +\infty.$$

Hence, the sequence $\{\beta_k\}$ is both upper bounded and bounded below away from zero; that is, we have

$$\inf_k \{\beta_k\} > 0, \quad \sup_k \{\beta_k\} < +\infty.$$

3.4 Prediction-Correction Reformulation Framework

In this section, we reformulate the proposed Algorithm 1 as a prediction-correction framework; some lemmas are proved accordingly. Note that this prediction-correction framework is simply a theoretical revisit to the scheme (3.8) that is more convenient for the coming convergence analysis; it is not necessary to obey this framework for implementing the scheme (3.12). In the following, we propose a key lemma of this paper, which immediately enables us to reformulate the Algorithm 1 with adaptive variable penalty parameters as a prediction-correction framework.

Lemma 3.1 *Let u^{k+1} and $\tilde{w}^k$ are generated by Algorithm 1 with given u^k, together with $u_k(w^k) = \mathcal{A}_k z^k$ and $u_k(\tilde{w}^k) = \mathcal{A}_k \tilde{z}^k$. Then we have*

1. *Prediction step: for all $w \in \mathcal{W}$*

$$\theta(x) - \theta(\tilde{x}^k) + \left(w - \tilde{w}^k\right)^{\mathrm{T}} F(\tilde{w}^k) \geq \left(u_k(w) - u_k(\tilde{w}^k)\right)^{\mathrm{T}} \mathcal{Q}\left(u_k(w^k) - u_k(\tilde{w}^k)\right); \quad (3.25)$$

2. *Correction-step: relationship between $u_k(w^{k+1})$ and $u_k(\tilde{w}^k)$ with $\alpha \in (0, 1)$*

$$u_k(w^{k+1}) = u_k(w^k) - \alpha\mathcal{M}\left(u_k(w^k) - u_k(\tilde{w}^k)\right); \quad (3.26)$$

3. *Based on matrices $\mathcal{Q}$ and $\mathcal{M}$ in (3.19), we define matrix $\mathcal{H}$ and $\mathcal{G}$*

$$\mathcal{H} = \mathcal{Q}\mathcal{M}^{-1} \in \mathbb{R}^{m\ell\times m\ell}, \quad \mathcal{G} = \mathcal{Q}^{\mathrm{T}} + \mathcal{Q} - \mathcal{M}^{\mathrm{T}}\mathcal{H}\mathcal{M} \in \mathbb{R}^{m\ell\times m\ell}, \quad (3.27)$$

where matrix $\mathcal{H}$ is symmetric and positive definite, while matrix $\mathcal{G}$ is also symmetric but positive semi-definite.

Proof The *prediction-step* (3.25) can be obtained through the same inference as [35, Theorem 3.3] except that $u_k(w)$ is based on the adaptive penalty parameter β_k. So that we ignore the detailed proof here, please check [35, Theorem 3.3] for more details.

As for the *correction-step*, it can be directly obtained by using the notations $u_k(w^k)$, $u_k(\tilde{w}^k)$, $u_k(w^{k+1})$ and $\mathcal{N}$. Also, the positive semi-definiteness of $\mathcal{H}$ and $\mathcal{G}$ can be obtained through some basic matrix calculations. Almost the same technique has been used in [12, 35, 37]. □

Moreover, based on the above Lemma 3.1 and more structure-driven properties of the above-defined notations, we prove two contraction-type convergence properties.

Lemma 3.2 *Let u^{k+1} and $\tilde{w}^k$ are generated by Algorithm 1 with given u^k, together with $u_k(w^k) = \mathcal{A}_k z^k$ and $u_k(\tilde{w}^k) = \mathcal{A}_k \tilde{z}^k$. Then we have*

1. *Basic-contraction-function: for all $w \in \mathcal{W}$*

$$\begin{aligned}&\theta(x) - \theta(\tilde{x}^k) + \left(w - \tilde{w}^k\right)^{\mathrm{T}} F\left(\tilde{w}^k\right) \\ &\geq \frac{1}{2\alpha}\left(\left\|u_k(w) - u_k(w^{k+1})\right\|_{\mathcal{H}}^2 - \left\|u_k(w) - u_k(w^k)\right\|_{\mathcal{H}}^2\right) + \frac{1-\alpha}{2\alpha^2}\left\|u_k(w^k) - u_k(w^{k+1})\right\|_{\mathcal{H}}^2;\end{aligned} \tag{3.28}$$

2. *Strictly-contraction-function: for all $w^* \in \mathcal{W}^*$*

$$\left\|u_k(w^{k+1}) - u_k(w^*)\right\|_{\mathcal{H}}^2 \leq \left\|u_k(w^k) - u_k(w^*)\right\|_{\mathcal{H}}^2 - \frac{1-\alpha}{\alpha}\left\|u_k(w^k) - u_k(w^{k+1})\right\|_{\mathcal{H}}^2. \tag{3.29}$$

Proof For the first *basic-contraction-function*, recall (3.25) and the definition of $\mathcal{H}$, we have for all $w \in \mathcal{W}$

$$\theta(x) - \theta(\tilde{x}^k) + \left(w - \tilde{w}^k\right)^{\mathrm{T}} F\left(\tilde{w}^k\right) \geq \frac{1}{\alpha}\left(u_k(w) - u_k(\tilde{w}^k)\right)^{\mathrm{T}} \mathcal{H}\left(u_k(w^k) - u_k(w^{k+1})\right).$$

Further applying the identity

$$(a-b)^{\mathrm{T}} \mathcal{H}(c-d) = \frac{1}{2}\left\{\|a-d\|_{\mathcal{H}}^2 - \|a-c\|_{\mathcal{H}}^2\right\} + \frac{1}{2}\left\{\|c-b\|_{\mathcal{H}}^2 - \|d-b\|_{\mathcal{H}}^2\right\},$$

to the right-hand side of the above inequality with

$$a = u_k(w),\ b = u_k(\tilde{w}^k),\ c = u_k(w^k),\ d = u_k(w^{k+1}),$$

we thus obtain

$$\begin{aligned}&\left(u_k(w) - u_k(\tilde{w}^k)\right)^{\mathrm{T}} \mathcal{H}\left(u_k(w^k) - u_k(w^{k+1})\right) \\ &= \frac{1}{2\alpha}\left(\left\|u_k(w) - u_k(w^{k+1})\right\|_{\mathcal{H}}^2 - \left\|u_k(w) - u_k(w^k)\right\|_{\mathcal{H}}^2\right) \\ &\qquad + \frac{1}{2\alpha}\left(\left\|u_k(w^k) - u_k(\tilde{w}^k)\right\|_{\mathcal{H}}^2 - \left\|u_k(w^{k+1}) - u_k(\tilde{w}^k)\right\|_{\mathcal{H}}^2\right).\end{aligned}$$

As for the last term, we can prove that

$$
\begin{aligned}
&\left\|u_k(w^k) - u_k(\tilde{w}^k)\right\|_{\mathcal{H}}^2 - \left\|u_k(w^{k+1}) - u_k(\tilde{w}^k)\right\|_{\mathcal{H}}^2 \\
&= \left\|u_k(w^k) - u_k(\tilde{w}^k)\right\|_{\mathcal{H}}^2 - \left\|\left(u_k(w^k) - u_k(\tilde{w}^k)\right) - \left(u_k(w^k) - u_k(w^{k+1})\right)\right\|_{\mathcal{H}}^2 \\
&\overset{(3.2)}{=} \left\|u_k(w^k) - u_k(\tilde{w}^k)\right\|_{\mathcal{H}}^2 - \left\|\left(u_k(w^k) - u_k(\tilde{w}^k)\right) - \alpha\mathcal{M}\left(u_k(w^k) - u_k(\tilde{w}^k)\right)\right\|_{\mathcal{H}}^2 \\
&= 2\alpha\left(u_k(w^k) - u_k(\tilde{w}^k)\right)^{\mathrm{T}}\mathcal{H}\mathcal{M}\left(u_k(w^k) - u_k(\tilde{w}^k)\right) - \alpha^2\left\|u_k(w^k) - u_k(\tilde{w}^k)\right\|_{\mathcal{M}^{\mathrm{T}}\mathcal{H}\mathcal{M}}^2 \\
&= \alpha\left(u_k(w^k) - u_k(\tilde{w}^k)\right)^{\mathrm{T}}(\mathcal{Q}^{\mathrm{T}} + \mathcal{Q} - \mathcal{M}^{\mathrm{T}}\mathcal{H}\mathcal{M})\left(u_k(w^k) - u_k(\tilde{w}^k)\right) \\
&\qquad + \alpha(1-\alpha)\left(u_k(w^k) - u_k(\tilde{w}^k)\right)^{\mathrm{T}}\mathcal{M}^{\mathrm{T}}\mathcal{H}\mathcal{M}\left(u_k(w^k) - u_k(\tilde{w}^k)\right) \\
&\overset{(3.2)}{=} \alpha\left\|u_k(w^k) - u_k(\tilde{w}^k)\right\|_{\mathcal{G}}^2 + \frac{1-\alpha}{\alpha}\left\|u_k(w^k) - u_k(w^{k+1})\right\|_{\mathcal{H}}^2 \\
&\geq \frac{1-\alpha}{\alpha}\left\|u_k(w^k) - u_k(w^{k+1})\right\|_{\mathcal{H}}^2,
\end{aligned}
\tag{3.30}
$$

with $\alpha \in (0, 1)$ and further the assertion of the first part in this theorem is proved. In the following, recall (3.28) and set $w = w^*$, we get

$$
\begin{aligned}
&\|u_k(w^k) - u_k(w^*)\|_{\mathcal{H}}^2 - \|u_k(w^{k+1}) - u_k(w^*)\|_{\mathcal{H}}^2 \\
&\geq \frac{1-\alpha}{\alpha}\|u_k(w^k) - u_k(w^{k+1})\|_{\mathcal{H}}^2 + 2\alpha\left\{\theta(\tilde{x}^k) - \theta(x^*) + \left(\tilde{w}^k - w^*\right)^{\mathrm{T}} F\left(\tilde{w}^k\right)\right\} \\
&\geq \frac{1-\alpha}{\alpha}\|u_k(w^k) - u_k(w^{k+1})\|_{\mathcal{H}}^2 + 2\alpha\left\{\theta(\tilde{x}^k) - \theta(x^*) + \left(\tilde{w}^k - w^*\right)^{\mathrm{T}} F\left(w^*\right)\right\} \\
&\geq \frac{1-\alpha}{\alpha}\|u_k(w^k) - u_k(w^{k+1})\|_{\mathcal{H}}^2,
\end{aligned}
$$

where the second last inequality is based on the monotonicity of $F(w)$ and the last inequality is obtained by using the optimality of w^*. Thus, the assertion (3.29) follows directly. □

3.5 Theoretical Analysis

In this section, we will establish the theoretical results for ADMM-G-V, which includes the global convergence and the worst case $O\left(\frac{1}{k}\right)$ convergence rate in both ergodic and non-ergodic senses.

3.5.1 Global Convergence

The proved lemmas are adequate for establishing the global convergence of the proposed ADMM-G-V algorithm, and the analytic framework is standard in the context of contractive-type methods. Before presenting the main theorem, we introduce the variable u, which has no H_k after comparing with $u_k(w)$. Actually, $\{u^k\}$ is the sequence that we should pay more attention to, which is iteratively computed in ADMM-G-V.

Theorem 3.2 *Let $\{u(w^k)\}$ and $\{\tilde{w}^k\}$ be the sequence generated by the proposed ADMM-G-V. Then, we have*

1. $\|u(w^k) - u(w^{k+1})\|_{\mathcal{H}} \overset{k\to\infty}{\longrightarrow} 0$, $\left\|\sum_{i=1}^{m} A_i \tilde{x}_i^k - b\right\|_2 \overset{k\to\infty}{\longrightarrow} 0$;[1]
2. $\theta(\tilde{x}^k) - \theta(x^*) \overset{k\to\infty}{\longrightarrow} 0$ *for any given* $x^* \in \mathcal{X}^*$. *where* $\mathcal{A}_k = diag$ $(A_2, A_3, \ldots, A_m, I_\ell)$.

Proof Recall the *strictly-contraction-function* property, e.g., for all $w^* \in \mathcal{W}^*$ we have

$$\left\|u_k(w^{k+1}) - u_k(w^*)\right\|_{\mathcal{H}}^2 \le \left\|u_k(w^k) - u_k(w^*)\right\|_{\mathcal{H}}^2 - \frac{1-\alpha}{\alpha}\left\|u_k(w^k) - u_k(w^{k+1})\right\|_{\mathcal{H}}^2.$$

Mention again the step-size property (3.20), we have

$$\left\|u_k(w) - u_k(w^k)\right\|_{\mathcal{H}}^2 \le (1+\eta_{k-1}) \cdot \left\|u_{k-1}(w) - u_{k-1}(w^k)\right\|_{\mathcal{H}}^2.$$

By applying this property in the above contractive property with $w = w^*$, we can obtain

$$\begin{aligned}\frac{1}{\gamma_1^{k+1}}\left\|u_{k+1}(w^{k+1}) - u_{k+1}(w^*)\right\|_{\mathcal{H}}^2 &\le \frac{1+\eta_k}{\gamma_1^{k+1}}\left\|u_k(w^{k+1}) - u_k(w^*)\right\|_{\mathcal{H}}^2\\ &\le \frac{1}{\gamma_1^k}\left\|u_k(w^k) - u_k(w^*)\right\|_{\mathcal{H}}^2 - \frac{1-\alpha}{\alpha\gamma_1^k}\left\|u_k(w^k) - u_k(w^{k+1})\right\|_{\mathcal{H}}^2,\end{aligned}$$

where

$$\gamma_k^{t-1} = \prod_{j=k}^{t-1}(1+\eta_j), \quad \gamma_t^{t-1} = 1.$$

[1] $\|u(w^k) - u(w^{k+1})\|_{\mathcal{H}}$ has no relationship with the adaptive β_k, and is only determined by the iterative sequence $\{u(w^k)\}$.

Summing the above inequality from $k = 1$ to $k = t$, we have

$$\begin{aligned}
&\frac{1}{\gamma_1^{t+1}} \left\| u_{t+1}(w^{t+1}) - u_{t+1}(w^*) \right\|_{\mathcal{H}}^2 \\
&\le \frac{1}{\gamma_1^1} \left\| u_1(w^1) - u_1(w^*) \right\|_{\mathcal{H}}^2 - \sum_{k=1}^{t} \frac{1-\alpha}{\alpha\gamma_1^k} \left\| u_k(w^k) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2 \\
&\le \frac{1}{\gamma_1^1} \left\| u_1(w^1) - u_1(w^*) \right\|_{\mathcal{H}}^2 - \sum_{k=1}^{t} \frac{1-\alpha}{\alpha C_p} \left\| u_k(w^k) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2 .
\end{aligned} \tag{3.31}$$

As a result, we obtain that for all iterations k

$$\begin{aligned}
\sum_{k=1}^{t} \frac{(1-\alpha)\min\left\{\beta_k, \frac{1}{\beta_k}\right\}}{\alpha C_p} \left\| u\left(w^k\right) - u\left(w^{k+1}\right) \right\|_{\mathcal{H}}^2 &\le \sum_{k=1}^{t} \frac{1-\alpha}{\alpha C_p} \left\| u_k(w^k) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2 \\
&\le \frac{1}{\gamma_1^1} \left\| u_1(w^1) - u_1(w^*) \right\|_{\mathcal{H}}^2 < \infty.
\end{aligned} \tag{3.32}$$

Further more with the unified boundedness of the sequence $\left\{ \frac{(1-\alpha)\min\left\{\beta_k, \frac{1}{\beta_k}\right\}}{\alpha C_p} \right\}_k$, we can guarantee that

$$\left\| u\left(w^k\right) - u\left(w^{k+1}\right) \right\|_{\mathcal{H}}^2 \overset{k\to\infty}{\longrightarrow} 0. \tag{3.33}$$

As follows, we have $\left\| \lambda^k - \lambda^{k+1} \right\| \overset{k\to\infty}{\longrightarrow} 0$, which indicates

$$\left\| \sum_{i=1}^{m} A_i \tilde{x}_i^k - b \right\|_2 \overset{k\to\infty}{\longrightarrow} 0.$$

Recall the optimality condition with respect to $\tilde{x}_i^k$ and x_i^* respectively, we have

$$\begin{gathered}
\theta_i\left(x_i\right) - \theta_i\left(\tilde{x}_i\right) + \left(x_i - \tilde{x}_i^k\right)^{\mathrm{T}} \left\{ -A_i^{\mathrm{T}} \tilde{\lambda}^k + \beta A_i^{\mathrm{T}} \left(\sum_{j=2}^{i} A_j^{\mathrm{T}} \left(\tilde{x}_j^k - x_j^k \right) \right) \right\} \ge 0, \\
\theta_i\left(x_i\right) - \theta_i\left(x_i^*\right) + \left(x_i - x_i^*\right)^{\mathrm{T}} \left(-A_i^{\mathrm{T}} \lambda^*\right) \ge 0.
\end{gathered}$$

Then by subscribing x_i^* and $\tilde{x}_i^k$ into the above two inequalities respectively, we can obtain

$$\begin{aligned}
&\left(\tilde{x}_i^k - x_i^*\right)^{\mathrm{T}} \left(A_i^{\mathrm{T}} \lambda^*\right) \le \theta_i\left(\tilde{x}_i^k\right) - \theta_i\left(x_i^*\right) \\
&\qquad \le \left(\tilde{x}_i^k - x_i^*\right)^{\mathrm{T}} \left(A_i^{\mathrm{T}} \tilde{\lambda}^k\right) + \left(x_i^* - \tilde{x}_i^k\right)^{\mathrm{T}} \left[\beta_k A_i^{\mathrm{T}} \left(\sum_{j=2}^{i} A_j \left(\tilde{x}_j^k - x_j^k \right) \right) \right].
\end{aligned} \tag{3.34}$$

Summing the above inequality from $i = 1$ to $i = m$, we can further obtain

$$\begin{aligned}
&\frac{1}{\beta_k}\left(\sum_{i=1}^{m} A_i\tilde{x}_i^k - b\right)^{\mathrm{T}} \lambda^* \le \theta\left(\tilde{x}^k\right) - \theta\left(x^*\right) \\
&\le \frac{1}{\beta_k}\left(\sum_{i=1}^{m} A_i\tilde{x}_i^k - b\right)^{\mathrm{T}} \tilde{\lambda}^k + \sum_{i=1}^{m}\left\{\left(x_i^* - \tilde{x}_i^k\right)^{\mathrm{T}}\left[\beta_k A_i^{\mathrm{T}}\left(\sum_{j=2}^{i} A_j\left(\tilde{x}_j^k - x_j^k\right)\right)\right]\right\} \\
&= \frac{1}{\beta_k}\left(\sum_{i=1}^{m} A_i\tilde{x}_i^k - b\right)^{\mathrm{T}} \tilde{\lambda}^k + \beta_k\left[u\left(w^*\right) - u\left(\tilde{w}^k\right)\right]^{\mathrm{T}} \mathcal{L}\left[u\left(\tilde{w}^k\right) - u\left(w^k\right)\right],
\end{aligned} \tag{3.35}$$

where matrix $\mathcal{L}$ is defined as

$$\mathcal{L} = \begin{pmatrix} I_\ell & 0 & \cdots & \cdots & 0 \\ I_\ell & I_\ell & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ I_\ell & \cdots & \cdots & I_\ell & 0 \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}.$$

We know that

$$\left\|\sum_{i=1}^{m} A_i\tilde{x}_i^k - b\right\|_2 \overset{k\to\infty}{\longrightarrow} 0, \quad \left\|u\left(\tilde{w}^k\right) - u\left(w^k\right)\right\| \overset{k\to\infty}{\longrightarrow} 0,$$

together the boundedness of sequence $u\left(w^k\right)$, $u\left(\tilde{w}^k\right)$ and $u\left(w^*\right)$ (strictly-contraction-function property), we can conclude that both sides of (3.35) converge to 0. As a result, we guarantee that

$$\theta(\tilde{x}^k) - \theta(x^*) \overset{k\to\infty}{\longrightarrow} 0, \quad \forall x^* \in \mathcal{X}^*.$$

This finishes the proof of this theorem. □

Besides the global convergence of ADMM-G-V, the convergence rate is also another focus of attention. In the following two subsections, we will establish the worst-case iteration complexity in both the ergodic sense and the non-ergodic sense.

3.5.2 Convergence Rate in the Ergodic Sense

In Theorem 3.1, we can prove the equivalence of the following two conditions (see [36, Theorem 2.3.5] or [12, Theorem 2.1]), e.g., for all $w \in \mathcal{W}$

$$\tilde{w} \in \mathcal{W}, \quad \theta(x) - \theta(\tilde{x}) + (w - \tilde{w})^{\mathrm{T}} F(\tilde{w}) \geq 0,$$

which is equivalent to

$$\theta(x) - \theta(\tilde{x}) + (w - \tilde{w})^{\mathrm{T}} F(w) \geq 0.$$

We use the late one to define the approximate solution of VI. Namely, for given $\epsilon > 0$, $\tilde{w} \in \mathcal{W}$ is called an ϵ-approximate solution of VI $(\mathcal{W}, F, \theta)$, if it satisfies for all $w \in \mathcal{W}_{(\tilde{w})} := \left\{ w \in \mathcal{W} \;\middle|\; \|w - \tilde{w}\| \leq 1 \right\}$

$$\theta(x) - \theta(\tilde{x}) + (w - \tilde{w})^{\mathrm{T}} F(w) \geq -\epsilon.$$

We need to show that for given $\epsilon > 0$, after some iterations, it can offer a $\tilde{w} \in \mathcal{W}$, such that

$$\tilde{w} \in \mathcal{W}, \quad \sup_{w \in \mathcal{W}_{(\tilde{w})}} \left\{ \theta(\tilde{u}) - \theta(u) + (\tilde{w} - w)^{\mathrm{T}} F(w) \right\} \leq \epsilon. \tag{3.36}$$

Lemma 3.2 is the base for the convergence rate proof. Using the monotonicity of F, we have

$$\left(w - \tilde{w}^k\right)^{\mathrm{T}} F(w) \geq \left(w - \tilde{w}^k\right)^{\mathrm{T}} F\left(\tilde{w}^k\right).$$

Substituting it in (3.28), we obtain for all $w \in \mathcal{W}$

$$\theta(x) - \theta(\tilde{x}^k) + \left(w - \tilde{w}^k\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_k(w) - u_k(w^k) \right\|_{\mathcal{H}}^2 \geq \frac{1}{2\alpha} \left\| u_k(w) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2. \tag{3.37}$$

Lemma 3.3 *Let $\{w^k\}$ be the sequence generated by* **Algorithm-G-V** *for the problem (3.1) and $\tilde{w}^k$ is obtained in the $k+1$-th iteration, together with $u_k(w^k) = \mathcal{A}_k v^k$ and $u_k(\tilde{w}^k) = \mathcal{A}_k \tilde{v}^k$. Then, we have*

$$\sum_{k=0}^{t} \gamma_k^{t-1} \theta(\tilde{x}^k) - \sum_{k=0}^{t} \gamma_k^{t-1} \theta(x) + \left(\sum_{k=0}^{t} \gamma_k^{t-1} \tilde{w}^k - \sum_{k=0}^{t} \gamma_k^{t-1} w \right)^{\mathrm{T}} F(w) \leq \frac{\gamma_0^{t-1}}{2\alpha} \left\| u_0(w) - u_0(w^0) \right\|_{\mathcal{H}}^2, \tag{3.38}$$

where

$$\gamma_k^{t-1} = \prod_{j=k}^{t-1} (1 + \eta_j), \quad \gamma_t^{t-1} = 1. \tag{3.39}$$

Proof First, it holds that $\tilde{w}^k \in \mathcal{W}$ for all $k \geq 0$ and for all $w \in \mathcal{W}$

$$\theta(x) - \theta(\tilde{x}^k) + \left(w - \tilde{w}^k\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_k(w) - u_k(w^k) \right\|_{\mathcal{H}}^2 \geq \frac{1}{2\alpha} \left\| u_k(w) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2 .$$

According to (3.20), we have

$$\left\| u_k(w) - u_k(w^k) \right\|_{\mathcal{H}}^2 \leq (1 + \eta_{k-1}) \cdot \left\| u_{k-1}(w) - u_{k-1}(w^k) \right\|_{\mathcal{H}}^2 , \tag{3.40}$$

consequently, it follows that for all $w \in \mathcal{W}$

$$\begin{aligned} &\theta(\tilde{x}^k) - \theta(x) + \left(\tilde{w}^k - w\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_k(w) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2 \\ &\leq \frac{1}{2\alpha} \left\| u_k(w) - u_k(w^k) \right\|_{\mathcal{H}}^2 \leq \frac{1}{2\alpha} (1 + \eta_{k-1}) \left\| u_{k-1}(w) - u_{k-1}(w^k) \right\|_{\mathcal{H}}^2 . \end{aligned} \tag{3.41}$$

Therefore, we have the following sequential inequalities

$$\begin{cases} \theta(\tilde{x}^0) - \theta(x) + \left(\tilde{w}^0 - w\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_0(w) - u_0(w^1) \right\|_{\mathcal{H}}^2 \leq \frac{1}{2\alpha} \left\| u_0(w) - u_0(w^0) \right\|_{\mathcal{H}}^2 , \\ \theta(\tilde{x}^1) - \theta(x) + \left(\tilde{w}^1 - w\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_1(w) - u_1(w^2) \right\|_{\mathcal{H}}^2 \leq \frac{1}{2\alpha} (1 + \eta_0) \left\| u_0(w) - u_0(w^1) \right\|_{\mathcal{H}}^2 , \\ \qquad \vdots \qquad\qquad \vdots \\ \theta(\tilde{x}^k) - \theta(x) + \left(\tilde{w}^k - w\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_k(w) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2 \leq \frac{1}{2\alpha} (1 + \eta_{k-1}) \left\| u_{k-1}(w) - u_{k-1}(w^k) \right\|_{\mathcal{H}}^2 , \\ \qquad \vdots \qquad\qquad \vdots \\ \theta(\tilde{x}^t) - \theta(x) + \left(\tilde{w}^t - w\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_t(w) - u_t(w^{t+1}) \right\|_{\mathcal{H}}^2 \leq \frac{1}{2\alpha} (1 + \eta_{t-1}) \left\| u_{t-1}(w) - u_{t-1}(w^t) \right\|_{\mathcal{H}}^2 . \end{cases}$$

For the (k+1)-th inequality, we multiply it by a desirable factor $\gamma_k^{t-1} = \prod_{j=k}^{t-1} \left(1 + \eta_j\right)$, we can get

$$\begin{cases} \gamma_0^{t-1} \left\{\theta(\tilde{x}^0) - \theta(x) + \left(\tilde{w}^0 - w\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_0(w) - u_0(w^1) \right\|_{\mathcal{H}}^2 \right\} \leq \frac{1}{2\alpha} \gamma_0^{t-1} \left\| u_0(w) - u_0(w^0) \right\|_{\mathcal{H}}^2 , \\ \gamma_1^{t-1} \left\{\theta(\tilde{x}^1) - \theta(x) + \left(\tilde{w}^1 - w\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_1(w) - u_1(w^2) \right\|_{\mathcal{H}}^2 \right\} \leq \frac{1}{2\alpha} (1 + \eta_0) \gamma_1^{t-1} \left\| u_0(w) - u_0(w^1) \right\|_{\mathcal{H}}^2 , \\ \qquad \vdots \\ \gamma_k^{t-1} \left\{\theta(\tilde{x}^k) - \theta(x) + \left(\tilde{w}^k - w\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_k(w) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2 \right\} \leq \frac{1}{2\alpha} (1 + \eta_{k-1}) \gamma_k^{t-1} \left\| u_{k-1}(w) - u_{k-1}(w^k) \right\|_{\mathcal{H}}^2 , \\ \qquad \vdots \\ \gamma_t^{t-1} \left\{\theta(\tilde{x}^t) - \theta(x) + \left(\tilde{w}^t - w\right)^{\mathrm{T}} F(w) + \frac{1}{2\alpha} \left\| u_t(w) - u_t(w^{t+1}) \right\|_{\mathcal{H}}^2 \right\} \leq \frac{1}{2\alpha} (1 + \eta_{t-1}) \gamma_t^{t-1} \left\| u_{t-1}(w) - u_{t-1}(w^t) \right\|_{\mathcal{H}}^2 . \end{cases}$$

Using the notations in (3.39), the above inequalities can be rewritten as

$$\begin{cases} \gamma_0^{t-1}\left\{\theta(\tilde{x}^0)-\theta(x)+\left(\tilde{w}^0-w\right)^{\mathrm{T}}F(w)\right\}+\frac{\gamma_0^{t-1}}{2\alpha}\left\|u_0(w)-u_0(w^1)\right\|_{\mathcal{H}}^2 \le \frac{\gamma_0^{t-1}}{2\alpha}\left\|u_0(w)-u_0(w^0)\right\|_{\mathcal{H}}^2, \\ \gamma_1^{t-1}\left\{\theta(\tilde{x}^1)-\theta(x)+\left(\tilde{w}^1-w\right)^{\mathrm{T}}F(w)\right\}+\frac{\gamma_1^{t-1}}{2\alpha}\left\|u_1(w)-u_1(w^2)\right\|_{\mathcal{H}}^2 \le \frac{\gamma_0^{t-1}}{2\alpha}\left\|u_0(w)-u_0(w^1)\right\|_{\mathcal{H}}^2, \\ \qquad\qquad\qquad\qquad\qquad\qquad \vdots \qquad \vdots \\ \gamma_k^{t-1}\left\{\theta(\tilde{x}^k)-\theta(x)+\left(\tilde{w}^k-w\right)^{\mathrm{T}}F(w)\right\}+\frac{\gamma_k^{t-1}}{2\alpha}\left\|u_k(w)-u_k(w^{k+1})\right\|_{\mathcal{H}}^2 \le \frac{\gamma_{k-1}^{t-1}}{2\alpha}\left\|u_{k-1}(w)-u_{k-1}(w^k)\right\|_{\mathcal{H}}^2, \\ \qquad\qquad\qquad\qquad\qquad\qquad \vdots \qquad \vdots \\ \gamma_t^{t-1}\left\{\theta(\tilde{x}^t)-\theta(x)+\left(\tilde{w}^t-w\right)^{\mathrm{T}}F(w)\right\}+\frac{\gamma_t^{t-1}}{2\alpha}\left\|u_t(w)-u_t(w^{t+1})\right\|_{\mathcal{H}}^2 \le \frac{\gamma_{t-1}^{t-1}}{2\alpha}\left\|u_{t-1}(w)-u_{t-1}(w^t)\right\|_{\mathcal{H}}^2. \end{cases}$$

Adding all the above inequalities together, we get (3.38) and the lemma is proved. □

Theorem 3.3 *Let $\{w^k\}$ be the sequence generated by* **Algorithm-G-V** *for the problem (3.1) and $\tilde{w}^k$ is obtained in the $k+1$-th iteration, together with $u_k(w^k)=\mathcal{A}_k v^k$ and $u_k(\tilde{w}^k)=\mathcal{A}_k\tilde{v}^k$. Then, for any integer number $t>0$, we have for all $w\in\mathcal{W}$*

$$\theta(\tilde{x}_t)-\theta(x)+(\tilde{w}_t-w)^{\mathrm{T}}F(w)\le\frac{\gamma_0^{t-1}}{2\alpha\Upsilon_t}\left\|u_0(w)-u_0(w^1)\right\|_{\mathcal{H}}^2, \tag{3.42}$$

where

$$\tilde{w}_t=\frac{1}{\Upsilon_t}\sum_{k=0}^{t}\gamma_k^{t-1}\tilde{w}^k, \qquad \Upsilon_t=\sum_{k=0}^{t}\gamma_k^{t-1}, \tag{3.43}$$

and γ_k^{t-1} is defined in (3.39).

Proof Use the notation of $\tilde{w}_t$, (3.38) can be written as for all $w\in\mathcal{W}$

$$\frac{1}{\Upsilon_t}\sum_{k=0}^{t}\gamma_k^{t-1}\theta(\tilde{x}^k)-\theta(x)+(\tilde{w}_t-w)^{\mathrm{T}}F(w)\le\frac{\gamma_0^{t-1}}{2\alpha\Upsilon_t}\left\|u_0(w)-u_0(w^1)\right\|_{\mathcal{H}}^2. \tag{3.44}$$

Since $\theta(u)$ is convex and

$$\tilde{x}_t=\frac{1}{\Upsilon_t}\sum_{k=0}^{t}\gamma_k^{t-1}\tilde{x}^k,$$

we have that

$$\theta(\tilde{x}_t)\le\frac{1}{\Upsilon_t}\sum_{k=0}^{t}\gamma_k^{t-1}\theta(\tilde{x}^k).$$

Substituting it in (3.44), the assertion of this theorem follows directly. □

Further since

$$\gamma_0^{t-1}=\prod_{j=0}^{t-1}(1+\eta_j)\le C_p \quad \text{and} \quad \Upsilon_t=\sum_{k=0}^{t}\gamma_k^{t-1}\ge t+1,$$

and it follows from (3.42) that for all $w \in \mathcal{W}$

$$\theta(\tilde{x}_t) - \theta(u) + (\tilde{w}_t - w)^{\mathrm{T}} F(w) \le \frac{C_p}{2\alpha(t+1)} \left\| u_0(w) - u_0(w^1) \right\|_{\mathcal{H}}^2 \sim O\left(\frac{1}{t}\right).$$

3.5.3 Convergence Rate in a Non-ergodic Sense

At the beginning, we first discuss that the quantity $\left\| u_k(w^k) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2$ can be used to measure the accuracy of the iterate $\tilde{w}^k$ to a solution point of $VI(\mathcal{W}, F, \theta)$. More specifically, since $\mathcal{H}$ is positive definite, we conclude that $u_k(w^k) - u_k(w^{k+1}) = 0$ and $u_k(w^k) - u_k(\tilde{w}^k) = 0$ if $\left\| u_k(w^k) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2 = 0$. In other words, recall (3.25), we can obtain the following variational inequality characterization

$$\tilde{w}^k \in \mathcal{W},\ \theta(x) - \theta(\tilde{x}^k) + \left(w - \tilde{w}^k\right)^{\mathrm{T}} F\left(\tilde{w}^k\right) \ge 0, \quad \forall w \in \mathcal{W},$$

which indicates $\tilde{w}^k$ is a solution of $VI(\mathcal{W}, F, \theta)$. Therefore, $\left\| u_k(w^k) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2$ can be considered as an error measurement after k iterates of the ADMM-G-V scheme, and it is reasonable to seek an upper bound of $\left\| u_k(w^k) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2$ in term of the quantity of $O\left(\frac{1}{k}\right)$ for the purpose of investigating the convergence rate of ADMM-G-V. Based on this fact, we have the following theorem.

Theorem 3.4 *Let $\{w^k\}$ be the sequence generated by* **Algorithm-G-V** *for the problem (3.1) and $\tilde{w}^k$ is obtained in the $k+1$-th iteration, together with $u_k(w^k) = \mathcal{A}_k v^k$ and $u_k(\tilde{w}^k) = \mathcal{A}_k \tilde{v}^k$. Then, for any integer number $t > 0$, we have for all $w \in \mathcal{W}$*

$$\min_{1 \le k \le t} \left\{ \left\| u(w^k) - u(w^{k+1}) \right\|_{\mathcal{H}}^2 \right\} \le \frac{\alpha C_p^2 \beta_0}{(1-\alpha)\gamma_1^1 \cdot t} \left\| u_1(w^1) - u_1(w^*) \right\|_{\mathcal{H}}^2 \sim O\left(\frac{1}{t}\right). \tag{3.45}$$

Proof Recall the *strictly-contraction-function* property in Lemma 3.2, for any $w^* \in \mathcal{W}^*$

$$\left\| u_k(w^{k+1}) - u_k(w^*) \right\|_{\mathcal{H}}^2 \le \left\| u_k(w^k) - u_k(w^*) \right\|_{\mathcal{H}}^2 - \frac{1-\alpha}{\alpha} \left\| u_k(w^k) - u_k(w^{k+1}) \right\|_{\mathcal{H}}^2.$$

Mention again the step-size property (3.40)

$$\left\| u_k(w) - u_k(w^k) \right\|_{\mathcal{H}}^2 \le (1 + \eta_{k-1}) \cdot \left\| u_{k-1}(w) - u_{k-1}(w^k) \right\|_{\mathcal{H}}^2.$$

By applying this property in the above contractive property with $w = w^*$, we can obtain

$$\begin{aligned}
\frac{1}{\gamma_1^{k+1}}\left\|u_{k+1}(w^{k+1})-u_{k+1}(w^*)\right\|_{\mathcal{H}}^2 &\le \frac{1+\eta_k}{\gamma_1^{k+1}}\left\|u_k(w^{k+1})-u_k(w^*)\right\|_{\mathcal{H}}^2 \\
&\le \frac{1}{\gamma_1^k}\left\|u_k(w^k)-u_k(w^*)\right\|_{\mathcal{H}}^2 - \frac{1-\alpha}{\alpha\gamma_1^k}\left\|u_k(w^k)-u_k(w^{k+1})\right\|_{\mathcal{H}}^2 .
\end{aligned}$$

Summing the above inequality from $k = 1$ to $k = t$, we have

$$\begin{aligned}
\frac{1}{\gamma_1^{t+1}}\left\|u_{t+1}(w^{t+1})-u_{t+1}(w^*)\right\|_{\mathcal{H}}^2 &\le \frac{1}{\gamma_1^1}\left\|u_1(w^1)-u_1(w^*)\right\|_{\mathcal{H}}^2 - \sum_{k=1}^{t}\frac{1-\alpha}{\alpha\gamma_1^k}\left\|u_k(w^k)-u_k(w^{k+1})\right\|_{\mathcal{H}}^2 \\
&\le \frac{1}{\gamma_1^1}\left\|u_1(w^1)-u_1(w^*)\right\|_{\mathcal{H}}^2 - \sum_{k=1}^{t}\frac{1-\alpha}{\alpha C_p}\left\|u_k(w^k)-u_k(w^{k+1})\right\|_{\mathcal{H}}^2 ,
\end{aligned}$$

further we have

$$\begin{aligned}
& t\cdot\min\left\{\frac{1-\alpha}{\alpha C_p^2\beta_0}\left\|u(w^k)-u(w^{k+1})\right\|_{\mathcal{H}}^2\right\} \\
&\le t\cdot\min\left\{\frac{(1-\alpha)\min\left\{\beta_k,\frac{1}{\beta_k}\right\}}{\alpha C_p}\left\|u(w^k)-u(w^{k+1})\right\|_{\mathcal{H}}^2\right\} \\
&\le t\cdot\min\left\{\frac{1-\alpha}{\alpha C_p}\left\|u_k(w^k)-u_k(w^{k+1})\right\|_{\mathcal{H}}^2\right\} \\
&\le \sum_{k=1}^{t}\frac{1-\alpha}{\alpha C_p}\left\|u_k(w^k)-u_k(w^{k+1})\right\|_{\mathcal{H}}^2 \\
&\le \frac{1}{\gamma_1^1}\left\|u_1(w^1)-u_1(w^*)\right\|_{\mathcal{H}}^2 ,
\end{aligned}$$

which directly supports the above assertion (3.45). □

3.6 Numerical Experiments

We consider the consensus problem over a network $\mathcal{N}$ as follows

$$\min_{x\in\mathbb{R}^n} f(x) := \sum_{i=1}^{m} f_i(x). \tag{3.46}$$

This multi-agent system with m agents and the set of agents is denoted by $\mathcal{V} = \{1,\cdots,m\}$. The agents communicate with each other via this undirected and connected graph topology $\mathcal{G} = (\mathcal{V},\mathcal{E})$ where $\mathcal{E}$ is the edge set with $\mathcal{E}\subset\mathcal{V}\times\mathcal{V}$. Each agent i has a local objective function $f_i(x)$. The agents coordinate with each other to minimize the global objective function $f(x)$ defined above. In order to apply our proposed algorithm framework ADMM-G-V, we first equivalently reformulate (3.46) into the following linear constrained separable problem, i.e.,

$$\begin{aligned} \min_{x_i \in \mathbb{R}^n} \ & \sum_{i=1}^{m} f_i(x_i), \\ s.t. \ & x_i = x_j, (i, j) \in \mathcal{E}, i < j. \end{aligned} \tag{3.47}$$

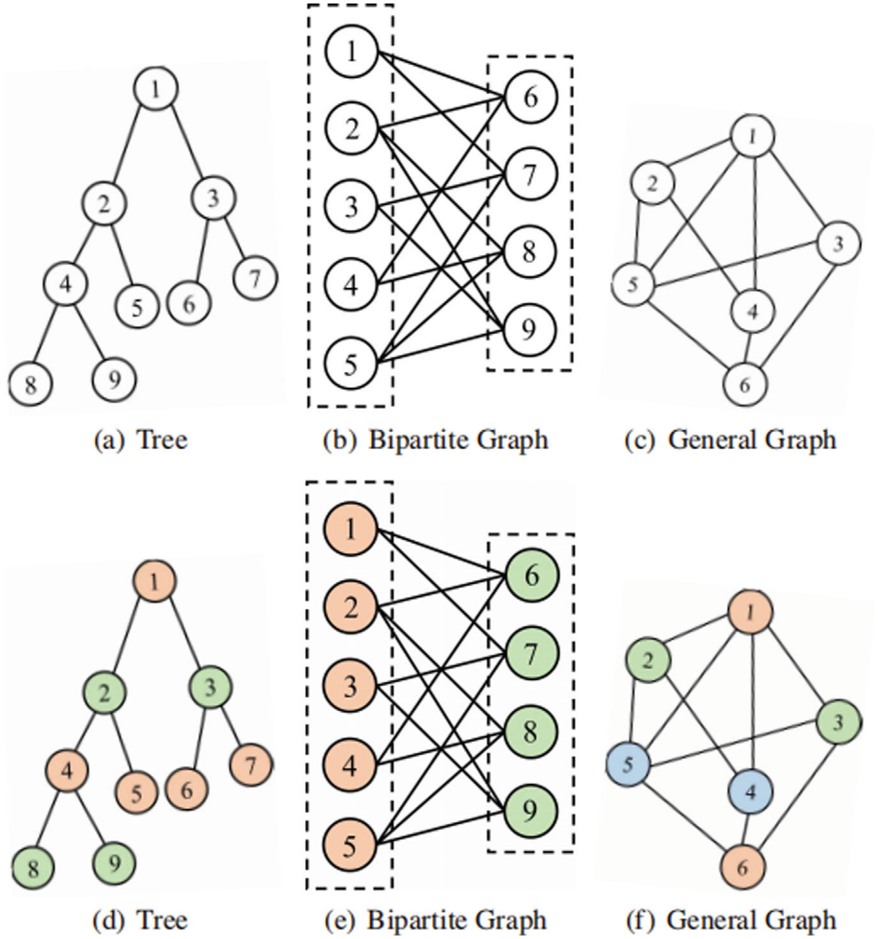
In this reformulation, each edge is considered only once; as a result, the $i < j$ constraint is additionally added.

3.6.1 Undirected and Connected Graph $\mathcal{G}$

Equation (3.47) can be considered as an m-blocks separable problem which is a special case of the general problem (3.1). However, if we take into consideration the network structure, (3.47) could be formulated as a multiple block separable problem with fewer blocks. By taking the following three specific graph examples in Fig. 3.3, we obtain the following observations:

1. **Tree**: When the network structure is a tree graph as in Fig. 3.3a, we can separate the nodes into two groups, e.g., $\mathcal{V}_1 = \{1, 4, 5, 6, 7\}$ and $\mathcal{V}_2 = \{2, 3, 8, 9\}$ as in Fig. 3.3d. It is obvious that the nodes in $\mathcal{V}_1$ and $\mathcal{V}_2$ only have a relationship with the nodes in another group, not with the nodes in their own group. In other words, problem (3.47) can be formulated into a two-block separable problem, not a multiple block problem.
2. **Bipartite Graph**: When the network structure is a bipartite graph as in Fig. 3.3b, we can separate the nodes into two groups, e.g., $\mathcal{V}_1 = \{1, 2, 3, 4, 5\}$ and $\mathcal{V}_2 = \{6, 7, 8, 9\}$ as in Fig. 3.3e. It is obvious that the nodes in $\mathcal{V}_1$ and $\mathcal{V}_2$ only have a relationship with the nodes in another group, not with the nodes in their own group. In other words, problem (3.47) can also be formulated into a two-block separable problem, not a multiple block problem;
3. **General Graph**: When the network structure is a general graph as in Fig. 3.3c with 6 nodes, we can separate the nodes into three groups, e.g., $\mathcal{V}_1 = \{1, 6\}$, $\mathcal{V}_2 = \{2, 3\}$ and $\mathcal{V}_3 = \{4, 5\}$ as in 3.3f. It is obvious that the nodes in $\mathcal{V}_1$, $\mathcal{V}_2$, and $\mathcal{V}_3$ only have relationships with the nodes in other groups, not with the nodes in their own group. In other words, problem (3.47) can be formulated into a three-block separable problem while $\{x_1, x_6\}$, $\{x_2, x_3\}$ and $\{x_4, x_5\}$ are considered as three variable blocks.

All these examples show that with proper rearrangement (3.47) can be considered as a multiple block problem with much less block number than the node number in the constraint graph $\mathcal{G}$. The greedy algorithm can be used to determine the groups in



(a) Tree (b) Bipartite Graph (c) General Graph

(d) Tree (e) Bipartite Graph (f) General Graph

Fig. 3.3 Undirected and connected graph examples $\mathcal{G}$

$\mathcal{G}$ by considering this problem as the graph coloring problem [38]. We assume that the nodes $\mathcal{V}$ in graph $\mathcal{G}$ can be separated into m_r groups, i.e., $\left\{\mathcal{V}_1, \ldots, \mathcal{V}_{m_r}\right\}$, as a result problem (3.47) can be further reformulated as

$$\min_{x_i \in \mathbb{R}^n} \sum_{i=1}^{m_r} \left\{ \sum_{j \in \mathcal{V}_i} f_j(x_j) \right\}, \quad s.t. \quad \sum_{i=1}^{m_r} \left\{ \sum_{j \in \mathcal{V}_i} \mathcal{A}_j x_j \right\} = 0, \tag{3.48}$$

where $\mathcal{A}_j \in \mathbb{R}^{mn \times n}$ and

$$\mathcal{A}_j = \begin{bmatrix} A_{1j} \\ \vdots \\ A_{mj} \end{bmatrix}, \quad A_{\ell j} \in \mathbb{R}^{n\times n}$$

and

$$A_{\ell j} = \begin{cases} -\mathcal{I}_{n\times n} & \text{if } (\ell, j) \in \mathcal{E},\ \ell < j, \\ \mathcal{I}_{n\times n} & \text{if } (\ell, j) \in \mathcal{E},\ \ell > j, \\ O_{n\times n} & \text{otherwise,} \end{cases} \qquad \forall\, 1 \le \ell \le n, 1 \le j \le n.$$

3.6.2 Distributed Logistic Regression

Logistic regression (LR) is a popular classification method that has been widely introduced to lots of machine learning applications, including computer vision, natural language processing and etc. The distributed Logistic regression problem considered in this paper implies that the datasets are separated in different nodes; however, we need to learn the overall LR parameter in order to guarantee a complete model. Given m distributed datasets $\{(\mathbf{A_i}, \mathbf{b_i})\}_{i=1}^m$, where $\mathbf{A_i} = \left\{\mathbf{a_i^j}\right\}_{j=1}^{N_i}$ denotes the data samples distributed in node i with $\mathbf{a_i^j} \in \mathbb{R}^\mathbf{n}$ and $\mathbf{b_i} = \left\{\mathbf{b_i^j}\right\}_{j=1}^{N_i}$ denotes the label set with $\mathbf{b_i^j} \in \{+\mathbf{1}, -\mathbf{1}\}$. We have the following formulation of each f_i, i.e.,

$$f_i(x_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} \log\left(1 + \exp\left(-\mathbf{b_i^j}\left(\mathbf{x_i^T a_i^j}\right)\right)\right),$$

where $x_i \in \mathbb{R}^n$ denotes the parameter with respect to the i-th dataset.

3.6.3 Datasets, Network Structure and Computing Environment

We apply the model (3.47) on two real datasets. The first is the RCV1 dataset, which is used for text binary classification research [39]. The other dataset is the large-scale URL dataset, which is used for identifying suspicious URLs [40]. The detailed sample number and feature dimension are stated in the following Table 3.2. We notice that both the sample number and feature dimension are huge for the URL dataset, but because of the limitation of processing capacity in each node, we only randomly choose 160,000 data samples for this numerical experiment.

After establishing the graph $\mathcal{G}$, we will separate the datasets into the nodes in $\mathcal{G}$.

Table 3.2 Description of real datasetsDatasets

Dataset	RCV1	URL
#Examples	20000	2.4M
#Features	47236	3.2M
Label ratio +1: -1	1:1	1:2

The connected network $\mathcal{G}$ is randomly generated with $m = 100$ nodes and connectivity ratio $r = 0.2$. Both datasets are randomly and uniformly separated into m nodes, i.g., $N_i = 200$ and $N_i = 24,000$ for each dataset, respectively. By applying the greedy algorithm, we can dye the generated graph with κ colors (without loss of generality, we assume $\kappa > 2$). Further, we can consider the guaranteed problem (3.47) as a κ-block linear constrained separable problem ($\kappa = 17$ for our generated problem). Our algorithm framework will be employed to solve this distributed Logistic regression problem on the generated network $\mathcal{E}$.

All the numerical experiments were implemented on a Laptop with Intel(R) Core(TM) i5-6300U CPU@ 2.40GHz 2.50GHz and 8.00 GB Memory. All the codes were written in MATLAB2016A. We compare with two other algorithms: direct extension of multiple block ADMM (**D-ADMM**) and Algorithm 1 with constant β (**Constant-beta**). All the three β adjustment schemes (3.22), (3.23), (3.24) will the compared, which all start from $\beta = 1$. The stopping criteria for all the algorithms are set to be

$$\max\left\{\|p_k\|_2, \|d_k\|_2\right\} \le 10^{-6},$$

while the iteration number ("Iter. #"), objective function value ("Objective"), the constraint violation value $\|d_k\|_2$ ("Constraint") and the computation time ("Time(s)") will be presented in the following tables.

3.6.4 Numerical Performance

In the above Table 3.3, generally **Algorithm 1** with self-adaptive β schemes perform better than multiple block ADMM with Gaussian back substitution technique, even better the direct extension of multiple block ADMM. In Table 3.4, we compare **Algorithm 1** for β schemes (3.22) (3.23) and (3.24) with respect to different t. We can find that although for some cases (like (3.22) with $t = 2$ and (3.23) with $t = 2$), a less self-adaptive frequency can guarantee faster convergence. However, overall the self-adaptive β scheme is not computationally consuming for our experiment, so that performing the self-adaptive scheme in each iteration ($t = 1$) will accelerate the algorithm framework.

Table 3.3 Numerical comparison of **D-ADMM** and **Algorithm 1** for different β scheme

Dataset	Algorithm	β_k	Iter. #	Objective	Constraint	Time(s)
RCV1	D-ADMM	10	721	230.0599	9.76e-07	551.15
		1	337	230.0599	9.71e-07	283.14
		0.1	2000	229.8048	2.46e-06	1565.57
	Algorithm 1	10	811	230.0602	9.35e-07	606.42
		1	382	230.0600	9.37e-07	327.93
		0.1	1839	230.0600	9.76e-07	1695.98
		(3.22)	223	230.0600	9.29e-07	236.93
		(3.23)	331	230.0600	9.46e-07	276.69
		(3.24)	136	230.0600	9.82e-07	153.21
URL	D-ADMM	10	889	478.4982	9.78e-07	1122.43
		1	529	478.4974	9.65e-07	632.14
		0.1	2000	477.9725	1.95e-05	2756.87
	Algorithm 1	10	1188	478.4980	9.33e-07	1671.57
		1	654	478.4978	9.45e-07	776.42
		0.1	1976	478.4878	9.69e-07	3097.53
		(3.22)	373	478.4978	9.65e-07	527.82
		(3.23)	362	478.4978	9.57e-07	496.45
		(3.24)	228	478.4978	9.54e-07	332.39

Table 3.4 Numerical comparison of **Algorithm 1** for β schemes (3.22), (3.23) and (3.24) with respect to different t

Dataset	Algorithm	β_k	t	Iter. #	Objective	Constraint	Time(s)
RCV1	**Algorithm 1**	(3.22)	1	223	230.0600	9.29e-07	236.93
			2	246	230.0600	9.67e-07	224.38
			5	313	230.0600	9.31e-07	245.29
		(3.23)	1	331	230.0600	9.46e-07	276.69
			2	347	230.0600	9.49e-07	280.16
			5	362	230.0600	9.37e-07	311.84
		(3.24)	1	136	230.0600	9.82e-07	153.21
			2	175	230.0600	9.75e-07	176.87
			5	224	230.0600	9.81e-07	226.25
URL	**Algorithm 1**	(3.22)	1	373	478.4978	9.65e-07	527.82
			2	423	478.4978	9.57e-07	578.57
			5	469	478.4978	9.69e-07	620.67
		(3.23)	1	362	478.4978	9.57e-07	496.45
			2	396	478.4978	9.29e-07	503.54
			5	423	478.4978	9.42e-07	559.58
		(3.24)	1	228	478.4978	9.54e-07	332.39
			2	247	478.4978	9.46e-07	329.63
			5	284	478.4978	9.39e-07	393.54

3.7 Conclusion and Future Work

In this paper, we propose a theoretical analysis of the alternating direction method of multipliers with a Gaussian back substitution step for the variable penalty parameters case. Both non-ergodic and ergodic worst-case convergence rates are established. Specifically, classical two-block ADMM with variable penalty parameters can be included in the framework. A better penalty parameter tuning mechanism should be further analyzed and considered.

Acknowledgements This work was supported by the NSFC with No. 12071145.

References

1. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)
2. Peng, Y.G., Ganesh, A., Wright, J., Wenli, X., Ma, Y.: RASL: robust alignment by sparse and low-rank decomposition for linearly correlated images. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2233–2246 (2012)
3. Zhang, Z., Ganesh, A., Liang, X., Ma, Y.: TILT: transform invariant low-rank textures. Int. J. Comput. Vis. **99**(1), 1–24 (2012)
4. Glowinski, R., Marrocco, A.: Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problémes non linéaires. R.A.I.R.O. R2 **60**(8), 41–76 (1975)
5. Eckstein, J., Yao, W.: Understanding the convergence of the alternating direction method of multipliers: theoretical and computational perspectives. Pac. J. Optim. **11**, 619–644 (2015)
6. Glowinski, R.: On alternating direction methods of multipliers: a historical perspective. In: Modeling, Simulation and Optimization for Science and Technology, pp. 59–82 (2014)
7. Glowinski, R., Marrocco, A.: Chapter IX applications of the method of multipliers to variational inequalities. Stud. Math. Appl. **15**, 299–331 (1983)
8. Glowinski, R., Oden, J.T.: Numerical methods for nonlinear variational problems. J. Appl. Mech. **52**, 739 (1985)
9. Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**(1), 293–318 (1992)
10. Martinet, B.: Brève communication. régularisation d'inéquations variationnelles par approximations successives. ESAIM: Mathematical Modelling and Numerical Analysis—Modélisation Mathématique et Analyse Numérique **4**(R3), 154–158 (1970)
11. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control. Optim. **14**(5), 877–898 (1976)
12. He, B.S., Yuan, X.M.: On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. SIAM J. Numer. Anal. **50**(2), 700–709 (2012)
13. He, B.S., Yuan, X.M.: On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers. Numerische Mathematik **130**(3), 567–577 (2015)
14. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. **4**(5), 303–320 (1969)
15. Tao, M., Yuan, X.M.: Recovering low-rank and sparse components of matrices from incomplete and noisy observations. SIAM J. Optim. **21**(1), 57–81 (2011)
16. Han, D.R., Cai, X.J., Yuan, X.M.: On the convergence of the direct extension of ADMM for three-block separable convex minimization models with one strongly convex function. Comput. Optim. Appl. **66**(1), 39–73 (2017)

17. Han, D.R., Yuan, X.M.: A note on the alternating direction method of multipliers. J. Optim. Theory Appl. **155**(1), 227–238 (2012)
18. Sun, D.F., Li, M., Toh, K.-C.: A convergent 3-block semi-proximal ADMM for convex minimization problems with one strongly convex block. Asia Pac. J. Oper. Res. **32**(4), 1550024 (2015)
19. Hong, M.Y., Luo, Z.-Q.: On the linear convergence of the alternating direction method of multipliers. Math. Program. **162**(1), 165–199 (2017)
20. Yuan, X., Ye, Y., He, B., Chen, C.: The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. Math. Program. **155**(1–2), 57–79 (2016)
21. Yang, H., He, B.S., Wang, S.L.: Alternating direction method with self-adaptive penalty parameters for monotone variational inequalitiest. J. Optim. Theory Appl. **106**(2), 337–356 (2000)
22. Yuan, X.M., Ng, M.K., Zhang, W.X.: Coupled variational image decomposition and restoration model for blurred cartoon-plus-texture images with missing pixels. IEEE Trans. Image Process. **22**(6), 2233–2246 (2013)
23. He, B., Liao, L.-Z., Han, D., Yang, H.: A new inexact alternating directions method for monotone variational inequalities. Math. Program. **92**(1), 103–118 (2002)
24. He, B.S., Yang, H.: Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities. Oper. Res. Lett. **23**(3), 151–161 (1998)
25. Kontogiorgis, S., Meyer, R.R.: A variable-penalty alternating directions method for convex optimization. Math. Program. **83**(1–3), 29–153 (1998)
26. Yoon, S., Song, C., Pavlovic, V.: Fast ADMM algorithm for distributed optimization with adaptive penalty. In: AAAI, pp. 753–759 (2016)
27. Figueiredo, M., Xu, Z., Goldstein, T.: Adaptive ADMM with spectral penalty parameter selection. In: AISTATS, pp. 718–727 (2017)
28. Ye, J., Li, X., Hu, Y., Zhang, D., He, X.: Fast and accurate matrix completion via truncated nuclear norm regularization. IEEE Trans. Pattern Anal. Mach. Intell. **35**(9), 2117–2130 (2013)
29. Meshi, O., Globerson, A.: An alternating direction method for dual MAP LP relaxation. In: Machine Learning and Knowledge Discovery in Databases, pp. 470–483 (2011)
30. Wohlberg, B.: Efficient algorithms for convolutional sparse representations. IEEE Trans. Image Process. **25**(1), 301–315 (2016)
31. Guibas, L., Solomon, J., Rustamov, R., Butscher, A.: Earth mover's distances on discrete surfaces. ACM Trans. Graph. **33**(4), 67 (2014)
32. Zhang, B., Sun, H., Zheng, W., Wu, W., Liu, Y.: A fully distributed reactive power optimization and control method for active distribution networks. IEEE Trans. Smart Grid **7**(2), 1021–1033 (2016)
33. Bioucas-Dias, J.M., Iordache, M.D., Plaza, A.: Collaborative sparse regression for hyperspectral unmixing. IEEE Trans. Geosci. Remote. Sens. **52**(1), 341–354 (2014)
34. Wu, Y., Tang, W., Shi, Z., Zhang, C.: Sparse unmixing of hyperspectral data using spectral a priori information. IEEE Trans. Geosci. Remote. Sens. **53**(2), 770–783 (2015)
35. Tao, M., He, B.S., Yuan, X.M.: Convergence rate analysis for the alternating direction method of multipliers with a substitution procedure for separable convex programming. Math. Oper. Res. **42**(3), 662–691 (2017)
36. Facchinei, F., Pang, J.-S.: Finite-Dimensional Variational Inequalities and Complementarity Problems, vol. 1. Springer Science & Business Media (2007)
37. Tao, M., He, B.S., Yuan, X.M.: Alternating direction method with Gaussian back substitution for separable convex programming. SIAM J. Optim. **22**(2), 313–340 (2012)
38. West, D.B.: Introduction to Graph Theory, 2nd edn. Prentice Hall (1996)
39. Rose, T.G., Lewis, D.D., Yang, Y., Li, F.: Rcv1: a new benchmark collection for text categorization research. J. Mach. Learn. Res. **5**, 361–397 (2004)
40. Savage, S., Ma, J., Saul, L.K., Voelker, G.M.: Identifying suspicious URLs: an application of large-scale online learning. In: ICML, pp. 681–688 (2009)

Chapter 4
Inertial Alternating Direction Method of Multipliers with Logarithmic-Quadratic Proximal Regularization

Zhongming Wu

Abstract It has been demonstrated that the alternating direction method of multipliers (ADMM) with logarithmic-quadratic proximal (LQP) regularization is efficient in solving a specific class of separable convex optimization problems. This method capitalizes on the individual separable properties and transforms the constrained subproblems into more manageable unconstrained subproblems during the iterative process. In this paper, we investigate the application of the inertial proximal point method and focus on studying the inertial ADMM and symmetric ADMM with LQP regularization for solving constrained separable convex optimization problems. These approaches employ ADMM or symmetric ADMM on extrapolated points with appropriate step sizes to accelerate the convergence rate. Under some mild conditions, we establish the global convergence of the proposed methods.

Keywords Alternating direction method of multipliers · Logarithmicquadratic proximal regularization · Inertial · Convergence analysis

4.1 Introduction

In this paper, we consider the following separable constrained optimization problem

$$\min \left\{ \theta_1(x) + \theta_2(y) \mid Ax + By = b, x \in \mathbb{R}^n_+, y \in \mathbb{R}^p_+ \right\}, \tag{4.1}$$

where $\theta_1 : \mathbb{R}^n_+ \to \mathbb{R}$ and $\theta_2 : \mathbb{R}^p_+ \to \mathbb{R}$ are closed convex functions, $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times p}$ and $b \in \mathbb{R}^m$. The solution set of (4.1) is assumed to be nonempty throughout our discussions in this paper. The augmented Lagrangian function of the problem (4.1) is defined by

Z. Wu (✉)
School of Management Science and Engineering, Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China
e-mail: wuzm@nuist.edu.cn

X. Wang et al. (eds.), *Splitting Optimization*, ICIAM2023 Springer Series 2,
https://doi.org/10.1007/978-981-96-8574-5_4

$$\mathcal{L}_\beta(x, y, \lambda) = \theta_1(x) + \theta_2(y) - \lambda^\top(Ax + By - b) + \frac{\beta}{2}\|Ax + By - b\|^2, \quad (4.2)$$

where $\lambda \in \mathbb{R}^m$ is the Lagrange multiplier associated with the linear constraint in (4.1) and $\beta > 0$ is a penalty parameter. Problem (4.1) captures many applications in various fields such as financial portfolio optimization [34] and traffic management [40, 47].

The alternating direction method of multipliers (ADMM), originally proposed in [22, 23], is widely recognized for its efficient solution to (4.1). This method leverages the individual properties of θ_1 and θ_2. For a comprehensive understanding of ADMM's theoretical analysis and diverse applications, we refer readers to [11, 13, 20, 26, 29, 30, 46] and the relevant references therein. In this paper, we introduce a cyclically equivalent form of ADMM, with further details on this cyclic equivalence provided in [12, 14]. The iterative scheme of ADMM in "$x \to \lambda \to y$" order for (4.1) can be read as

$$\begin{cases} x^{k+1} = \arg\min\{\mathcal{L}_\beta(x, y^k, \lambda^k) \mid x \in \mathbb{R}^n_+\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^k - b), \\ y^{k+1} = \arg\min\{\mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+1}) \mid y \in \mathbb{R}^p_+\}. \end{cases} \quad (4.3)$$

Another influential and important method for solving problem (4.1) is the Peaceman-Rachford splitting (PRS) method, initially proposed in [37, 41]. When applying the PRS method to the dual of the considered problem, we obtain another alternating method, known as symmetric ADMM [18, 24, 28, 33, 49, 50]. The iterative scheme of symmetric ADMM for solving (4.1) can be read as follows:

$$\begin{cases} x^{k+1} = \arg\min\{\mathcal{L}_\beta(x, y^k, \lambda^k) \mid x \in \mathbb{R}^n_+\}, \\ \lambda^{k+1/2} = \lambda^k - \beta(Ax^{k+1} + By^k - b), \\ y^{k+1} = \arg\min\{\mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+1/2}) \mid y \in \mathbb{R}^p_+\}, \\ \lambda^{k+1} = \lambda^{k+1/2} - \beta(Ax^{k+1} + By^{k+1} - b). \end{cases} \quad (4.4)$$

The symmetric ADMM (4.4) differs from ADMM due to the additional Lagrange multiplier update step, as analyzed in [22]. It has been observed that the symmetric ADMM converges faster than ADMM if it does converge [21, 27]. However, symmetric ADMM (4.4) may fail to converge without further assumptions. To address this limitation, He et al. [27] introduced an underdetermined relaxation factor $\gamma \in (0, 1)$ to both Lagrange multiplier updating steps of (4.4). The parameter $\gamma \in (0, 1)$ plays a crucial role in ensuring the strict contractiveness of the generated sequence. Under this condition, they established convergence results and demonstrated the same sublinear convergence rate as that of ADMM [29].

For the generic convex objective function in (4.1), the constrained subproblems in (4.3) and (4.4) typically require iterative and approximate solutions. However, due to the specific nature of the constraints in these (symmetric) ADMM subproblems, several studies [6–8, 36, 38, 47, 51] propose appropriate regularization of the objective functions to strictly confine the solutions within the interiors of $\mathbb{R}^n_+$ and $\mathbb{R}^p_+$. This regularization allows the constrained subproblems in (4.3) and (4.4) to be converted into unconstrained subproblems. An excellent choice for such regularization is the logarithmic-quadratic proximal (LQP) regularization, initially proposed in [6], and extensively studied in various articles such as [4, 5, 51]. The iterative scheme of ADMM with LQP regularization is as follows:

$$\begin{cases} x^{k+1} = \arg\min\{\mathcal{L}_\beta(x, y^k, \lambda^k) + rd(x, x^k) \mid x \in \mathbb{R}^n_+\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^k - b), \\ y^{k+1} = \arg\min\{\mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+1}) + sd(y, y^k) \mid y \in \mathbb{R}^p_+\}, \end{cases} \tag{4.5}$$

and the iterative scheme of symmetric ADMM with LQP regularization is

$$\begin{cases} x^{k+1} = \arg\min\{\mathcal{L}_\beta(x, y^k, \lambda^k) + rd(x, x^k) \mid x \in \mathbb{R}^n_+\}, \\ \lambda^{k+1/2} = \lambda^k - \beta(Ax^{k+1} + By^k - b), \\ y^{k+1} = \arg\min\{\mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+1/2}) + sd(y, y^k) \mid y \in \mathbb{R}^p_+\}, \\ \lambda^{k+1} = \lambda^{k+1/2} - \beta(Ax^{k+1} + By^{k+1} - b), \end{cases} \tag{4.6}$$

where $d(\cdot, \cdot)$ is the LQP function defined by

$$d(z', z) := \begin{cases} \sum\limits_{j=1}^{N} \left[\frac{1}{2}\left(z'_j - z_j\right)^2 + \mu\left(z_j^2 \log \frac{z_j}{z'_j} + z'_j z_j - z_j^2\right)\right], & \text{if } z' \in \mathbb{R}^N_{++}; \\ +\infty, & \text{otherwise}, \end{cases} \tag{4.7}$$

for any $z \in \mathbb{R}^N_{++}$, r and s are two positive scalars. Indeed, the LQP regularization terms $rd(x, x^k)$ and $sd(y, y^k)$ inherently constrain x^{k+1} and y^{k+1} to belong to $\mathbb{R}^{n_1}_{++}$ and $\mathbb{R}^{n_2}_{++}$, respectively, rendering the constraints $x \in \mathbb{R}^{n_1}_+$ and $y \in \mathbb{R}^{n_2}_+$ inactive. Consequently, only two unconstrained minimization subproblems are involved in (4.5). The convergence and the rate of convergence of these two methods have been widely discussed in the literature [15, 35, 36, 44].

Proximal point algorithm (PPA) [43, 52] is a closely related method with the existing splitting methods such as ADMM and Douglas-Rachford splitting method [19], which is to minimize a differentiable function $\psi : \mathbb{R}^n \to \mathbb{R}$, it can be interpreted as an implicit one-step discretization method for the ordinary differential equation $w' + \nabla\psi(w) = 0$, where $w : \mathbb{R} \to \mathbb{R}^n$ is differentiable, w' denotes its derivative and $\nabla\psi$ denotes the gradient of ψ. To accelerate the convergence speed of the PPA, one can adopt the multi-step methods, which

$$w^{k+1} = (I + \sigma\nabla\psi)^{-1}(w^k + \alpha(w^k - w^{k-1})), \tag{4.8}$$

where $\sigma = h^2/(1 + \tau h), \alpha = 1/(1 + \tau h)$ and I represents the identity operator. Note that the iterative scheme (4.8) can be regarded as applying the PPA to the extrapolated point $w^k + \alpha(w^k - w^{k-1})$, which is usually the so-called inertial PPA [1, 16]. Indeed, this inertial technique can be traced back to [42], and recently there has been extensive work in studying inertial-type algorithms, including the inertial forward-backward splitting method [2, 10, 39, 48], inertial Douglas-Rachford splitting method [9], and so forth.

Inspired by the inertial PPA aforementioned, Chen et al. [14] introduced an inertial proximal ADMM to enhance the convergence speed of ADMM (4.3). This approach can be seen as applying ADMM (4.3) to an extrapolated point with an appropriate step size. Subsequently, the inertial ADMM has been further investigated in [3], where the authors proposed an inertial proximal ADMM with an appropriate adjustment of the viscosity and proximal parameters. The fast convergence properties, as well as the convergence of the iterates to saddle points of the Lagrangian function, can be guaranteed. Recently, the inertial acceleration techniques have been extended to handle nonconvex optimization problems with linear constraints [17, 31, 32, 45]. However, the resulting method still requires iterative and approximate solutions for the constrained subproblems.

In this paper, we apply the inertial technique to ADMM and symmetric ADMM with LQP regularization for solving the linearly constrained separable convex optimization problem (4.1). These proposed methods not only enforce the solutions of the constrained subproblems in (4.3) and (4.4) to remain strictly within the interiors of $\mathbb{R}^n_+$ and $\mathbb{R}^p_+$, facilitating their conversion into easier unconstrained subproblems, but also accelerate the convergence speed of the iterative schemes (4.5) and (4.6) through extrapolation-based steps. Under mild assumptions, we establish the global convergence of the proposed methods.

The remainder of this paper is organized as follows. Section 4.2 provides a summary of relevant preliminary results and introduces the definition of LQP regularization. In Sects. 4.3 and 4.4, we present the inertial ADMM and symmetric ADMM with LQP regularization, respectively, and analyze their convergence properties. Finally, Sect. 4.5 concludes this paper.

4.2 Preliminaries

We first summarize some useful preliminaries known in the literature and present some simple conclusions for further analysis.

4.2.1 The Logarithmic-Quadratic Proximal Regularization

In this subsection, we give some basic knowledge about the LQP regularization. More details are also provided in [6]. Let us define

$$\varphi(c) := \begin{cases} \frac{1}{2}(c-1)^2 + \mu(c - \log c - 1), & \text{if } c > 0; \\ +\infty, & \text{otherwise,} \end{cases}$$

where $\mu \in (0, 1)$ is a given constant. Associated with φ, we define

$$\Phi'(z, z') := (z_1\varphi'(z'_1/z_1), \ldots, z_N\varphi'(z'_N/z_N))^\top, \quad \forall z, z' \in \mathbb{R}^N_{++},$$

where

$$\varphi'(z'_j/z_j) = z'_j/z_j - 1 + \mu(1 - z'_j/z_j), \quad j = 1, 2, \ldots, N.$$

For any $z', z \in \mathbb{R}^N_{++}$, we have $d(z', z) \geq \|z' - z\|^2/2$ and $d(z', z) = 0$ if and only if $z' = z$. Moreover, the function $d(\cdot, \cdot)$ defined in (4.7) can be rewritten as

$$d(z', z) = \sum_{j=1}^{N} z_j^2 \varphi(z'_j/z_j), \quad \forall z', z \in \mathbb{R}^N_{++},$$

and then we have

$$\Phi'(z, z') = \nabla_{z'} d(z', z) = (z' - z) + \mu[z - Z^2(z')^{-1}],$$

where $Z := \text{diag}(z_1, z_2, \ldots, z_N) \in \mathbb{R}^{N\times N}$, $(z')^{-1} \in \mathbb{R}^N$ is a vector whose j-th element is $1/z'_j$.

We now summarize an important lemma, proven in [36], which will be useful for the convergence analysis in subsequent sections.

Lemma 4.1 *Let* $P := \text{diag}(p_1, p_2, \ldots, p_N) \in \mathbb{R}^{N\times N}$ *be a positive definite diagonal matrix,* $q(z) \in \mathbb{R}^N$ *be a monotone mapping of* z *with respect to* $\mathbb{R}^N_+$, *and* $\vartheta : \mathbb{R}^N \to \mathbb{R}$. *Let* $\mu \in (0, 1)$ *be a constant. For given* $\bar{z}, z \in \mathbb{R}^N_{++}$, *we define* $\bar{Z} := \text{diag}(\bar{z}_1, \bar{z}_2, \ldots, \bar{z}^N)$, $z^{-1} := (1/z_1, \ldots, 1/z_N)^\top$ *and*

$$\Phi'(\bar{z}, z) := (z - \bar{z}) + \mu(\bar{z} - \bar{Z}^2 z^{-1}).$$

Then, the variational inequality

$$\vartheta(z') - \vartheta(z) + (z' - z)^\top[q(z) + P\Phi'(\bar{z}, z)] \geq 0, \quad \forall z' \in \mathbb{R}^N_+,$$

has the unique positive solution z. *Besides, for this positive solution* $z \in \mathbb{R}^N_{++}$ *and any* $z' \in \mathbb{R}^N_+$, *we have*

$$\vartheta(z') - \vartheta(z) + (z' - z)^\top q(z) \geq (1 + \mu)(\bar{z} - z)^\top P(z' - z) - \mu\|\bar{z} - z\|^2_P. \quad (4.9)$$

4.2.2 Variational Reformulation

In this subsection, we use a variational inequality (VI) reformulation of the model (4.1) and a characterization of its solution set. Let the Lagrangian function of (4.1) be

$$\mathcal{L}(x, y, \lambda) = \theta_1(x) + \theta_2(y) - \lambda^\top(Ax + By - b),$$

where $x \in \mathbb{R}^n_+$, $y \in \mathbb{R}^p_+$ and $\lambda \in \mathbb{R}^m$ be the Lagrange multiplier. Then finding a saddle point of $\mathcal{L}(x, y, \lambda)$ is to find $(x^*, y^*, \lambda^*) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ \times \mathbb{R}^m$ such that

$$\mathcal{L}(x^*, y^*, \lambda) \le \mathcal{L}(x^*, y^*, \lambda^*) \le \mathcal{L}(x, y, \lambda^*), \quad \forall x^* \in \mathbb{R}^n_+,\ y^* \in \mathbb{R}^p_+,\ \lambda^* \in \mathbb{R}^m.$$

Therefore, solving (4.1) is equivalent to finding $w^* = (x^*, y^*, \lambda^*) \in \Omega := \mathbb{R}^n_+ \times \mathbb{R}^p_+ \times \mathbb{R}^m$ such that

$$\mathrm{VI}(\Omega, \mathrm{F}, \theta):\ \theta(u) - \theta(u^*) + (w - w^*)^\top F(w^*) \ge 0, \ \ \forall w \in \Omega, \tag{4.10}$$

where

$$\theta(u) = \theta_1(x) + \theta_2(y),\ \ u = \begin{pmatrix} x \\ y \end{pmatrix},\ \ w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix} \text{ and } F(w) = \begin{pmatrix} -A^\top \lambda \\ -B^\top \lambda \\ Ax + By - b \end{pmatrix}. \tag{4.11}$$

Since the mapping $F(w)$ defined in (4.11) is affine with a skew-symmetric matrix, it is monotone. We denote by Ω^* the solution set of $\mathrm{VI}(\Omega, F, \theta)$, and suppose that it is nonempty.

4.2.3 *Some Notations*

For notational simplicity, we first define two matrices which will simplify our further analysis in the following sections. Specifically, let

$$G := \begin{pmatrix} (1+\mu)R & 0 & 0 \\ 0 & \beta B^\top B + (1+\mu)S & -B^\top \\ 0 & -B & \frac{1}{\beta} I_m \end{pmatrix}, \quad N := \begin{pmatrix} \mu R & 0 & 0 \\ 0 & \mu S & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{4.12}$$

and

$$M := \begin{pmatrix} (1+\mu)R & 0 & 0 \\ 0 & \frac{\gamma+\rho-\gamma\rho}{\gamma+\rho}\beta B^\top B + (1+\mu)S & -\frac{\gamma}{\gamma+\rho} B^\top \\ 0 & -\frac{\gamma}{\gamma+\rho} B & \frac{1}{(\gamma+\rho)\beta} I_m \end{pmatrix}, \tag{4.13}$$

where $R := rI_n$, $S := sI_p$ and $\gamma,\ \rho$ are two positive constants. Indeed, it is not difficult to verify that the matrices G, N, and M are positive symmetric definite when $r,\ s > 0$ and $\gamma,\ \rho \in (0, 1)$.

4.3 Inertial ADMM with LQP Regularization

In this section, we first introduce the inertial ADMM with LQP regularization as a solution approach for problem (4.1). We then analyze its convergence using a variational inequality framework.

4.3.1 Inertial ADMM with LQP Regularization

We now present the inertial ADMM with LQP regularization. The method first extrapolates the current point in the direction of the last movement and then applies the ADMM with LQP regularization to the extrapolated point. Below, we summarize the overall algorithm framework.

Algorithm 1 Inertial ADMM with LQP regularization

```
**Initialization:** Choose the constants $r, s > 0$ and $\beta > 0$, and a sequence of nonnegative parameters $\{\alpha_k\}_{k=0}^{\infty}$. Let $(x^{-1}, y^{-1}, \lambda^{-1}) := (x^0, y^0, \lambda^0) \in \mathbb{R}^n_{++} \times \mathbb{R}^p_{++} \times \mathbb{R}^m$.
**for** $k = 0, 1, 2, \ldots$ do
    Step 1. Update
        $\tilde{w}^k = w^k + \alpha_k(w^k - w^{k-1}),$    (4.14)
    ensuring that $\tilde{w}^k \in \mathbb{R}^n_{++} \times \mathbb{R}^p_{++} \times \mathbb{R}^m$.
    Step 2. Update the new iterate $w^{k+1}$ by
        $\begin{cases} x^{k+1} = \arg\min\{\mathcal{L}_\beta(x, \tilde{y}^k, \tilde{\lambda}^k) + rd(x, \tilde{x}^k) \mid x \in \mathbb{R}^n_+\}, \\ \lambda^{k+1} = \tilde{\lambda}^k - \beta(Ax^{k+1} + B\tilde{y}^k - b), \\ y^{k+1} = \arg\min\{\mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+1}) + sd(y, \tilde{y}^k) \mid y \in \mathbb{R}^p_+\}. \end{cases}$    (4.15)
**end**
**Return:** $w^{k+1}$.
```

Remark 4.1 Note that Lemma 4.1 guarantees the existence of a unique solution, denoted as $x^{k+1} \in \mathbb{R}^n_{++}$ for subproblem (4.15), and $y^{k+1} \in \mathbb{R}^p_{++}$ for subproblem (4.15). Additionally, the condition $\tilde{w}^k \in \mathbb{R}^n_{++} \times \mathbb{R}^p_{++} \times \mathbb{R}^m$ can be satisfied by choosing $\alpha_k = \max\left\{0, \min_i\left\{\frac{w_i^{k-1} - w_i^k}{w_i^k}\right\}\right\}$. Moreover, the effectiveness of LQP regularization has been showcased in [47] for a traffic management problem, and the benefits of the inertial technique have been extensively confirmed in various real-world applications, as seen in, for instance, [9, 10, 14]. Hence, we exclude the numerical experiments in this paper.

We now introduce the following assumption for the sequence of parameters $\{\alpha_k\}_{k=0}^{\infty}$.

Assumption 4.1 Suppose that the sequence of parameters $\{\alpha_k\}_{k=0}^{\infty}$ is chosen such that

(i) for all $k \geq 0$, $0 \leq \alpha_k \leq \alpha$ for some $\alpha \in [0, 1)$,
(ii) the sequence $\{w^k\}_{k=0}^{\infty}$ generated by (4.15) satisfies

$$\sum_{k=0}^{\infty} \alpha_k \|w^k - w^{k-1}\|_G^2 < \infty. \tag{4.16}$$

Note that in order to ensure Assumption 4.1, we can choose $\{\alpha_k\}_{k=0}^{\infty}$ adaptively based on the historical iterative information in practice such that (4.16) holds. Alternatively, it is simultaneously satisfied if $\{\alpha_k\}_{k=0}^{\infty}$ satisfies some further conditions; see, e.g., [1, Prop. 2.1].

4.3.2 Convergence Analysis

Below, we start our convergence analysis by proving some properties for the sequence $\{w^k\}$ according to the first-order optimality condition.

Lemma 4.2 *Let the sequence $\{w^k\}$ be generated by the iterative scheme of the inertial ADMM with LQP regularization given in Algorithm 1, and G be defined in* (4.12). *Then, for any $w \in \Omega$, we have $w^{k+1} \in \Omega$ and*

$$\theta(u) - \theta(u^{k+1}) + (w - w^{k+1})^\top [F(w^{k+1}) + G(w^{k+1} - \tilde{w}^k)] + \|w^{k+1} - \tilde{w}^k\|_N^2 \geq 0. \quad (4.17)$$

Proof From the first-order optimality condition of the x-subproblem in (4.15), we have $x^{k+1} \in \mathbb{R}_+^n$, and for any $x \in \mathbb{R}_+^n$, it holds that

$$\begin{aligned}&\theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^\top \\ &\{-A^\top[\tilde{\lambda}^k - \beta(Ax^{k+1} + B\tilde{y}^k - b)] + r\Phi'(\tilde{x}^k, x^{k+1})\} \geq 0.\end{aligned}$$

Applying Lemma 4.1 to above inequality with $P = R$, $\bar{z} = \tilde{x}^k$, $z = x^{k+1}$, $z' = x$, $\vartheta = \theta_1$ and

$$q(z) = -A^\top[\tilde{\lambda}^k - \beta(Ax^{k+1} + B\tilde{y}^k - b)]$$

in (4.9), then for any $x \in \mathbb{R}_+^n$, we have $x^{k+1} \in \mathbb{R}_{++}^n$ and

$$\begin{aligned}&\theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^\top(-A^\top\lambda^{k+1}) \\ &\geq (1+\mu)(x - x^{k+1})R(\tilde{x}^k - x^{k+1}) - \mu\|x^{k+1} - \tilde{x}^k\|_R^2.\end{aligned}$$

Rearranging the inequality, we have

$$\begin{aligned}&\theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^\top[-A^\top\lambda^{k+1} + (1+\mu)R(x^{k+1} - \tilde{x}^k)] \\ &\quad + \mu\|x^{k+1} - \tilde{x}^k\|_R^2 \geq 0.\end{aligned} \quad (4.18)$$

Similarly, for the y-subproblem (4.15), we have $y^{k+1} \in \mathbb{R}_{++}^p$, and for any $y \in \mathbb{R}_+^p$, it holds that

$$\begin{aligned}\theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^\top\{&-B^\top[\lambda^{k+1} - \beta(Ax^{k+1} + By^{k+1} - b)] \\ &+ (1+\mu)S(y^{k+1} - \tilde{y}^k)\} + \mu\|y^{k+1} - \tilde{y}^k\|_S^2 \geq 0.\end{aligned} \quad (4.19)$$

Moreover, it follows from (4.15) that

$$\beta(Ax^{k+1} + By^{k+1} - b) = \tilde{\lambda}^k - \lambda^{k+1} + \beta B(y^{k+1} - \tilde{y}^k) = 0, \tag{4.20}$$

then the inequality (4.19) can be rewritten as

$$\begin{aligned}\theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^\top \Big\{ & - B^\top\lambda^{k+1} + [\beta B^\top B + (1+\mu)S](y^{k+1} - \tilde{y}^k) \\ & - B^\top(\lambda^{k+1} - \tilde{\lambda}^k)\Big\} + \mu\|y^{k+1} - \tilde{y}^k\|_S^2 \geq 0.\end{aligned} \tag{4.21}$$

Combining (4.18), (4.20) and (4.21), and using the notations in (4.11) and (4.12), we obtain the assertion (4.17) immediately. This completes the proof.

□

Theorem 4.1 *Suppose that $\{\alpha_k\}_{k=0}^\infty$ satisfies Assumption 4.1. Let the sequence $\{w^k\}_{k=0}^\infty$ be generated by the inertial ADMM with LQP regularization given in Algorithm 1. Then, it holds that*

(i) for any $w^* \in \Omega^*$, $\lim\limits_{k\to\infty} \|w^k - w^*\|_G$ exists,
(ii) $\sum_{k=0}^\infty \|w^{k+1} - \tilde{w}^k\|_G^2 < \infty$.

Proof (i) Setting $w = w^*$ in (4.17) and $w = w^{k+1}$ in (4.10), adding these two inequalities and using the monotonicity of F, we obtain

$$(w^{k+1} - w^*)^\top G(w^{k+1} - \tilde{w}^k) - \|w^{k+1} - \tilde{w}^k\|_N^2 \leq 0. \tag{4.22}$$

It follows from (4.14) that

$$(w^{k+1} - w^*)^\top G(w^{k+1} - \tilde{w}^k) = (w^{k+1} - w^*)^\top G[w^{k+1} - w^k - \alpha_k(w^k - w^{k-1})]. \tag{4.23}$$

Define $\varphi_k = \frac{1}{2}\|w^k - w^*\|_G^2$, then we have

$$(w^{k+1} - w^*)^\top G(w^{k+1} - w^k) = \varphi_{k+1} - \varphi_k + \frac{1}{2}\|w^{k+1} - w^k\|_G^2,$$

and

$$\begin{aligned}&(w^{k+1} - w^*)^\top G(w^k - w^{k-1}) \\ &= \varphi_k - \varphi_{k-1} + \frac{1}{2}\|w^k - w^{k-1}\|_G^2 + (w^{k+1} - w^k)^\top G(w^k - w^{k-1}).\end{aligned}$$

Together with (4.22) and (4.23), we have

$$\varphi_{k+1} - \varphi_k - \alpha_k(\varphi_k - \varphi_{k-1}) \leq -\frac{1}{2}\|w^{k+1} - w^k - \alpha_k(w^k - w^{k-1})\|_G^2 \\ + \frac{1}{2}(\alpha_k + \alpha_k^2)\|w^k - w^{k-1}\|_G^2 + \|w^{k+1} - \tilde{w}^k\|_N^2.$$

It follows from (4.14) and $\frac{1}{2}(\alpha_k + \alpha_k^2) \leq \alpha_k$ for any $\alpha_k \in [0, 1)$ that

$$\begin{aligned} &\varphi_{k+1} - \varphi_k - \alpha_k(\varphi_k - \varphi_{k-1}) \\ &\leq -\frac{1}{2}\|w^{k+1} - \tilde{w}^k\|_G^2 + \alpha_k\|w^k - w^{k-1}\|_G^2 + \|w^{k+1} - \tilde{w}^k\|_N^2 \\ &\leq \alpha_k\|w^k - w^{k-1}\|_G^2, \end{aligned} \tag{4.24}$$

where the second inequality follows from $G \succeq 2N$ since $\mu \in (0, 1)$, G and N are defined in (4.12).

Define $\nu_k := \varphi_k - \varphi_{k-1}$ and $\delta_k := \alpha_k\|w^k - w^{k-1}\|_G^2$. Then, the inequality (4.24) implies that $\nu_{k+1} \leq \alpha_k\nu_k + \delta_k \leq \alpha[\nu_k]_+ + \delta_k$, where $[t]_+ = \max\{t, 0\}$ for $t \in \mathbb{R}$. Also, we have

$$[\nu_{k+1}]_+ \leq \alpha[\nu_k]_+ + \delta_k \leq \cdots \leq \alpha^{k+1}[\nu_0]_+ + \sum_{j=0}^{k} \alpha^j \delta_{k-j}.$$

Note that $w^0 = w^{-1}$, which implies that $\nu_0 = [\nu_0]_+ = 0$, $\delta_0 = 0$. Thus, it follows from the above inequalities and Assumption 4.1 (ii) that

$$\sum_{k=0}^{\infty}[\nu_{k+1}]_+ \leq \frac{1}{1-\alpha}\sum_{k=1}^{\infty}\delta_k < \infty. \tag{4.25}$$

Let $\sigma_k := \varphi_k - \sum_{j=1}^{k}[\nu_j]_+$. Then, it follows from (4.25) and $\varphi_k \geq 0$ that σ_k is bounded from below. Besides, we have

$$\begin{aligned} \sigma_{k+1} &= \varphi_{k+1} - [\nu_{k+1}]_+ - \sum_{j=1}^{k}[\nu_j]_+ \\ &\leq \varphi_{k+1} - \nu_{k+1} - \sum_{j=1}^{k}[\nu_j]_+ \leq \varphi_k - \sum_{j=1}^{k}[\nu_j]_+ = \sigma_k, \end{aligned}$$

thus σ_k is nonincreasing. Consequently, $\{\sigma_k\}_{k=0}^{\infty}$ converges as $k \to \infty$, and the limit

$$\lim_{k\to\infty}\varphi_k = \lim_{k\to\infty}\left(\sigma_k + \sum_{j=1}^{k}[\nu_j]_+\right) = \lim_{k\to\infty}\sigma_k + \sum_{k=1}^{\infty}[\nu_k]_+$$

exists. This completes the proof of assertion (i).

(ii) Applying the identity

$$\|a\|_G^2 - \|b\|_G^2 = \|a-b\|_G^2 + 2b^\top G(a-b),$$

and setting

$$a := w^* - \tilde{w}^k, \;\; b := w^* - w^{k+1},$$

we have

$$(w^{k+1} - w^*)^\top G(w^{k+1} - \tilde{w}^k) = \frac{1}{2}\|w^{k+1} - \tilde{w}^k\|_G^2 + \frac{1}{2}\Big(\|w^{k+1} - w^*\|_G^2 - \|\tilde{w}^k - w^*\|_G^2\Big).$$

Together with (4.22), we obtain

$$\|\tilde{w}^k - w^*\|_G^2 - \|w^{k+1} - w^*\|_G^2 \geq \|w^{k+1} - \tilde{w}^k\|_G^2 - 2\|w^{k+1} - \tilde{w}^k\|_N^2.$$

Recalling the definition of G and N in (4.12) and $\mu \in (0, 1)$, there exists a positive constant $0 < c \leq \frac{1-\mu}{1+\mu} < 1$ such that $G - 2N \succeq cG$. Then, the above inequality can be written as

$$\|\tilde{w}^k - w^*\|_G^2 - \|w^{k+1} - w^*\|_G^2 \geq c\|w^{k+1} - \tilde{w}^k\|_G^2. \tag{4.26}$$

Since

$$(w^k - w^*)^\top G(w^k - w^{k-1}) = \frac{1}{2}\|w^k - w^*\|_G^2 - \frac{1}{2}\|w^{k-1} - w^*\|_G^2 + \frac{1}{2}\|w^k - w^{k-1}\|_G^2. \tag{4.27}$$

Combining (4.26), (4.27) and (4.14), we obtain

$$\begin{aligned}
&\|w^{k+1} - \tilde{w}^k\|_G^2 \\
&\leq \frac{1}{c}\Big[\|w^k - w^*\|_G^2 - \|w^{k+1} - w^*\|_G^2 + \alpha_k \nu_k + (\alpha_k + \alpha_k^2)\|w^k - w^{k-1}\|_G^2\Big] \\
&\leq \frac{1}{c}\Big[\|w^k - w^*\|_G^2 - \|w^{k+1} - w^*\|_G^2 + \alpha[\nu_k]_+ + 2\alpha_k\|w^k - w^{k-1}\|_G^2\Big].
\end{aligned}$$

By taking sum over k, we obtain

$$\sum_{k=1}^{\infty} \|w^{k+1} - \tilde{w}^k\|_G^2 \leq \frac{1}{c}\Big[\varphi_1 + \sum_{k=1}^{\infty}\Big([\alpha \nu_k]_+ + 2\delta_k\Big)\Big].$$

Together with (4.25) and Assumption 4.1 (ii), we obtain $\sum_{k=0}^{\infty} \|w^{k+1} - \tilde{w}^k\|_G^2 < \infty$ immediately. □

The following lemma presents the feasibility and convergence of the objective function value for the inertial ADMM with LQP regularization, and the proof can refer to [14, Theorem 4.3]. Here, we omit the details.

Lemma 4.3 *Suppose that $\{\alpha_k\}_{k=0}^{\infty}$ satisfies Assumption 4.1. Let the sequence $\{w_k\}_{k=0}^{\infty}$ be generated by the inertial ADMM with LQP regularization given in Algorithm 1. Then, we have*

(i) $\sum\limits_{k=0}^{\infty} \|Ax^k + By^k - b\|^2 < \infty$, *and hence* $\lim\limits_{k\to\infty} \|Ax^k + By^k - b\| = 0$.

(ii) The objective function value $f(x^k) + g(y^k)$ converges to the optimal value of (4.1) as $k \to \infty$.

Note that the results presented in Theorem 4.1 does not ensure the convergence of $\{w^k\}_{k=0}^{\infty}$. In the following theorem, we present the convergence analysis for the sequence $\{w^k\}_{k=0}^{\infty}$ generated by inertial ADMM with LQP regularization scheme (4.15).

Theorem 4.2 *Suppose that $\{\alpha_k\}_{k=0}^{\infty}$ satisfies Assumption 4.1. Let the sequence $\{w^k\}_{k=0}^{\infty}$ be generated by the inertial ADMM with LQP regularization given in Algorithm 1. Then, $\{w^k\}_{k=0}^{\infty}$ converges to a point $w^* \in \Omega^*$ as $k \to \infty$.*

Proof It follows from Theorem 4.1 that $\lim\limits_{k\to\infty} \|w^k - w^*\|_G$ exists for any $w^* \in \Omega^*$, thus the sequence $\{Gw^k\}_{k=0}^{\infty}$ is bounded. By the definition of G in (4.12), we have that the sequences $\{Rx^k\}_{k=0}^{\infty}$, $\{Sy^k\}_{k=0}^{\infty}$ and $\{By^k - \frac{\lambda^k}{\beta}\}_{k=0}^{\infty}$ are bounded, and thus $\{x^k\}_{k=0}^{\infty}$, $\{y^k\}_{k=0}^{\infty}$ are bounded by the definition of R and S, respectively. Besides, the assertion of Lemma 4.3 (i) implies that $\{Ax^k + \frac{\lambda^k}{\beta}\}_{k=0}^{\infty}$ is bounded. On the other hand, setting $x = x^*$ in (4.18), we obtain

$$\begin{aligned}\theta_1(x^*) - \theta_1(x^{k+1}) + (x^* - x^{k+1})^\top[-A^\top\lambda^{k+1} + (1+\mu)R(x^{k+1} - \tilde{x}^k)]\\ + \mu\|x^{k+1} - \tilde{x}^k\|_R^2 \geq 0.\end{aligned}$$

Since $w^* = (x^*, y^*, \lambda^*) \in \Omega^*$, thus setting $x = x^{k+1}$, $y = y^*$ and $\lambda = \lambda^*$ in (4.10), we have

$$\theta_1(x^{k+1}) - \theta_1(x^*) + (x^{k+1} - x^*)^\top(-A^\top\lambda^*) \geq 0.$$

Adding the above two inequalities, it holds that

$$\langle A(x^{k+1} - x^*), \lambda^{k+1} - \lambda^*\rangle \geq \langle x^{k+1} - x^*, (1+\mu)R(x^{k+1} - \tilde{x}^k)\rangle - \mu\|x^{k+1} - \tilde{x}^k\|_R^2.$$

From the above inequality, we have

$$\langle A(x^k - x^*), \lambda^k - \lambda^*\rangle \geq -\frac{1+\mu}{2}\Big(\|x^k - x^*\|_R^2 + \|x^k - \tilde{x}^{k-1}\|_R^2\Big) - \mu\|x^k - \tilde{x}^{k-1}\|_R^2.$$

By the boundedness of $\{Rx^k\}_{k=0}^{\infty}$ and the assertion in Theorem 4.1 (ii), we know that $\langle A(x^k - x^*), \lambda^k - \lambda^* \rangle$ is bounded from below for $k \geq 1$. Then, according to

$$\|A(x^k - x^*\|^2 + \|(\lambda^k - \lambda^*)/\beta\|^2 = \|(Ax^k + \lambda^k/\beta) - (Ax^* + \lambda^*/\beta)\|^2 - \frac{2}{\beta}\langle A(x^k - x^*), \lambda^k - \lambda^* \rangle.$$

It holds that $\{Ax^k\}_{k=0}^{\infty}$ and $\{\lambda^k\}_{k=0}^{\infty}$ are bounded. Therefore, the sequence $\{w^k\}_{k=0}^{\infty}$ is bounded and at least has a limit point. Let w^* be any limit point of $\{w^k\}_{k=0}^{\infty}$ and thus there exists a sequence $w^{k_j} \to w^*$ as $j \to \infty$. First, we have $w^* \in \Omega$ since Ω is closed. Furthermore, taking limits over $k = k_j \to \infty$ in (4.17) and note that $G(w^{k_j} - \tilde{w}^{k_j - 1}) \to 0$, we have

$$\theta(w) - \theta(w^*) + \langle w - w^*, F(w^*) \rangle \geq 0.$$

Since we can choose $w \in \Omega$ vary arbitrarily, thus $w^* \in \Omega^*$. Next, it is routine to prove the uniqueness of the limit points of $\{w^k\}_{k=0}^{\infty}$ based on the results presented in [1, Lemma 2.1] and [14, Theorem 4.7]. Therefore, the sequence $\{w_k\}_{k=0}^{\infty}$ converges to a point $w^* \in \Omega^*$ as $k \to \infty$. □

Remark 4.2 Note that the convergence result does not need any assumptions for the matrices A and B; the result also holds when $A = 0$ or/and $B = 0$ in problem (4.1). Compared with the algorithm presented in [14], the inertial ADMM with LQP regularization not only can force the subproblems in (4.15) converted to unconstrained, but also needs less assumptions for deriving the convergence results.

4.4 Inertial Symmetric ADMM with LQP Regularization

In this section, we propose the inertial symmetric ADMM with LQP regularization for solving the problem (4.1), and analyze its convergence in a similar way as that in Sect. 4.3.

4.4.1 Inertial Symmetric ADMM with LQP Regularization

We now present the inertial symmetric ADMM with LQP regularization, which differs from Algorithm 1 with the additional Lagrange multiplier updating step. Besides, we add two contractive factors to the dual step sizes to guarantee the convergence as that in [27, 36]. More details can be found in the following algorithm framework.

Algorithm 2 Inertial symmetric ADMM with LQP regularization

```
Initialization: Choose the constants $r, s > 0$, $\gamma, \rho \in (0, 1)$ and $\beta > 0$, and a sequence of nonnegative parameters $\{\alpha_k\}_{k=0}^{\infty}$. Let $(x^{-1}, y^{-1}, \lambda^{-1}) := (x^0, y^0, \lambda^0) \in \mathbb{R}^n_{++} \times \mathbb{R}^p_{++} \times \mathbb{R}^m$.
for $k = 0, 1, 2, \ldots$ do
    Step 1. Update
        $\tilde{w}^k = w^k + \alpha_k(w^k - w^{k-1}),$    (4.28)
    ensuring that $\tilde{w}^k \in \mathbb{R}^n_{++} \times \mathbb{R}^p_{++} \times \mathbb{R}^m$.
    Step 2. Update the new iterate $w^{k+1}$ by
        $x^{k+1} = \arg\min\{\mathcal{L}_\beta(x, \tilde{y}^k, \tilde{\lambda}^k) + rd(x, \tilde{x}^k) \mid x \in \mathbb{R}^n_+\},$
        $\lambda^{k+1/2} = \tilde{\lambda}^k - \gamma\beta(Ax^{k+1} + B\tilde{y}^k - b),$
        $y^{k+1} = \arg\min\{\mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+1/2}) + sd(y, \tilde{y}^k) \mid y \in \mathbb{R}^p_+\},$    (4.29)
        $\lambda^{k+1} = \lambda^{k+1/2} - \rho\beta(Ax^{k+1} + By^{k+1} - b).$
end
return $w^{k+1}$.
```

Remark 4.3 Note that there exists a unique solution $x^{k+1} \in \mathbb{R}^n_{++}$ and $y^{k+1} \in \mathbb{R}^p_{++}$ in subproblems (4.28) and (4.29) respectively as that in Algorithm 1. Besides, the Assumption 4.1 also should be satisfied for Algorithm 2. In addition, when choosing $\alpha_k = 0$, the algorithm reduces to the method studied in [36]. Note that Algorithm 2 cannot include Algorithm 1 as a special case, since the step sizes of the Lagrange multiplier update are constrained to $(0, 1)$. This implies that we need to conduct a separate convergence analysis for both algorithms.

4.4.2 Convergence Analysis

Similar to Lemma 4.2 and [25, Lemma 3.1], we give the following lemma which is mainly based on the first-order optimality conditions of (4.28) and (4.29).

Lemma 4.4 *Let the sequence $\{w^k\}$ be generated by the iterative scheme of the inertial symmetric ADMM with LQP regularization given in Algorithm 2, N and M are defined in (4.12) and (4.13), respectively. Then, for any $w \in \Omega$, we have $w^{k+1} \in \Omega$ and*

$$\begin{aligned} \theta(u) - \theta(u^{k+1}) + (w - w^{k+1})^\top[F(w^{k+1}) + M(w^{k+1} - \tilde{w}^k)] + \|w^{k+1} - \tilde{w}^k\|_N^2 \\ \geq (w^{k+1} - w)^\top \begin{pmatrix} A^\top \\ B^\top \\ 0 \end{pmatrix} [(1-\gamma)\beta(\tilde{y}^k - y^{k+1}) + (1-\gamma-\rho)\beta r^{k+1}], \end{aligned} \tag{4.30}$$

where $r^{k+1} := Ax^{k+1} + By^{k+1} - b$.

Proof From the first-order optimality condition of (4.29), and applying the Lemma 4.1, we have $x^{k+1} \in \mathbb{R}^n_{++}$, and for any $x \in \mathbb{R}^n_{+}$, it holds that

$$\theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^\top \Big\{ - A^\top \tilde{\lambda}^k + \beta A^\top r^{k+1} + \beta A^\top B(\tilde{y}^k - y^{k+1}) \\ + (1+\mu) R(x^{k+1} - \tilde{x}^k) \Big\} + \mu \|x^{k+1} - \tilde{x}^k\|_R^2 \geq 0. \tag{4.31}$$

Similarly, from (4.29), for any $y \in \mathbb{R}^p_{+}$, we have $y^{k+1} \in \mathbb{R}^p_{++}$ and

$$\theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^\top \Big\{ - B^\top \lambda^{k+1/2} + \beta B^\top r^{k+1} \\ + (1+\mu) S(y^{k+1} - \tilde{y}^k) \Big\} + \mu \|y^{k+1} - \tilde{y}^k\|_S^2 \geq 0. \tag{4.32}$$

It follows from (4.29) that

$$\lambda^{k+1/2} = \lambda^{k+1} + \rho \beta r^{k+1}$$

and

$$\tilde{\lambda}^k = \lambda^{k+1} + (\gamma + \rho)\beta r^{k+1} + \gamma \beta B(\tilde{y}^k - y^{k+1}). \tag{4.33}$$

Substituting the above equalities into (4.31) and (4.32), we have

$$\theta_1(x) - \theta_1(x^{k+1}) + (x - x^{k+1})^\top \Big\{ - A^\top \lambda^{k+1} + (1 - \gamma - \rho)\beta A^\top r^{k+1} \\ + (1-\gamma)\beta A^\top B(\tilde{y}^k - y^{k+1}) + (1+\mu) R(x^{k+1} - \tilde{x}^k) \Big\} + \mu \|x^{k+1} - \tilde{x}^k\|_R^2 \geq 0. \tag{4.34}$$

and

$$\theta_2(y) - \theta_2(y^{k+1}) + (y - y^{k+1})^\top \Big\{ - B^\top \lambda^{k+1} + (1-\rho)\beta B^\top r^{k+1} \\ + (1+\mu) S(y^{k+1} - \tilde{y}^k) \Big\} + \mu \|y^{k+1} - \tilde{y}^k\|_S^2 \geq 0. \tag{4.35}$$

On the other hand, it follows from (4.33) that

$$r^{k+1} - \frac{\gamma}{\gamma + \rho} B(y^{k+1} - \tilde{y}^k) + \frac{1}{(\gamma+\rho)\beta}(\lambda^{k+1} - \tilde{\lambda}^k) = 0. \tag{4.36}$$

Combining (4.34), (4.35) and (4.36), the assertion (4.30) is obtained immediately. □

Theorem 4.3 *Suppose that* $\{\alpha_k\}_{k=0}^\infty$ *satisfies Assumption 4.1. Let the sequence* $\{w^k\}_{k=0}^\infty$ *be generated by the inertial symmetric ADMM with LQP regularization given in Algorithm 2. Then, we have* *(i) for any* $w^* \in \Omega^*$, $\lim_{k\to\infty} \|w^k - w^*\|_M$ *exists,*

(ii) $\sum_{k=0}^{\infty} \|w^{k+1} - \tilde{w}^k\|_M^2 < \infty$.

Proof (i) Setting $w = w^*$ in (4.30) and $w = w^{k+1}$ in (4.10), adding these two inequalities, using the monotonicity of F and the fact $Ax^* + By^* - b = 0$, we obtain

$$\begin{aligned}&(w^{k+1} - w^*)^\top M(w^{k+1} - \tilde{w}^k)\\ &\le -(1-\gamma)\beta(r^{k+1})^\top(\tilde{y}^k - y^{k+1}) - (1-\gamma-\rho)\beta\|r^{k+1}\|^2 + \|w^{k+1} - \tilde{w}^k\|_N^2.\end{aligned} \tag{4.37}$$

Let

$$H := \frac{1}{\gamma+\rho}\begin{pmatrix} (\gamma+\rho-\gamma\rho)\beta B^\top B & -\gamma B^\top \\ -\gamma B & \frac{1}{\beta} I_m \end{pmatrix} \text{ and } C := \begin{pmatrix} I_p & 0 \\ \gamma\beta B & (\gamma+\rho)\beta I_m \end{pmatrix}. \tag{4.38}$$

Then, it is easy to verify that H is positive semidefinite if $0 \le \gamma \le 1$ and $\rho > 0$, and

$$C^\top HC := \frac{1}{\gamma+\rho}\begin{pmatrix} (1-\gamma)\beta B^\top B & 0 \\ 0 & (\gamma+\rho)\beta I_m \end{pmatrix}. \tag{4.39}$$

Let $v := (y, \lambda)$, together with (4.33) and (4.38), we have

$$\tilde{v}^k - v^{k+1} = \begin{pmatrix} \tilde{y}^k - y^{k+1} \\ \tilde{\lambda}^k - \lambda^{k+1} \end{pmatrix} = C\begin{pmatrix} \tilde{y}^k - y^{k+1} \\ r^{k+1} \end{pmatrix}. \tag{4.40}$$

With (4.39) and (4.40), we obtain

$$\|\tilde{v}^k - v^{k+1}\|_H^2 = (1-\gamma)\beta\|B(\tilde{y}^k - y^{k+1})\|^2 + (\gamma+\rho)\beta\|r^{k+1}\|^2. \tag{4.41}$$

Applying the inequality $2a^\top b \le \|a\|^2 + \|b\|^2$, we have

$$\begin{aligned}&-(1-\gamma)\beta(r^{k+1})^\top(\tilde{y}^k - y^{k+1}) - (1-\gamma-\rho)\beta\|r^{k+1}\|^2\\ &\le \frac{\gamma+2\rho-1}{2}\beta\|r^{k+1}\|^2 + \frac{1-\gamma}{2}\beta\|B(\tilde{y}^k - y^{k+1})\|^2.\end{aligned} \tag{4.42}$$

Combining (4.41), (4.42) and $\gamma \in [0, 1)$, we conclude

$$-(1-\gamma)\beta(r^{k+1})^\top(\tilde{y}^k - y^{k+1}) - (1-\gamma-\rho)\beta\|r^{k+1}\|^2 \le \frac{1}{2}\|v^{k+1} - \tilde{v}^k\|_H.$$

It follows from (4.37) that

$$(w^{k+1} - w^*)^\top M(w^{k+1} - \tilde{w}^k) \le \frac{1}{2}\|w^{k+1} - \tilde{w}^k\|_{\widetilde{H}+2N}^2, \tag{4.43}$$

where

$$\widetilde{H} := \begin{pmatrix} 0 & 0 \\ 0 & H \end{pmatrix}.$$

Then, we can derive the assertion (i) similar to the proof of Theorem 4.1 (i) and omit it here.

(ii) Applying the identity

$$\|a\|_G - \|b\|_G = \|a-b\|_G + 2b^\top G(a-b),$$

and setting

$$a := w^* - \tilde{w}^k, \;\; b := w^* - w^{k+1},$$

we have

$$\begin{aligned}(w^{k+1}-w^*)^\top M(w^{k+1}-\tilde{w}^k) &= \frac{1}{2}\|w^{k+1}-\tilde{w}^k\|_M^2 \\ &\quad + \frac{1}{2}\Big(\|w^{k+1}-w^*\|_M^2 - \|\tilde{w}^k - w^*\|_M^2\Big).\end{aligned}$$

Together with (4.37), we obtain

$$\begin{aligned}&\|\tilde{w}^k - w^*\|_M^2 - \|w^{k+1}-w^*\|_M^2 \\ &\geq \|w^{k+1}-\tilde{w}^k\|_M^2 + 2(1-\gamma)\beta(r^{k+1})^\top B(\tilde{y}^k - y^{k+1}) \\ &\quad + 2(1-\gamma-\rho)\beta\|r^{k+1}\|^2 - 2\|w^{k+1}-\tilde{w}^k\|_N^2 \\ &= (1-\mu)\|w^{k+1}-\tilde{w}^k\|_N^2 + (1-\gamma)\beta\|B(\tilde{y}^k - y^{k+1})\|^2 \\ &\quad + 2(1-\gamma)\beta(r^{k+1})^\top B(\tilde{y}^k - y^{k+1}) + (2-\gamma-\rho)\beta\|r^{k+1}\|^2 \\ &= (1-\mu)\|w^{k+1}-\tilde{w}^k\|_N^2 + (\tilde{v}^k - v^{k+1})^\top C^{-T}KC(\tilde{v}^k - v^{k+1}),\end{aligned} \tag{4.44}$$

the last equality follows from (4.40), where

$$K := \begin{pmatrix} (1-\gamma)\beta B^\top B & (1-\gamma)\beta B^\top \\ (1-\gamma)\beta B & (2-\gamma-\rho)\beta I_m \end{pmatrix},$$

and C is defined in (4.38). It is easy to verify that K is positive definite if $0 \leq \gamma < 1$ and $0 < \rho < 1$. Moreover, by some simple calculations, we have

$$K \succeq \tilde{c}C^\top HC$$

when taking $\tilde{c} := \frac{1-\sqrt{1-\gamma+\rho(1-\rho)}}{\gamma+\rho} \in (0,1)$. On the other hand, if $0 < \bar{c} \leq \frac{1-\mu}{1+\mu} < 1$, we have $(1-\mu)R \succeq \bar{c}(1+\mu)R$ and $(1-\mu)S \succeq \bar{c}(1+\mu)S$. Therefore, taking $\eta = \min\{\tilde{c}, \bar{c}\}$, by the definition of M in (4.13), and together with (4.44), we have

$$\|\tilde{w}^k - w^*\|_M^2 - \|w^{k+1}-w^*\|_M^2 \geq \eta\|w^{k+1}-\tilde{w}^k\|_M^2.$$

Then, it is easy to obtain the assertion (ii) similar to the proof of Theorem 4.1 (ii). This completes the proof. □

Similar to Lemma 4.3, we first obtain the results of the feasibility and the convergence of the objective function value for the inertial symmetric ADMM with LQP regularization.

Lemma 4.5 *Suppose that $\{\alpha_k\}_{k=0}^{\infty}$ satisfies Assumption 4.1. Let the sequence $\{w^k\}_{k=0}^{\infty}$ be generated by the inertial symmetric ADMM with LQP regularization given in Algorithm 2. Then, we have*

(i) $\sum\limits_{k=0}^{\infty} \|Ax^k + By^k - b\|^2 < \infty$, *and hence* $\lim\limits_{k\to\infty} \|Ax^k + By^k - b\| = 0$.

(ii) The objective function value $f(x^k) + g(y^k)$ converges to the optimal value of (4.1) as $k \to \infty$.

The following theorem presents the convergence analysis for the sequence $\{w^k\}_{k=0}^{\infty}$ generated by inertial symmetric ADMM with LQP regularization scheme (4.29).

Theorem 4.4 *Suppose that $\{\alpha_k\}_{k=0}^{\infty}$ satisfies Assumption 4.1. Let the sequence $\{w^k\}_{k=0}^{\infty}$ be generated by the inertial symmetric ADMM with LQP regularization given in Algorithm 2. Then, $\{w^k\}_{k=0}^{\infty}$ converges to a point $w^* \in \Omega^*$ as $k \to \infty$.*

Proof We prove it in a similar way as that in Theorem 4.2. It follows from Theorem 4.3 that $\lim\limits_{k\to\infty} \|w^k - w^*\|_M$ exists for any $w^* \in \Omega^*$, thus the sequence $\{Mw^k\}_{k=0}^{\infty}$ is bounded. By the definition of M in (4.13), we have that the sequences $\{Rx^k\}_{k=0}^{\infty}$, $\{Sy^k\}_{k=0}^{\infty}$ and $\{\gamma By^k - \frac{\lambda^k}{\beta}\}_{k=0}^{\infty}$ are bounded, and thus $\{x^k\}_{k=0}^{\infty}$, $\{y^k\}_{k=0}^{\infty}$ are bounded by the definition of R and S, respectively. Besides, the assertion of Lemma 4.5 (i) implies that $\{\gamma Ax^k + \frac{\lambda^k}{\beta}\}_{k=0}^{\infty}$ is bounded. On the other hand, setting $x = x^*$ in (4.34), we obtain

$$\begin{aligned}\theta_1(x^*) - \theta_1(x^{k+1}) + (x^* - x^{k+1})^\top \Big\{ - A^\top\lambda^{k+1} + (1-\gamma-\rho)\beta A^\top r^{k+1} \\ + (1-\gamma)\beta A^\top B(\tilde{y}^k - y^{k+1}) + (1+\mu)R(x^{k+1} - \tilde{x}^k)\Big\} + \mu\|x^{k+1} - \tilde{x}^k\|_R^2 \geq 0.\end{aligned}$$

Since $w^* = (x^*, y^*, \lambda^*) \in \Omega^*$, thus setting $x = x^{k+1}$, $y = y^*$ and $\lambda = \lambda^*$ in (4.10), we have

$$\theta_1(x^{k+1}) - \theta_1(x^*) + (x^{k+1} - x^*)^\top(-A^\top\lambda^*) \geq 0.$$

Adding the above two inequalities, and let $k := k-1$, it holds that

$$\begin{aligned}&\langle A(x^k - x^*), \lambda^k - \lambda^* + (\gamma+\rho-1)\beta r^k + (1-\gamma)\beta B(y^k - \tilde{y}^{k-1})\rangle \\ &\geq \langle x^k - x^*, (1+\mu)R(x^k - \tilde{x}^{k-1})\rangle - \mu\|x^k - \tilde{x}^{k-1}\|_R^2 \\ &\geq -\frac{1+\mu}{2}\Big(\|x^k - x^*\|_R^2 + \|x^k - \tilde{x}^{k-1}\|_R^2\Big) - \mu\|x^k - \tilde{x}^{k-1}\|_R^2.\end{aligned}$$

By the boundedness of $\{Rx^k\}_{k=0}^{\infty}$ and the assertion in Theorem 4.3 (ii), we know that

$$\langle A(x^k - x^*), \lambda^k - \lambda^* + (\gamma + \rho - 1)\beta A^\top r^k + (1 - \gamma)\beta A^\top B(y^k - \tilde{y}^{k-1})\rangle$$

is bounded from below for $k \geq 1$. On the other hand, we have

$$\begin{aligned}
&\|\gamma A(x^k - x^*)\|^2 + \|(\lambda^k - \lambda^*)/\beta + (\gamma + \rho - 1)\beta A^\top r^k + (1 - \gamma)\beta A^\top B(y^k - \tilde{y}^{k-1})\|^2 \\
&\leq 2\|(\gamma Ax^k + \lambda^k/\beta) - (\gamma Ax^* + \lambda^*/\beta)\|^2 \\
&\quad + 2\|(\gamma + \rho - 1)\beta r^k + (1 - \gamma)\beta B(y^k - \tilde{y}^{k-1})\|^2 \\
&\quad - \frac{2\gamma}{\beta}\langle A(x^k - x^*), \lambda^k - \lambda^* + (\gamma + \rho - 1)\beta^2 r^k + (1 - \gamma)\beta^2 B(y^k - \tilde{y}^{k-1})\rangle.
\end{aligned} \tag{4.45}$$

It follows from Theorem 4.3 (ii) that $\lim_{k\to\infty} \|v^k - \tilde{v}^{k-1}\|_H^2 = 0$, and it also have $\lim_{k\to\infty} r^k = 0$ follows from Lemma 4.5 (i), thus it holds that $\lim_{k\to\infty} B(y^k - \tilde{y}^{k-1}) = 0$, and

$$\lim_{k\to\infty} 2\|(\gamma + \rho - 1)\beta r^k + (1 - \gamma)\beta B(y^k - \tilde{y}^{k-1})\|^2 = 0.$$

Together with the boundedness of $\{\gamma Ax^k + \frac{\lambda^k}{\beta}\}_{k=0}^{\infty}$ and (4.45), it holds that $\{Ax^k\}_{k=0}^{\infty}$ and $\{\lambda^k\}_{k=0}^{\infty}$ are bounded. Therefore, the sequence $\{w^k\}_{k=0}^{\infty}$ is bounded and at least has a limit point. Let w^* be any limit point of $\{w^k\}_{k=0}^{\infty}$ and thus exists a sequence $w^{k_j} \to w^*$ as $j \to \infty$. First, we have $w^* \in \Omega$ since Ω is closed. Furthermore, taking limits over $k = k_j \to \infty$ in (4.17) and note that $M(w^{k_j} - \tilde{w}^{k_j-1}) \to 0$, we have

$$\theta(w) - \theta(w^*) + \langle w - w^*, F(w^*)\rangle \geq 0.$$

Since we can choose $w \in \Omega$ vary arbitrarily, thus $w^* \in \Omega^*$. Next, it is routine to prove the uniqueness of the limit points of $\{w^k\}_{k=0}^{\infty}$ based on the results presented in [1, Lemma 2.1] and [14, Theorem 4.7]. Therefore, the sequence $\{w^k\}_{k=0}^{\infty}$ converges to a point $w^* \in \Omega^*$ as $k \to \infty$. □

4.5 Conclusion

We revisited the alternating direction method of multipliers (ADMM) with logarithmic- quadratic proximal (LQP) regularization, which can be utilized to solve a class of convex optimization problems with linear constraints. By incorporating the inertial extrapolation step, we proposed an inertial ADMM and an inertial symmetric ADMM with LQP regularization, respectively. Additionally, the convergence of the proposed methods was established under the framework of variational inequality.

Acknowledgements This work was supported by the National Natural Science Foundation of China grants 12471291, 12001286, and the China Postdoctoral Science Foundation grants 2022M7116

Competing Interests The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Alvarez, F., Attouch, H.: An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping. Set-Valued Anal. **9**, 3–11 (2001)
2. Attouch, H., Peypouquet, J., Redont, P.: A dynamical approach to an inertial forward-backward algorithm for convex minimization. SIAM J. Optim. **24**(1), 232–256 (2014)
3. Attouch, H., Chbani, Z., Fadili, J., Riahi, H.: Fast convergence of dynamical ADMM via time scaling of damped inertial dynamics. J. Optim. Theory Appl. **193**(1), 704–736 (2022)
4. Auslender, A., Teboulle, M.: Entropic proximal decomposition methods for convex programs and variational inequalities. Math. Program. **91**, 33–47 (2001)
5. Auslender, A., Teboulle, M.: The log-quadratic proximal methodology in convex optimization algorithms and variational inequalities. In: Equilibrium Problems and Variational Models, pp. 19–52 (2003)
6. Auslender, A., Teboulle, M., Ben-Tiba, S.: A logarithmic-quadratic proximal method for variational inequalities. In: Computational Optimization: A Tribute to Olvi Mangasarian, vol. I, pp. 31–40 (1999)
7. Bai, J., Ma, Y., Sun, H., Zhang, M.: Iteration complexity analysis of a partial LQP-based alternating direction method of multipliers. Appl. Numer. Math. **165**, 500–518 (2021)
8. Bnouhachem, A.: A self-adaptive descent LQP alternating direction method for the structured variational inequalities. Numer. Algorithms **86**, 303–324 (2021)
9. Bot, R.I., Csetnek, E.R., Hendrich, C.: Inertial Douglas-Rachford splitting for monotone inclusion problems. Appl. Math. Comput. **256**, 472–487 (2015)
10. Bot, R.I., Csetnek, E.R., László, S.C.: An inertial forward-backward algorithm for the minimization of the sum of two nonconvex functions. EURO J. Comput. Optim. **4**, 3–25 (2016)
11. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)
12. Cai, X., Gu, G., He, B., Yuan, X.: A proximal point algorithm revisit on the alternating direction method of multipliers. Sci. China Math. **56**, 2179–2186 (2013)
13. Cai, X., Guo, K., Jiang, F., Wang, K., Wu, Z., Han, D.: The developments of proximal point algorithms. J. Operat. Res. Soc. China **10**(2), 197–239 (2022)
14. Chen, C., Chan, R.H., Ma, S., Yang, J.: Inertial proximal ADMM for linearly constrained separable convex optimization. SIAM J. Imag. Sci. **8**(4), 2239–2267 (2015)
15. Chen, C., Li, M., Yuan, X.: Further study on the convergence rate of alternating direction method of multipliers with logarithmic-quadratic proximal regularization. J. Optim. Theory Appl. **166**, 906–929 (2015)
16. Chen, C., Ma, S., Yang, J.: A general inertial proximal point algorithm for mixed variational inequality problem. SIAM J. Optim. **25**(4), 2120–2142 (2015)
17. Dang, Y., Chen, L., Gao, Y.: Multi-block relaxed-dual linear inertial ADMM algorithm for nonconvex and nonsmooth problems with nonseparable structures. Numer. Algorithms 1–35 (2024)
18. Deng, Z., Han, D.: Multi-step inertial strictly contractive PRSM algorithms for convex programming problems with applications. J. Comput. Appl. Math. **437**, 115469 (2024)
19. Douglas, J., Rachford, H.H.: On the numerical solution of heat conduction problems in two and three space variables. Trans. Am. Math. Soc. **82**(2), 421–439 (1956)

20. Eckstein, J., Yao, W.: Understanding the convergence of the alternating direction method of multipliers: theoretical and computational perspectives. Pacific J. Optim. **11**(4), 619–644 (2015)
21. Gabay, D.: Applications of the method of multipliers to variational inequalities. In: Fortinand, M., Glowinski, R. (eds.) Augmented Lagrange Methods: Applications to the Solution of Boundary-Valued Problems, pp. 299–331. North-Holland, Amsterdam (1983)
22. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput. Math. Appl. **2**(1), 17–40 (1976)
23. Glowinski, R., Marroco, A.: Sur lapproximation, parlments nis dordre un, et la rsolution, par penalisation-dualit, dune classe de problmes de Dirichlet non linaires. Revue Francaise d'automatique, Informatique, Recherche Operationnelle. Analyse Numerique **9**(R2), 41–76 (1975)
24. Goldfarb, D., Ma, S., Scheinberg, K.: Fast alternating linearization methods for minimizing the sum of two convex functions. Math. Program. **141**(1), 349–382 (2013)
25. Gu, Y., Jiang, B., Han, D.: A semi-proximal-based strictly contractive Peaceman-Rachford splitting method, pp. 1–20 (2015). arXiv:1506.02221
26. Han, D.: A survey on some recent developments of alternating direction method of multipliers. J. Oper. Res. Soc. China 1–52 (2022)
27. He, B., Liu, H., Wang, Z., Yuan, X.: A strictly contractive Peaceman-Rachford splitting method for convex programming. SIAM J. Optim. **24**(3), 1011–1040 (2014)
28. He, B., Ma, F., Yuan, X.: Convergence study on the symmetric version of ADMM with larger step sizes. SIAM J. Imag. Sci. **9**(3), 1467–1501 (2016)
29. He, B., Yuan, X.: On the O(1/n) convergence rate of the Douglas-Rachford alternating direction method. SIAM J. Numer. Anal. **50**(2), 700–709 (2012)
30. He, B., Yuan, X.: On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers. Numer. Math. **130**(3), 567–577 (2015)
31. Hien, L.T.K., Papadimitriou, D.: An inertial ADMM for a class of nonconvex composite optimization with nonlinear coupling constraints. J. Glob. Optim. 1–22 (2024)
32. Hien, L., Phan, D.N., Gillis, N.: Inertial alternating direction method of multipliers for non-convex non-smooth optimization. Comput. Optim. Appl. **83**(1), 247–285 (2022)
33. Jiang, F., Wu, Z., Cai, X.: Generalized ADMM with optimal indefinite proximal term for linearly constrained convex optimization. J. Ind. Manag. Optim. **16**(2), 835–856 (2020)
34. Lai, Z.R., Yang, P.Y., Fang, L., Wu, X.: Short-term sparse portfolio optimization based on alternating direction method of multipliers. J. Mach. Learn. Res. **19**(63), 1–28 (2018)
35. Li, M., Li, X., Yuan, X.: Convergence analysis of the generalized alternating direction method of multipliers with logarithmic-quadratic proximal regularization. J. Optim. Theory Appl. **164**, 218–233 (2015)
36. Li, M., Yuan, X.: A strictly contractive Peaceman-Rachford splitting method with logarithmic-quadratic proximal regularization for convex programming. Math. Oper. Res. **40**(4), 842–858 (2015)
37. Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. **16**(6), 964–979 (1979)
38. Liu, Y., Guo, K., Yang, M.: Convergence study on the logarithmic-quadratic proximal regularization of strictly contractive Peaceman-Rachford splitting method with larger step-size. Int. J. Comput. Math. **97**(8), 1744–1766 (2020)
39. Lorenz, D.A., Pock, T.: An inertial forward-backward algorithm for monotone inclusions. J. Math. Imaging Vis. **51**, 311–325 (2015)
40. Nagurney, A., Zhang, D.: Projected Dynamical Systems and Variational Inequalities with Applications, vol. 2. Springer Science & Business Media (2012)
41. Peaceman, D.W., Rachford, H.H., Jr.: The numerical solution of parabolic and elliptic differential equations. J. Soc. Ind. Appl. Math. **3**(1), 28–41 (1955)
42. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Comput. Math. Math. Phys. **4**(5), 1–17 (1964)
43. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control. Optim. **14**(5), 877–898 (1976)

44. Tao, M., Yuan, X.: On the O(1/t) convergence rate of alternating direction method with logarithmic-quadratic proximal regularization. SIAM J. Optim. **22**(4), 1431–1448 (2012)
45. Wang, X., Shao, H., Liu, P., Wu, T.: An inertial proximal partially symmetric ADMM-based algorithm for linearly constrained multi-block nonconvex optimization problems with applications. J. Comput. Appl. Math. **420**, 114821 (2023)
46. Wu, Z., Cai, X., Han, D.: Linearized block-wise alternating direction method of multipliers for multiple-convex programming. J. Ind. Manag. Optim. **14**(3), 833–855 (2018)
47. Wu, Z., Li, M.: An LQP-based symmetric alternating direction method of multipliers with larger step sizes. J. Oper. Res. Soc. China **7**, 365–383 (2019)
48. Wu, Z., Li, M.: General inertial proximal gradient method for a class of nonconvex nonsmooth optimization problems. Comput. Optim. Appl. **73**, 129–158 (2019)
49. Wu, Z., Li, M., Wang, D.Z., Han, D.: A symmetric alternating direction method of multipliers for separable nonconvex minimization problems. Asia-Pacific J. Oper. Res. **34**(06), 1750030 (2017)
50. Wu, Z., Liu, F., Li, M.: A proximal Peaceman-Rachford splitting method for solving the multi-block separable convex minimization problems. Int. J. Comput. Math. **96**(4), 708–728 (2019)
51. Yuan, X., Li, M.: An LQP-based decomposition method for solving a class of variational inequalities. SIAM J. Optim. **21**(4), 1309–1318 (2011)
52. Zaslavski A.J.: Proximal Point Algorithm. Springer International Publishing (2016)

Chapter 5
A Class of Augmented-Lagrangian-Type Algorithms for Solving Generalized Nash Equilibrium Problems

Xiaoxi Jia, Shiwei Wang, and Lingling Xu

Abstract In this paper, we study the solution of convex generalized Nash equilibrium problems (GNEP) with shared linear constraints, and propose a class of regularized augmented Lagrangian methods. The idea is to penalize the shared linear constraints into the augmented Lagrangian function of each player, so as to construct a convex Nash equilibrium subproblem (NEP). Under the strong monotonicity and Lipschitz continuity assumptions of pseudo-gradient, we prove the Fejér monotonicity of iterative points with respect to the set of solutions. Under the cocoercivity assumption of pseudo-gradient, the iterative scheme of the algorithm is equivalent to the forward-backward splitting algorithm for solving the zero of an operator. If one more correction step is added, the cocoercivity hypothesis of pseudo-gradient can be weakened to the Lipschitz continuity hypothesis. Some numerical examples are given to verify the effectiveness of the algorithm.

Keywords Generalized Nash equilibrium problem · Augmented Lagrangian method · Forward-backward splitting · Convergence analysis

5.1 Introduction

We consider the GNEP of N players with common linear equality constraints, and the optimization problem of each player is as follows

X. Jia
Department of Mathematics and Computer Science, Saarland University, Saarbrcken, Germany
e-mail: xiaoxijia@math.uni-sb.de

S. Wang
School of Mathematical Sciences, Ministry of Education Key Laboratory for NSLSCS, Nanjing Normal University, Nanjing, China

L. Xu (✉)
School of Mathematical Sciences, Ministry of Education Key Laboratory for NSLSCS, Nanjing Normal University, Nanjing, China
e-mail: xulingling@njnu.edu.cn

X. Wang et al. (eds.), *Splitting Optimization*, ICIAM2023 Springer Series 2,
https://doi.org/10.1007/978-981-96-8574-5_5

$$\min_{x_\nu \in \mathcal{X}_\nu} \theta_\nu(x_\nu, x_{-\nu}) \qquad \text{s.t.} \qquad \sum_{\nu=1}^{N} A_\nu x_\nu = b. \tag{5.1}$$

More generally, common linear inequality constraints are also considered

$$\min_{x_\nu \in \mathcal{X}_\nu} \theta_\nu(x_\nu, x_{-\nu}) \qquad \text{s.t.} \qquad \sum_{\nu=1}^{N} A_\nu x_\nu \le b, \tag{5.2}$$

where $x_\nu \in \mathbb{R}^{n_\nu}$ denotes the decision set vector of player v, $\nu = 1, 2, \ldots, N, \sum_{\nu=1}^{N} n_\nu = n$. Write $x := (x_\nu, x_{-\nu})$, where $x_{-\nu}$ represents all decision vectors except x_ν. This notation is used to emphasize the role of the sub-vector block x_ν. Therefore, we have $x = (x_\nu, x_{-\nu}) = (x_1, x_2, \ldots, x_N), \quad (y_\nu, x_{-\nu}) = (x_1, \ldots, x_{\nu-1}, y_\nu, x_{\nu+1}, \ldots, x_N)$. $\mathcal{X}_\nu \subseteq \mathbb{R}^{n_\nu}$ is a non-empty closed convex set. $\theta_\nu : \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_N} \to \mathbb{R}$ is a continuous differentiable function. For the fixed $x_{-\nu}$, $\theta_\nu(\cdot, x_{-\nu})$ is convex. $A_\nu \in \mathbb{R}^{m \times n_\nu}$ and $b \in \mathbb{R}^m$ are constraint matrix and constraint vector, respectively.

For convenience, we introduce notation

$$\mathcal{X} := \mathcal{X}_1 \times \ldots \times \mathcal{X}_N \subseteq \mathbb{R}^n, \quad x := (x_1, x_2, \ldots, x_N) \in \mathbb{R}^n,$$
$$A := (A_1, A_2, \ldots, A_N) \in \mathbb{R}^{m \times n}, \quad \mathcal{F} := \{x \in \mathcal{X} \mid Ax = b\}.$$

The general GNEP is expressed as (5.3).

$$\min_{x_\nu \in \mathcal{X}_\nu} \theta_\nu(x_\nu, x_{-\nu}) \qquad \text{s.t.} \quad x_\nu \in X_\nu(x_{-\nu}), \tag{5.3}$$

where X_ν represents the feasible set mapping of the ν-th player. After the other players' strategies $x_{-\nu}$ are given, the ν-th player's feasible strategy set can be obtained through X_ν. Generally, the solution of GNEP is to find a strategy vector $x^* = (x_1^*, \ldots, x_N^*)$, such that for any $\nu = 1, \ldots, N$, we have

$$\theta_\nu(x^*) \le \theta_\nu(x_\nu, x_{-\nu}^*) \quad \forall x_\nu \in X_\nu(x_{-\nu}^*).$$

As early as 1955, Nikaido and Isoda [1] proposed the NI function to reformulate the general GNEP in a constrained quasi-optimization problem, where the objective function can be regarded as an evaluation function constructed from the NI function. Here, "quasi-optimization" means that the problem is not a standard optimization problem because the set of constraints is closely related to the decision variables. In 1974, Bensoussan [2] transformed convex GNEP into a quasi-variational inequality problem (QVI for short). Correspondingly, it is also obtained that the convex NEP can be equivalent to a certain variational inequality (VI for short) problem. But the solution for QVI is also very difficult. Rosen [3] considered the jointly convex case in 1965. At this time, the constraint expressions of each player are characterized as the same non-positive convex function. The idea still needs to minimize the evaluation function, so the solution to this problem is still quite difficult. Under the assumption of joint convexity, a solution of QVI, which is the so-called variational equilibrium or regular equilibrium, is the definite solution of GNEP. In addition to the algorithms for solving the GNEP problem, research on the existence, uniqueness, stability, and KKT-type conditions of its solutions is also being followed up. For details, please refer to the two review papers in 2010 and 2014 [4, 5].

The algorithms of convex GNEP include augmented Lagrangian methods (ALM), penalty function methods, Newton methods, split methods, etc. In GNEP, when some players have common constraints, inherent singularities arise when solving GNEP, see [6] for details, which causes much difficulty in designing proper second-order methods where Hessian or Hessian approximation is necessary, such as Newton-type methods [6, 7]. This difficulty inspired us to start looking for algorithms with at least good information about differentiation. Penalty function algorithms are one of them. The first penalty function algorithm for GNEP was proposed by Fukushima [8], and the literature [9, 10] also proposed similar penalty algorithms. But these algorithms have a common defect. Their subproblems are non-smooth NEP, so it is difficult to solve numerically. However, this disadvantage can be overcome by introducing the multiplier, that is, applying the ALM-type methods.

The ALM methods (or multiplier penalty function methods) are the classic methods for solving constrained optimization problems. This type of method is often applied to GNEP. For details, please refer to the review literature [4, 5]. The subproblems of ALM-type methods are generally highly smooth and therefore easier to solve. Pang and Fukushima [11] used ALM-type methods to solve quasi-variational inequality (QVI) problems, and GNEP is a special class of QVI problems. [12] proposed an improved ALM algorithm for QVI. Kanzow in [13] also proposed an ALM algorithm directly for GNEP.

The splitting-type approach can be applied to NEP with relative ease. For example, applying the Gauss-Seidel-type algorithms to NEP, i.e., for participants $\nu = 1, 2, \ldots, N$, do

$$x_\nu^{k+1} = \arg\min_{x_\nu \in \mathcal{X}_\nu} \theta_\nu(x_1^{k+1}, \ldots, x_{\nu-1}^{k+1}, x_\nu, x_{\nu-1}^k, \ldots, x_N^k).$$

But its generalization on GNEP becomes more complicated because the feasible set of each participant in GNEP is influenced by the decision variables of the remaining participants. The literature [14] applied this type of splitting method to a class of potential game problems and obtained the corresponding convergence analysis, where the inner semi-continuity of feasible set-valued maps is required, which, however, is usually not easy to verify. The so-called ADMM is also proposed in the literature [15, 16] where the (multiplier) penalty term was employed to get rid of the assumption of inner semi-continuity.

The ADMM method is a well-known algorithm for large-scale optimization problems with block structures. It is efficient for the problems with two-block structures, but for problems with three or more blocks, the classical iterative schemes should be properly corrected or more restricted assumptions should be addicted. For details, please refer to the literature [17–25]. Recently, Kanzow et al. proposed Jacobi-ADMM-type methods [26] and Gauss-Seidel-ADMM-type methods [27] for GNEP with common linear constraints in an infinite-dimensional space, referring to ADMM-type algorithms for solving classical optimization problems.

The Jacobi-ADMM method minimizes the augmented Lagrangian function of the linearized objective function for each participant with an additional regularization term, and then solves it in parallel. Its algorithm is shown in Algorithm 1. Börgens and Kanzow [26] prove that its iterative format (5.4), (5.5) is equivalent to a forward-backward splitting algorithm for solving the zero point of an operator. And the convergence of iterative schemes (5.4) and (5.5) can be obtained by the related theory of operator splitting.

Algorithm 1 Regularized linearized Jacobi-type ADMM method

```
1: Choose a starting point $(x^0, \lambda^0) \in \mathcal{X} \times \mathbb{R}^m$, parameters $\beta,\ \gamma > 0$.
2: If a suitable termination criterion is satisfied: STOP.
3: For $\nu = 1, 2, 3, \dots, N$, compute
   $$x_\nu^{k+1} := \arg\min_{x_\nu \in \mathcal{X}_\nu} \Big\{ \Big\langle \nabla_{x_\nu}\theta_\nu(x^k), x_\nu - x_\nu^k \Big\rangle + \langle \lambda^k, A_\nu x_\nu \rangle + \frac{\beta}{2}\big(\|A_\nu x_\nu + \sum_{i \neq \nu} A_i x_i^k - b\|^2 + \gamma \|x_\nu - x_\nu^k\|^2\big)\Big\}. \tag{5.4}$$
4: Compute
   $$\lambda^{k+1} := \lambda^k + \beta \left( \sum_{\nu=1}^{N} A_\nu x_\nu^{k+1} - b \right). \tag{5.5}$$
5: Set $k := k + 1$, and go to Step 1.
```

The Gauss-Seidel-ADMM-type method is shown in Algorithm 2. Different from Algorithms 1 and 2 only adds a regular term to the augmented Lagrangian function, and then solves the subproblems one by one in sequence. Under strong assumptions (such as strong monotonicity of pseudo-gradients, etc.), Fejér monotonicity of iterates can be obtained.

Algorithm 2 Regularized Gauss-Seidel-type ADMM method

```
1: Choose a starting point $(x^0, \lambda^0) \in \mathcal{X} \times \mathbb{R}^m$, parameters $\beta,\ \gamma > 0$.
2: If a suitable termination criterion is satisfied: STOP.
3: For $\nu = 1, 2, 3, \ldots, N$, compute
```

$$\begin{aligned} x_\nu^{k+1} := \arg\min_{x_\nu \in X_\nu} \big\{ &\theta_\nu(x_1^{k+1}, \ldots, x_{\nu-1}^{k+1}, x_\nu, x_{\nu+1}^k, \ldots, x_N^k) \\ &+ \langle \lambda^k, A_\nu x_\nu \rangle + \frac{\gamma}{2} \left\| x_\nu - x_\nu^k \right\|^2 \\ &+ \frac{\beta}{2} \| A_\nu x_\nu + \sum_{i=1}^{\nu-1} A_i x_i^{k+1} + \sum_{i=\nu+1}^{N} A_i x_i^k - b \|^2 \big\}. \end{aligned} \tag{5.6}$$

```
4: Compute
```

$$\lambda^{k+1} = \lambda^k + \beta \left(\sum_{\nu=1}^{N} A_\nu x_\nu^{k+1} - b \right). \tag{5.7}$$

```
5: Set $k := k + 1$, and go to Step 1.
```

Based on these two ADMM-type methods, we propose the corresponding augmented Lagrangian method in this paper. The difference is that we do not solve the subproblems with a block-by-block strategy like the two ADMM-type algorithms, but solve a convex NEP. The connection between the proposed algorithms and operator splitting methods is discussed. We prove the convergence of the algorithms under mild assumptions.

This paper contains four sections. The first section describes the GNEP with a linear equality constraint and introduces several existing related algorithms. In Sect. 5.2 we recall some basic knowledge and notation. In the third section, several types of augmented Lagrangian-type algorithms are proposed, namely, the regularized linearized augmented Lagrangian algorithm, the regularized augmented Lagrangian algorithm, and the corrected regularized linearized augmented Lagrangian algorithm, where a correction step is added to the regularized linearized augmented Lagrangian algorithm. Under the mild assumptions, their Fejér monotonicity of the iterative point sequence is obtained. Section 5.4 carries out the numerical experiments. Seven numerical examples of GNEP are solved by several augmented Lagrangian-type algorithms.

5.2 Preliminaries

In this section, some basic definitions and notations are listed first. Then, a reformulation of the GNEP is given, which aims to find the zero point of the corresponding operator. Finally, the splitting algorithm is briefly introduced for tracking the zero point.

5.2.1 Basic Definitions

Let $\mathbb{R}^n$ be a real n dimensional vector space, $\langle \cdot, \cdot \rangle$ is the inner product defined on $\mathbb{R}^n$, i.e. $\langle x, y \rangle = x^T y$. Define $\|x\| = \sqrt{x^T x}$. Given a symmetric positive definite matrix Q, we define the inner product induced by Q as

$$\langle x, y \rangle_Q := \langle x, Qy \rangle = x^T Qy.$$

The norm induced by the inner product is denoted as $\|x\|_Q = \sqrt{x^T Qx}$. When the sign of the norm has no subscript, it still defaults to 2-norm.

For the theoretical proof, we need to recall some knowledge of operator theory. Let $\mathcal{H}$ be a Hilbert space. For the set-valued operator $T : \mathcal{H} \to 2^{\mathcal{H}}$, its domain is defined as $dom\ T := \{x \in \mathcal{H} \mid T(x) \neq \emptyset\}$. The graph of the operator T is defined as $grap\ T := \{(x, u) \in \mathcal{H} \times \mathcal{H} \mid u \in T(x)\}$. Its zero point set is defined as $zer\ T := \{x \in \mathcal{H} \mid 0 \in T(x)\}$.

An operator T is called monotonic if it satisfies

$$\langle u - v, x - y \rangle \geq 0 \quad \forall (x, u), (y, v) \in grap\ T.$$

On the basis of monotonicity, we give the definition of a maximal monotone operator.

Definition 5.1 Let the operator $A : \mathcal{H} \to 2^{\mathcal{H}}$ be a monotone operator. A is called a maximal monotone operator if there is no monotone operator $B : \mathcal{H} \to 2^{\mathcal{H}}$ that satisfies $grap\ A \subsetneqq grap\ B$. That is, for all $(x, u) \in \mathcal{H} \times \mathcal{H}$, we have

$$(x, u) \in grap\ A \iff \langle x - y, u - v \rangle \geq 0 \quad \forall (y, v) \in grap\ A. \tag{5.8}$$

The set-valued operator T is called strongly monotonic, if there exists a constant $\rho > 0$, such that

$$\langle u - v, x - y \rangle \geq \rho \|x - y\|^2 \quad \forall (x, u), (y, v) \in grap\ T.$$

The subdifferential of a convex function $f : \mathcal{H} \to (-\infty, \infty]$ at a point x_0 is defined as

$$\partial f(x_0) := \{g \in \mathcal{H} \mid f(x) - f(x_0) \geq \langle g, x - x_0 \rangle, \forall x \in \mathcal{H}\}.$$

It is easy to know that the subdifferential operator of a closed, convex function is a maximal monotone operator.

The single-valued operator $T : \mathcal{H} \to \mathcal{H}$ is called α-cocoercive, if for $\alpha > 0$, we have

$$\langle T(x) - T(y), x - y \rangle \geq \alpha \|T(x) - T(y)\|^2 \quad \forall x, y \in \mathcal{H}.$$

By the Cauchy-Schwarz inequality, it is easy to prove that the α-cocoercive operator is also $1/\alpha$-Lipschitz continuous. And it is easy to prove that an operator satisfying ρ-strong monotonicity and L-Lipschitz continuity is ρ/L^2-cocoercive. On the other hand, cocoercivity cannot infer strong monotonicity. In the GNEP for inequality constraints proposed later, after the introduction of new variables, the gradient of the objective function will no longer maintain strong monotonicity, but the cocoercivity can be retained.

Given a set $\mathcal{X} \subseteq \mathcal{H}$, its normal cone is defined as

$$N_{\mathcal{X}}(x) := \begin{cases} \{s \in \mathcal{H} \mid \langle s, y - x \rangle \leq 0, \forall y \in \mathcal{X}\} & \text{if } x \in \mathcal{X}, \\ \emptyset \quad \text{if } x \notin \mathcal{X}. \end{cases}$$

Now we introduce an important concept about iterative sequences.

Definition 5.2 (*Fejér monotonicity*) Let C be a non-empty subset of $\mathcal{H}$, $\{x_k\}_{k\in\mathbb{N}}$ is the point sequence in $\mathcal{H}$, we call that $\{x_k\}_{k\in\mathbb{N}}$ is Fejér monotone about the set C, if

$$\|x^{k+1} - x\| \leq \|x^k - x\| \quad \forall k \in \mathbb{N},\ x \in C.$$

There is an important result about Fejér monotone sequences. See the proof in [28].

Lemma 5.1 *Let $\{x_k\}_{k\in\mathbb{N}}$ be the sequence in $\mathcal{H}$, and C is a non-empty subset of $\mathcal{H}$. If $\{x_k\}_{k\in\mathbb{N}}$ is Fejér monotone about the set C, and each cluster point of the sequence is in the set C, then $\{x_k\}_{k\in\mathbb{N}}$ converges to a certain point in C.*

Some common basic inequalities, such as the Cauchy-Schwarz inequality and the following Young inequality, will also be used in the theoretical analysis.

Lemma 5.2 (Young's inequality) *For all $a, b \in \mathbb{R}$, $\varepsilon > 0$, we have $|a \cdot b| \leq \frac{\varepsilon}{2}a^2 + \frac{1}{2\varepsilon}b^2$.*

5.2.2 Reformulation of GNEP

In this subsection, we will reformulate the GNEP (5.1) considered in this paper as the problem of finding the zeros of a maximal monotone operator. For this purpose, two special types of solutions to GNEP (5.1) need to be considered.

First, two classes of convexity assumptions about general GNEP are introduced. For the general GNEP (5.3), the constraint $X_\nu(x_{-\nu}) = \{x_\nu \in \mathcal{X}_\nu \mid g_\nu(x_\nu, x_{-\nu}) \leq 0\}$, then GNEP (5.3) can be reformulated as

$$\min_{x_\nu \in \mathcal{X}_\nu} \theta_\nu(x_\nu, x_{-\nu}) \qquad \text{s.t.} \quad g_\nu(x_\nu, x_{-\nu}) \leq 0, \tag{5.9}$$

where $g_\nu : \mathcal{H} \to \mathbb{R}^{m_\nu}$ is the constraint function of the ν-th player, and the objective function is continuously differentiable. The problem is said to satisfy the player convexity, if for fixed $x_{-\nu}$, $\theta_\nu(\cdot, x_{-\nu})$ is convex, and the feasible set $X_\nu(x_{-\nu})$ is also convex. Under the framework of linear equality or inequality constraints in this paper, the player convexity is obviously satisfied.

Joint convexity means that the GNEP is first of all player convex, and there is a closed convex set $\mathcal{F} \subseteq \mathcal{H}$, such that $X_\nu(x_{-\nu}) = \{x_\nu \in \mathcal{X}_\nu \mid (x_\nu, x_{-\nu}) \in \mathcal{F}\}$. Under the framework of the problem in this paper, joint convexity is obviously satisfied. Some properties of joint convex GNEP are given below.

Let us now define the pseudo-gradient of GNEP (or NEP), which represents the re-aggregation of the gradient of each player's objective function with respect to its own decision variable, i.e.,

$$\hat{P}(x) = \begin{pmatrix} \nabla_{x_1}\theta_1(x) \\ \vdots \\ \nabla_{x_N}\theta_N(x) \end{pmatrix}.$$

For the jointly convex GNEP, we define a corresponding variational inequality (VI) problem, that is, to find $x^* \in \mathcal{F}$, such that

$$\langle x - x^*, \hat{P}(x^*)\rangle \geq 0 \quad \forall x \in \mathcal{F}.$$

Let us write this variational inequality problem as VI($\mathcal{F}$, $\hat{P}$). According to the literature [4], for joint convex GNEP, the solution of this VI is also the solution of GNEP. The solution of GNEP corresponding to this VI is called the variational equilibrium or normalized equilibrium.

Lemma 5.3 ([26]) *x^* is the variational equilibrium of joint convex GNEP, which is equivalent to* $0 \in \hat{P}(x^*) + N_{\mathcal{F}}(x^*)$.

In this manuscript, we consider the constraints as $\mathcal{F} = \{x \in \mathcal{X} \mid Ax = b\}$, that is, a linear constraints $Ax = b$ and an abstract constraint $x \in \mathcal{X}$. For this purpose, the concept of variational KKT point is introduced.

Definition 5.3 (*Variational KKT point*) The point $(x^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^m$ is called a variational KKT point (or a variational KKT pair) of GNEP (5.1), if it satisfies the KKT-type condition

$$0 \in \hat{P}(x^*) + A^T\lambda^* + N_{\mathcal{X}}(x^*) \qquad Ax^* - b = 0. \tag{5.10}$$

From this definition, it can be intuitively seen that the variational KKT point can be regarded as the KKT point in the case where the multipliers of each player are the same. Regarding the relationship between the variational KKT point and the variational equilibrium, the following results are obtained.

Proposition 5.1 *If (x^*, λ^*) is the variational KKT point of GNEP (5.1), then x^* is also the variational equilibrium for GNEP.*

Note that the converse of Proposition 5.1 does not hold, which can be achieved under the regularity conditions [26].

The above results tell us that a variational KKT point is the variational equilibrium, and a variational equilibrium is the solution of GNEP. Therefore, a solution of the variational KKT point is a candidate for obtaining the solution of GNEP.

Now we write the variational KKT condition in a compact form. Let

$$\mathcal{W} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_N \times \mathbb{R}^m, \quad w := (x_1, \ldots, x_N, \lambda).$$

The pseudo-gradient is consequently expanded and regarded as a function of w, denoted as

$$P(w) := \begin{pmatrix} \hat{P}(x) \\ 0 \end{pmatrix}.$$

Define operator $G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$ as

$$G(w) := \begin{pmatrix} A_1^T \lambda \\ \vdots \\ A_N^T \lambda \\ b - \sum_{\nu=1}^N A_\nu x_\nu \end{pmatrix}. \tag{5.11}$$

and

$$G_0 := \begin{pmatrix} 0 & \cdots & 0 & A_1^T \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & A_N^T \\ -A_1 & \cdots & -A_N & 0 \end{pmatrix}, \quad b_0 := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ b \end{pmatrix}.$$

Therefore, we have $G(w) = G_0 w + b_0$. Since G_0 is an antisymmetric matrix, so we have

$$\langle Gx, x \rangle = 0 \quad \forall x \in \mathbb{R}^n.$$

With the above notation, the variational KKT condition can also be expressed in a compact form, see the following lemma.

Lemma 5.4 ([26]) *The point $w^* = (x^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^m$ is a variational KKT point of GNEP (5.1) if and only if $w^* \in \mathcal{W}^*$, where $\mathcal{W}^* = \{w \in \mathbb{R}^n \times \mathbb{R}^m \mid 0 \in P(w) + G(w) + N_{\mathcal{W}}(w)\}$.*

Define a set-valued operator

$$T(w) := P(w) + G(w) + N_{\mathcal{W}}(w). \tag{5.12}$$

Its domain is obviously the non-empty set $\mathcal{W}$, and its zero point is the variational KKT point of GNEP (5.1).

We next prove the maximal monotonicity of the operator T. This conclusion can be proved by proving that each part of the operator T is maximally monotone, and each part satisfies the condition of the addition operation, maintaining maximal monotonicity. Please see the following lemmas, which are generally from [28, 29].

Lemma 5.5 *Let the single-valued operator $A : \mathcal{H} \to \mathcal{H}$ be continuous and monotone, then A is a maximal monotone operator.*

We then illustrate that the operator G defined in (5.11) is maximally monotone.

Lemma 5.6 *Let $\mathcal{W} \subset \mathcal{H}$ be a non-empty closed convex set, $A : \mathcal{W} \to \mathcal{H}$ is a continuous monotone operator, then the operators $N_{\mathcal{W}}(\cdot)$ and $A + N_{\mathcal{W}}(\cdot)$ are both maximally monotone.*

Lemma 5.7 *Let $A, B : \mathcal{H} \to 2^{\mathcal{H}}$ be two maximal monotone operators, if any of the following conditions hold:*

1. *$dom B = \mathcal{H}$,*
2. *$dom A \cap int dom B \neq \emptyset$,*

Then $A + B$ is also a maximal monotone operator.

The maximal monotonicity of the operator P also needs to be explained below. Since P is a single-valued mapping, from Lemma 5.5, it is only necessary to assume that it is a monotone operator.

Proposition 5.2 *Suppose operator $P : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$ is monotone, then the set-valued operator T is maximally monotone.*

Till now, we have already obtained that operator T is maximally monotone provided that P is monotone. In the finite-dimensional Euclidean space, the maximal monotonicity of the operator T can lead to the closedness of its image. However, this conclusion is not true in infinite dimensions, and the strong or weak convergence of the sequence needs to be taken into account. Under the condition of infinite dimension, the strong-weak sequential closedness of the operator T can be derived. See the following lemma, and its proof, see [28].

Lemma 5.8 *Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be a maximal monotone operator, and $\{(x_k, u_k)\}_{k\in\mathbb{N}}$ is a bounded sequence in $gra\ A$, for $(x, u) \in \mathcal{H} \times \mathcal{H}$, if any of the following conditions are true:*

1. *$x_k \to x$, $u_k \rightharpoonup u$,*
2. *$x_k \rightharpoonup x$, $u_k \to u$,*

then we have $(x, u) \in gra\ A$.

5.2.3 Splitting-Type Methods

This section introduces some basics about the forward-backward splitting algorithm. We first recall the classic forward-backward splitting algorithm, which is employed for solving the zeros of the set-valued operator $T : \mathcal{H} \to 2^{\mathcal{H}}$, where T can be expressed as $A + B$, with $A : \mathcal{H} \to 2^{\mathcal{H}}$, operator $B : \mathcal{H} \to \mathcal{H}$. Let $x^{k+1} := (I + \beta A)^{-1}(I - \beta B)x^k$. This iteration can also be divided into two steps, denoted as $y^k := x^k - \beta Bx^k$, $x^{k+1} := (I + \beta A)^{-1} y^k$. The explicit calculation of y^k is called the forward step, and the solution of x^{k+1} is called the backward step, which is the so-called forward-backward splitting algorithm.

Under appropriate conditions, the forward-backward splitting algorithm can converge to the zero point of the operator T.

Lemma 5.9 ([29]) (Forward-backward splitting) *Let $T : \mathcal{H} \to 2^{\mathcal{H}}$ be a set-valued map with at least one zero. $T := A + B$, where the mapping $A : \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone, the mapping $B : \mathcal{H} \to \mathcal{H}$ is single-valued and α-cocoercive. Let $x^0 \in \mathcal{H}$, and generate the sequence $\{x^k\}_{k\in\mathbb{N}}$ as follows*

$$\begin{cases} y^k = (I - c_k B)(x^k), \\ x^{k+1} = J_{c_k A}(y^k), \end{cases}$$

where the parameter c_k satisfies the existence of $M > m > 0$, such that

$$0 < m \leq c_k \leq M < 2\alpha \quad \forall k.$$

Then the sequence $\{x^k\}_{k\in\mathbb{N}}$ converges to a zero of T.

Another class of split-type algorithms is proposed by Tseng in [28], a simplified version is that $x^{k+1} = [(I - \beta B)(I + \beta A)^{-1}(I - \beta B) + \beta B]x^k$. Respectively denote $y^k := x^k - \beta Bx^k$, $z^k = (I + \beta A)^{-1} y^k$, then $x^{k+1} := z^k + \beta(Bx^k - Bz^k)$. This algorithm is also known as the forward-backward-forward algorithm.

Under some mild conditions, the algorithm can converge to the zero of the operator T.

Lemma 5.10 ([28]) *Let D be a non-empty subset of $\mathcal{H}$, f, g are appropriate convex lower semicontinuous functions on $\mathcal{H}$, $dom\partial f \subset D$, g is G-differentiable on D, $argmin$ $(f + g) \neq \emptyset$, and ∇g is $1/\alpha$-Lipschitz continuous on $dom\partial f$ $(\alpha > 0)$. Let $x^0 \in \mathcal{H}$, $\beta \in (0, \alpha)$, and generate the sequence $\{x^k\}_{k\in\mathbb{N}}$ as follows*

$$\begin{cases} y^k = x^k - \beta\nabla g(x^k), \\ z^k = Prox_{\beta f}(y^k), \\ x^{k+1} = z^k + \beta(\nabla g(x^k) - \nabla g(z^k)). \end{cases} \tag{5.13}$$

Then the sequence $\{x^k\}_{k\in\mathbb{N}}$ converges to a zero of T.

5.2.4 Assumptions

This subsection introduces four assumptions, which are the basic assumptions that need to be satisfied throughout the paper, as well as the strong monotonicity assumption, the cocoercivity assumption, and the Lipschitz continuity assumption of the pseudo-gradients. They will be used successively in the following convergence proofs.

Assumption 5.1 *For all $\nu = 1, \ldots, N,\ A_\nu \in \mathbb{R}^{m\times n_\nu},\ b \in \mathbb{R}^m$, assuming $\theta_\nu(\cdot, x_{-\nu})$ is convex and continuously differentiable, and its gradient is continuous about $x = (x_1, \ldots, x_N)$, the pseudo-gradient $\hat{P}$ is monotone with respect to $x = (x_1, \ldots, x_N)$. Let $\mathcal{X}$ be a non-empty closed convex set, and the feasible set $\mathcal{F} = \{x \in \mathcal{X} \mid Ax = b\}$ is non-empty.*

Assumption 5.2 *Suppose the pseudo-gradient $\hat{P}$ is ρ-strongly monotone at the solution point x^*, that is, for $\rho > 0$, we have*

$$\langle \hat{P}(x) - \hat{P}(x^*), x - x^*\rangle \geq \rho\|x - x^*\|^2 \quad \forall x \in \mathbb{R}^n. \tag{5.14}$$

Assumption 5.3 *Suppose the pseudo-gradient $\hat{P}$ is α-cocoercive , that is, for $\alpha > 0$, we have*

$$\langle \hat{P}(x) - \hat{P}(y), x - y\rangle \geq \alpha\|\hat{P}(x) - \hat{P}(y)\|^2 \quad \forall x, y \in \mathbb{R}^n. \tag{5.15}$$

Assumption 5.4 *For all $\nu = 1, \ldots, N$, assuming gradient $\nabla_{x_\nu}\theta_\nu$ is L_ν-Lipschitz continuous. That is, for $L_\nu > 0,\ \nu = 1, \ldots, N$, we have*

$$\|\nabla_{x_\nu}\theta_\nu(x) - \nabla_{x_\nu}\theta_\nu(y)\| \leq L_\nu\|x - y\| \quad \forall x, y \in \mathbb{R}^n. \tag{5.16}$$

Strong monotonicity and Assumption 5.4 are employed in [27] for Algorithm 2. Strong monotonicity, Assumptions 5.3, and 5.4 are employed for Algorithm 1 in [26], respectively. Note that the assumption of strong monotonicity in [26, 27] has been weakened in this paper, i.e., which only needs to be satisfied at the solution point.

5.3 Augmented Lagrangian-Type Algorithms

In this section, several types of augmented Lagrangian-type algorithms for solving convex generalized Nash equilibrium problems with separable linearly common equality constraints are presented, which are called regularized linearized augmented Lagrangian algorithm (RLALM), regularized augmented Lagrangian algorithm (RALM), and corrected regularized linearization augmented Lagrangian algorithm (CRLALM), respectively.

Under the assumptions of basic convexity, strong monotonicity, cocoercivity, and Lipschitz continuity of pseudo-gradients, several convergence proofs are given respectively. Specifically, under the assumption of basic convexity, strong monotonicity and Lipschitz continuity of pseudo-gradients, the Fejér monotonicity of the iterative sequence of RLALM can be proved. Similarly, we can prove the Fejér monotonicity of the iterative sequence of RALM only with the basic convexity assumption. Under the coercivity condition, it can be proved that RLALM is equivalent to the forward-backward splitting algorithm, thus proving the convergence. In addition, if an additional correction step is added to RLALM, which is CRLALM, then only the basic convexity assumption and Lipschitz continuity assumption of the gradient (or pseudo-gradient) can be used for convergence. In addition, if an additional correction step is added to RLALM, which is CRLALM, then only the basic convexity assumption and Lipschitz continuity assumption of the gradient (or pseudo-gradient) can be used for convergence. At the end of this section, we take a closer look at GNEP with inequality constraints will also be given. The idea is to convert inequality constraints into equality constraints and then use the algorithms proposed in this section.

For each player, we define the regularized linearized (partial) augmented Lagrangian function as

$$\begin{aligned} L_\nu(x, x^k, \lambda^k) := \nabla_{x_\nu}\theta_\nu(x^k)^T(x_\nu - x_\nu^k) + (\lambda^k)^T\left(\sum_{\nu=1}^N A_\nu x_\nu - b\right) \\ + \frac{\beta}{2}\|\sum_{\nu=1}^N A_\nu x_\nu - b\|^2 + \frac{\gamma}{2}\|x_\nu - x_\nu^k\|^2, \end{aligned} \tag{5.17}$$

where β is penalty parameter, and γ is the regularization parameter. Note that each player will have their own augmented Lagrangian function, and each function is still related to the decision variables of all participants, but no longer has constraints besides their own abstract constraints. In this way, the subproblems of the algorithm will become a convex Nash equilibrium problem (NEP) with convex abstract constraints.

5.3.1 Regularized Linearized Augmented Lagrangian Algorithm

The first augmented Lagrangian algorithm proposed in this paper is shown in Algorithm 3, where subproblems (5.18) should be solved at each iteration, we then update the multiplier, where the parameters γ and β are fixed.

Algorithm 3 Regularized linearized augmented Lagrangian algorithm

```
1: Choose a starting point $(x^0, \lambda^0) \in \mathcal{X} \times \mathbb{R}^m$, parameters $\beta,\ \gamma > 0$.
2: If a suitable termination criterion is satisfied: STOP.
3: Calculate the solution $x^{k+1}$ of the following NEP problem
       $$\min_{x_\nu \in \mathcal{X}_\nu} L_\nu(x, x^k, \lambda^k), \quad \nu = 1, 2, 3, \dots, N. \tag{5.18}$$
4: Compute
       $$\lambda^{k+1} = \lambda^k + \beta \left( \sum_{\nu=1}^{N} A_\nu x_\nu^{k+1} - b \right). \tag{5.19}$$
5: Set $k := k + 1$, and go to Step 1.
```

Note that the subproblems of this algorithm are a convex NEP problem. According to [4], the solution of the NEP is equivalent to the solution of the variational inequality VI($\mathcal{X}$, P_L^k), where P_L^k is defined as

$$P_L^k(x) := \begin{pmatrix} \nabla_{x_1} L_1(x, x^k, \lambda^k) \\ \vdots \\ \nabla_{x_N} L_N(x, x^k, \lambda^k) \end{pmatrix}.$$

Then the subproblem is equivalent to find $x^{k+1} \in \mathcal{X}$, such that

$$\langle x - x^{k+1}, P_L^k(x^{k+1}) \rangle \geq 0 \quad \forall x \in \mathcal{X}. \tag{5.20}$$

The optimality condition of the variational inequality VI($\mathcal{X}$, P_L^k) is

$$\begin{aligned} 0 \in & \begin{pmatrix} \nabla_{x_1} \theta_1(x^k) \\ \vdots \\ \nabla_{x_N} \theta_N(x^k) \end{pmatrix} + \begin{pmatrix} A_1^T \lambda^k \\ \vdots \\ A_N^T \lambda^k \end{pmatrix} + \beta \begin{pmatrix} A_1^T (\sum_{\nu=1}^N A_\nu x_\nu^{k+1} - b) \\ \vdots \\ A_N^T (\sum_{\nu=1}^N A_\nu x_\nu^{k+1} - b) \end{pmatrix} \\ & + \gamma \begin{pmatrix} x_1^{k+1} - x_1^k \\ \vdots \\ x_N^{k+1} - x_N^k \end{pmatrix} + \begin{pmatrix} N_{\mathcal{X}_1}(x_1^{k+1}) \\ \vdots \\ N_{\mathcal{X}_N}(x_N^{k+1}) \end{pmatrix}. \end{aligned} \tag{5.21}$$

The third term in (5.21) reflects the essential difference between the algorithm and the Jacobi-type ADMM algorithm in Algorithm 1 and the Gauss-Seidel-ADMM-type algorithm in Algorithm 2. First, review the optimality conditions for the subproblems of Algorithms 1 and 2. The optimality conditions for the subproblems of the Jacobi-type ADMM in Algorithm 1 are given below.

$$0 \in \begin{pmatrix} \nabla_{x_1}\theta_1(x^k) \\ \vdots \\ \nabla_{x_N}\theta_N(x^k) \end{pmatrix} + \begin{pmatrix} A_1^T\lambda^k \\ \vdots \\ A_N^T\lambda^k \end{pmatrix} + \beta \begin{pmatrix} A_1^T(A_1x_1^{k+1} + \sum_{\mu\neq 1} A_\mu x_\mu^k - b) \\ \vdots \\ A_N^T(A_N x_N^{k+1} + \sum_{\mu\neq N} A_\mu x_\mu^k - b) \end{pmatrix}$$
$$+ \gamma \begin{pmatrix} x_1^{k+1} - x_1^k \\ \vdots \\ x_N^{k+1} - x_N^k \end{pmatrix} + \begin{pmatrix} N_{\mathcal{X}_1}(x_1^{k+1}) \\ \vdots \\ N_{\mathcal{X}_N}(x_N^{k+1}) \end{pmatrix}. \tag{5.22}$$

The difference between (5.22) and (5.21) is the third term; the third term of (5.21) is more concise.

The optimality conditions for the subproblems of the Gauss-Seidel-type ADMM in Algorithm 2 are given below.

$$0 \in \begin{pmatrix} \nabla_{x_1}\theta_1(x_1^{k+1}, x_{>1}^k) \\ \vdots \\ \nabla_{x_\nu}\theta_\nu(x_{\leq\nu}^{k+1}, x_{>\nu}^{k+1}) \\ \vdots \\ \nabla_{x_N}\theta_N(x^{k+1}) \end{pmatrix} + \begin{pmatrix} A_1^T\lambda^k \\ \vdots \\ A_\nu^T\lambda^k \\ \vdots \\ A_N^T\lambda^k \end{pmatrix}$$
$$+\beta \begin{pmatrix} A_1^T(A_1x_1^{k+1} + \sum_{\mu>1} A_\mu x_\mu^{k+1} - b) \\ \vdots \\ A_\nu^T(\sum_{\mu\leq\nu} A_\mu x_\mu^{k+1} + \sum_{\mu>\nu} A_\mu x_\mu^{k+1} - b) \\ \vdots \\ A_N^T(\sum_{\mu=1}^N A_\mu x_\mu^{k+1}) \end{pmatrix} \tag{5.23}$$
$$+\gamma \begin{pmatrix} x_1^{k+1} - x_1^k \\ \vdots \\ x_\nu^{k+1} - x_\nu^k \\ \vdots \\ x_N^{k+1} - x_N^k \end{pmatrix} + \begin{pmatrix} N_{\mathcal{X}_1}(x_1^{k+1}) \\ \vdots \\ N_{\mathcal{X}_\nu}(x_\nu^{k+1}) \\ \vdots \\ N_{\mathcal{X}_N}(x_N^{k+1}) \end{pmatrix}.$$

In the (5.23), $x_{\leq\nu}$ represents all decision variable blocks whose subscripts do not exceed ν, and other similar symbols have similar meanings. The difference between (5.23) and (5.21) lies in the first term and the third term. Due to the characteristics of serial calculation of subproblems of the Gauss-Seidel-type ADMM algorithm, the form of its optimality condition is more complicated.

Since the subproblem of Algorithm 3 is to solve a NEP as a whole, the third term in the (5.21) only appears in the iteration point x^{k+1} of the $(k+1)$-th step, which is well symmetry. And combining with (5.19), the second term and the third term on the right-hand side of the above formula can be directly combined, so that the above optimality is simplified.

$$0 \in \begin{pmatrix} \nabla_{x_1}\theta_1(x^k) \\ \vdots \\ \nabla_{x_N}\theta_N(x^k) \end{pmatrix} + \begin{pmatrix} A_1^T\lambda^{k+1} \\ \vdots \\ A_N^T\lambda^{k+1} \end{pmatrix} + \gamma \begin{pmatrix} x_1^{k+1} - x_1^k \\ \vdots \\ x_N^{k+1} - x_N^k \end{pmatrix} + \begin{pmatrix} N_{\mathcal{X}_1}(x_1^{k+1}) \\ \vdots \\ N_{\mathcal{X}_N}(x_N^{k+1}) \end{pmatrix}. \tag{5.24}$$

Note that (5.19) can also be transformed into a form consistent with the above. Use square brackets $[\cdot]$ to group the terms of (5.19), which becomes

$$0 = [0] + \left[b - \sum_{\nu=1}^{N} A_\nu x_\nu^{k+1} \right] + \left[\frac{1}{\beta}(\lambda^{k+1} - \lambda^k) \right] + [0], \tag{5.25}$$

and considering $N_{\mathbb{R}^m}(\lambda^{k+1}) = 0$, replacing the last term on the right-hand side of the above equation, the optimality condition can be extended to

$$\begin{aligned} 0 \in & \begin{pmatrix} \nabla_{x_1}\theta_1(x^k) \\ \vdots \\ \nabla_{x_N}\theta_N(x^k) \\ 0 \end{pmatrix} + \begin{pmatrix} A_1^T\lambda^{k+1} \\ \vdots \\ A_N^T\lambda^{k+1} \\ b - \sum_{\nu=1}^{N} A_\nu x_\nu^{k+1} \end{pmatrix} \\ & + \begin{pmatrix} \gamma(x_1^{k+1} - x_1^k) \\ \vdots \\ \gamma(x_N^{k+1} - x_N^k) \\ \frac{1}{\beta}(\lambda^{k+1} - \lambda^k) \end{pmatrix} + \begin{pmatrix} N_{\mathcal{X}_1}(x_1^{k+1}) \\ \vdots \\ N_{\mathcal{X}_N}(x_N^{k+1}) \\ N_{\mathbb{R}^m}(\lambda^{k+1}) \end{pmatrix}. \end{aligned} \tag{5.26}$$

The matrix $Q \in \mathbb{R}^{(n+m)\times(n+m)}$ is introduced as follows.

$$Q = \begin{pmatrix} \gamma I_{n_1} & & & \\ & \ddots & & \\ & & \gamma I_{n_N} & \\ & & & \frac{1}{\beta} I_m \end{pmatrix} = \begin{pmatrix} \gamma I_n & \\ & \frac{1}{\beta} I_m \end{pmatrix}. \tag{5.27}$$

Hence, (5.26) can be reformulated by

$$0 \in P(w^k) + Gw^{k+1} + Q(w^{k+1} - w^k) + N_{\mathcal{W}}(w^{k+1}). \tag{5.28}$$

Based on this formula and different assumptions, two types of proofs are given below. That is to prove the Fejér monotonicity of the iterative point or prove that (5.28) is equivalent to finding the zero of the operator T using a forward-backward splitting algorithm.

Under Assumptions 5.1, 5.2, and 5.4, we will prove the Fejér monotonicity of the iterative sequence of Algorithm 3 , and then use Lemma 5.1 to prove the convergence of Algorithm 3. Now introduce the notation

$$\Delta(w^{k+1}, w^k) = Q(w^{k+1} - w^k) + P(w^k) - P(w^{k+1}). \tag{5.29}$$

By (5.12), $T(w) := P(w) + G(w) + N_{\mathcal{W}}(w)$. Optimality condition (5.28) can be rewritten as

$$0 \in T(w^{k+1}) + \Delta(w^{k+1}, w^k). \tag{5.30}$$

Lemma 5.11 *Let $w^* = (x^*, \lambda^*)$ be the variational KKT point of GNEP (5.1), the pseudo-gradient $\hat{P}(x)$ is the ρ-strongly monotone operator, then for any $w = (x, \lambda) \in \mathcal{X} \times \mathbb{R}^m$, we have*

$$\langle P(w) + Gw, w - w^* \rangle \geq \rho \|x - x^*\|^2. \tag{5.31}$$

Proof According to the characterization of the variational KKT point by Lemma 5.4, we have

$$\langle P(w^*) + Gw^*, w - w^* \rangle \geq 0 \quad \forall w = (x, \lambda) \in \mathcal{X} \times \mathbb{R}^m. \tag{5.32}$$

When the pseudo-gradient $\hat{P}(x)$ is strong monotone, combined with the properties (5.2.2) of G, we have

$$\langle P(w) + Gw - (P(w^*) + Gw^*), w - w^* \rangle \geq \rho \|x - x^*\|^2. \tag{5.33}$$

Adding (5.32) and (5.33) results in (5.31). □

According to the definition of normal cone (5.2.1), the optimality condition can be rewritten in an equality form, see the following lemma.

Lemma 5.12 *The iterative steps (5.18) and (5.19), i.e. (5.26), is equivalent to find $w^{k+1} = (x_1^{k+1}, \ldots, x_N^{k+1}, \lambda^{k+1})$, such that*

$$\begin{aligned} &\langle P(w^{k+1}) + Gw^{k+1} + Q(w^{k+1} - w^k), w - w^{k+1} \rangle \\ &+ \sum_{\nu=1}^{N} \langle \nabla_{x_\nu} \theta_\nu(x^k) - \nabla_{x_\nu} \theta_\nu(x^{k+1}), x_\nu - x_\nu^{k+1} \rangle \geq 0 \quad \forall w \in \mathcal{W}. \end{aligned} \tag{5.34}$$

Based on the above results, the convergence analysis of Algorithm 3 can be given.

Theorem 5.1 *Assume that GNEP (5.1) has variational KKT points. Assumptions 5.1, 5.2, and 5.4 hold. Parameters $\gamma > \varepsilon L > 0,\ \beta > 0$, then the iterative point w^{k+1} generated by Algorithm 3 converges to the variational KKT point $(x_1^*, \ldots, x_N^*, \lambda^*)$ of (5.1).*

Proof Since the parameters $\gamma > 0,\ \beta > 0$, it can be seen that the matrix Q is a symmetric positive definite matrix.

Using Lemma 5.11, we have

$$\langle P(w^{k+1}) + Gw^{k+1}, w^{k+1} - w^* \rangle \geq \rho \|x^{k+1} - x^*\|^2. \tag{5.35}$$

Taking $w = w^*$ into (5.34), using Cauchy-Schwarz inequality and Young's inequality (Lemma 5.2), for parameters $\varepsilon,\ \rho$satisfying $\varepsilon > \frac{1}{2\rho}$ (consequently $2\rho - \frac{1}{\varepsilon} > 0$), we have

$$\begin{aligned}
0 \leq & \langle P(w^{k+1}) + Gw^{k+1} + Q(w^{k+1} - w^k), w^* - w^{k+1}\rangle \\
& + \sum_{\nu=1}^{N} \langle \nabla_{x_\nu}\theta_\nu(x^k) - \nabla_{x_\nu}\theta_\nu(x^{k+1}), x_\nu^* - x_\nu^{k+1}\rangle \\
\leq & \langle w^{k+1} - w^k, w^* - w^{k+1}\rangle_Q - \rho\|x^{k+1} - x^*\|^2 \\
& + \sum_{\nu=1}^{N} \langle \nabla_{x_\nu}\theta_\nu(x^k) - \nabla_{x_\nu}\theta_\nu(x^{k+1}), x_\nu^* - x_\nu^{k+1}\rangle \\
\leq & \langle w^{k+1} - w^k, w^* - w^{k+1}\rangle_Q - \rho\|x^{k+1} - x^*\|^2 \\
& + \sum_{\nu=1}^{N} \|\nabla_{x_\nu}\theta_\nu(x^k) - \nabla_{x_\nu}\theta_\nu(x^{k+1})\| \cdot \|x_\nu^* - x_\nu^{k+1}\| \\
\leq & \langle w^{k+1} - w^k, w^* - w^{k+1}\rangle_Q - \rho\|x^{k+1} - x^*\|^2 \\
& + \sum_{\nu=1}^{N} \frac{\varepsilon}{2}\|\nabla_{x_\nu}\theta_\nu(x^k) - \nabla_{x_\nu}\theta_\nu(x^{k+1})\|^2 + \frac{1}{2\varepsilon}\|x_\nu^* - x_\nu^{k+1}\|^2 \\
\leq & \langle w^{k+1} - w^k, w^* - w^{k+1}\rangle_Q - \rho\|x^{k+1} - x^*\|^2 \\
& + \sum_{\nu=1}^{N} \frac{\varepsilon}{2}L_\nu^2\|x^k - x^{k+1}\|^2 + \frac{1}{2\varepsilon}\|x_\nu^* - x_\nu^{k+1}\|^2 \\
= & - \langle w^{k+1} - w^k, w^{k+1} - w^*\rangle_Q - (\rho - \frac{1}{2\varepsilon})\|x^{k+1} - x^*\|^2 \\
& + \frac{\varepsilon}{2}L\|x^k - x^{k+1}\|^2.
\end{aligned}$$

From the identity $2\langle w, v\rangle = \|w\|^2 + \|v\|^2 - \|w - v\|^2$, we have

$$\begin{aligned}
0 \leq & \frac{1}{2}\|w^k - w^*\|_Q - \frac{1}{2}\|w^{k+1} - w^k\|_Q - \frac{1}{2}\|w^{k+1} - w^*\|_Q \\
& - \left(\rho - \frac{1}{2\varepsilon}\right)\|x^{k+1} - x^*\|^2 + \frac{\varepsilon}{2}L\|x^k - x^{k+1}\|^2.
\end{aligned}$$

After arranging the above equation, we can get

$$\begin{aligned}\|w^{k+1}-w^*\|_Q^2 \leq & \|w^k-w^*\|_Q^2-\|w^{k+1}-w^k\|_Q^2\\ & -\left(2\rho-\frac{1}{\varepsilon}\right)\|x^{k+1}-x^*\|^2+\varepsilon L\|x^k-x^{k+1}\|^2\\ = & \|w^k-w^*\|_Q^2-\gamma\|x^{k+1}-x^k\|^2-\frac{1}{\beta}\|\lambda^{k+1}-\lambda^k\|^2\\ & -\left(2\rho-\frac{1}{\varepsilon}\right)\|x^{k+1}-x^*\|^2+\varepsilon L\|x^k-x^{k+1}\|^2\\ = & \|w^k-w^*\|_Q^2-(\gamma-\varepsilon L)\|x^{k+1}-x^k\|^2-\frac{1}{\beta}\|\lambda^{k+1}-\lambda^k\|^2\\ & -\left(2\rho-\frac{1}{\varepsilon}\right)\|x^{k+1}-x^*\|^2.\end{aligned} \tag{5.36}$$

Note that in the above steps, when using Young's inequality. On the other hand, after selecting the parameter ε, the regular term parameter γ can be sufficiently large, so that $\gamma > \varepsilon L$, then we have $\gamma - \varepsilon L > 0$. In this way, the Fejér monotonicity of the iterative point sequence with respect to the variational KKT point set is obtained. Thus, the sequence $\{w^{k+1}\}$ is bounded, so it has cluster points. Below, we will prove that the cluster point of the sequence is a variational KKT point.

Let $\{w^{k+1}\}_{\mathcal{I}}$ be the subsequence convergent to the cluster point $\bar{w}$ with $\mathcal{I} \subset \mathbb{N}$, denote as $w^{k+1} \to_{\mathcal{I}} \bar{w}$. Since $\mathcal{X}_1, \ldots, \mathcal{X}_N$ are closed convex sets, so $\mathcal{W} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_N \times \mathbb{R}^m$ is also closed convex set. So $\bar{w} \in \mathcal{W}$.

Let $\hat{\rho} = 2\rho - \frac{1}{\varepsilon} > 0,\ \hat{\gamma} = \gamma - \varepsilon L > 0$, then Fejér monotonicity can be rewritten as

$$\hat{\gamma}\|x^{k+1}-x^k\|^2+\frac{1}{\beta}\|\lambda^{k+1}-\lambda^k\|^2+\hat{\rho}\|x^{k+1}-x^*\|^2 \leq \|w^k-w^*\|_Q^2-\|w^{k+1}-w^*\|_Q^2. \tag{5.37}$$

In (5.37), summation of $k = 0, 1, \ldots, t$ yields that

$$\begin{aligned}&\sum_{k=0}^{t}(\hat{\gamma}\|x^{k+1}-x^k\|^2+\frac{1}{\beta}\|\lambda^{k+1}-\lambda^k\|^2+\hat{\rho}\|x^{k+1}-x^*\|^2)\\ &\leq \sum_{k=0}^{t}(\|w^k-w^*\|_Q^2-\|w^{k+1}-w^*\|_Q^2)\\ &= \|w^0-w^*\|_Q^2-\|w^{t+1}-w^*\|_Q^2\\ &\leq \|w^0-w^*\|_Q^2.\end{aligned} \tag{5.38}$$

Let $t \to \infty$ in the above formula, we can get

$$\|x^{k+1}-x^k\| \to 0, \quad \|\lambda^{k+1}-\lambda^k\| \to 0, \quad \|x^{k+1}-x^*\| \to 0. \tag{5.39}$$

So we have

$$\|w^{k+1} - w^k\| \to 0, \quad x^{k+1} \to x^*. \tag{5.40}$$

Combining $w^{k+1} \to_{\mathcal{I}} \bar{w}$, we get $w^k \to_{\mathcal{I}} \bar{w}$. Then by Assumption 5.4, i.e., the Lipschitz continuity of $\nabla_{x_\nu}\theta_\nu$, we have

$$\|\nabla_{x_\nu}\theta_\nu(x^k) - \nabla_{x_\nu}\theta_\nu(x^{k+1})\| \leq L_\nu\|x^k - x^{k+1}\| \to 0. \tag{5.41}$$

That is

$$\|P(x^k) - P(x^{k+1})\| \to 0, \quad k \to \infty. \tag{5.42}$$

So

$$\|\Delta(w^{k+1}, w^k)\| \leq \|Q\|\|w^k - w^{k+1}\| + \|P(x^k) - P(x^{k+1})\| \to 0, \quad k \to \infty. \tag{5.43}$$

So there is $\Delta(w^{k+1}, w^k) \to_{\mathcal{I}} 0$. (5.30) can be rewritten as $-\Delta(w^{k+1}, w^k) \in T(w^{k+1})$. Using the Lemma 5.8, and $\Delta(w^{k+1}, w^k) \to_{\mathcal{I}} 0,\ w^{k+1} \to_{\mathcal{I}} \bar{w}$, we get $0 \in T(\bar{w})$. So it is proved that $\bar{w}$ is a variational KKT point. By Lemma 5.1, it is known that the iterative sequence $\{w^{k+1}\}$ converges to a certain variational KKT point. □

So far, we have proved the convergence of RLALM under the strong monotonicity and Lipschitz continuity assumptions of pseudo-gradients. Next, we consider "weakening" the assumptions. Note that "weakening" here is not strictly speaking, since the strong monotonicity assumption of pseudo-gradients is only satisfied at the solution points. Below we will illustrate that Algorithm 3 is equivalent to the forward-backward splitting algorithm based on (5.28) under Assumptions 5.1 and 5.3, the convergence analysis will be implied.

Let

$$\overline{Q} := \beta Q = \begin{pmatrix} \beta\gamma I_n & \\ & I_m \end{pmatrix}. \tag{5.44}$$

The parameter $\beta > 0$, so $\overline{Q}$ is also a positive definite matrix. Then the inner product $\langle\cdot,\cdot\rangle_{\bar{Q}}$ and its induced norm $\|\cdot\|_{\overline{Q}}$ can be defined. Convergence is proved under this norm later in this subsection. In finite-dimensional Euclidean space, by the equivalence of any two norms, the corresponding convergence of different norms is also equivalent.

Rewrite (5.28) with $\overline{Q}$:

$$\begin{aligned} 0 &\in P(w^k) + Gw^{k+1} + Q(w^{k+1} - w^k) + N_{\mathcal{W}}(w^{k+1}) \\ &= \frac{1}{\beta}\overline{Q}w^k - P(w^k) \in Gw^{k+1} + \frac{1}{\beta}\overline{Q}w^{k+1} + N_{\mathcal{W}}(w^{k+1}) \\ &= (I - \beta\overline{Q}^{-1}P)(w^k) \in (I + \beta\overline{Q}^{-1}(G + N_{\mathcal{W}}))(w^{k+1}). \end{aligned}$$

Let $T = P + G + N_{\mathcal{W}} = T_1 + T_2$, where $T_1 = P,\ T_2 = G + N_{\mathcal{W}}$. By Assumption 5.1, Lemmas 5.5, and 5.7, we know that $T_1,\ T_2$ are both maximally monotonic. And denote

$$A := \overline{Q}^{-1}T_2 = \overline{Q}^{-1}(G + N_{\mathcal{W}}), \qquad B := \overline{Q}^{-1}T_1 = \overline{Q}^{-1}P.$$

So

$$A + B = \overline{Q}^{-1}(T_1 + T_2) = \overline{Q}^{-1}T. \tag{5.45}$$

Since $\overline{Q}^{-1}$ is positive definite, it is an invertible matrix. So the zero of the operator $\overline{Q}^{-1}T$ is the same as the zero of T. That is, the zero point of $A + B$ is equivalent to the zero point of T. Below we will prove that Algorithm 3 is equivalent to the forward-backward splitting algorithm that finds the zero point of $\overline{Q}^{-1}T = A + B$.

Although $T_1,\ T_2$ are maximal monotone operators, but $A = \overline{Q}^{-1}T_2,\ B = \overline{Q}^{-1}T_1$ is not necessarily a maximal monotone operator. After defining the induced norm $\|\cdot\|_{\overline{Q}}$, it can be proved that $A,\ B$ are maximally monotone under the induced norm.

Lemma 5.13 *Let $Q \in \mathbb{R}^n$ be a positive definite matrix, the operator $T : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is a maximal monotone operator about the norm $\|\cdot\|$, then the operator $Q \cdot T$ is a maximal monotone operator about the norm $\|\cdot\|_{Q^{-1}}$*

Proof According to the definition of the maximal monotone operator in view of (5.8), for the operator $Q \cdot T$, and the point $(x, u) \in \mathbb{R}^n \times \mathbb{R}^n$ proves this equivalence.

On the one hand, if $u \in QAx$, that is, $Q^{-1}u \in Ax$, for $\forall v \in QAy$, i.e. $Q^{-1}v \in Ay$, using the maximal monotonicity of A, we have

$$\langle x - y, Q^{-1}u - Q^{-1}v\rangle \geq 0,$$

That is

$$\langle x - y, u - v\rangle_{Q^{-1}} \geq 0.$$

Conversely, for all $v \in QAy$, we have

$$\langle x - y, u - v\rangle_{Q^{-1}} \geq 0,$$

Then for all $v' \in Ay$, there is $Qv' \in QAy$, and

$$\langle x - y, Q^{-1}u - v'\rangle = \langle x - y, u - Qv'\rangle_{Q^{-1}} \geq 0.$$

From the maximal monotonicity of A, we can get $Q^{-1}u \in Ax$ and hence $u \in QAx$. The proof is complete. □

According to Lemma 5.13, the following conclusions can be drawn.

Proposition 5.3 *Let Assumption 5.1 hold, parameters $\beta > 0,\ \gamma > 0$, then under the norm $\|\cdot\|_{\overline{Q}}$, the operators $A = \overline{Q}^{-1}T_2,\ B = \overline{Q}^{-1}T_1$ are both maximally monotonic.*

Based on the above notation, (5.28) can be further rewritten as

$$(I - \beta B)(w^k) \in (I + \beta A)(w^{k+1}),$$

where the operator A is maximally monotonic, then its resolution $J_{\beta A} = (I + \beta A)^{-1}$ is a single-valued mapping. In addition, the operator $T_1 = P$ is obviously a single-valued mapping, so $B = \overline{Q}^{-1} T_1$ is also a single-valued mapping. Then the above formula is equivalent to

$$w^{k+1} = (I + \beta A)^{-1}(I - \beta B)(w^k).$$

This gives the iterative form of the forward-backward splitting algorithm for finding the zeros of $A + B$.

To prove the convergence, we also need to consider the cocoercivity of the operator B. Under Assumption 5.3, i.e.,$\hat{P}$ is α-cocoercive, obviously P is also α-cocoercive. The following results give the effect of the composite matrix $\overline{Q}^{-1}$ on the cocoercivity.

Lemma 5.14 ([26]) *Let the operator $P : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ be α-cocoercive about the inner product $\langle \cdot, \cdot \rangle$, Q is a symmetric positive definite matrix of order n, then the operator $Q^{-1}P$ is $\alpha / \| Q^{-1} \|$-cocoercive about the inner product $\langle \cdot, \cdot \rangle_Q$.*

The inverse of the matrix $\overline{Q}$ can be directly obtained:

$$\overline{Q}^{-1} = \begin{pmatrix} \frac{1}{\beta\gamma} I_n & \\ & I_m \end{pmatrix}.$$

The eigenvalues of $\overline{Q}^{-1}$ are $1/\beta\gamma$ and 1, so

$$\|\overline{Q}^{-1}\| = max\left\{\frac{1}{\beta\gamma}, 1\right\},$$

$$\frac{1}{\|\overline{Q}^{-1}\|} = min\left\{\beta\gamma, 1\right\}.$$

We want the operator $B = \overline{Q}^{-1} P$ to be α-cocoercive. From the definition of cocoercivity, it can be seen that the operator of δ-cocoercive ($\delta > \alpha$) is also α-cocoercive. So just let $1/\|\overline{Q}^{-1}\| \geq 1$, i.e. $\beta\gamma \geq 1$, then we get the α-cocoercivity of the operator B.

Therefore, applying the convergence Theorem 5.9 of the forward-backward splitting algorithm, the following result can be obtained.

Theorem 5.2 *Assume that GNEP (5.1) has variational KKT points, and that Assumptions 5.1 and 5.3 hold. Parameters $\gamma > 0$, $\beta > 0$, and satisfy $\beta\gamma \geq 1$, $\beta \in (0, 2\alpha)$. Then the iteration point w^{k+1} generated by Algorithm 3 converges to the variational KKT point $(x_1^*, \ldots, x_N^*, \lambda^*)$ of the problem under inner product $\langle \cdot, \cdot \rangle_{\bar{Q}}$ and its induced norm $\| \cdot \|_{\overline{Q}}$.*

5.3.2 Regularized Augmented Lagrangian Algorithm

This subsection will propose the second augmented Lagrangian-type algorithm. Considering that the augmented Lagrangian function is not linearized in the subproblem (5.18), then in addition to Assumption 5.1, the convergence can be proved without any additional conditions. That is, using the following augmented Lagrangian function

$$\begin{aligned}\bar{L}_\nu(x, x^k, \lambda^k) =& \theta_\nu(x) + (\lambda^k)^T \big(\sum_{\nu=1}^{N} A_\nu x_\nu - b\big)\\ &+ \frac{\beta}{2}\|\sum_{\nu=1}^{N} A_\nu x_\nu - b\|^2 + \frac{\gamma}{2}\|x_\nu - x_\nu^k\|^2.\end{aligned}$$

The algorithm is modified to the following Algorithm 4.

Algorithm 4 Regularized augmented Lagrangian algorithm

```
1: Choose a starting point $(x^0, \lambda^0) \in \mathcal{X} \times \mathbb{R}^m$, parameters $\beta,\ \gamma > 0$.
2: If a suitable termination criterion is satisfied: STOP.
3: Calculate the solution $x^{k+1}$ of the following NEP problem
       $$\min_{x_\nu \in \mathcal{X}_\nu} \bar{L}_\nu(x, x^k, \lambda^k), \quad \nu = 1, 2, 3, \ldots, N. \tag{5.46}$$
4: Compute
       $$\lambda^{k+1} = \lambda^k + \beta\left(\sum_{\nu=1}^{N} A_\nu x_\nu^{k+1} - b\right).$$
5: Set $k := k + 1$, and go to Step 1.
```

The optimality condition of (5.46) can be rewritten as

$$0 \in P(w^{k+1}) + Gw^{k+1} + Q(w^{k+1} - w^k) + N_{\mathcal{W}}(w^{k+1}).$$

Redefine the residual notation $\Delta(w^{k+1}, w^k) = Q(w^{k+1} - w^k)$. Then the optimality condition is still rewritten as $0 \in T(w^{k+1}) + \Delta(w^{k+1}, w^k)$. Its form is simpler, and convergence can be easily proved by a similar method.

Similar to Lemma 5.11, we have the following result.

Lemma 5.15 *Let $w^* = (x^*, \lambda^*)$ be the variational KKT point of GNEP (5.1), if Assumption 5.1 is established, then for any $w = (x, \lambda) \in \mathcal{X} \times \mathbb{R}^m$, we have*

$$\langle P(w) + Gw, w - w^*\rangle \geq 0. \tag{5.47}$$

Proof According to the characterization of the variational KKT point by Lemma 5.4, we have

$$\langle P(w^*) + Gw^*, w - w^* \rangle \geq 0. \tag{5.48}$$

When Assumption 5.1 is satisfied, by the monotonicity of $\hat{P}(x)$ and the property (5.2.2) of G, we get

$$\langle P(w) + Gw - (P(w^*) + Gw^*), w - w^* \rangle \geq 0. \tag{5.49}$$

Adding (5.48) and (5.49) together yields (5.47). □

Due to the changes made in this lemma and the lack of linearization, compared with the proof of the convergence theorem under the strong monotonic assumption, there will be no more two differences about P and strong monotonic parameter terms in the process. Then we have

$$\begin{aligned} &0 \leq \langle Q(w^{k+1} - w^k), w^* - w^{k+1} \rangle \\ \Leftrightarrow &\|w^{k+1} - w^*\|_Q^2 \leq \|w^k - w^*\|_Q^2 - \|w^{k+1} - w^k\|_Q^2. \end{aligned}$$

That is, the Fejér monotonicity of the variational KKT point set with a simpler form. The following proofs are basically the same. So the convergence theorem of Algorithm 4 can be obtained.

Theorem 5.3 *Assume that GNEP (5.1) has variational KKT points, and that Assumption 5.1 holds, parameters* $\gamma > 0$, $\beta > 0$. *Then the iterative point* w^{k+1} *generated by Algorithm 4 converges to the variational KKT point* $(x_1^*, \ldots, x_N^*, \lambda^*)$ *of (5.1).*

For ALM without linearization in Algorithm 4, the corresponding convergence is easier to prove; however, the solution of algorithmic subproblems tends to be more complicated. The solution method of the subproblems is still similar to the inner iterative solution method, but each inner iteration will recall the pseudo-gradient function once, which will increase a lot of computation. But for small-scale problems, its numerical performance is often better.

5.3.3 Corrected Regularized Linearized Augmented Lagrangian Algorithm

This section will focus on the corrected regularized linearized augmented Lagrangian algorithm, which requires weaker assumptions to guarantee the corresponding convergence results. Referring to Tseng's algorithm (Lemma 5.10), which is a forward-backward splitting algorithm, where a correction step is added at each iteration. In this case, the cocoercivity assumption about operator B can be weakened into the

Lipschitz continuity assumption. Inspired by this, the corresponding correction step is added to Algorithms 3, and 5 is obtained.

Algorithm 5 Corrected regularized linearized augmented Lagrangian algorithm

```
1: Choose a starting point $(x^0, \lambda^0) \in \mathcal{X} \times \mathbb{R}^m$, parameters $\beta,\ \gamma > 0$.
2: If a suitable termination criterion is satisfied: STOP.
3: Calculate the solution $y^k$ of the following NEP problem
       $\min_{x_\nu \in \mathcal{X}_\nu} L_\nu(x, x^k, \lambda^k), \qquad \nu = 1, 2, 3, \ldots, N.$   (5.50)
4: Compute
       $\lambda^{k+1} = \lambda^k + \beta \left( \sum_{\nu=1}^{N} A_\nu y_\nu^k - b \right).$   (5.51)
5: Compute
       $x^{k+1} = y^k + \frac{1}{\gamma} \left( \hat{P}(x^k) - \hat{P}(y^k) \right).$   (5.52)
6: Set $k := k + 1$, and go to Step 1.
```

A correction step (5.52) has been added to Algorithm 5, which can be calculated explicitly. This section will give the equivalence of Algorithm 5 and Tseng's splitting algorithm under Assumptions 5.1 and 5.4 to prove the convergence. Denote

$$v^{k+1} := \begin{pmatrix} y^k \\ \lambda^{k+1} \end{pmatrix}.$$

The correction step (5.52) should be

$$w^{k+1} = v^{k+1} + \beta \left(B(w^k) - B(v^{k+1}) \right) = v^{k+1} + \beta \left(\bar{Q}^{-1} P(w^k) - \bar{Q}^{-1} P(v^{k+1}) \right).$$

Then the update of the decision variable x should be

$$x^{k+1} = y^k + \beta \cdot \frac{1}{\beta\gamma} (\overline{P}(x^k) - \overline{P}(y^k) = y^k + \frac{1}{\gamma} (\overline{P}(x^k) - \overline{P}(y^k)).$$

This is the correction step in Algorithm 5.

Next, we discuss the Lipschitz continuity of the operator $B = \overline{P}^{-1} T_1 = \overline{P}^{-1} P$. Based on Assumption 5.4, for all $\nu = 1, \ldots, N$, gradient $\nabla_{x_\nu} \theta_\nu$ is L_ν-Lipschitz continuous. Then for the operator P, for $\forall w = (x, \lambda),\ v = (y, \mu) \in \mathbb{R}^{n+m}$, we have

$$\begin{aligned}\|P(w)-P(v)\| &= \left\| \begin{pmatrix} \nabla_{x_1}\theta_1(x)-\nabla_{x_1}\theta_1(y) \\ \vdots \\ \nabla_{x_N}\theta_N(x)-\nabla_{x_N}\theta_N(y) \\ 0 \end{pmatrix} \right\| \\ &= \sum_{\nu=1}^{N} \|\nabla_{x_\nu}\theta_\nu(x)-\nabla_{x_\nu}\theta_\nu(y)\| \\ &\leq \left(\sum_{\nu=1}^{N} L_\nu\right)\|x-y\| \leq \left(\sum_{\nu=1}^{N} L_\nu\right)\|w-v\|.\end{aligned}$$

Denote $L=\sum_{\nu=1}^{N} L_\nu$, we know that P is L-Lipschitz continuous. Obviously, we know that the operator $B=\overline{Q}^{-1}P$ is also $\|\overline{Q}^{-1}\|L$-Lipschitz continuous. So we just require $\|\overline{Q}^{-1}\|\leq 1$, i.e. $\beta\gamma\geq 1$ to get the L-Lipschitz continuity of the operator B.

Therefore, using Lemma 5.10 of Tseng's splitting algorithm, the following results can be obtained.

Theorem 5.4 *Suppose GNEP (5.1) has variational KKT points, and Assumptions 5.1 and 5.4 hold. Denote $L=\sum_{\nu=1}^{N} L_\nu$, parameters $\gamma>0$, $\beta>0$, and satisfy $\beta\gamma\geq 1$, $\beta\in(0,L)$. Then the iteration point w^{k+1} generated by Algorithm 5 converges to the variational KKT point $(x_1^*,\dots,x_N^*,\lambda^*)$ of the problem (5.1).*

5.3.4 Treatment of Inequality Constraints

So far, we have considered algorithms for convex GNEP with common linear equality constraints. Now generalize the previous results to linear inequality constraints. By introducing new variables, the inequality constraints are transformed into equality constraints, and the constraints of the new variables are incorporated into the abstract constraints. This allows the use of algorithms for equality constraints.

Börgens and Kanzow [26] give two transformation methods to convert GNEP (5.2) to linear equality-constrained GNEP. One is to introduce a new decision variable s_ν for each player, that is, for $\nu=1,\dots,N$, solve

$$\min_{\substack{x_\nu\in\mathcal{X}_\nu \\ s_\nu\in\mathbb{R}^m_-}} \theta_\nu(x_\nu,x_{-\nu}) \qquad \text{s.t.} \qquad \sum_{\nu=1}^{N} A_\nu x_\nu - b - \sum_{\nu=1}^{N} s_\nu = 0. \tag{5.53}$$

Or we just introduce a new decision variable s for the last player, i.e., for $\nu=1,\dots,N-1$, solve

$$\min_{x_\nu\in\mathcal{X}_\nu} \theta_\nu(x_\nu,x_{-\nu}) \qquad \text{s.t.} \qquad \sum_{\nu=1}^{N} A_\nu x_\nu - b - s = 0. \tag{5.54a}$$

And for the N-th player, solve

$$\min_{\substack{x_N \in X_N \\ s \in \mathbb{R}^m_-}} \theta_N(x_N, x_{-N}) \quad \text{s.t.} \quad \sum_{\nu=1}^{N} A_\nu x_\nu - b - s = 0. \tag{5.54b}$$

Regarding the relationship between their solutions and the solution of the original problem, we have the following results.

Proposition 5.4 ([26]) *The following statements are equivalent:*

1. $x^* = (x_1^*, \ldots, x_N^*)$ *is the solution of GNEP (5.2).*
2. *For some* $s_\nu^* \in \mathbb{R}^m_-,\ \nu = 1, \ldots, N,\ (x_1^*, s_1^*, \ldots, x_N^*, s_N^*)$ *is the solution of GNEP (5.53).*
3. *For some* $s^* \in \mathbb{R}^m_-,\ (x_1^*, \ldots, x_N^*, s^*)$ *is the solution of GNEP (5.54).*

Regarding the relationship between their variational KKT points and the variational KKT points of the original problem, we have the following results.

Proposition 5.5 ([26]) *The following statements are equivalent:*

1. $(x^*, \lambda^*) = (x_1^*, \ldots, x_N^*)$ *is the variational KKT point of GNEP (5.2).*
2. *For some* $s_\nu^* \in \mathbb{R}^m_-,\ \nu = 1, \ldots, N,\ ((x_1^*, s_1^*, \ldots, x_N^*, s_N^*), \lambda^*)$ *is the variational KKT point of GNEP (5.53).*
3. *For some* $s^* \in \mathbb{R}^m_-,\ ((x_1^*, \ldots, x_N^{ast}, s^*), \lambda^*)$ *is the variational KKT point of GNEP (5.54).*

Note that after employing new variables, the gradient of the objective function with respect to the new decision variables no longer maintains strong monotonicity, but its cocoercivity can still be maintained. In addition, regarding the solution of the subproblems at each iteration, the idea of this paper is to solve the old and new decision variables separately.

5.4 Numerical Experiment

In this section, some numerical examples will be given to verify the effectiveness of the proposed algorithms.

5.4.1 *Numerical Examples*

Example 5.1 The first example was proposed by Facchinei and Fischer [6] and tested by Facchinei and Kanzow [9] and Han et al. [30]. There are two players in this question, each player ν controls a decision variable $x_\nu \in \mathbb{R}$. The objective function and constraints of each player are as follows

$$\begin{aligned} \min_{x_1} \quad & (x_1 - 1)^2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 1, \end{aligned}$$

and

$$\begin{aligned} \min_{x_2} \quad & (x_2 - \frac{1}{2})^2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 1. \end{aligned}$$

The generalized Nash equilibrium point set of this example is $\{(s, 1 - s) \mid s \in [\frac{1}{2}, 1]\}$, and the unique variational equilibrium point is $(\frac{3}{4}, \frac{1}{4})$.

Example 5.2 The second example, taken from [6, 9], also consists of two players. The first player controls a two-dimensional decision variable $x_1 = (x_{11}, x_{12})^T =: (y_1, y_2)^T \in \mathbb{R}^2$, another participant controls a one-dimensional decision variable $x_2 =: y_3 \in \mathbb{R}$. The objective functions of the two participants are

$$\theta_1(x) = y_1^2 + y_1 y_2 + y_2^2 + (y_1 + y_2)y_3 - 25y_1 - 38y_2,$$

and

$$\theta_2(x) = y_3^2 + (y_1 + y_2)y_3 - 25y_3.$$

They all have non-negative constraints $y_1, y_2, y_3 \geq 0$, and common linear constraints

$$\begin{aligned} y_1 + 2y_2 - y_3 &\leq 14, \\ 3y_1 + 2y_2 + y_3 &\leq 30. \end{aligned}$$

The generalized Nash equilibrium point set of this example is $\{(s, 11 - s, 8 - s) \mid s \in [0, 2]\}$, and the only variational equilibrium point is $(0, 11, 8)$.

Example 5.3 The third example is a modified version [26] of the duopoly model introduced by Krawczyk and Uryasev [31]. Two players ν control a decision variable $x_\nu \in \mathbb{R}$, representing their production of a given product. The objective function representing their profit is given by

$$\theta_\nu(x) = x_\nu(\rho(x_1 + x_2) + \lambda - d), \qquad \nu = 1, 2.$$

The productive capacity of each participant is limited by individual constraints $x_\nu \in [0, 10]$. Furthermore, both participants have a common resource limit constraint $x_1 + x_2 \leq r$. Here we choose parameters $d = 20,\ \lambda = 4,\ \rho = 1,\ r = 9$.

Example 5.4 The fourth example is from Harker [32]. There are two players in this problem, player ν controls a decision variable $x_\nu \in \mathbb{R}$. The objective functions and constraints of each player are as follows

Table 5.1 Parameters of a river basin pollution model

Player ν	$c_{1\nu}$	$c_{2\nu}$	e_ν	$u_{\nu 1}$	$u_{\nu 2}$
1	0.10	0.01	0.50	6.5	4.583
2	0.12	0.05	0.25	5.0	6.250
3	0.15	0.01	0.75	5.5	3.750

$$\begin{aligned} \min_{x_1} \quad & (x_1)^2 + \frac{8}{3}x_1x_2 - 34x_1 \\ \text{s.t.} \quad & 0 \le x_1 \le 10, \\ & x_1 + x_2 \le 15, \end{aligned}$$

and

$$\begin{aligned} \min_{x_2} \quad & (x_2)^2 + \frac{5}{4}x_1x_2 - 24.25x_1 \\ \text{s.t.} \quad & 0 \le x_2 \le 10, \\ & x_1 + x_2 \le 15. \end{aligned}$$

The generalized Nash equilibrium point set for this example is $\{(5, 9)\} \cup \{(s, 15 - s) \mid s \in [9, 10]\}$. The variational equilibrium is $(5, 9)$.

Example 5.5 This example is a river basin pollution model introduced by Krawczyk and Uryasev [31] and tested by Han [30] et al. There are three players in this problem, player ν controls a decision variable $x_\nu \in \mathbb{R}$. The objective function is given by

$$\theta_\nu(x) = x_\nu(c_{1\nu} + c_{2\nu}x_\nu - d_1 + d_2(x_1 + x_2 + x_3)), \qquad \nu = 1, 2, 3.$$

And there are common linear constraints

$$\begin{pmatrix} u_{11}e_1 & u_{21}e_2 & u_{31}e_3 \\ u_{12}e_1 & u_{22}e_2 & u_{32}e_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \le \begin{pmatrix} K_1 \\ K_2 \end{pmatrix}.$$

The parameters $c_{1\nu},\ c_{2\nu},\ e_\nu,\ u_{\nu 1},\ u_{\nu 2}$ are shown in Table 5.1, and setting the parameters $K_1 = K_2 = 100, d_1 = 3, d_2 = 0.01$. An approximate solution for this example is (21.1448, 16.0279, 2.7260).

Example 5.6 This example is the oligopoly model described in the literature [33], which was tested by Facchinei and Kanzow [9]. The problem has N players, each player has a decision variable $x_\nu \in \mathbb{R}$, and the objective function is as follows

$$\theta_\nu(x) = f_\nu(x_\nu) - 5000^{1/\eta}x_\nu(x_1 + x_2 + \ldots + x_N)^{-1/\eta}, \quad \nu = 1, 2, \ldots, N,$$

where

Table 5.2 Oligopoly model parameters

Player ν	1	2	3	4	5
c_ν	10	8	6	4	2
K_ν	5	5	5	5	5
δ_ν	1.2	1.1	1.0	0.9	0.8

$$f_\nu(x_\nu) = c_\nu x_\nu + \frac{\delta_\nu}{1+\delta_\nu} K_\nu^{-1/\delta_\nu} (x_\nu)^{\frac{1+\delta_\nu}{\delta_\nu}}, \quad \nu = 1, 2, \ldots, N.$$

All of them have non-negative constraints $x_\nu \geq 0,\ \nu = 1, 2, \ldots, N$, and common linear constraints

$$x_1 + x_2 + \ldots + x_N \leq P.$$

Take the parameters $N = 5,\ \ \eta = 1.1,\ \ P = 75$. The remaining parameters $c_\nu,\ K_\nu,\ \delta_\nu$ are shown in Table 5.2. An approximate solution for this example is

$$(10.4038, 13.0359, 15.4074, 17.3815, 18.7713).$$

Example 5.7 A final example is the electricity market model. It is presented in [11] and further studied in [34]. The problem has 2 players, each with a power plant on 2 of the 3 regions, which can be defined on the nodes of the graph (this is a simplified model of the literature [11]). First denote

$$\begin{aligned}
S_1 &= 40 - \frac{40}{500}(x_1 + x_4 + x_7 + x_{10}),\\
S_2 &= 35 - \frac{35}{400}(x_2 + x_5 + x_8 + x_{11}),\\
S_3 &= 32 - \frac{32}{600}(x_3 + x_6 + x_9 + x_{12}).
\end{aligned}$$

The objective functions of the two players are as follows

$$\theta_1(x) = (15 - S_1)(x_1 + x_4) + (15 - S_2)(x_2 + x_5) + (15 - S_3)(x_3 + x_6),$$

$$\theta_2(x) = (15 - S_1)(x_7 + x_{10}) + (15 - S_2)(x_8 + x_{11}) + (15 - S_3)(x_9 + x_{12}).$$

They all have non-negative constraints $x_\nu \geq 0,\ \nu = 1, 2, \ldots, 12$, and common linear constraints

$$\begin{aligned}
x_1 + x_2 + x_3 &\leq 100,\\
x_4 + x_5 + x_6 &\leq 50,\\
x_7 + x_8 + x_9 &\leq 100,\\
x_{10} + x_{11} + x_{12} &\leq 50,
\end{aligned}$$

$$S_j - S_i \leq 0, \quad \forall i, j = 1, 2, 3, \quad i \neq j.$$

After testing, an approximate solution for this example is

$$(43.5364, 28.1381, 28.3255, 26.8698, 11.4714, 11.6588,\\ 43.5364, 28.1381, 28.3255, 26.8698, 11.4714, 11.6588).$$

5.4.2 Numerical Results

We test all examples in Example 5.6 using the regularized linearized augmented Lagrangian algorithm in Algorithm 3 (RLALM), the regularized augmented Lagrangian algorithm in Algorithm 4 (RALM), the corrected regularized linear augmented Lagrangian algorithm in 5 (CRLALM), the Jacobi-type ADMM method in Algorithm 1 (JADMM), and the Gauss-Seidel-type ADMM method in Algorithm 2 (GADMM).

The subproblems for each algorithm are solved by the self-adjusting ratio projected gradient method. The termination condition of the inner loop is that the difference of the iterates $\|x^k - x^{k-1}\|$ is less than 10^{-8}. The termination condition of the outer iteration is taken as

$$\|P(x^k) + A^T\lambda^k\| \leq 10^{-6},$$
$$\|\min(\lambda^k, b - Ax^k)\| \leq 10^{-6}.$$

Regarding the adjustment of parameters, JADMM adopts the same parameter settings in [26] on the one hand, that is, for the α-cocoercive operator P, taking $\beta = \{\alpha, 0.2\},\ \gamma = 1/\beta^2 + \|M\|$. On the other hand, this paper will manually adjust its parameters, denoted JADMMa, to further obtain better results. For GADMM, we take the adaptive strategy of the parameter γ in [27]. The residual of the iterates is

$$r^k := \sum_{\nu=1}^{N} \|\nabla_{x_\nu}\theta_\nu(x^k) + A_\nu^T\lambda^k + v_\nu^k\|^2 + \|\sum_{\nu=1}^{N} A_\nu x_\nu^k - b\|^2,$$

where $v_\nu^k \in N_{\mathcal{X}_\nu}(x^k)$. This residual characterizes how well the iterate satisfies the variational KKT condition. Börgens and Kanzow [27] proposed that the condition that γ is sufficiently large is not a necessary condition for convergence, so they proposed an adaptive method to update γ when the following three conditions are satisfied at the same time:

1. γ is less than the given upper bound.
2. Residuals do not fall sufficiently: $r^{k+1} > \varepsilon' r^k$, where parameter $\varepsilon' \in (0, 1)$.
3. γ has not been updated in the previous κ steps, where κ is a given constant.

Table 5.3 Numerical results of RLALM with different ways of introducing new variables

Example	RLALM		RLALM2	
	Iter.	Time	Iter.	Time
3.1	14	5.4	14	4.9
3.2	22	34.1	22	26.8
3.3	8	3.8	8	2.7
3.4	92	80.8	92	62.4
3.5	23	77.9	31	90.7
3.6	23	44.5	23	48.9
3.7	46	124.6	46	105.0

Table 5.4 Numerical results of several ALM-type algorithms

Example	RLALM		RALM		CRLALM	
	Iter.	Time	Iter.	Time	Iter.	Time
4.1	14	5.4	6	4.1	47	15.2
4.2	22	34.1	9	36.0	93	98.8
4.3	8	3.8	7	3.0	29	11.7
4.4	92	80.8	5	11.5	326	88.8
4.5	23	77.9	10	66.8	97	92.4
4.6	23	47.5	7	29.5	346	149.8
4.7	46	124.6	13	131.1	114	194.8

γ can be updated by accumulation or multiplication, in other words, $\gamma^{k+1} := \gamma^k + \tau$ or $\gamma^{k+1} := \tau \cdot \gamma^k$. Here, we use the accumulation principle to update γ.

First, we test and compare the two ways of converting inequality constraints into equality constraints.

In Table 5.3, the column RLALM represents the transformation of inequality constraints into equality constraints by introducing N block new variables, then using the regularized linearized augmented Lagrangian algorithm for equality constraints proposed in this paper to solve the problem. The column RLALM2 indicates that only one new variable is introduced, and then the regularized linearized augmented Lagrangian algorithm is used to solve it. It can be seen that the numerical performance of the two is relatively close in terms of the number of iterations and time. All the following algorithms will uniformly adopt the method of introducing N blocks of new variables (Table 5.4).

Several augmented Lagrangian-type algorithms proposed in this paper are compared. It can be found from Algorithm 5.4 that for RALM, generally when the regular parameter γ is relatively small, the performance is better. Compared with RLALM, the number of iterations of RLALM is significantly smaller, and the iteration time of some examples is also significantly shorter. The parameters of CRLALM have to

Table 5.5 Numerical results for JADMM and RLALM

Example	JADMM		JADMMa		RLALM	
	Iter.	Time	Iter.	Time	Iter.	Time
4.1	53	21.3	22	8.5	14	5.4
4.2	87	99.1	34	31.0	22	34.1
4.3	112	39.6	23	9.5	8	3.8
4.4	536	171.0	202	81.6	92	80.8
4.5	4511	13868.4	361	499.7	23	101.5
4.6	–	–	156	175.0	23	61.0
4.7	788	874.4	190	261.9	46	124.6

Table 5.6 Numerical results for GADMM and RLALM

Example	GADMM		LGADMM		RLALM	
	Iter.	Time	Iter.	Time	Iter.	Time
4.1	22	10.5	35	18.8	14	5.4
4.2	36	32.5	366	512.3	22	34.1
4.3	36	29.5	197	93.7	8	3.8
4.4	100	72.0	364	191.2	92	80.8
4.5	500	772.9	711	990.2	23	101.5
4.6	60	86.9	187	221.8	23	61.0
4.7	91	428.5	160	580.6	46	124.6

be re-adjusted to have better results. Its numerical performance is obviously inferior to that of RLALM.

Next, we examine the comparison between JADMM in [26] and RLALM in this paper (Table 5.5).

It can be seen from Algorithm 5.5 that the iteration of JADMM using the original parameter settings in [26] is too many, and JADMMa with manual parameter adjustment can greatly speed up the convergence. And RLALM is also significantly faster than the tuned JADMM. Note that since Example 5.6 is difficult to track the cocoercivity parameters of the pseudo-gradient, the numerical results of JADMM are not given.

Finally, we will examine the numerical performance comparison between the Gauss-Seidel-type ADMM method of the literature [27] and the RLALM (Table 5.6).

Both GADMM and LGADMM in Algorithm 5.6 use the adaptive strategy in the literature [27] to adjust the parameters γ. LGADMM represents the linearization of the objective function part of the GADMM algorithm. We examine the effect of this operation on numerical performance. The parameter requirements of LGADMM are different from those of GADMM, and it needs a larger initial γ and growth span than GADMM in order to have better convergence performance. But its numerical performance is not as good as GADMM. In addition, the numerical performance of RLALM is obviously better than that of GADMM.

5.5 Conclusions

In this paper, we propose several augmented Lagrange-type algorithms for solving generalized Nash equilibrium problems with linear common equality or inequality constraints. Under different assumptions, there were different technical methods for convergence, i.e., the Fejér monotonicity of the corresponding iterative sequences or the equivalence with forward-backward splitting theoretically. In the future, we may consider adapting the parameters to give full play to the efficiency of the algorithms.

Acknowledgements This work was supported by National Natural Science Foundation of China grants 11971238.

References

1. Nikaido, H., Isoda, K.: Note on noncooperative convex games. Pac. J. Math. **5**, 807–815 (1955)
2. Bensoussan, A.: Points de Nash dans le cas de fonctionnelles quadratiqueset jeux differentiels lineaires a N personnes. SIAM J. Control Optim. **12**, 460–499 (1974)
3. Rosen, J.B.: Existence and uniqueness of equilibrium points for concave n-person games. Econometrica **33**, 520–534 (1965)
4. Facchinei, F., Kanzow, C.: Generalized Nash equilibrium problems. Ann. Oper. Res. **175**(1), 177–211 (2010)
5. Fischer, A., Herrich, M., Schönefeld, K.: Generalized Nash equilibrium problems—recent advances and challenges. Ann. Oper. Res. **34**(3), 521–558 (2014)
6. Facchinei, F., Fischer, A., Piccialli, V.: Generalized Nash equilibrium problems and Newton methods. Math. Program. **117**, 163–194 (2009)
7. Santos, P.J.S., Santos, P.S.M., Scheimberg, S.: A newton-type method for quasi-equilibrium problems and applications. Optimization **71**(1), 7–32 (2022)
8. Fukushima, M.: Restricted generalized Nash equilibria and controlled penalty algorithm. CMS **8**(3), 201–218 (2011)
9. Facchinei, F., Kanzow, C.: Penalty methods for the solution of generalized Nash equilibrium problems. SIAM J. Optim. **5**, 2228–2253 (2010)
10. Facchinei, F., Lampariello, L.: Partial penalization for the solution of generalized Nash equilibrium problems. J. Global Optim. **50**(1), 39–57 (2011)
11. Pang, J., Fukushima, M.: Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games. Math. Program. **2**, 21–56 (2005)
12. Kanzow, C.: On the multiplier-penalty-approach for quasi-variational inequalities. Math. Program. **160**(1–2), 33–63 (2016)
13. Kanzow, C., Steck, D.: Augmented Lagrangian methods for the solution of generalized Nash equilibrium problems. SIAM J. Optim. **26**, 2034–2058 (2016)
14. Facchinei, F., Piccialli, V., Sciandrone, M.: Decomposition algorithms for generalized potential games. Comput. Optim. Appl. **50**, 237–262 (2011)

15. Lei, J., Shanbhag, U.V., Pang, J., Sen, S.: On synchronous, asynchronous, and randomized best-response schemes for stochastic Nash games. Math. Oper. Res. **45**, 157–190 (2020)
16. Shanbhag, U.V., Pang, J., Sen, S.: Inexact best-response schemes for stochastic Nash games: linear convergence and iteration complexity analysis. In: Proceedings of the 55th Conference on Decision and Control, pp. 3591–3596. IEEE (2016)
17. Boţ, R.I., Csetnek, E.R., Nguyen, D.-K.: A proximal minimization algorithm for structured nonconvex and nonsmooth problems. SIAM J. Optim. **29**, 1300–1329 (2019)
18. Boţ, R.I., Nguyen, D.-K.: The proximal alternating direction method of multipliers in the nonconvex setting: convergence analysis and rates. Math. Oper. Res. **54**, 682–712 (2020)
19. Börgens, E., Kanzow, C.: Regularized Jacobi-type ADMM-methods for a class of separable convex optimization problems in Hilbert spaces. Comput. Optim. Appl. **73**, 755–790 (2019)
20. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**, 1–122 (2011)
21. Eckstein, J., Yao, W.: Understanding the convergence of the alternating direction method of multipliers: theoretical and computational perspectives. Pac. J. Optim. **11**, 619–644 (2015)
22. He, B., Hou, L., Yuan, X.: On full Jacobian decomposition of the augmented Lagrangian method for separable convex programming. SIAM J. Optim. **25**, 2274–2312 (2015)
23. He, B., Tao, M., Yuan, X.: Alternating direction method with Gaussian back substitution for separable convex programming. SIAM J. Optim. **22**, 313–340 (2012)
24. He, B., Xu, H.-K., Yuan, X.: On the proximal Jacobian decomposition of ALM for multipleblock separable convex minimization problems and its relationship to ADMM. J. Sci. Comput. **66**, 1204–1217 (2016)
25. He, B.: PPA-like contraction methods for convex optimization: a framework using variational inequality approach. J. Oper. Res. Soc. China **3**, 391–420 (2015)
26. Börgens, E., Kanzow, C.: A distributed regularized Jacobi-type ADMM-method for generalized Nash equilibrium problems in Hilbert spaces. Numer. Funct. Anal. Optim. **39**, 1316–1349 (2018)
27. Börgens, E., Kanzow, C.: ADMM-type methods for generalized Nash equilibrium problems in Hilbert spaces. SIAM J. Optim. **31**, 377–403 (2021)
28. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer (2010)
29. Facchinei, F., Pang, J.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer (2003)
30. Han, D., Zhang, H., Qian, G., Xu, L.: An improved two-step method for solving generalized Nash equilibrium problems. Eur. J. Oper. Res. **216**, 613–623 (2012)
31. Krawczyk, J.B., Uryasev, S.: Relaxation algorithms to find Nash equilibria with economic applications. Environ. Model. Assess. **5**(1), 63–73 (2000)
32. Harker, P.T.: Generalized Nash games and quasi-variational inequalities. Eur. J. Oper. Res. **54**, 81–94 (1991)
33. Outrta, J.V., Kocvara, M., Zowe, J.: Nonsmooth approach to optimization problems with equilibrium constraints. J. Environ. Manag. **103**(4), 74–82 (1997)
34. Nabetani, K., Tseng, P., Fukushima, M.: Parametrized variational inequality approaches to generalized Nash equilibrium problems with shared constraints. Comput. Optim. Appl. **48**(3), 423–452 (2011)