Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a uniform approach

Guoyong Gu* Bingsheng He† Xiaoming Yuan‡

October 30, 2011

Abstract. This paper takes a uniform look at the customized applications of proximal point algorithm (PPA) to two classes of problems: the linearly constrained convex minimization problem with a generic or separable objective function and a saddle-point problem. We model these two classes of problems uniformly by a mixed variational inequality, and show how PPA with customized proximal parameters can yield favorable algorithms, which are able to exploit the structure of the models fully. Our customized PPA revisit turns out to be a uniform approach in designing a number of efficient algorithms, which are competitive with, or even more efficient than some benchmark methods in the existing literature such as the augmented Lagrangian method, the alternating direction method, the split inexact Uzawa method, and a class of primal-dual methods, etc. From the PPA perspective, the global convergence and the $O(1/t)$ convergence rate for this series of algorithms are established in a uniform way.

Keywords. Convex minimization, saddle-point problem, proximal point algorithm, $O(1/t)$ convergence rate, customized algorithms, splitting algorithms.

1 Introduction

Let Ω be a closed convex subset in $\Re^l$ and $F : \Re^l \to \Re^l$ be a monotone mapping; $\Re^{l'}$ $(l' \leq l)$ be a subspace of $\Re^l$ and $\theta : \Re^{l'} \to \Re$ be a closed convex but not necessarily smooth function. Let u denote the sub-vector of w in $\Re^{l'}$ for any $w \in \Omega$. We consider the mixed variational inequality (MVI): Find $w^* \in \Omega$ such that

$$\theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega. \tag{1.1}$$

The MVI (1.1) includes the ordinary variational inequality (see [30]) as a special case with $\theta \equiv 0$, and it has been well studied in various fields such as the partial differential equations, economics and mathematical programming [40, 53, 66]. Throughout, the solution set of (1.1), denoted by Ω^*, is assumed to be nonempty.

We do not discuss the generic case of MVI (1.1). Instead, we focus on some fundamental optimization models which all turn out to be special cases of (1.1) where there are specific properties/structures associated with the function θ, the mapping F and the set Ω. We thus discuss how to develop customized algorithms in accordance with these properties/structures for these optimization models. But, the MVI (1.1) serves as a uniform mathematical model for our theoretical analysis, and it enables us to show the convergence results uniformly while presenting different algorithms for

*Email: ggu@nju.edu.cn. This author was supported by the NSFC grant 11001124.

†Email: hebma@nju.edu.cn. This author was supported by the NSFC grant 10971095.

‡Corresponding author. Email: xmyuan@hkbu.edu.hk. This author was supported by a research grant from Hong Kong General Research Fund.

various models individually. More specifically, we consider two classes of problems: a) the convex minimization problem with linear constraints and a generic or separable objective function, and b) a saddle-point problem.

1). The generic convex minimization problem with linear constraints

$$\min\{\theta(x) \mid Ax = b\ ,\ x \in \mathcal{X}\}, \tag{1.2}$$

where $A \in \Re^{m\times n}$, $b \in \Re^m$, $\mathcal{X} \subseteq \Re^n$ is convex, and $\theta : \Re^n \to \Re$ is a closed convex but not necessarily smooth function. The objective function in (1.2) is generic, and there is no further separable structure assumed.

2). The particular separable case of (1.2) where the objective function is separable into two individual functions without coupled variables. For this case, by decomposing the linear constraints into two parts accordingly, we consider the model

$$\min\{\theta(x) := \theta_1(x_1) + \theta_2(x_2) \mid A_1x_1 + A_2x_2 = b,\ x = (x_1, x_2) \in \mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2\}, \tag{1.3}$$

where $A_1 \in \Re^{m\times n_1}$, $A_2 \in \Re^{m\times n_2}$, $b \in \Re^m$, $\mathcal{X}_1 \subseteq \Re^{n_1}$ and $\mathcal{X}_2 \subseteq \Re^{n_2}$ are convex sets, $n_1+n_2 = n$, $\theta_1 : \Re^{n_1} \to \Re$ and $\theta_2 : \Re^{n_2} \to \Re$ are closed convex but not necessarily smooth functions. With the purposes of exploiting the separable structure fully and developing more customized algorithms, the philosophy of algorithmic design for the separable case (1.3) is different from that for the generic case (1.2). Thus, the separable case (1.3) is worth a particular investigation.

3). The general separable case of (1.2) where the objective function is separable into more than two individual functions without coupled variables. Again, by rewriting the linear constraints in accordance with the separable objective function, we consider the model

$$\min\left\{\sum_{i=1}^{K}\theta_i(x_i) \mid \sum_{i=1}^{K}A_ix_i = b,\ x_i \in \mathcal{X}_i,\ i = 1, \cdots, K\right\}, \tag{1.4}$$

where $A_i \in \Re^{m\times n_i}$ $(i = 1, \ldots, K)$, $b \in \Re^m$, $\mathcal{X}_i \subseteq \Re^{n_i}$ $(i = 1, \ldots, K)$ are convex sets, $\sum_{i=1}^{K} n_i = n$, and $\theta_i : \Re^{n_i} \to \Re$ $(i = 1, \ldots, K)$ are closed convex but not necessarily smooth. Note that (1.3) is a special case of (1.4) with $K = 2$. We consider (1.3) individually because of its own wide applications in various fields and its unique speciality in algorithmic design (as referred to Sections 6 and 7) which are not extendable to the general case (1.4) with $K \geq 3$. Alternatively, with the purpose of exploiting the properties of θ_i's individually in the procedure of algorithmic design, the model (1.4) deserves specific attention mainly due to the failure of extending those algorithms applicable for (1.3) to (1.4) straightforwardly, see e.g. [39, 43, 44, 45].

4). The saddle-point problem

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}}\{\theta_1(x) - y^TAx - \theta_2(y)\}, \tag{1.5}$$

where $A \in \Re^{m\times n}$, $\mathcal{X} \subseteq \Re^n$, $\mathcal{Y} \subseteq \Re^m$, $\theta_1 : \Re^n \to \Re$ and $\theta_2 : \Re^m \to \Re$ are closed convex but not necessarily smooth functions. The saddle-point problem (1.5) captures a broad spectrum of applications in various fields such as image restoration problems with the total variation (TV) regularization introduced in [70] (see e.g. [15, 29, 82, 91]), fluid dynamics or linear elasticity problems in the contexts of partial differential equations (see e.g. [1, 31]), and Nash equilibrium problems in game theory (see e.g. [55, 61]). In particular, finding a saddle point of the Lagrange function of the model (1.2) is a special case of (1.5).

In Section 2.1, we specify how to reformulate the models (1.2)-(1.5) as special cases of the MVI (1.1) case by case.

The proximal point algorithm (PPA), which dates back to [57] and was firstly introduced to the optimization community in [56], has been playing fundamental roles both theoretically and algorithmically in the optimization area, including of course the models (1.2)-(1.5) under our consideration, see e.g. [37, 38, 69] for a few of seminal works. Let us make more concrete our motivation of relating PPA to the mentioned target models.

1). For solving (1.2), the augmented Lagrangian method (ALM, [51, 65]) is a benchmark method in the literature. More specifically, the iterate scheme of ALM for (1.2) is

$$\begin{cases} x^{k+1} = \text{Argmin}\,\{\theta(x) - (\lambda^k)^T(Ax-b) + \frac{\beta}{2}\|Ax-b\|^2 \,\big|\, x \in \mathcal{X}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1}-b), \end{cases} \tag{1.6}$$

where λ^k is the Lagrange multiplier and $\beta > 0$ is the penalty parameter for the violation of the linear constraints. In [68], it was shown that the ALM (1.6) is exactly the application of PPA to the dual problem of (1.2).

2). For solving (1.3), it is not wise to apply the generic-purpose ALM directly and a dominating method in the literature is the alternating direction method (ADM) proposed originally in [32]. The iterative scheme of ADM for (1.3) is

$$\begin{cases} x_1^{k+1} = \arg\min\{\theta_1(x_1) - (\lambda^k)^T(A_1x_1 + A_2x_2^k - b) + \frac{\beta}{2}\|A_1x_1 + A_2x_2^k - b\|^2 \mid x \in \mathcal{X}_1\}, \\ x_2^{k+1} = \arg\min\{\theta_2(x_2) - (\lambda^k)^T(A_1x_1^{k+1} + A_2x_2 - b) + \frac{\beta}{2}\|A_1x_1^{k+1} + A_2x_2 - b\|^2 \mid x_2 \in \mathcal{X}_2\}, \\ \lambda^{k+1} = \lambda^k - \beta(A_1x_1^{k+1} + A_2x_2^{k+1} - b). \end{cases} \tag{1.7}$$

Obviously, the ADM (1.7) is a splitting version of the ALM (1.6) in accordance with the separable structure of (1.3). By decomposing the ALM subproblem into two subproblems in the Gauss-Seidel fashion at each iteration, the variables x_1 and x_2 can be solved separably in the alternating order. Since the functions $\theta_1(x_1)$ and $\theta_2(x_2)$ often have specific properties for a particular application of (1.3), the decomposition treatment of ADM makes it possible to exploit these particular properties separately. This feature has inspired the recent burst of ADM's novel applications in various areas, see e.g. [4, 19, 20, 27, 46, 59, 60, 73, 83, 88] and references cited therein. In particular, we refer to [27, 72] for the relationship between ADM and the split Bregman iteration scheme [34] which is very influential in the area of image processing.

In [33], it was elucidated that the ADM (1.7) is essentially the application of the Douglas-Rachford splitting method [25] (which is a special form of PPA, as demonstrated in [26]) to the dual problem of (1.3).

3). For solving (1.4), a natural idea is to extend the ADM (1.7) in a straightforward way, yielding the iterative scheme

$$\begin{cases} x_1^{k+1} = \arg\min\big\{\theta_1(x_1) + \frac{\beta}{2}\|(A_1x_1 + \sum_{j=2}^K A_jx_j^k - b) - \frac{1}{\beta}\lambda^k\|^2 \,\big|\, x_1 \in \mathcal{X}_1\big\}; \\ x_2^{k+1} = \arg\min\big\{\theta_2(x_2) + \frac{\beta}{2}\|(A_1x_1^{k+1} + A_2x_2 + \sum_{j=3}^K A_jx_j^k - b) - \frac{1}{\beta}\lambda^k\|^2 \big| x_2 \in \mathcal{X}_2\big\}; \\ \quad\vdots \\ x_i^{k+1} = \arg\min\big\{\theta_i(x_i) + \frac{\beta}{2}\|(\sum_{j=1}^{i-1} A_jx_j^{k+1} + A_ix_i + \sum_{j=i+1}^K A_jx_j^k - b) - \frac{1}{\beta}\lambda^k\|^2 \big| x_i \in \mathcal{X}_i\big\}; \\ \quad\vdots \\ x_K^{k+1} = \arg\min\big\{\theta_K(x_K) + \frac{\beta}{2}\|(\sum_{j=1}^{K-1} A_jx_j^{k+1} + A_Kx_K - b) - \frac{1}{\beta}\lambda^k\|^2 \,\big|\, x_K \in \mathcal{X}_K\big\}; \\ \lambda^{k+1} = \lambda^k - \beta(\sum_{j=1}^K A_jx_j^{k+1} - b). \end{cases} \tag{1.8}$$

The extended ADM scheme (1.8), which comes from the straightforward splitting of the ALM subproblem in the alternating order, preserves the advantages of the original ADM scheme (1.7) such that θ_i's properties can be exploited individually and thus the subproblems might be easy. Unfortunately, the convergence of (1.8) remains a challenge without further assumptions on the model (1.4). This difficulty thus has inspired us to develop a series of splitting algorithms [39, 43, 44, 45] recently, the common purpose of which is to preserve the decomposition nature as (1.8). Among these work, the splitting method in [44] is inspired by PPA.

4). As analyzed in [15, 29, 91], the saddle-point problem (1.5) can be regarded as the primal-dual formulation of a nonlinear programming problem, and this fact has inspired a series of primal-dual algorithms in the particular literature of image restoration problems with total variational regularization. We refer to, e.g. [15, 17, 29, 35, 64, 89, 90, 91], for their numerical efficiency. In [49], we revisit these primal-dual algorithms from the PPA perspective. It turns out that this PPA revisit simplifies the convergence analysis for this type of algorithms substantially and makes it possible to relax the involved parameters (step sizes and proximal parameters) greatly, as acknowledged instantly by some most recent works [22, 63, 81].

Because of the aforementioned individual applications, we are interested in studying the PPA's applications for the models (1.2)-(1.5) in a uniform way, by means of the uniform model (1.1). Our aim is to show that by choosing the proximal parameters judiciously in accordance with the specific structures of the MVI reformulations of the models (1.2)-(1.5), a series of customized PPAs can be yielded. These customized PPAs are fully capable of taking advantage of the available structures of the considered models, and they are competitive with, or even more efficient than, some benchmark methods designed particularly for these models. In addition, this customized PPA approach enables us to establish the global convergence and the $O(1/t)$ convergence rate uniformly for this series of algorithms.

To elucidate the application of PPA to the MVI (1.1), let $G \in \Re^{l\times l}$ be a symmetric positive definite matrix and the G-norm be defined as $\|w\|_G := \sqrt{w^T G w}$. With the given $w^k \in \Omega$, the PPA in the metric form for (1.1) generates the new iterate w^{k+1} via solving the subproblem

$$\text{(PPA)} \quad w^{k+1} \in \Omega, \;\; \theta(u) - \theta(u^{k+1}) + (w - w^{k+1})^T\big(F(w^{k+1}) + G(w^{k+1} - w^k)\big) \geq 0, \;\; \forall\, w \in \Omega, \quad (1.9)$$

where G is called the metric proximal parameter. A popular choice of G is $G = rI$ where $r > 0$ is a scale and $I \in \Re^{l\times l}$ is the identity matrix. In the literature there are intensive investigations on how to determine a value of r to guarantee convergence theoretically or how to adjust it dynamically for numerical acceleration. This simplest choice of G essentially means that the proximal regularization makes no difference on different coordinates of w and thus all the coordinates of w are proximally regularized with equivalent weights. On the other hand, there are some impressive works on PPA with metric proximal regularization, e.g. [3, 6, 7, 54] and references therein, which mainly discuss theoretical restrictions or numerical choices on the metric proximal parameter.

For the generic form of (1.1) where particular properties/structures of θ, F and Ω are not specified, there is no hint to determine any customized choice for the metric proximal parameter and thus the simplest choice of G is enough on theoretical purposes. But, for the particular models (1.2)-(1.5) under our consideration, their MVI reformulations in the form of (1.1) enjoy favorable splitting structure in θ, F and Ω. Thus, customized choices of G in accordance with the separable structures of their MVI reformulations are potential to decompose the generic PPA task (1.9) into smaller and easier subproblems. Accordingly, it becomes possible to exploit fully the particular properties of the models (1.2)-(1.5) for algorithmic benefits. This is our philosophy of algorithmic design.

The rest of the paper is organized as follows. In Section 2, we provide some preliminaries which are necessary for further discussions. In Section 3, we present the conceptual algorithmic framework based on the relaxed PPA in [36] for the models (1.2)-(1.5). Then, we elucidate the customization of this relaxed PPA for the model (1.2) in Section 4. The customization of this relaxed PPA for the model (1.3) is analyzed in Section 5, while further customizations for some special cases of (1.3) are discussed in Sections 6 and 7. Similar discussions for the model (1.4) are completed in Section 8. Afterwards, we analyze the customization for the model (1.5) in Section 9. After individual discussions on the customization for the models (1.2)-(1.5), we establish uniformly the global convergence in Section 10 and the $O(1/t)$ convergence rate in Section 11 for these new algorithms. In Section 12, we list some concrete applications of the models (1.2)-(1.5) arising in different areas. Finally, some conclusions are made in Section 13.

2 Preliminaries

In this section, we review some preliminaries which are useful later. First, we specify the MVI reformulation for the models (1.2)-(1.5). Then, we take a brief look at the generic application of PPA to this variational reformulation, and in particular, we recall the relaxed PPA in [36] which blends the original PPA with a simple relaxation step. After that, we show a characterization on the solution set of the MVI (1.1) which is a cornerstone for proving the $O(1/t)$ convergence rate of the new methods. Finally, we supplement some useful notations and preliminaries.

2.1 The MVI reformulations of (1.2)-(1.5)

We first show that all the models (1.2)-(1.5) can be reformulated as specifical cases of the MVI (1.1).

1). Let $\lambda \in \Re^m$ be the Lagrange multiplier associated with the linear constraints in (1.2). It is easy to see that solving (1.2) amounts to finding $w^* = (u^*, \lambda^*)$ such that

$$w^* \in \Omega, \quad \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega, \tag{2.1a}$$

where

$$w = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \qquad F(w) = \begin{pmatrix} -A^T\lambda \\ Ax - b \end{pmatrix} \text{ and } \Omega = \mathcal{X} \times \Re^m. \tag{2.1b}$$

Therefore, (2.1) is a special case of (1.1) with $l = m + n$, $u = x$, $w = (u, \lambda)$, $F(w)$ and Ω are given in (2.1b).

2). Similarly, we have that solving the model (1.3) is equivalent to finding $w^* = (x_1^*, x_2^*, \lambda^*)$ such that

$$w^* \in \Omega, \quad \theta_1(x_1) - \theta_1(x_1^*) + \theta_2(x_2) - \theta_2(x_2^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega, \tag{2.2a}$$

where

$$w = \begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix}, \qquad F(w) = \begin{pmatrix} -A_1^T\lambda \\ -A_2^T\lambda \\ A_1x_1 + A_2x_2 - b \end{pmatrix} \text{ and } \Omega = \mathcal{X}_1 \times \mathcal{X}_2 \times \Re^m. \tag{2.2b}$$

Therefore, (2.2) is a special case of (1.1) with $l = n_1 + n_2 + m$, $u = (x_1, x_2)$, $\theta(u) = \theta_1(x_1) + \theta_2(x_2)$, $w = (x_1, x_2, \lambda)$, $F(w)$ and Ω are given in (2.2b).

3). Also, the Lagrange function of (1.4) is

$$L(x_1, x_2, \ldots, x_K, \lambda) = \sum_{i=1}^{K} \theta_i(x_i) - \lambda^T(\sum_{i=1}^{K} A_i x_i - b),$$

which is defined on

$$\Omega = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_K \times \Re^m. \tag{2.3}$$

It is evident that finding a saddle point of $\mathcal{L}(x_1, x_2, \ldots, x_K, \lambda)$ is equivalent to finding a vector $w^* = (x_1^*, x_2^*, ..., x_K^*, \lambda^*) \in \Omega$ such that

$$\left\{ \begin{array}{rcl} \theta_1(x_1) - \theta_1(x_1^*) & + & (x_1 - x_1^*)^T(-A_1^T\lambda^*) \geq 0, \quad \forall\, x_1 \in \mathcal{X}_1, \\ \theta_2(x_2) - \theta_2(x_2^*) & + & (x_2 - x_2^*)^T(-A_2^T\lambda^*) \geq 0, \quad \forall\, x_2 \in \mathcal{X}_2, \\ & \vdots & \\ \theta_K(x_K) - \theta_K(x_K^*) & + & (x_K - x_K^*)^T(-A_K^T\lambda^*) \geq 0, \quad \forall\, x_K \in \mathcal{X}_K, \\ & & (\lambda - \lambda^*)^T\{\sum_{i=1}^K A_i x_i^* - b\} \geq 0, \quad \forall \lambda \in \Re^m. \end{array} \right. \tag{2.4}$$

More compactly, (2.4) can be written into the following VI:

$$\theta(u) - \theta(u^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega, \tag{2.5a}$$

which is a special case of (1.1) with $l = \sum_{i=1}^K n_i + m$,

$$u = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{pmatrix}, \quad \theta(u) = \sum_{i=1}^K \theta_i(x_i), \quad w = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \\ \lambda \end{pmatrix}, \; F(w) = \begin{pmatrix} -A_1^T\lambda \\ -A_2^T\lambda \\ \vdots \\ -A_K^T\lambda \\ \sum_{i=1}^K A_i x_i - b \end{pmatrix} \tag{2.5b}$$

and Ω being given in (2.3).

4). Note that solving (1.5) is equivalent to finding $w^* = (x^*, y^*)$ such that

$$w^* \in \Omega, \quad \theta_1(x) - \theta_1(x^*) + \theta_2(y) - \theta_2(y^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega, \tag{2.6a}$$

where

$$w = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad F(w) = \begin{pmatrix} -A^T y \\ Ax \end{pmatrix}, \qquad \Omega = \mathcal{X} \times \mathcal{Y}. \tag{2.6b}$$

Therefore, the model (2.6) is a special case of the MVI (1.1) with $l = n + m$, $u = w = (x, y)$, $\theta(u) = \theta_1(x) + \theta_2(y)$, $F(w)$ and Ω are given in (2.6b).

Note that it is trivial to verify that the operators $F(w)$ given in (2.1b), (2.2b), (2.5b) and (2.6b) are all monotone.

2.2 A characterization of the solution set of (1.1)

As in [9, 41, 50], for the purpose of proving the $O(1/t)$ convergence rate, it is useful to follow Theorem 2.3.5 of [30] (see (2.3.2) in pp.159) and characterize the solution set of the MVI (1.1) by an intersection set. For completeness, we provide the full proof in Theorem 2.1.

Theorem 2.1. *The solution set of (1.1), i.e., Ω^*, is convex and it can be represented by*

$$\Omega^* = \bigcap_{w \in \Omega} \big\{ \tilde{w} \in \Omega : \theta(u) - \theta(\tilde{u}) + (w - \tilde{w})^T F(w) \geq 0 \big\}. \tag{2.7}$$

Proof. Indeed, if $\tilde{w} \in \Omega^*$, we have

$$\theta(u) - \theta(\tilde{u}) + (w - \tilde{w})^T F(\tilde{w}) \geq 0, \ \ \forall\, w \in \Omega.$$

By using the monotonicity of F on Ω, this implies

$$\theta(u) - \theta(\tilde{u}) + (w - \tilde{w})^T F(w) \geq 0, \ \ \forall\, w \in \Omega.$$

Thus, $\tilde{w}$ belongs to the right-hand-side set in (2.7). Conversely, suppose $\tilde{w}$ belongs to the right-hand-side set in (2.7). Let $w \in \Omega$ be arbitrary. The vector

$$\bar{w} = \alpha\tilde{w} + (1-\alpha)w$$

belong to Ω for all $\alpha \in (0,1)$. Thus we have

$$\theta(\bar{u}) - \theta(\tilde{u}) + (\bar{w} - \tilde{w})^T F(\bar{w}) \geq 0. \tag{2.8}$$

Because $\theta(\cdot)$ is convex, we have

$$\theta(\bar{u}) \leq \alpha\theta(\tilde{u}) + (1-\alpha)\theta(u).$$

Substituting it into (2.8), we get

$$\theta(u) - \theta(\tilde{u}) + (w - \tilde{w})^T F(\alpha\tilde{w} + (1-\alpha)w) \geq 0,$$

for all $\alpha \in (0,1)$. Letting $\alpha \to 1$ yields

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

Thus $\tilde{w} \in \Omega^*$. Now, we turn to prove the convexity of Ω^*. For each fixed but arbitrary $w \in \Omega$, the set

$$\{\tilde{w} \in \Omega : \ \theta(\tilde{u}) + \tilde{w}^T F(w) \leq \theta(u) + w^T F(w)\},$$

and its equivalent expression

$$\{\tilde{w} \in \Omega : \ \theta(u) - \theta(\tilde{u}) + (w - \tilde{w})^T F(w) \geq 0\}$$

is convex. Since the intersection of any number of convex sets is convex, it follows that Ω^* is convex. □

Theorem 2.1 thus implies that $\tilde{w} \in \Omega$ is an approximate solution of the MVI (1.1) with the accuracy $\epsilon > 0$ if it satisfies

$$\theta(u) - \theta(\tilde{u}) + F(w)^T(w - \tilde{w}) \geq -\epsilon, \ \ \forall w \in \Omega.$$

In other words, for a substantial compact set $\mathcal{D} \subset \Omega$, after t iterations of an algorithm, if we can find $\tilde{w} \in \Omega$ such that

$$\tilde{w} \in \Omega \quad \text{and} \quad \sup_{w \in \mathcal{D}} \big\{\theta(\tilde{u}) - \theta(u) + (\tilde{w} - w)^T F(w)\big\} \leq \epsilon,$$

where $\epsilon = O(1/t)$, then the $O(1/t)$ convergence rate of this algorithm is derived. In the papers [9, 41, 50], we have shown the $O(1/t)$ convergence rate for some algorithms including the ADM (1.7) and the split inexact Uzawa method in [89, 90]. In this paper, we will follow this line of research and prove the $O(1/t)$ convergence rate for the application of PPA to the MVI (1.1), and thus the $O(1/t)$ convergence rate of a series of algorithms are established uniformly.

2.3 Some additional notations

In this subsection we supplement some useful notations for the convenience of further analysis.

First, revisiting the iterative schemes of the ALM (1.6), it is easy to observe that the variable x plays just an intermediate role and only the sequence $\{\lambda^k\}$ is required to execute the scheme. Similarly, for the ADM (1.7), the variable x_1 is an intermediate variable and only the sequence $\{x_2^k, \lambda^k\}$ is involved in the iteration. Thus, for the variable w in (1.1), we conceptually classify all the coordinates into two categories: the intermediate and essential coordinates, depending on whether or not they are required during the iteration of a certain algorithm. Here, we introduce the variable v, an appropriate sub-vector of w, to collect all the essential coordinates of w, i.e., v represents all the coordinates of w which are really involved in iterations. Accordingly, the intended meaning of the notations v^k, $\tilde{v}^k$, v^*, $\mathcal{V}$ and $\mathcal{V}^*$ should be clear from the context. For example, when the ALM (1.6) is considered, we have

$$v = \lambda; \;\; \mathcal{V} = \Re^m;$$
$$v^k = \lambda^k; \quad \tilde{v}^k = \tilde{\lambda}^k, \; \forall k \in \mathcal{N};$$
$$v^* = \lambda^*; \;\; \mathcal{V}^* = \{\lambda^* \,|\, (x^*, \lambda^*) \in \Omega^*\}.$$

When the ADM (1.7) is considered, we have

$$v = (x_2, \lambda); \;\; \mathcal{V} = \mathcal{X}_2 \times \Re^m;$$
$$v^k = (x_2^k, \lambda^k); \quad \tilde{v}^k = (\tilde{x}_2^k, \tilde{\lambda}^k), \; \forall k \in \mathcal{N};$$
$$v^* = (x_2^*, \lambda^*); \;\; \mathcal{V}^* = \{(x_2^*, \lambda^*) \,|\, (x_1^*, x_2^*, \lambda^*) \in \Omega^*\}.$$

Later, for the algorithms to be proposed, we will establish the global convergence in the context of the essential coordinates v. That is, we shall show that the sequence $\{v^k\}$ generated by each of the algorithms to be proposed converges to a point in $\mathcal{V}^*$. The consideration of investigating the convergence on the essential coordinates is inspired by the well-known convergence results of the ALM in [68] and the ADM in [4, 33, 47]. In these literatures, the convergence analysis for the ALM is conducted in term of the sequence $\{\lambda^k\}$ in [68], and for the ADM in term of the sequence $\{x_2^k, \lambda^k\}$ in [4, 33, 47].

On the other hand, it is apparent that the proximal regularization on the intermediate coordinates of w is profitless or redundant in (1.9). Thus, we only need to regularize proximally the essential coordinates of w. In other words, the metric proximal parameter G in (1.9) are not necessarily square with the dimensionality $l \times l$. In fact, its number of columns corresponds to the dimensionality of the vector v while its number of rows are still in the dimensionality of l. Thus, in general, the metric proximal parameter could be a thin matrix in (1.1) under our consideration for the models (1.2)-(1.5). Accordingly, by denoting the partial metric proximal parameter as Q, we rewrite (1.9) as

$$w^{k+1} \in \Omega, \;\; \theta(u) - \theta(u^{k+1}) + (w - w^{k+1})^T \{F(w^{k+1}) + Q(v^{k+1} - v^k)\} \geq 0, \;\; \forall\, w' \in \Omega. \tag{2.9}$$

Indeed, this idea of partially proximal regularization has been implemented in [9] for the particular case of (1.3).

Finally, we also recall the definition of the projection under the Euclidean norm. For a convex set $\Omega \subseteq \Re^l$, the projection onto Ω under the Euclidean norm is defined by

$$P_\Omega(w) = \text{Argmin}\{\frac{1}{2}\|z - w\|^2 | z \in \Omega\}. \tag{2.10}$$

By the first-order optimality condition, we can easily derive the inequality

$$(z - P_\Omega(w))^T (P_\Omega(w) - w) \geq 0, \quad \forall\; z \in \Omega, \forall\, w \in \Re^l, \tag{2.11}$$

which will be used often in the coming analysis.

3 Conceptual algorithmic framework

In this section, we propose the conceptual algorithmic framework by applying PPA to the MVI (1.1). In particular, we are interested in the relaxed PPA [36] which generates the new iterate by relaxing the output of the original PPA (1.9) appropriately. More specifically, let the solution of (1.9) be denoted by $\tilde{w}^k$, then the relaxed PPA in [36] yields the new iterate via

$$w^{k+1} = w^k - \gamma(w^k - \tilde{w}^k), \tag{3.1}$$

where $\gamma \in (0, 2)$ is a relaxation factor, and it is called an under-relaxation (when $\gamma \in (0, 1)$) or over-relaxation factor (when $\gamma \in (1, 2)$). Obviously, the relaxed PPA (3.1) reduces to the original PPA (1.9) when $\gamma = 1$. Our numerical results in [9, 48, 49] has already shown the effectiveness of acceleration contributed by this relaxed step (3.1) with $\gamma \in (1, 2)$ numerically.

Recall that we consider the PPA subproblem (2.9) with partial proximal regularization merely on the essential coordinates of w, i.e., the sub-vector v. Thus, instead of the full relaxation on w in (3.1), we only relax v^k in the relaxation step accordingly. Overall, we propose conceptually the following algorithmic framework for the MVI (1.1) by applying the relaxed PPA in [36] but with partial proximal regularization.

The conceptual algorithmic framework of relaxed PPA for (1.1) with partial proximal regularization.

Let the partial metric proximal parameter Q be positive semi-definite, the relaxation factor $\gamma \in (0, 2)$ and v be an appropriate sub-vector of w. With the initial iterate v^0, the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\tilde{w}^k \in \Omega, \quad \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T \{F(\tilde{w}^k) + Q(\tilde{v}^k - v^k)\} \geq 0, \quad \forall\, w \in \Omega. \tag{3.2}$$

2. **Relaxation step**: generate the new iterate v^{k+1} via

$$v^{k+1} = v^k - \gamma(v^k - \tilde{v}^k). \tag{3.3}$$

As we have mentioned, our convergence analysis will be mainly conducted in the context of the essential coordinates v. For each concrete algorithm to be proposed, the following requirement should be met for the convenience of theoretical analysis.

A requirement associated with the conceptual relaxed PPA (3.2)-(3.3).

We need to identify a positive semi-definite, square, and symmetric sub-matrix of Q, which is denoted by H, such that

$$w^T Q v = v^T H v, \quad \forall w \in \Omega. \tag{3.4}$$

Remark 3.1. Recall that the partial metric proximal parameter Q could be a thin matrix (when $v \neq w$), and it is not necessary to be square and positive definite. This is different from a traditional metric proximal parameter in the PPA literature, where the square and positive definiteness requirements are always assumed. However, if v coincides with w, i.e., all the coordinates of w are essential, then Q is square. For this case, we simply choose $H = Q$. Nevertheless, as we shall show later, if Q is a thin matrix, it is also very easy to find the corresponding matrix H in accordance with the essential coordinates of w.

Remark 3.2. We refer to [9, 48, 49] for some preliminary discussions on the relaxed PPA for the models (1.2), (1.3) and (1.5).

4 Relaxed customized PPAs for (1.2)

Now, we start to specify the conceptual relaxed PPA (3.2)-(3.3) with customized metric proximal parameters in accordance with the structures of the models (1.2)-(1.5). In this section, we focus on the generic convex minimization model (1.2), and show that several scalable algorithms can be derived by choosing different forms of the metric proximal parameter Q in (3.2).

4.1 A relaxed augmented Lagrangian method

Recall that the ALM (1.6) is an application of PPA to the dual problem of (1.2). In this subsection, we demonstrate that the ALM (1.6) can be recovered by taking a specific metric proximal parameter in (3.2). Consequently, the ALM (1.6) itself is also a special case of the direct application of PPA to the primal problem (1.2). The PPA illustration of the ALM (1.6) thus makes it possible to combine the ALM (1.6) with a relaxation step, yielding immediately the a relaxed ALM to be proposed.

Denoting by $(\tilde{x}^k, \tilde{\lambda}^k)$ the output of the ALM (1.6), we can rewrite the iterative scheme (1.6) of ALM as

$$\begin{cases} \tilde{x}^k = \text{Argmin}\{\theta(x) + \frac{\beta}{2}\|Ax - b - \frac{1}{\beta}\lambda^k\|^2 \mid x \in \mathcal{X}\}, \\ \tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k - b). \end{cases} \tag{4.1}$$

According to the scheme (4.1), we have

$$\tilde{x}^k \in \mathcal{X}, \quad \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\{\beta A^T[A\tilde{x}^k - b - \frac{1}{\beta}\lambda^k]\} \geq 0, \ \ \forall x \in \mathcal{X},$$

or equivalently,

$$\tilde{x}^k \in \mathcal{X}, \quad \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T\big(-A^T\tilde{\lambda}^k\big) \geq 0, \ \ \forall x \in \mathcal{X}. \tag{4.2}$$

Note that $\tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k - b)$ can be written as

$$(A\tilde{x}^k - b) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) = 0 \tag{4.3}$$

Combining (4.2) and (4.3) together, we get $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega := \mathcal{X} \times \Re^m$ and

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0, \ \ \forall (x, \lambda) \in \Omega. \tag{4.4}$$

Obviously, (4.4) coincides with (3.2) according to the specification given in (2.1b) and the metric proximal parameter given by

$$Q = \begin{pmatrix} 0 \\ \frac{1}{\beta}I_m \end{pmatrix}.$$

We thus can obtain a relaxed ALM for (1.2) by specifying the conceptual algorithmic framework of the relaxed PPA (3.2)-(3.3). Note that the scheme (4.1) only requires λ^k during its iterations. Thus, the essential coordinates of w is $v = \lambda$ in (4.4) and the metric proximal regularization matrix Q is thin. Accordingly, we only relax the coordinates λ in the relaxation step.

Remark 4.1. Because of the simplicity of the relaxation step (4.6), the proposed Algorithm 4.1 and the ALM (1.6) are of the same difficulty to implement numerically.

Remark 4.2. Note that the requirement (3.4) is met by choosing $H = \frac{1}{\beta}I_m$.

Algorithm 4.1 A relaxed augmented Lagrangian method for (1.2)

Let $\gamma \in (0,2)$ and $\beta > 0$ be given. With the initial iterate λ^0, the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving
$$\begin{cases} \tilde{x}^k = \text{Argmin}\{\theta(x) + \frac{\beta}{2}\|Ax - b - \frac{1}{\beta}\lambda^k\|^2 \mid x \in \mathcal{X}\}, \\ \tilde{\lambda}^k = \lambda^k - \beta(A\tilde{x}^k - b). \end{cases} \tag{4.5}$$

2. **Relaxation step**: generate the new iterate λ^{k+1} via
$$\lambda^{k+1} = \lambda^k - \gamma(\lambda^k - \tilde{\lambda}^k). \tag{4.6}$$

4.2 The relaxed customized PPA for (1.2) in the primal-dual order in [48]

In various areas, we witness such a situation of (1.2) where the objective function $\theta(x)$ itself is easy in the sense that its resolvent operator has a closed-form representation or it can be efficiently solved up to a high precision. Here, the resolvent operator of the convex function θ is defined as
$$\big(I + \frac{1}{\beta}\partial\theta\big)^{-1}(a) = \text{Argmin}\{\theta(x) + \frac{\beta}{2}\|x - a\|^2 \mid x \in \Re^n\}, \tag{4.7}$$
for any given $a \in \Re^n$ and $\beta > 0$, see [67]. However, even for a function $\theta(x)$ whose resolvent operator is easy to evaluate, the evaluation of
$$\big(A^TA + \frac{1}{\beta}\partial\theta\big)^{-1}(A^Ta) = \text{Argmin}\{\theta(x) + \frac{\beta}{2}\|Ax - a\|^2 \mid x \in \Re^n\},$$
could be still difficult provided that the matrix A is not identity. An illustrative example is the basis pursuit problem (see [21]) which falls exactly into the model (1.2) with $\theta(x) = \|x\|_1$ ($\|x\|_1 := \sum_{i=1}^n |u_i|$ for inducing the sparsity) and $A \in \Re^{m\times n}$ (with $m << n$). For such a scenario, although the ALM (1.6) alleviates the difficulty resulted by the linear constrains, the resulting ALM subproblem (1.6) is still difficult enough to require inner iterations to pursuit an approximate solution, and the implementation of the ALM (1.6) might be resistive from the numerical point of view. This difficulty, however, can be removed completely by the customized PPA developed in [48]. In [48], it was suggested to choose
$$Q = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix},$$
for (3.2) where the positive parameters r and s are required to satisfy $rs > \|A^TA\|$ for the purpose of ensuring the positive definiteness of Q. With this customized choice of Q, the PPA subproblem (3.2) reduces to
$$\begin{cases} \tilde{x}^k = \text{Argmin}\big\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T\lambda^k\big]\big\|^2 \,\big|\, x \in \mathcal{X}\big\}, \\ \tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A(2\tilde{x}^k - x^k) - b], \end{cases} \tag{4.8}$$
whose difficulty of implementation amounts to evaluating the resolvent operator of $\theta(x)$. Therefore, the customized PPA in [48] alleviates the difficulty of the ALM subproblem in (1.6) tangibly for some concrete applications.

Note that $\tilde{x}^k$ and $\tilde{\lambda}^k$ are both required by the scheme (4.8). Accordingly, all the coordinates of w should be regularized proximally, the entire variable w should be relaxed, and the requirement (3.4) is met by taking $H = Q$. For completeness, we list the customized PPA in [48].

Algorithm 4.2 A relaxed customized PPA for (1.2) in the primal-dual order in [48]

.

Let $\gamma \in (0,2)$ be given, the positive scalars r and s be required to satisfy $rs > \|A^T A\|$. With the initial iterate $w^0 = (x^0, \lambda^0)$, the iterate scheme is

1. **PPA step**: generate $(\tilde{x}^k, \tilde{\lambda}^k)$ via

$$\begin{cases} \tilde{x}^k = \text{Argmin}\{\theta(x) + \frac{r}{2}\|x - [x^k + \frac{1}{r} A^T\lambda^k]\|^2 \mid x \in \mathcal{X}\}, \\ \tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A(2\tilde{x}^k - x^k) - b]. \end{cases} \tag{4.9}$$

2. **Relaxation step**: generate the new iterate w^{k+1} via

$$w^{k+1} = w^k - \gamma(w^k - \tilde{w}^k). \tag{4.10}$$

4.3 A relaxed customized PPA for (1.2) in the dual-primal order

In this subsection, we continue the idea of developing a customized PPA for (1.2) such that the resulting subproblem at each iteration is of the same difficulty as evaluating the resolvent operator of $\theta(x)$. The new algorithm differs from Algorithm 4.2 in the update order of the primal and dual variables. Thus, it is a symmetric counterpart of Algorithm 4.2 by applying the same idea of algorithmic design.

Recall the MVI reformulation (2.1) of the model (1.2). If we choose

$$Q = \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix},$$

in (3.2), then the subproblem (3.2) is specified as

$$\begin{cases} \tilde{\lambda}^k = \lambda^k - \frac{1}{s}(Ax^k - b). \\ \tilde{x}^k = \text{Argmin}\{\theta(x) + \frac{r}{2}\|x - [x^k + \frac{1}{r} A^T(2\tilde{\lambda}^k - \lambda^k)]\|^2 \mid x \in \mathcal{X}\}, \end{cases}$$

which is of the same difficulty as (4.9) but with a different order of updating the variables. Note that with this choice of Q, it is implied that we choose $v = w$ in (3.2) and thus all the coordinates of w should be relaxed in the relaxation step. We thus derive a new relaxed customized PPA for (1.2) in the dual-primal order.

Remark 4.3. Note that the requirement (3.4) is met by choosing $H = Q$.

5 Relaxed customized PPAs for (1.3)

In this section, we investigate how to develop customized PPAs for the separable model (1.3). As we have mentioned, the ADM (1.7) is a benchmark solver for (1.3) and it is more efficient than the straightforward application of the ALM (1.6) to (1.3).

We first review the relaxed customized PPA proposed in [9]. In that paper, we demonstrate that when the partial metric proximal parameter in (3.2) is chosen as

$$Q = \begin{pmatrix} 0 & 0 \\ \beta A_2^T A_2 & -A_2^T \\ -A_2 & \frac{1}{\beta} I \end{pmatrix},$$

Algorithm 4.3 A relaxed customized PPA for (1.2) in the dual-primal order

Let $\gamma \in (0,2)$ be given, the positive scalars r and s be required to satisfy $rs > \|A^TA\|$. With the initial iterate $w^0 = (x^0, \lambda^0)$, the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\begin{cases} \tilde{\lambda}^k = \lambda^k - \frac{1}{s}(Ax^k - b). \\ \tilde{x}^k = \text{Argmin}\big\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T(2\tilde{\lambda}^k - \lambda^k)\big]\big\|^2 \,\big|\, x \in \mathcal{X}\big\}. \end{cases} \tag{4.11}$$

2. **Relaxation step**: generate the new iterate w^{k+1} via

$$w^{k+1} = w^k - \gamma(w^k - \tilde{w}^k). \tag{4.12}$$

then the decomposed subproblems are equally effective in exploiting the separable structure of (1.3), equally efficient in numerical senses and equally easy to implement as the ADM's subproblems in (1.7). Thus, with the combination of the relaxation step (3.3), the relaxed customized PPA in [9] outperforms the ADM (1.7) numerically. Note that with this customized choice of Q, it is implied that $v = (x_2, \lambda)$ in (3.2). For completeness, we list the relaxed customized PPA in [9] below as Algorithm 5.1.

Algorithm 5.1 The relaxed customized PPA in [9] for (1.3) in the $x_1 - \lambda - x_2$ order

Let $\gamma \in (0,2)$ and $\beta > 0$ be given. With the initial iterate (x_2^0, λ^0), the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\begin{cases} \tilde{x}_1^k = \text{Argmin}\big\{\theta_1(x_1) + \frac{\beta}{2}\big\|(A_1x_1 + A_2x_2^k - b) - \frac{1}{\beta}\lambda^k\big\|^2 \,\big|\, x_1 \in \mathcal{X}_1\big\}, \\ \tilde{\lambda}^k = \lambda^k - \beta(A_1\tilde{x}_1^k + A_2x_2^k - b), \\ \tilde{x}_2^k = \text{Argmin}\big\{\theta_2(x_2) + \frac{\beta}{2}\big\|(A_1\tilde{x}_1^k + A_2x_2 - b) - \frac{1}{\beta}\tilde{\lambda}^k\big\|^2 \,\big|\, x_2 \in \mathcal{X}_2\big\}. \end{cases} \tag{5.1}$$

2. **Relaxation step**: generate the new iterate v^{k+1} by

$$\begin{pmatrix} x_2^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x_2^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{5.2}$$

Remark 5.1. Note that the requirement (3.4) is met for Algorithm 5.1 by choosing

$$H = \begin{pmatrix} \beta A_2^TA_2 & -A_2^T \\ -A_2 & \frac{1}{\beta}I \end{pmatrix},$$

which is a symmetric square sub-matrix of Q. In addition, it is easy to verify that

$$H = \frac{1}{\beta}\begin{pmatrix} \beta A_2^T \\ -I \end{pmatrix}\begin{pmatrix} \beta A_2 & -I \end{pmatrix}.$$

Thus, it is positive semi-definite for any $\beta > 0$.

Algorithm 5.1 treats the variable x_1 as an intermediate variable. Thus, the first n_1 rows of the partial metric proximal matrix Q are all zero. Alternatively, we can choose the variable x_2 as the intermediate variable, i.e., in (3.2) we choose $v = (x_1, \lambda)$ and

$$Q = \begin{pmatrix} \beta A_1^T A_1 & -A_1^T \\ 0 & 0 \\ -A_1 & \frac{1}{\beta} I \end{pmatrix}.$$

We thus obtain a symmetric counterpart of of Algorithm 5.1.

Algorithm 5.2 A relaxed customized PPA for (1.3) in the $x_2 - \lambda - x_1$ order

---

Let $\gamma \in (0, 2)$ and $\beta > 0$ be given. With the initial iterate (x_1^0, λ^0), the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\begin{cases} \tilde{x}_2^k = \operatorname{Argmin}\big\{\theta_2(x_2) + \frac{\beta}{2}\big\|(A_1 x_1^k + A_2 x_2 - b) - \frac{1}{\beta}\lambda^k\big\|^2 \,\big|\, x_2 \in \mathcal{X}_2\big\}, \\ \tilde{\lambda}^k = \lambda^k - \beta(A_1 x_1^k + A_2 \tilde{x}_2^k - b), \\ \tilde{x}_1^k = \operatorname{Argmin}\big\{\theta_1(x_1) + \frac{\beta}{2}\big\|(A_1 x_1 + A_2 \tilde{x}_2^k - b) - \frac{1}{\beta}\tilde{\lambda}^k\big\|^2 \,\big|\, x_1 \in \mathcal{X}_1\big\}. \end{cases} \tag{5.3}$$

2. **Relaxation step**: generate the new iterate v^{k+1} via

$$\begin{pmatrix} x_1^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x_1^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{5.4}$$

Remark 5.2. The requirement (3.4) is met by choosing

$$H = \begin{pmatrix} \beta A_1^T A_1 & -A_1^T \\ -A_1 & \frac{1}{\beta} I \end{pmatrix}.$$

Since we have

$$\begin{pmatrix} \beta A_1^T A_1 & -A_1^T \\ -A_1 & \frac{1}{\beta} I \end{pmatrix} = \frac{1}{\beta} \begin{pmatrix} \beta A_1^T \\ -I \end{pmatrix} \begin{pmatrix} \beta A_1 & -I \end{pmatrix},$$

the matrix H is positive semi-definite for any $\beta > 0$.

6 Linearized versions of Algorithms 5.1 and 5.2

The proposed Algorithms 5.1 and 5.2 are both customized PPAs with the consideration of the separable structure in the model (1.3), and they are more scalable than the ADM in the sense that their subproblems (5.1) and (5.3) are of the same difficulty as the ADM subproblem (1.7) while their relaxation steps may lead to numerical acceleration. The essential subproblems arising in both Algorithms 5.1 and 5.2 (also the ADM (1.7)) are in the form of

$$\min\{\theta_i(x_i) + \frac{\beta}{2}\|A_i x_i - a\|^2 \mid x_i \in \mathcal{X}_i\}, \; i = 1, 2, \tag{6.1}$$

where $a \in \Re^m$ is a given vector. Thus, the difficulty of implementing Algorithms 5.1 and 5.2 is determined by the difficulty of evaluating $(\frac{1}{\beta}\partial\theta_i + A_i^T A_i)^{-1}$ with $i = 1, 2$.

In this and next sections, inspired mainly by a wide range of concrete applications in the fields such as statistical learning, image processing and numerical linear algebra, we discuss some special scenarios of (1.3) and investigate how to develop more customized PPAs than Algorithms 5.1 and 5.2 in accordance with the specificity of these special scenarios.

This section focuses on the case where one of the functions (say $\theta_1(x_1)$) in the objective of (1.3) is hard in the sense that its corresponding task of (6.1) with $i = 1$ has no closed-form solution whenever A_1 is not the identity matrix (e.g., $\theta_1(x_1) = \|x_1\|_1$), while the other function $\theta_2(x_2)$ is easy in the sense that its corresponding task of (6.1) with $i = 2$ is still easy even if A_2 is not the identity matrix (e.g., $\theta_2(x_2)$ is a least-square term). For this special case of (1.3), one subproblem (the x_1-related subproblem if $\theta_1(x_1)$ is the easy function) at each iteration of Algorithm 5.1 or 5.2 is hard, as it requires inner iterations to pursuit an approximate solution, and this difficulty may cause numerical inefficiency for the implementation of Algorithms 5.1 or 5.2. We now discuss how to alleviate the resulting hard subproblems under certain circumstance, and accordingly two more customized versions of Algorithms 5.1 and 5.2 are proposed for this special case of (1.3).

6.1 Linearized version of Algorithm 5.1

We first discuss the case where $\theta_1(x_1)$ is hard while $\theta_2(x_2)$ is easy in the mentioned senses. Inspired by a lot of applications, our discussion in this subsection is under the circumstance that the resolvent operator of $\theta_1(x_1)$ does admit a closed-form representation. Such an example is $\theta_1(x_1) = \|x_1\|_1$ which has popular applications in sparsity-induced optimization problems. Thus, our purpose to alleviate the difficulty associated with $\theta_1(x_1)$ is to take advantage of the availability of the closed-form representation of its resolvent operator. A natural and simple idea is to linearize the quadratic term in (6.1) such that the linearized subproblem amounts to evaluating the resolvent operator of $\theta_1(x_1)$ (thus, the closed-form solution is available), see e.g. [18, 80, 85, 89, 90]. Therefore, we can improve Algorithm 5.1 by linearizing the quadratic term of its x_1-subproblem at each iteration. Interestingly, this blend of Algorithm 5.1 and linearization inside turns out be also a special case of the relaxed PPA (3.4)-(3.3) with a customized metric proximal parameter in (3.4), as to be shown.

In fact, by linearizing the quadratic term in the objective function of the x_1-related subproblem in (5.1), Algorithm 5.1 is altered to the following iterative scheme where the x_1-subproblem simply amounts to evaluating the resolvent operator of $\theta_1(x_1)$ when $\mathcal{X}_1 = \Re^n$.

Below we show that the step (6.2) is a special case of the PPA step (3.2) with a customized partial metric proximal parameter. Hence, Algorithm 6.1 is a specified case of the relaxed PPA (3.4)-(3.3) for the model (1.3) under our aforementioned circumstance. In fact, by deriving the first-order optimality condition for the x_1-subproblem in (6.2), we have

$$\tilde{x}_1^k \in \mathcal{X}_1, \quad \theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T \{A_1^T[\beta(A_1 x_1^k + A_2 x_2^k - b) - \lambda^k] + r(\tilde{x}_1^k - x_1^k)\} \geq 0, \ \forall\, x_1 \in \mathcal{X}_1,$$

and by using the fact $\tilde{\lambda}^k = \lambda^k - \beta(A_1\tilde{x}_1^k + A_2 x_2^k - b)$, it can be written as

$$\tilde{x}_1^k \in \mathcal{X}_1, \quad \theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T \big\{-A_1^T\tilde{\lambda}^k + (rI - \beta A_1^T A_1)(\tilde{x}_1^k - x^k)\big\} \geq 0, \ \forall\, x_1 \in \mathcal{X}_1. \quad (6.4)$$

Similarly, by deriving the first-order optimality condition for the x_2-subproblem in (6.2), we obtain

$$\tilde{x}_2^k \in \mathcal{X}_2, \quad \theta_2(x_2) - \theta_2(\tilde{x}_2^k) + (x_2 - \tilde{x}_2^k)^T \{A_2^T[\beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) - \tilde{\lambda}^k]\} \geq 0, \quad \forall\, x_2 \in \mathcal{X}_2.$$

Substituting the relationship $\tilde{\lambda}^k = \lambda^k - \beta(A_1\tilde{x}_1^k + A_2 x_2^k - b)$ into the last inequality, we obtain

$$\tilde{x}_2^k \in \mathcal{X}_2, \ \theta_2(x_2) - \theta_2(\tilde{x}_2^k) + (x_2 - \tilde{x}_2^k)^T\{-A_2^T\tilde{\lambda}^k + [\beta A_2^T A_2(\tilde{x}_2^k - x_2^k) - A_2^T(\tilde{\lambda}^k - \lambda^k)]\} \geq 0, \ \forall\, x_2 \in \mathcal{X}_2. \quad (6.5)$$

Last, note that $\tilde{\lambda}^k = \lambda^k - \beta(A_1\tilde{x}_1^k + A_2 x_2^k - b)$ can be written as

$$(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) + [-A_2(\tilde{x}_2^k - x_2^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k)] = 0. \quad (6.6)$$

Algorithm 6.1 Linearized relaxed customized PPA for (1.3) in the $x_1 - \lambda - x_2$ order

Let $\gamma \in (0,2)$ and $\beta > 0$ be given, and $r \geq \beta \|A_1^T A_1\|$. With the initial iterate $(x_1^0, x_2^0, \lambda^0)$, the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\begin{cases} \tilde{x}_1^k = \text{Argmin}\big\{\theta_1(x_1) + \frac{r}{2}\big\|x_1 - [x_1^k + \frac{\beta}{r}(A_1^T(A_1x_1^k + A_2x_2^k - b) - \frac{1}{\beta}\lambda^k)]\big\|^2 \,\big|\, x_1 \in \mathcal{X}_1\big\}, \\ \tilde{\lambda}^k = \lambda^k - \beta(A_1\tilde{x}_1^k + A_2x_2^k - b), \\ \tilde{x}_2^k = \text{Argmin}\big\{\theta_2(x_2) + \frac{\beta}{2}\big\|(A_1\tilde{x}_1^k + A_2x_2 - b) - \frac{1}{\beta}\tilde{\lambda}^k\big\|^2 \,\big|\, x_2 \in \mathcal{X}_2\big\}. \end{cases} \tag{6.2}$$

2. **Relaxation step**: generate the new iterate w^{k+1} by

$$\begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k \\ x_2^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{6.3}$$

Combining (6.4), (6.5) and (6.6) together, we get

$$\tilde{w}^k \in \Omega,\ \theta_1(x_1) - \theta_1(\tilde{x}_1^k) + \theta_2(x_2) - \theta_2(\tilde{x}_2^k) + \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A_1^T\tilde{\lambda}^k \\ -A_2^T\tilde{\lambda}^k \\ A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b \end{pmatrix} + \right.$$
$$\left. + \begin{pmatrix} (rI - \beta A_1^T A_1) & 0 & 0 \\ 0 & \beta A_2^T A_2 & -A_2^T \\ 0 & -A_2 & \frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} \tilde{x}_1^k - x_1^k \\ \tilde{x}_2^k - x_2^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix} \right\} \geq 0, \quad \forall\, w \in \Omega,$$

where $\Omega = \mathcal{X}_1 \times \mathcal{X}_2 \times \Re^m$. Hence, the iterative scheme (6.2) is a special case of the PPA subproblem (3.2) with the specification in (2.2b) and the partial metric proximal parameter given by

$$Q = \begin{pmatrix} (rI - \beta A_1^T A_1) & 0 & 0 \\ 0 & \beta A_2^T A_2 & -A_2^T \\ 0 & -A_2 & \frac{1}{\beta} I \end{pmatrix}.$$

Remark 6.1. According to the scheme (6.2), all the iterates $(x_1^k, x_2^k, \lambda^k)$ are required for the $(k+1)$-th iteration. Thus, all the coordinates of w should be regularized proximally; the vector $v = w$; and we relax all the coordinates of w in the relaxation step (6.3). The requirement (3.4) thus can be met by choosing $H = Q$. Obviously, for any $\beta > 0$ and $r \geq \beta\|A_1^T A_1\|$, the matrix H is positive semi-definite.

6.2 Linearized version of Algorithm 5.2

Symmetrically, we propose the linearized version of Algorithm 5.2 under the circumstance that $\theta_1(x_1)$ is easy in the sense that the corresponding task of (6.1) with $i = 1$ has a closed-form solution even when A_1 is not an identity matrix, while $\theta_2(x_2)$ is hard in the sense that the corresponding task of (6.1) with $i = 2$ has no closed-form solution whenever A_2 is not an identity matrix. Similarly, our discussion of this subsection is under the assumption that the resolvent operator of $\theta_2(x_2)$ has a closed-form representation, and taking advantage of this fact is the main idea of alleviating the x_2-subproblem at each iteration of Algorithm 5.2. In fact, by linearizing the quadratic term of

Algorithm 6.2 Linearized relaxed customized PPA for (1.3) in the $x_2 - \lambda - x_1$ order

.

Let $\gamma \in (0,2)$ and $\beta > 0$ be given, and $s \geq \beta\|A_2^T A_2\|$. With the initial iterate $(x_1^0, x_2^0, \lambda^0)$, the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\begin{cases} \tilde{x}_2^k = \text{Argmin}\big\{\theta_2(x_2) + \frac{s}{2}\big\|x_2 - [\frac{\beta}{s}(A_2^T(A_1x_1^k + A_2x_2^k - b) - \frac{1}{\beta}\lambda^k) + x_2^k]\big\|^2 \,\big|\, x_2 \in \mathcal{X}_2\big\}, \\ \tilde{\lambda}^k = \lambda^k - \beta(A_1x_1^k + A_2\tilde{x}_2^k - b), \\ \tilde{x}_1^k = \text{Argmin}\big\{\theta_1(x_1) + \frac{\beta}{2}\big\|(A_1x_1 + A_2\tilde{x}_2^k - b) - \frac{1}{\beta}\tilde{\lambda}^k\big\|^2 \,\big|\, x_1 \in \mathcal{X}_1\big\}. \end{cases} \tag{6.7}$$

2. **Relaxation step**: generate the new iterate v^{k+1} by

$$\begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k \\ x_2^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ x_2^k - \tilde{x}_2^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{6.8}$$

---

the x_2-subproblem in (5.3), Algorithm 5.2 is altered to the following iterative scheme where the x_2-subproblem simply amounts to evaluating the resolvent operator of $\theta_2(x_2)$ when $\mathcal{X}_2 = \Re^m$.

Similar to Algorithm 6.1, it is easy to show the step (6.7) is a special case of the PPA step (3.2) with the customized metric proximal parameter

$$Q = \begin{pmatrix} \beta A_1^T A_1 & 0 & -A_1^T \\ 0 & (sI - \beta A_2^T A_2) & 0 \\ -A_1 & 0 & \frac{1}{\beta}I \end{pmatrix}.$$

Hence, Algorithm 6.2 is also a specified case of the relaxed PPA (3.2)-(3.3) for the model (1.3) under our aforementioned circumstance.

Remark 6.2. According to the scheme (6.7), all the iterates $(x_1^k, x_2^k, \lambda^k)$ are required for the $(k+1)$-th iteration. Thus, all the coordinates of w should be regularized proximally; the vector $v = w$; and we relax all the coordinates of w in the relaxation step (6.8). The requirement (3.4) thus can be met by choosing $H = Q$. Obviously, for any $\beta > 0$ and $s \geq \beta\|A_2^T A_2\|$, the matrix H is positive semi-definite.

7 Relaxed customized PPAs for (1.3) with $A_2 = -I$

In this section, we take a particular look at the special case of (1.3) with $A_2 = -I$, i.e., the model

$$\min\{\theta(x) := \theta_1(x_1) + \theta_2(x_2) \mid A_1x_1 - x_2 = b,\, x = (x_1, x_2) \in \mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2\}, \tag{7.1}$$

and we discuss how to design more customized PPAs for this special case by taking advantage of the specificity $A_2 = -I$. The reason we pay particular attention to the model (7.1) is that it is the mathematical model of many applications problems in various areas. For example, (7.1) with $b = 0$, $\mathcal{X}_1 = \Re^n$ and $\mathcal{X}_2 = \Re^m$ reduces to the model

$$\min\{\theta_1(x_1) + \theta_2(A_1x_1) \mid x_1 \in \Re^n\}, \tag{7.2}$$

where $\theta_1(x_1)$ often reflects the data-fidelity (e.g., the least-squares term), $\theta_2(A_1x_1)$ usually represents certain regularization to avoid ill-posedness (e.g., the Tikhinove regularization [78] where $\theta_2(A_1x_1) =$

$\|x_1\|^2$, or the total variational regularization [70] where $\theta_2(A_1x_1) = \|\nabla x_1\|_1$ and ∇ is the matrix representation of the discrete gradient operator). A typical application of (7.2) is the model of least-squares problem with l_1-norm regularization

$$\min\{\frac{1}{2}\|Ax_1 - b\|^2 + \tau\|x_1\|_1 \mid x_1 \in \Re^n\}, \tag{7.3}$$

where $\tau > 0$ is a trade-off parameter between the data-fidelity and regularization terms. The model (7.3) can be illustrated as finding a sparse solution of an underdetermined system of linear equations (when $m \ll n$), and it has a lot of applications such as the the unconstrained model of the basis pursuit problem [21].

The proposed Algorithms 6.1 and 6.2 are conceptually applicable for the model (7.1). In fact, because of the simplicity of A_2, the difficulty of implementing Algorithm 6.1 is equivalent to evaluating the resolvent operators of θ_1 and θ_2, while that of implementing Algorithm 6.2 is equivalent to evaluating $(\frac{1}{\beta}\partial\theta_1 + A_1^TA_1)^{-1}$ and the resolvent operator of θ_2. Recall that the task of $(\frac{1}{\beta}\partial\theta_1 + A_1^TA_1)^{-1}$ is assumed to be easy in Section 6.2.

Our purpose in this subsection is to show that: 1) a more customized PPA can be developed for (7.1) whose subproblems are of the same difficulty as those of Algorithm 6.1, while only the coordinates (x_1, λ) need to be proximally regularized and relaxed; and 2) without the assumption on the easiness of evaluating $(\frac{1}{\beta}\partial\theta_1 + A_1^TA_1)^{-1}$, we can alleviate the x_1-subproblem in Algorithm 6.2 by linearization and develop a relaxed version of Algorithm 6.2 such that the difficulty of the resulting subproblems amounts only to evaluating the resolvent operators of θ_1 and θ_2. The relaxed version of Algorithm 6.2 also excludes the coordinate of x_2 from the consideration of proximal regularization and relaxation. We achieve these goals by taking full advantage of the specificity $A_2 = -I$ in the model (7.1).

Recall the MVI reformulation (2.2) of the model (1.3). With $A_2 = -I$, then we can rewrite the MVI reformulation of (7.1) as finding $w^* = (x_1^*, x_2^*, \lambda^*)$ such that

$$w^* \in \Omega,\ \theta_1(x_1) - \theta_1(x_1^*) + \theta_2(x_2) - \theta_2(x_2^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega, \tag{7.4a}$$

where

$$w = \begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix}, \qquad F(w) = \begin{pmatrix} -A_1^T\lambda \\ \lambda \\ A_1x_1 - x_2 - b \end{pmatrix} \text{ and } \Omega = \mathcal{X}_1 \times \mathcal{X}_2 \times \Re^m. \tag{7.4b}$$

That is, (7.1) can be reformulated as a special case of the MVI (1.1) with $u = (x_1, x_2)$, $\theta(u) = \theta_1(x_1) + \theta_2(x_2)$, $w = (x_1, x_2, \lambda)$, F and Ω being given in (7.4b).

7.1 A customized PPA for (7.1) in the $x_1 - x_2 - \lambda$ order

In this subsection, we aim at presenting a customized PPA for (7.1) whose implementation is of the same difficulty as that of Algorithm 6.1 (i.e., evaluating the resolvent operators of θ_1 and θ_2), while only the coordinates (x_1, λ) need to be proximally regularized and relaxed. Thus, the new algorithm can be regarded as an improved version of Algorithm 6.1 with the customized consideration of the fact $A_2 = -I$ in (7.1).

In fact, by choosing the metric proximal parameter

$$Q = \begin{pmatrix} rI_n & A_1^T \\ 0 & 0 \\ A_1 & sI_m \end{pmatrix}$$

in (3.2) where the positive scalars r and s are required to satisfy $rs \geq \|A_1^T A_1\|$ in order to ensure the positive semi-definiteness, and recalling the specification (7.4) of the MVI reformulation, we easily conclude that the PPA subproblem (3.2) becomes

$$\theta(u)-\theta(\tilde{u}^k)+\begin{pmatrix} x_1-\tilde{x}_1^k \\ x_2-\tilde{x}_2^k \\ \lambda-\tilde{\lambda}^k \end{pmatrix}^T \left\{\begin{pmatrix} -A_1^T\tilde{\lambda}^k \\ \tilde{\lambda}^k \\ A_1\tilde{x}_1^k-\tilde{x}_2^k-b \end{pmatrix}+\begin{pmatrix} r(\tilde{x}_1^k-x_1^k)+A_1^T(\tilde{\lambda}^k-\lambda^k) \\ 0 \\ A_1(\tilde{x}_1^k-x_1^k)+s(\tilde{\lambda}^k-\lambda^k) \end{pmatrix}\right\} \geq 0, \forall w\in\Omega,$$

and it can be decomposed into

$$\tilde{x}_1^k \in \mathcal{X}_1,\ \ \theta_1(x_1)-\theta_1(\tilde{x}_1^k)+(x_1-\tilde{x}_1^k)^T\{[r(\tilde{x}_1^k-x_1^k)-A_1^T\lambda^k]\}\geq 0,\ \forall\, x_1\in\mathcal{X}_1; \tag{7.5}$$

$$\tilde{x}_2^k \in \mathcal{X}_2,\quad \theta_2(x_2)-\theta_2(\tilde{x}_2^k)+(x_2-\tilde{x}_2^k)^T(\tilde{\lambda}^k)\geq 0,\quad \forall\, x_2\in\mathcal{X}_2; \tag{7.6}$$

$$(A_1\tilde{x}_1^k-\tilde{x}_2^k-b)+[A_1(\tilde{x}_1^k-x_1^k)+s(\tilde{\lambda}^k-\lambda^k)]=0. \tag{7.7}$$

Note that (7.5) is equivalent to

$$\tilde{x}_1^k = \text{Argmin}\big\{\theta_1(x_1)+\frac{r}{2}\big\|x_1-\big[x_1^k+\frac{1}{r}A_1^T\lambda^k\big]\big\|^2 \,\big|\, x_1\in\mathcal{X}_1\big\}, \tag{7.8}$$

and this procedure can be executed only with the last iterative (x_1^k, λ^k). Thus, the new algorithm can start the iteration from (7.8) to obtain $\tilde{x}_1^k$. Then, notice that (7.7) implies that

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A_1(2\tilde{x}_1^k-x_1^k)-\tilde{x}_2^k-b]. \tag{7.9}$$

Substituting this equation into (7.6), we obtain

$$\tilde{x}_2^k\in\mathcal{X}_2,\quad \theta_2(x_2)-\theta_2(\tilde{x}_2^k)+(x_2-\tilde{x}_2^k)^T\big\{\frac{1}{s}[\tilde{x}_2^k-\big(A_1(2\tilde{x}_1^k-x_1^k)-b\big)]+\lambda^k\big\}\geq 0,\quad \forall\, x_2\in\mathcal{X}_2,$$

which can be equivalently expressed as

$$\tilde{x}_2^k = \text{Argmin}\big\{\theta_2(x_2)+\frac{1}{2s}\big\|x_2-\big(A_1(2\tilde{x}_1^k-x_1^k)-b\big)+s\lambda^k\big\|^2 \,\big|\, x_2\in\mathcal{X}_2\big\},\} \tag{7.10}$$

With the solved $\tilde{x}_1^k$ via (7.8), the procedure (7.10) can be executed to obtain $\tilde{x}_2^k$. Finally, the variable $\tilde{\lambda}^k$ can be updated via (7.9) with the updated $\tilde{x}_1^k$ and $\tilde{x}_2^k$.

Overall, the new algorithm starts the new iteration following the $x_1-x_2-\lambda$ order, and then a customized PPA for (7.1) whose subproblems are of the difficulty of evaluating the resolvent operators of θ_1 and θ_2 can be presented as in Algorithm 7.1.

Remark 7.1. Note that the second block row of the matrix Q for Algorithm 7.1 is zero. Therefore, the iteration of Algorithm 7.1 only requires (x_1,λ), and x_2 is an intermediate vector. Hence, only the essential coordinates $v=(x_1,\lambda)$ are proximally regularized in (7.11) and relaxed in (7.12). Accordingly, it is easy to verify that the requirement (3.4) is met by choosing

$$H=\begin{pmatrix} rI_n & A_1^T \\ A_1 & sI_m \end{pmatrix},$$

whose positive semi-definiteness is guaranteed whenever $rs\geq\|A_1^TA_1\|$.

Algorithm 7.1 A relaxed customized PPA for (7.1) in the $x_1 - x_2 - \lambda$ order

Let $\gamma \in (0, 2)$ be given, the positive scalars r and s be required to satisfy $rs \geq \|A_1^T A_1\|$. With the initial iterate $w^0 = (x_1^0, \lambda^0)$, the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\begin{cases} \tilde{x}_1^k = \text{Argmin}\big\{\theta_1(x_1) + \frac{r}{2}\big\|x_1 - \big[x_1^k + \frac{1}{r}A_1^T\lambda^k\big]\big\|^2 \,\big|\, x_1 \in \mathcal{X}_1\big\}, \\ \tilde{x}_2^k = \text{Argmin}\big\{\theta_2(x_2) + \frac{1}{2s}\big\|x_2 - \big(A_1(2\tilde{x}_1^k - x_1^k) - b\big) + s\lambda^k\big\|^2 \,\big|\, x_2 \in \mathcal{X}_2\big\}, \} \\ \tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A_1(2\tilde{x}_1^k - x_1^k) - \tilde{x}_2^k - b]. \end{cases} \tag{7.11}$$

2. **Relaxation step**: generate the new iterate v^{k+1} by

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{7.12}$$

Remark 7.2. Applying the ADM (1.7) directly to the model (7.1), we obtain the iterative scheme

$$\begin{cases} x_1^{k+1} = \arg\min\{\theta_1(x_1) - (\lambda^k)^T(Ax_1 - x_2^k - b) + \frac{\beta}{2}\|A_1x_1 - x_2^k - b\|^2 \mid x \in \mathcal{X}_1\}, \\ x_2^{k+1} = \arg\min\{\theta_2(x_2) - (\lambda^k)^T(A_1x_1^{k+1} - x_2 - b) + \frac{\beta}{2}\|A_1x_1^{k+1} - x_2 - b\|^2 \mid x_2 \in \mathcal{X}_2\}, \\ \lambda^{k+1} = \lambda^k - \beta(A_1x_1^{k+1} - x_2^{k+1} - b). \end{cases} \tag{7.13}$$

Therefore, the difficulty of implementing the scheme (7.13) is mainly due to the evaluation of $(\frac{1}{\beta}\partial\theta_1 + A_1^T A_1)^{-1}$. With the purpose of alleviating this difficulty to the task of evaluating only the resolvent operator of θ_1, the split inexact Uzawa method was proposed in [89, 90]. In particular, the iterative scheme of the split inexact Uzawa method is

$$\begin{cases} x_1^{k+1} = \arg\min\left\{ \begin{array}{c} \theta_1(x_1) - (\lambda^k)^T(Ax_1 - x_2^k - b) + \frac{r}{2}\|x_1 - x_1^k\|^2 \\ +(x_1 - x_1^k)^T(A_1^T\beta(A_1x_1^k - x_2^k - b)) \end{array} \middle| x_1 \in \mathcal{X}_1 \right\}, \\ x_2^{k+1} = \arg\min\{\theta_2(x_2) - (\lambda^k)^T(A_1x_1^{k+1} - x_2 - b) + \frac{\beta}{2}\|A_1x_1^{k+1} - x_2 - b\|^2 \mid x_2 \in \mathcal{X}_2\}, \\ \lambda^{k+1} = \lambda^k - \beta(A_1x_1^{k+1} - x_2^{k+1} - b), \end{cases} \tag{7.14}$$

where the parameter r is assumed to be $r > \beta\|A_1^T A_1\|$. Essentially, the split inexact Uzawa method is to linearize the quadratic term of the x_1-subproblem of the ADM (7.13) such that the difficulty of the x_1-subproblem is alleviated to evaluate only the resolvent operator of θ_1. In [50], we have shown that the split inexact Uzawa method and the ADM can be treated in a uniform way in order to establish the $O(1/t)$ convergence rate.

Compared to the schemes (7.14) and Algorithm 7.1, we find that their implementations share the same difficulty: evaluating the resolvent operators of θ_1 and θ_2 at each iteration, and they are equally effective to exploit the structure of the model (7.1).

7.2 A customized PPA for (7.1) in [75] in the $x_2 - x_1 - \lambda$ order

Recall that Algorithm 6.2 is proposed for the general case of the model (1.3) but under the assumption that it is easy to evaluate $(\frac{1}{\beta}\partial\theta_1 + A_1^T A_1)^{-1}$ even when A_1 is not an identity matrix. Considering that (7.1) is a special case of (1.3) with $A_2 = -I$, we are also interested in the case of (7.1) but without the additional assumption on the easiness of evaluating $(\frac{1}{\beta}\partial\theta_1 + A_1^T A_1)^{-1}$ when A_1 is not the identity matrix. In this subsection, we show that the algorithm in [75] can be regarded as a modified version of Algorithm 6.2, where the x_1-subproblem of Algorithm 6.2 is alleviated to the task of evaluating only the resolvent operator of $\theta_1(x_1)$.

We first show that the algorithm in [75] is recovered when we choose the customized metric proximal parameter Q as

$$Q = \begin{pmatrix} rI_n & -A_1^T \\ 0 & 0 \\ -A_1 & sI_m \end{pmatrix}$$

in (3.2) where the positive scalars r and s are required to satisfy $rs > \|A_1^T A_1\|$ in order to ensure the positive semi-definiteness. In fact, recall the specification (7.4) of the MVI reformulation of (7.1), we easily conclude that the PPA subproblem (3.2) becomes

$$\theta(u)-\theta(\tilde{u}^k)+\begin{pmatrix} x_1-\tilde{x}_1^k \\ x_2-\tilde{x}_2^k \\ \lambda-\tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A_1^T\tilde{\lambda}^k \\ \tilde{\lambda}^k \\ A_1\tilde{x}_1^k-\tilde{x}_2^k-b \end{pmatrix} + \begin{pmatrix} r(\tilde{x}_1^k-x_1^k)-A_1^T(\tilde{\lambda}^k-\lambda^k) \\ 0 \\ -A_1(\tilde{x}_1^k-x^k)+s(\tilde{\lambda}^k-\lambda^k) \end{pmatrix} \right\} \geq 0, \forall\, w \in \Omega,$$

which can be decomposed into

$$\tilde{x}_1^k \in \mathcal{X}_1,\;\; \theta_1(x_1)-\theta_1(\tilde{x}_1^k)+(x_1-\tilde{x}_1^k)^T\{-A_1^T\tilde{\lambda}^k+r(\tilde{x}_1^k-x_1^k)-A_1^T(\tilde{\lambda}^k-\lambda^k)\} \geq 0,\;\forall\, x_1 \in \mathcal{X}_1; \quad (7.15)$$

$$\tilde{x}_2^k \in \mathcal{X}_2, \quad \theta_2(x_2)-\theta_2(\tilde{x}_2^k)+(x_2-\tilde{x}_2^k)^T(\tilde{\lambda}^k) \geq 0, \quad \forall\, x_2 \in \mathcal{X}_2; \quad (7.16)$$

$$(A_1x_1^k-\tilde{x}_2^k-b)+s(\tilde{\lambda}^k-\lambda^k)=0. \quad (7.17)$$

So, (7.17) implies that

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A_1x_1^k-\tilde{x}_2^k-b]\}. \quad (7.18)$$

Substituting (7.18) into (7.16), we have

$$\tilde{x}_2^k \in \mathcal{X}_2, \quad \theta_2(x_2)-\theta_2(\tilde{x}_2^k)+(x_2-\tilde{x}_2^k)^T(\lambda^k-\frac{1}{s}[A_1x_1^k-\tilde{x}_2^k-b]) \geq 0, \quad \forall\, x_2 \in \mathcal{X}_2,$$

which equivalently means that

$$\tilde{x}_2^k = \text{Argmin}\big\{\theta_2(x_2)+\frac{1}{2s}\big\|x_2-\big(A_1x_1^k-b\big)+s\lambda^k\big\|^2 \,\big|\, x_2 \in \mathcal{X}_2\big\}, \quad (7.19)$$

Note that the procedure (7.18) only depends on the last iterate (x_1^k, λ^k). Thus, we can start the iteration from (7.18) to obtain $\tilde{x}_2^k$, then we get $\tilde{\lambda}^k$ based on (7.18), and finally obtain $\tilde{x}_1^k$ via (7.15) which can be rewritten as

$$\tilde{x}_1^k = \text{Argmin}\big\{\theta_1(x_1)+\frac{r}{2}\big\|x_1-\big[x_1^k+\frac{1}{r}A_1^T(2\tilde{\lambda}^k-\lambda^k)\big]\big\|^2 \,\big|\, x_1 \in \mathcal{X}_1\big\}. \quad (7.20)$$

Overall, the algorithm in [75] starts the new iteration following the $x_2-\lambda-x_1$ order, and the scheme is as follows. For completeness, we give the algorithm (Algorithm 7.2) in [75].

Remark 7.3. Note that the second block row of the matrix Q for Algorithm 7.2 is also zero. Therefore, like Algorithm 7.1, the iteration of Algorithm 7.2 only requires (x_1, λ), and x_2 is an intermediate vector. Hence, only the essential coordinates $v = (x_1, \lambda)$ are proximally regularized in (7.21) and relaxed in (7.22). Accordingly, it is easy to verify that the requirement (3.4) is met by choosing

$$H = \begin{pmatrix} rI_n & -A_1^T \\ -A_1 & sI_m \end{pmatrix},$$

whose positive semi-definiteness is guaranteed whenever $rs \geq \|A_1^T A_1\|$.

Remark 7.4. Despite the different orders of updating the variables, Algorithms 7.1 and 7.2 share the same difficulty of implementation: evaluating the resolvent operators of θ_1 and θ_2 at each iteration.

Algorithm 7.2 The relaxed customized PPA for (7.1) in [75] in the $x_2 - \lambda - x_1$ order

Let $\gamma \in (0,2)$ be given, the positive scalars r and s be required to satisfy $rs \ge \|A_1^T A_1\|$. With the initial iterate $w^0 = (x_1^0, \lambda^0)$, the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\left\{ \begin{array}{l} \tilde{x}_2^k = \text{Argmin}\big\{\theta_2(x_2) + \frac{1}{2s}\big\|x_2 - \big(A_1 x_1^k - b\big) + s\lambda^k\big\|^2 \,\big|\, x_2 \in \mathcal{X}_2\big\}, \\ \tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A_1 x_1^k - \tilde{x}_2^k - b]\}, \\ \tilde{x}_1^k = \text{Argmin}\big\{\theta_1(x_1) + \frac{r}{2}\big\|x_1 - \big[x_1^k + \frac{1}{r}A_1^T(2\tilde{\lambda}^k - \lambda^k)\big]\big\|^2 \,\big|\, x_1 \in \mathcal{X}_1\big\}. \end{array} \right. \tag{7.21}$$

2. **Relaxation step**: generate the new iterate v^{k+1} by

$$\begin{pmatrix} x_1^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{7.22}$$

7.3 Application of (7.1) for convex minimization with linear inclusion constraints

In this subsection, we show that the model (7.1) is a convenient reformulation of a more general model than (1.2), i.e., the convex minimization problem with linear inclusion constraints

$$\min\{\theta(x) \mid Ax \in \mathcal{B},\, x \in \mathcal{X}\}, \tag{7.23}$$

where $\mathcal{B}$ is a convex subset of $\Re^m$ and all other settings are the same as those in (1.2). Obviously, the model (7.23) reduces to the model (1.2) when the set $\mathcal{B}$ only incudes one vector b, and in particular, the model (7.23) includes as a special case the convex minimization problem with inequality constraints when $\mathcal{B} = \Re^m_+$. For many concrete applications of the model (7.23), we often encounter the case where the resolvent operator of the function $\theta(x)$ has a closed-form representation, and the sets $\mathcal{X}$ and $\mathcal{B}$ are simple in the sense that the projections onto them under the Euclidean norm are easy to compute.

By introducing an auxiliary variable $y = Ax$, the model (7.23) is converted into

$$\min\{\theta(x) \mid Ax - y = 0, x \in \mathcal{X}, y \in \mathcal{B}\}, \tag{7.24}$$

which is a special case of (7.1) with $x_1 = x$, $x_2 = y$, $\theta_1(x_1) = \theta(x)$, $\theta_2 \equiv 0$, $b = 0$, $A_1 = A$, $\mathcal{X}_1 = \mathcal{X}$ and $\mathcal{X}_2 = \mathcal{B}$. Therefore, the proposed Algorithms 7.1 and 7.2 are applicable for (7.24).

In fact, because of the specificity of the model (7.24), the applications of Algorithms 7.1 and 7.2 do not involve the auxiliary variable y, and only the original primal variable x and the dual variable λ appear during the iterations of Algorithms 7.1 and 7.2. Now we explain it. Since $\theta_2 \equiv 0$, $\mathcal{X}_2 = \mathcal{B}$ and $b = 0$, the x_2-subproblem (i.e, y) in (7.11) has a closed-form representation

$$\tilde{y}^k = P_{\mathcal{B}}[\big(A(2\tilde{x}^k - x^k)\big) - s\lambda^k]. \tag{7.25}$$

Substituting it into the λ-subproblem in (7.11) and using again $b = 0$, it becomes

$$\tilde{\lambda}^k = \frac{1}{s}\{P_{\mathcal{B}}[A(2\tilde{x}^k - x^k) - s\lambda^k] - [A(2\tilde{x}^k - x^k) - s\lambda^k]\}. \tag{7.26}$$

Note that the x_1-subproblem (i.e., x) in (7.11) does not require $\tilde{y}^k$ and y^k. Thus, for the particular case (7.23), the proposed Algorithm 7.1 reads as Algorithm 7.3. In the case $\mathcal{B} = \{b\}$, we have $P_{\mathcal{B}} = b$. In this special case, the PPA step of Algorithm 7.3 coincides with the PPA step of Algorithm of 4.2

Analogously, for the particular case (7.23), the proposed Algorithm 7.2 can be simplified as the Algorithm 7.4 without the auxiliary variable y.

Algorithm 7.3 The application of Algorithm 7.1 for (7.23)

Let $\gamma \in (0,2)$ be given, the positive scalars r and s be required to satisfy $rs \geq \|A_1^T A_1\|$. With the initial iterate $w^0 = (x^0, \lambda^0)$, the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\begin{cases} \tilde{x}^k = \text{Argmin}\{\theta(x) + \frac{r}{2}\|x - [x^k + \frac{1}{r}A^T\lambda^k]\|^2 \,|\, x \in \mathcal{X}\}, \\ \tilde{\lambda}^k = \frac{1}{s}\{P_{\mathcal{B}}[A(2\tilde{x}^k - x^k) - s\lambda^k] - [A(2\tilde{x}^k - x^k) - s\lambda^k]\}. \end{cases}$$

2. **Relaxation step**: generate the new iterate w^{k+1} by

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

Algorithm 7.4 The application of Algorithm 7.2 for (7.23)

Let $\gamma \in (0,2)$ be given, the positive scalars r and s be required to satisfy $rs \geq \|A_1^T A_1\|$. With the initial iterate $w^0 = (x^0, \lambda^0)$, the iterate scheme is

1. **PPA step**: generate $\tilde{w}^k$ via solving

$$\begin{cases} \tilde{\lambda}^k = \frac{1}{s}\{P_{\mathcal{B}}[Ax^k - s\lambda^k] - [Ax^k - s\lambda^k]\}, \\ \tilde{x}^k = \text{Argmin}\{\theta(x) + \frac{r}{2}\|x - [x^k + \frac{1}{r}A^T(2\tilde{\lambda}^k - \lambda^k)]\|^2 \,|\, x \in \mathcal{X}\}. \end{cases}$$

2. **Relaxation step**: generate the new iterate w^{k+1} by

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

8 Relaxed customized PPAs for (1.4)

In this section, we discuss how to develop customized PPAs for the model (1.4) whose MVI reformulation is given by (2.5). We show that the properties of θ_i's can be exploited individually by choosing the metric proximal parameters appropriately in (3.2), and consequently the relaxed PPA (3.2)-(3.3) can be specified into some concrete customized versions for the model (1.4). For simplicity, our discussion in this section only focuses on the circumstance where the tasks

$$\min\{\theta_i(x_i) + \frac{\beta}{2}\|A_i x_i - a\|^2 \,|\, x_i \in \mathcal{X}_i\}, \quad i = 1, \ldots, K, \tag{8.1}$$

are easy for any given $\beta > 0$ and $a \in \Re^m$.

8.1 A relaxed customized PPA for (1.4) in the primal-dual order

We first present a relaxed customized PPA for (1.4), namely, Algorithm 8.1, where the primal variables x_i's are updated prior to the dual variable λ.

Algorithm 8.1 A relaxed customized PPA for (1.4) in the primal-dual order

Let $\gamma \in (0,2)$ and $\beta > 0$ be given. With the initial iterate $w^0 = (x_1^0, \cdots, x_K^0, \lambda^0)$, the iterate scheme is

1. **PPA step:** generate $\tilde{w}^k$ via solving

$$\tilde{x}_i^k = \operatorname{Argmin}\big\{\theta_i(x_i) + \frac{\beta}{2K}\big\|[A_i(Kx_i - (K-1)x_i^k) + \sum_{j=1, j\neq i}^{K} A_j x_j^k - b] - \frac{1}{\beta}\lambda^k\big\|^2 \,\big|\, x_i \in \mathcal{X}_i\big\}, \quad (8.2a)$$

where $i = 1, , \cdots, K$;

$$\tilde{\lambda}^k = \lambda^k - \beta\big(\sum_{j=1}^{K} A_j \tilde{x}_j^k - b\big); \quad (8.2b)$$

2. **Relaxation step**: generate the new iterate w^{k+1} by

$$\begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_K^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^{k} \\ \vdots \\ x_K^{k} \\ \lambda^{k} \end{pmatrix} - \gamma \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ \vdots \\ x_K^k - \tilde{x}_K^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \quad (8.3)$$

Now, we show that the step (8.2) is essentially a special case of the PPA (3.2) with a customized choice of the metric proximal parameter Q. First, by deriving the first-order optimality condition of (8.2a), we have

$$\tilde{x}_i^k \in \mathcal{X}_i, \;\; \theta_i(x_i) - \theta_i(\tilde{x}_i^k) + (x_i - \tilde{x}_i^k)^T\big\{A_i^T[\beta\big(KA_i(\tilde{x}_i^k - x_i^k) + \sum_{j=1}^{K} A_j x_j^k - b\big) - \lambda^k]\big\} \geq 0, \; \forall\, x_i \in \mathcal{X}_i,$$

for $i = 1, 2, \cdots, K$. By using (8.2b), it can be rewritten as $\tilde{x}_i^k \in \mathcal{X}_i$ and

$$\theta_i(x_i) - \theta_i(\tilde{x}_i^k) + (x_i - \tilde{x}_i^k)^T\big\{-A_i^T\tilde{\lambda}^k + \beta A_i^T[KA_i(\tilde{x}_i^k - x_i^k) - \sum_{j=1}^{K} A_j(\tilde{x}_j^k - x_j^k)]\big\} \geq 0, \; \forall\, x \in \mathcal{X}. \quad (8.4)$$

Note that the equation (8.2b) can be written as

$$(\sum_{j=1}^{K} A_j \tilde{x}_j^k - b) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) = 0. \quad (8.5)$$

Combining (8.4) and (8.5) together, we get $\tilde{w}^k = (\tilde{x}^k, \cdots, \tilde{x}_K^k, \tilde{\lambda}^k) \in \Omega$ such that

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x_1 - \tilde{x}_1^k \\ \vdots \\ x_K - \tilde{x}_K^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A_1^T \tilde{\lambda}^k \\ \vdots \\ -A_K \tilde{\lambda}^k \\ \sum_{i=1}^K A_i \tilde{x}_i^k - b \end{pmatrix} + \right.$$

$$\left. + \begin{pmatrix} K\beta A_1^T A_1(\tilde{x}_1^k - x_1^k) - \beta A_1^T \sum_{j=1}^K A_j(\tilde{x}_j^k - x_j^k) \\ \vdots \\ K\beta A_K^T A_K(\tilde{x}_K^k - x_K^k) - \beta A_K^T \sum_{j=1}^K A_j(\tilde{x}_j^k - x_j^k) \\ \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0, \quad \forall\, w \in \Omega,$$

which coincides with (3.2) with the specification given in (2.3) and (2.5b), and the metric proximal parameter is

$$Q = \beta \begin{pmatrix} KA_1^T A_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & KA_K^T A_K & 0 \\ 0 & \cdots & 0 & \frac{1}{\beta^2} I_m \end{pmatrix} - \beta \begin{pmatrix} A_1^T A_1 & \cdots & A_1^T A_K & 0 \\ \vdots & \ddots & \vdots & \vdots \\ A_K^T A_1 & \cdots & A_K^T A_K & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{8.6}$$

Obviously, the matrix

$$K \cdot \mathrm{diag}\big(A_1^T A_1, \ldots, A_K^T A_K\big) - \begin{pmatrix} A_1^T A_1 & \cdots & A_1^T A_K \\ \vdots & \ddots & \vdots \\ A_K^T A_1 & \cdots & A_K^T A_K \end{pmatrix}$$

is positive semi-definite. Hence, the matrix Q given in (8.6) is also positive semi-definite for any $\beta > 0$.

Remark 8.1. Note that all the iterate $(x_1^k, x_2^k, \cdots, x_K^k, \lambda^k)$ are required to implement Algorithm 8.1. Thus, all the coordinates of w need to be proximally regularized in (8.2) and relaxed in (8.3). Accordingly, we take $H = Q$ and the requirement (3.4) can be met.

Remark 8.2. All the x_i-subproblems of Algorithm 8.1 are in the form of (8.1), and they are eligible for parallel computation. This feature is particularly favorable when K is large and the x_i-subproblems are of the same difficulty.

8.2 A relaxed customized PPA for (1.4) in the dual-primal order

The PPA framework also enables us to solve the model (1.4) in the dual-primal order, i.e., the dual variable λ is updated prior to the primal variables x_i's. Symmetrically, we propose a relaxed customized PPA for (1.4) in the dual-primal order, whose subproblems are of the same difficulty as those of Algorithm 8.1.

By similar analysis as that in last subsection, we can easily verify that the step (8.7) essentially

Algorithm 8.2 A relaxed customized PPA for (1.4) in the dual-primal order

Let $\gamma \in (0,2)$ and $\beta > 0$ be given. With the initial iterate $w^0 = (x_1^0, \cdots, x_K^0, \lambda^0)$, the iterate scheme is

1. **PPA step:** generate $\tilde{w}^k$ via solving

$$\left\{\begin{array}{l} \tilde{\lambda}^k = \lambda^k - \beta\big(\sum_{j=1}^K A_j x_j^k - b\big), \\ \tilde{x}_i^k = \operatorname{Arg\,min}\big\{\theta_i(x_i) + \frac{\beta}{2K}\big\|KA_i(x_i - x_i^k) - \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k)\big\|^2 \,\big|\, x_i \in \mathcal{X}_i\big\}, \\ \quad \text{where } i = 1, , \cdots, K; \end{array}\right. \tag{8.7}$$

2. **Relaxation step**: generate the new iterate w^{k+1} by

$$\begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_K^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k \\ \vdots \\ x_K^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x_1^k - \tilde{x}_1^k \\ \vdots \\ x_K^k - \tilde{x}_K^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{8.8}$$

---

is representable by the fact that $\tilde{w}^k = (\tilde{x}^k, \cdots, \tilde{x}_K^k, \tilde{\lambda}^k) \in \Omega$ such that

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x_1 - \tilde{x}_1^k \\ \vdots \\ x_K - \tilde{x}_K^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A_1^T \tilde{\lambda}^k \\ \vdots \\ -A_K \tilde{\lambda}^k \\ \sum_{i=1}^K A_i \tilde{x}_i^k - b \end{pmatrix} + \right.$$
$$\left. + \begin{pmatrix} K\beta A_1^T A_1(\tilde{x}_1^k - x_1^k) - A_1^T(\tilde{\lambda}^k - \lambda^k) \\ \vdots \\ K\beta A_K^T A_K(\tilde{x}_K^k - x_K^k) - A_K^T(\tilde{\lambda}^k - \lambda^k) \\ -\sum_{j=1}^K A_j(\tilde{x}_j^k - x_j^k) + \frac{1}{\beta}(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \geq 0, \quad \forall\, w \in \Omega,$$

which coincides with (3.2) with the specification given in (2.3) and (2.5b), and the metric proximal parameter is

$$Q = \begin{pmatrix} K\beta A_1^T A_1 & 0 & \cdots & 0 & -A_1^T \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & K\beta A_K^T A_K & -A_K^T \\ -A_1 & \cdots & \cdots & -A_K & \frac{1}{\beta} I_m \end{pmatrix}. \tag{8.9}$$

Note that the matrix Q in (8.9) can be expressed as $Q = \mathcal{Q}_D^T \tilde{Q} \mathcal{Q}_D$, where

$$\tilde{Q} = \begin{pmatrix} KI_m & 0 & \cdots & 0 & -I_m \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & KI_m & -I_m \\ -I_m & \cdots & \cdots & -I_m & I_m \end{pmatrix},$$

and

$$\mathcal{Q}_D = \mathrm{diag}(\sqrt{\beta}A_1, \ldots, \sqrt{\beta}A_K, \frac{1}{\sqrt{\beta}}I_m).$$

Thus, the matrix Q given by (8.9) is positive semi-definite for any $\beta > 0$.

Remark 8.3. Similar as Algorithm 8.1, all the iterate $(x_1^k, x_2^k, \cdots, x_K^k, \lambda^k)$ are required to implement Algorithm 8.2. Thus, all the coordinates of w need to be proximally regularized in (8.7) and relaxed in (8.8). Accordingly, we take $H = Q$ and the requirement (3.4) can be met.

Remark 8.4. Similar as Algorithm 8.1, all the x_i-subproblems of Algorithm 8.2 are in the form of (8.1), and they are eligible for parallel computation.

8.3 Linearized versions of Algorithms 8.1 and 8.2

Similar as Sections 6 and 7, we can alleviate the x_i-subproblems in Algorithms 8.1 and 8.2 to the task of evaluating only the resolvent operators of θ_i's. That is, the essential subproblems can be in the form of

$$\min\{\theta_i(x_i) + \frac{\beta}{2}\|x_i - a\|^2 \,|\, x_i \in \mathcal{X}_i\}, \quad i = 1, \ldots, K, \tag{8.10}$$

where $a \in \Re^m$. Moreover, when some matrices A_i in the linear constraints are simple (e.g, I or $-I$), we can develop more customized PPAs by taking advantage of this specificity. The idea and analysis of these extensions for (1.4) are analogous to those for Algorithms 6.1-7.2, but only with more complicated notations. Thus, we omit the elaboration for the sake of succinctness.

9 Relaxed customized PPAs for (1.5)

In this section, we discuss how to design customized PPAs for the saddle-point problem (1.5). Recall the MVI reformulation (2.6) of the model (1.5). Our purpose is to exploit the properties of θ_1 and θ_2 individually, and the resulting subproblems are of the same difficulty as evaluating the resolvent operators of θ_1 and θ_2 individually.

To begin with, we propose a relaxed customized PPA for (1.5) in the $x - y$ order, in which the iterative scheme reads as in Algorithm 9.1.

Algorithm 9.1 A relaxed customized PPA for (1.5) in the $x - y$ order

---

Let $\gamma \in (0, 2)$ be given, the positive scalars r and s be required to satisfy $rs \geq \|A^T A\|$. With the initial iterate $w^0 = (x^0, y^0)$, the iterate scheme is

1. **PPA step:** generate $\tilde{w}^k$ via solving

$$\tilde{x}^k = \mathrm{Argmin}\big\{\theta_1(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T y^k\big]\big\|^2 \,\big|\, x \in \mathcal{X}\big\}, \tag{9.1a}$$

$$\tilde{y}^k = \mathrm{Argmin}\big\{\theta_2(y) + \frac{s}{2}\big\|y - [y^k - \frac{1}{s}A(2\tilde{x}^k - x^k)\big\|^2 \,\big|\, y \in \mathcal{Y}\big\}. \tag{9.1b}$$

2. **Relaxation step**: generate the new iterate w^{k+1} by

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \end{pmatrix} - \gamma \begin{pmatrix} x^k - \tilde{x}^k \\ y^k - \tilde{y}^k \end{pmatrix}. \tag{9.2}$$

We show that the step (9.1) is a special case of the PPA (3.2) with a customized metric proximal parameter. Hence, the relaxation step (9.2) is effective. In fact, by deriving the first-order optimality condition, it follows from (9.1a) that

$$\tilde{x}^k \in \mathcal{X}, \;\; \theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T\{r(\tilde{x}^k - x^k) - A^T y^k\} \geq 0, \; \forall\, x \in \mathcal{X},$$

and it can be written as

$$\tilde{x}^k \in \mathcal{X}, \;\; \theta_1(x) - \theta_1(\tilde{x}^k) + (x - \tilde{x}^k)^T\{(-A^T\tilde{y}^k) + [r(\tilde{x}^k - x^k) + A^T(\tilde{y}^k - y^k)\} \geq 0, \; \forall\, x \in \mathcal{X}. \quad (9.3)$$

Similarly, from (9.1b)), we have

$$\tilde{y}^k \in \mathcal{Y}, \quad \theta_2(y) - \theta_2(\tilde{y}^k) + (y - \tilde{y}^k)^T\{s(\tilde{y}^k - y^k) + A(2\tilde{x}^k - x^k)\} \geq 0, \quad \forall\, y \in \mathcal{Y},$$

and it can be written as

$$\tilde{y}^k \in \mathcal{Y}, \quad \theta_2(y) - \theta_2(\tilde{y}^k) + (y - \tilde{y}^k)^T\{A\tilde{x}^k + s(\tilde{y}^k - y^k) + A(\tilde{x}^k - x^k)\} \geq 0, \quad \forall\, y \in \mathcal{Y}. \qquad (9.4)$$

Combining (9.3) and (9.4) together, we get $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k) \in \Omega$ such that

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{y}^k \\ A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) + A^T(\tilde{y}^k - y^k) \\ A(\tilde{x}^k - x^k) + s(\tilde{y}^k - y^k) \end{pmatrix} \right\} \geq 0, \forall\, w \in \Omega,$$

which coincides with (3.2) with the specification given in (2.6) and the metric proximal parameter is

$$Q = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix},$$

where the positive semi-definiteness of Q is ensured by the condition $rs \geq \|A^TA\|$.

Remark 9.1. Since the $(k+1)$-th iteration of Algorithm 9.1 requires both x^k and y^k, all the coordinates of w (i.e, x and y) need to be proximally regularized in the PPA step (9.1) and relaxed in the relaxation step (9.2). Accordingly, it is easy to verify that the requirement (3.4) is met by choosing $H = Q$.

Symmetrically, we propose a relaxed customized PPA (Algorithm 9.2) for (1.5) in the $y - x$ order. With similar analysis, we can verify that the PPA step (9.5) can be representable by finding $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k) \in \Omega$ such that

$$\theta(u) - \theta(\tilde{u}^k) + \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{y}^k \\ A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) - A^T(\tilde{y}^k - y^k) \\ -A(\tilde{x}^k - x^k) + s(\tilde{y}^k - y^k) \end{pmatrix} \right\} \geq 0, \forall\, w \in \Omega,$$

which coincides with (3.2) with the specification given in (2.6) and the metric proximal parameter is

$$Q = \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix},$$

where the positive semi-definiteness of Q is ensured by the condition $rs \geq \|A^TA\|$.

10 The global convergence

In this section, we establish the global convergence uniformly for all the proposed algorithms in the context of the uniform form (3.2)-(3.3). Our coming analysis uses the notation $\|v\|_H$ to denote $\sqrt{v^T H v}$ for notational convenience even though H may only be positive semi-definite, and we still

Algorithm 9.2 A relaxed customized PPA for (1.5) with the $y-x$ order

Let $\gamma \in (0,2)$ be given, the positive scalars r and s be required to satisfy $rs \geq \|A^TA\|$. With the initial iterate $w^0 = (x^0, y^0)$, the iterate scheme is

1. **PPA step:** generate $\tilde{w}^k$ via solving

$$\tilde{y}^k = \text{Argmin}\big\{\theta_2(y) + \frac{s}{2}\big\|y - [y^k - \frac{1}{s}Ax^k\big\|^2 \,\big|\, y \in \mathcal{Y}\big\}, \tag{9.5a}$$

$$\tilde{x}^k = \text{Argmin}\big\{\theta_1(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T(2\tilde{y}^k - y^k)\big]\big\|^2 \,\big|\, x \in \mathcal{X}\big\}. \tag{9.5b}$$

2. **Relaxation step**: generate the new iterate w^{k+1} by

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \end{pmatrix} - \gamma \begin{pmatrix} x^k - \tilde{x}^k \\ y^k - \tilde{y}^k \end{pmatrix}. \tag{9.6}$$

call it the H-norm by this slight abuse of notation. The notations u, v, w, θ, F, Ω, $\mathcal{V}^*$, Q and H are defined as before.

The proof follows the framework of contraction type methods (see [2] for the definition of contraction methods). That is, we show that the sequence $\{v^k\}$ generated by the conceptual algorithm (3.2)-(3.3) is contractive with respect to the set $\mathcal{V}^*$. Recall that for a given proposed algorithm, v denotes the set of all essential coordinates which are truly required by the iteration; and all the proposed algorithms are concrete cases of the conceptual algorithm (3.2)-(3.3) with specified u, v, w, θ, F, Ω, Q and H.

We first prove an inequality in Lemma 10.1 which is useful for establishing the global convergence.

Lemma 10.1. *The sequence $\{v^k\}$ generated by the scheme (3.2)-(3.3) satisfies*

$$(v^k - v^*)^T H(v^k - \tilde{v}^k) \geq \|v^k - \tilde{v}^k\|_H^2, \quad \forall\, v^* \in \mathcal{V}^*. \tag{10.1}$$

Proof. Let $w^* \in \Omega^*$. Note that $w^* \in \Omega$. By setting $w = w^*$ in (3.2), we get

$$(\tilde{w}^k - w^*)^T Q(v^k - \tilde{v}^k) \geq (\tilde{w}^k - w^*)^T F(\tilde{w}^k) + \theta(\tilde{u}^k) - \theta(u^*) \geq 0, \;\; \forall w^* \in \Omega^*.$$

By using the monotonicity of F and the fact that $w^* \in \Omega^*$, we obtain

$$(\tilde{w}^k - w^*)^T F(\tilde{w}^k) + \theta(\tilde{u}^k) - \theta(u^*) \geq (\tilde{w}^k - w^*)^T F(w^*) + \theta(\tilde{u}^k) - \theta(u^*) \geq 0.$$

Consequently, we have

$$(\tilde{w}^k - w^*)^T Q(v^k - \tilde{v}^k) \geq 0, \;\; \forall w^* \in \Omega^*.$$

Finally, recall the requirement (3.4), we obtain

$$(\tilde{v}^k - v^*)^T H(v^k - \tilde{v}^k) \geq 0, \;\; \forall v^* \in \mathcal{V}^*.$$

Therefore, the assertion (10.1) follows from the above inequality and the notation $\|v\|_H := \sqrt{v^T H v}$ immediately. □

With Lemma 10.1, we are ready to show an important inequality.

Lemma 10.2. *The sequence $\{v^k\}$ generated by the scheme (3.2)-(3.3) satisfies*

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \gamma(2-\gamma)\|v^k - \tilde{v}^k\|_H^2, \; \forall v^* \in \mathcal{V}^*. \tag{10.2}$$

Proof. It follows from (3.3) that

$$\begin{aligned}\|v^k - v^*\|_H^2 - \|v^{k+1} - v^*\|_H^2 &= \|v^k - v^*\|_H^2 - \|(v^k - v^*) - \gamma(v^k - \tilde{v}^k)\|_H^2 \\ &= 2\gamma(v^k - v^*)^T H(v^k - \tilde{v}^k) - \gamma^2\|v^k - \tilde{v}^k\|_H^2, \ \forall v^* \in \mathcal{V}^*.\end{aligned}$$

Recall (10.1). It follows from the last inequality that

$$\|v^k - v^*\|_H^2 - \|v^{k+1} - v^*\|_H^2 \geq \gamma(2-\gamma)\|v^k - \tilde{v}^k\|_H^2, \ \forall v^* \in \mathcal{V}^*. \tag{10.3}$$

The assertion of the theorem follows immediately. □

Remark 10.3. According to (10.2), it is clear to require $\gamma \in (0, 2)$ in the relaxation step (3.3) for the purpose of ensuring the contraction of $\{v^k\}$ with respect to $\mathcal{V}^*$. In addition, (10.2) suggests us to avoid too small or large value of γ, as the quantity $\gamma(2-\gamma)$ attains its maximal value at $\gamma = 1$.

Now, with Lemma 10.2, we show that the relaxation step (3.3) is actually effective for the contraction purpose. In other words, this step brings the new iterate v^{k+1} closer to the set $\mathcal{V}^*$ than v^k under the H-norm, making it true that the sequence $\{v^k\}$ generated by the scheme (3.2)-(3.3) is contractive with respect to the set $\mathcal{V}^*$. Hence, the techniques for establishing convergence in [2] apply.

Theorem 10.4. *The sequence $\{v^k\}$ generated by the scheme (3.2)-(3.3) is contractive with respect to $\mathcal{V}^*$, i.e.,*

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \frac{2-\gamma}{\gamma}\|v^k - v^{k+1}\|_H^2, \quad \forall\, v^* \in \mathcal{V}^*. \tag{10.4}$$

Proof. It follows from (3.3) that $v^k - \tilde{v}^k = \frac{1}{\gamma}(v^k - v^{k+1})$. Thus, the assertion (10.4) is an immediate conclusion of (10.2) □

Since H is positive semi-definite, we denote by $H^{1/2}$ its square root. We also define

$$z = H^{1/2}v \qquad \text{and} \qquad \mathcal{Z}^* = \{z^* = H^{1/2}v^* \,|\, v^* \in \mathcal{V}^*\}.$$

It is clear that $\mathcal{Z}^*$ is convex. By using the above definitions, the PPA step (3.2) can be rewritten as

$$\tilde{w}^k \in \mathcal{W}, \quad \theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (v - \tilde{v}^k)^T H^{1/2}(z^k - \tilde{z}^k), \quad \forall\, w \in \mathcal{W}, \tag{10.5}$$

and the inequality (10.2) can be restated as

$$\|z^{k+1} - z^*\|^2 \leq \|z^k - z^*\|^2 - \gamma(2-\gamma)\|z^k - \tilde{z}^k\|^2, \quad \forall\, z^* \in \mathcal{Z}^*. \tag{10.6}$$

Now, we are ready to establish the global convergence for the scheme (3.2)-(3.3).

Theorem 10.5. *Let the sequence $\{v^k\}$ be generated by the scheme (3.2)-(3.3). Then, there exists $z^\infty \in \mathcal{Z}^*$ such that*

$$\lim_{k\to\infty} z^k = z^\infty,$$

and the related vector w^∞ is a solution point of the MVI(1.1).

Proof. First, for an arbitrarily fixed $z^* \in \mathcal{Z}^*$, it follows from (10.6) that the sequence $\{z^k\}$ is bounded. Summing the inequality (10.6) over $k = 0, 1, \ldots$, we obtain

$$\sum_{k=0}^{\infty} \gamma(2-\gamma) \cdot \|z^k - \tilde{z}^k\|^2 \leq \|z^0 - z^*\|^2,$$

and thus

$$\lim_{k\to\infty} \|z^k - \tilde{z}^k\|^2 = 0.$$

Let z^∞ be a cluster point of $\{z^k\}$, because $\lim_{k\to\infty} \|z^k - \tilde{z}^k\| = 0$, z^∞ is also a cluster point of $\{\tilde{z}^k\}$. In addition, from (10.5) we obtain

$$\lim_{k\to\infty} \{\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k)\} \geq 0, \quad \forall\, w \in \mathcal{W},$$

and therefore, any w^∞ induced by z^∞ is a solution point of the MVI(1.1). By using (10.6), we have

$$\|z^{k+1} - z^\infty\|^2 \leq \|z^k - z^\infty\|^2 - c \cdot \|z^k - \tilde{z}^k\|^2,$$

and

$$\lim_{k\to\infty} z^k = z^\infty.$$

The theorem is proved. $\square$

It follows from Theorem 10.5 that the sequences $\{z^k\}$ and $\{\tilde{z}^k\}$ converge to a unique point $z^* \in \mathcal{Z}^*$. Since $\lim_{k\to\infty} z^k - \tilde{z}^k = 0$, substituting into (10.5) yields that

$$\lim_{k\to\infty} \{\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k)\} \geq 0, \quad \forall\, w \in \mathcal{W}.$$

Therefore, any w^* induced by z^* is a solution point of the MVI (1.1).

11 The $O(1/t)$ convergence rate

In this section, we establish the $O(1/t)$ convergence rate uniformly for all the proposed algorithms in the context of the conceptual algorithm (3.2)-(3.3). As we have mentioned in Section 2.2, for this purpose, we need to show that after t iterations of the scheme (3.2)-(3.3), we can find $\tilde{w} \in \Omega$ such that

$$\sup_{w\in\mathcal{D}} \big\{\theta(\tilde{u}) - \theta(u) + (\tilde{w} - w)^T F(w)\big\} \leq \epsilon, \tag{11.1}$$

where $\epsilon = O(1/t)$ and $\mathcal{D} \subset \Omega$ is an arbitrary substantial compact set.

We first present an identity which will be often used in the proof. Since the proof is elementary, we omit it.

Lemma 11.1. *Let $H \in \Re^{l\times l}$ be positive semi-definite and we use the notation $\|v\|_H := \sqrt{v^T H v}$. Then, we have*

$$(a-b)^T H(c-d) = \frac{1}{2}\big(\|a-d\|_H^2 - \|a-c\|_H^2\big) + \frac{1}{2}\big(\|c-b\|_H^2 - \|d-b\|_H^2\big),\ \forall a,b,c,d \in \Re^l. \tag{11.2}$$

Then, we show some inequalities in the following lemmas which are useful for the analysis of convergence rate.

Lemma 11.2. *The sequence generated by the scheme (3.2)-(3.3) satisfies*

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(w) \geq (v - \tilde{v}^k)^T H(v^k - \tilde{v}^k), \quad \forall\, w \in \Omega. \tag{11.3}$$

Proof. It follows from (3.2) that

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \ge (w - \tilde{w}^k)^T Q(v^k - \tilde{v}^k), \quad \forall\, w \in \Omega. \tag{11.4}$$

Because of the monotonicity of F, we have

$$(w - \tilde{w}^k)^T F(w) \ge (w - \tilde{w}^k)^T F(\tilde{w}^k),$$

and the requirement (3.4) implies that

$$(w - \tilde{w}^k)^T Q(v^k - \tilde{v}^k) = (v - \tilde{v}^k)^T H(v^k - \tilde{v}^k).$$

Hence, the assertion (11.3) follows from (11.4) directly. □

Lemma 11.3. *The sequence generated by the scheme (3.2)-(3.3) satisfies*

$$\gamma(v - \tilde{v}^k)^T H(v^k - \tilde{v}^k) + \frac{1}{2}\big(\|v - v^k\|_H^2 - \|v - v^{k+1}\|_H^2\big) \ge \gamma(1 - \frac{\gamma}{2})\|v^k - \tilde{v}^k\|_H^2, \quad \forall v \in \mathcal{V}. \tag{11.5}$$

Proof. Recall (3.3). In order to show (11.5), we need only to prove

$$(v - \tilde{v}^k)^T H(v^k - v^{k+1}) + \frac{1}{2}\big(\|v - v^k\|_H^2 - \|v - v^{k+1}\|_H^2\big) \ge \gamma(1 - \frac{\gamma}{2})\|v^k - \tilde{v}^k\|_H^2, \quad \forall v \in \mathcal{V}. \tag{11.6}$$

For the term $(v - \tilde{v}^k)^T H(v^k - v^{k+1})$, by using the identity (11.2), we get

$$(v - \tilde{v}^k)^T H(v^k - v^{k+1}) = \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) + \frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2\big). \tag{11.7}$$

Use the fact (3.3) again for the term $\frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2\big)$, we obtain

$$\begin{aligned} \frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2\big) &= \frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 - \|(v^k - \tilde{v}^k) - \gamma(v^k - \tilde{v}^k)\|_H^2\big) \\ &= \frac{1}{2}\gamma(2 - \gamma)\|v^k - \tilde{v}^k\|_H^2. \end{aligned}$$

Substituting it into the right-hand side of (11.7), we obtain (11.6) and the lemma is proved. □

Now, with the assertions in Lemmas 11.2 and 11.3, we can show the $O(1/t)$ convergence rate for the scheme (3.2)-(3.3).

Theorem 11.4. *For any integer $t > 0$, we define*

$$\tilde{w}_t = \frac{1}{t+1}\sum_{k=0}^{t} \tilde{w}^k, \tag{11.8}$$

where $\tilde{w}^k$ $(k = 1, 2, \cdots, t)$ are generated by the scheme (3.2)-(3.3). Then, we have $\tilde{w}_t \in \Omega$ and

$$\theta(\tilde{u}_t) - \theta(u) + (\tilde{w}_t - w)^T F(w) \le \frac{1}{2\gamma(t+1)}\|v - v^0\|_H^2, \quad \forall w \in \Omega, \tag{11.9}$$

Proof. First of all, combining (11.3) and (11.5), we get

$$\theta(u) - \theta(\tilde{u}^k) + (w - \tilde{w}^k)^T F(w) + \frac{1}{2\gamma}\|v - v^k\|_H^2 \ge \frac{1}{2\gamma}\|v - v^{k+1}\|_H^2, \quad \forall\, w \in \Omega. \tag{11.10}$$

Summing the inequality (11.10) over $k = 0, 1, \ldots, t$, we obtain

$$\big((t+1)\theta(u) - \sum_{k=0}^{t} \theta(\tilde{u}^k)\big) + \Big((t+1)w - \sum_{k=0}^{t} \tilde{w}^k\Big)^T F(w) + \frac{1}{2\gamma}\|v - v^0\|_H^2 \geq 0, \quad \forall w \in \Omega.$$

It follows that

$$\big(\sum_{k=0}^{t} \frac{\theta(\tilde{u}^k)}{t+1} - \theta(u)\big) + (\tilde{w}_t - w)^T F(w) \leq \frac{1}{2\gamma(t+1)}\|v - v^0\|_H^2, \quad \forall w \in \Omega. \tag{11.11}$$

Since

$$\tilde{u}_t = \frac{1}{t+1}\sum_{k=0}^{t} \tilde{u}^k \qquad \text{and} \qquad \theta(u) \text{ is convex,}$$

we have

$$\theta(\tilde{u}_t) \leq \frac{1}{t+1}\sum_{k=0}^{t} \theta(\tilde{u}^k).$$

Substituting it into (11.11), the assertion (11.9) follows directly. □

For an arbitrary substantial set $\mathcal{D} \subset \Omega$, after t iterations of the scheme (3.2)-(3.3), we find an approximate solution $\tilde{w}_t$ given by (11.8) for the MVI (1.1) such that

$$\sup_{w \in \mathcal{D}} \{\theta(\tilde{u}_t) - \theta(u) + (\tilde{w}_t - w)^T F(w)\} \leq \epsilon,$$

where $\epsilon = \frac{D^2}{2\gamma(t+1)}$ with $D = \sup\{\|v - v^0\| \mid w \in \mathcal{D}\}$. The $O(1/t)$ convergence rate of the scheme (3.2)-(3.3), and thus all the proposed algorithms, is established in the ergodic sense.

Remark 11.5. The result (11.9) suggests us to choose aggressive values of γ which are close to 2, in order to induce a smaller right-hand side in (11.9). On the other hand, recall Remark 10.3. We thus need to take a balance between these two informative properties of γ. In practice, as shown in [9, 48, 49], we recommend $\gamma \in (1.5, 1.9)$.

12 Applications

In this paper, we focus on the theoretical elaboration on a new methodology of algorithmic design for four models (1.2)-(1.5) in abstract forms. Each of the discussed models captures many concrete applications in various areas, and customized techniques in accordance with the given application (e.g., tuning appropriately the involved parameters) might be required to achieve satisfactory numerical performance when a proposed algorithm is implemented for a particular application. In particular, the efficiency of some of the proposed algorithms have been already verified in some earlier papers, e.g., [9, 48, 49, 75]. Therefore, instead of reporting some specific numerical results in details for special applications, we here just show the wide range of possible applications of our proposed algorithms by listing an incomplete set of known applications of the models (1.2)-(1.5) arising in such fields as image processing, statistics, machine learning and numerical linear algebra. The range of applications of the models under consideration is continuing to be expanding in various fields in the literature. Note that some of the applications below fit the models (1.2)-(1.5) with matrix variables. Obviously, our analysis can be trivially extended to the cases with matrix variables.

1. Applications of (1.2). The basis pursuit problem [21], the matrix completion problem [8, 14, 20], etc.

2. Applications of (1.3). A large number of distributed optimization problems and statistical learning problems listed in [4], the total variational image restoration problems [34, 59, 60, 84], image inpainting problem in wavelet domain [19], l_1-norm compressed sensing problems [11, 12], the sparse covariation problem [23, 24, 86], semidefinite programming problems [5, 46, 83, 73, 88], the least absolute shrinkage and selection operator (Lasso) problem [76], the Dantzig Selector problem [13, 80], the robust principal component analysis models [10, 16, 62, 87], the constrained linear least-squares problem [18, 58], the inverse eigenvalue problem [79], etc.

3. Applications of (1.4). The robust principal component analysis model with incomplete and noisy observations [74], the non-negative matrix factorization and dimensionality reduction on physical space [28], the image restoration removal problem with mixed noise [52], the fused Lasso problem [77], a multi-stage stochastic programming problem [71], etc.

4. Applications of (1.5). The total variational image restoration problem in [15, 17, 35, 49, 64, 82, 89, 90, 91], partial differential equations arising from fluid dynamics or linear elasticity problems in [1, 31], Nash equilibrium problems in game theory in [55, 61], etc.

13 Conclusions

We study both the convex minimization problem with linear constraints and the saddle-point problem uniformly via their MVI reformulations, and propose a uniform methodology to design structure-exploited algorithms based on the classical PPA. Our idea is to specify the PPA with customized choices of the metric proximal parameter in accordance with their MVI reformulations. The resulting algorithms are in the decomposition nature, with the possibility of exploiting the properties/structures of considered models fully. This uniform customized PPA approach makes it extremely easy to accelerate some existing benchmark methods (e.g. the augmented Lagrangian method, the alternating direction method, the split inexact Uzawa method and a class of primal-dual methods), and to develop some customized algorithms for the considered models as well. The global convergence and the $O(1/t)$ convergence rate in the ergodic sense for this series of algorithms are established easily in a uniform way.

In our analysis, we relax the conventional assumption of positive definiteness on a metric proximal parameter in the PPA literature to only positive semi-definiteness, and relax the full proximal regularization to only partial proximal regularization, see Q in (3.2). But, we still remain the symmetry requirement on H. That is, all the specified choices of the matrix H in Sections 4-9 are forced to be symmetric. Inspired by the asymmetric proximal parameter in [42], we are interested in relaxing H to be asymmetric, and also relaxing the requirement (3.4) to identifying a possibly asymmetric square sub-matrix of Q (denoted by M) such that

$$w^T Qv = v^T Mv.$$

With these relaxed requirements on the customized choices of metric proximal parameter, a new series of algorithms based on the same idea of customizing PPA can be proposed for the models (1.2)-(1.5). Because of the relaxation of the symmetric requirement, we expect that the involved parameters may be further relaxed. In the future, we shall investigate the details of this new series of algorithms and compare them numerically with the algorithms developed in this paper.

References

[1] M. Benzi, G. H. Golub and J. Liesen, Numerical solution of saddle point problems, Acta Numerica, pp. 1-137, 2005.

[2] E. Blum and W. Oettli, Mathematische Optimierung. Grundlagen und Verfahren. Ökonometrie und Unternehmensforschung, Springer-Verlag, Berlin-Heidelberg-New York, 1975.

[3] J. F. Bonnans, J. C. Gilbert, C. Lemarechal and C. A. Sagastizaba, A family of variable-metric proximal methods, Mathematical Programming, 68(1), pp. 15-47, 1995.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning, 3, pp. 1-122, 2010.

[5] S. Boyd and L. Xiao, Least-aquares covariance matrix adjustment, SIAM Journal on Matrix Analysis and Applications, 27, pp. 532-546, 2005.

[6] J. V. Burke and M. J. Qian, A variable metric proximal point algorithm for monotone operators, SIAM Journal on Control and Optimization, 37(2), 353-375, 1998.

[7] J. V. Burke and M. J. Qian, On the superlinear convergence of the variable metric proximal point algorithm using Broyden and BFGS matrix secant updating, Mathematical Programming, 88 (1), pp.157-181, 2000.

[8] J. F. Cai, E. J. Candès and Z. W. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization., 20, pp. 1956-1982, 2010.

[9] X. J. Cai, G. Gu, B. S. He and X. M. Yuan, A relaxed customized proximal point algorithm for separable convex programming, submitted, 2011.

[10] E. J. Candès, X. D. Li, Y. Ma and J. Wright, Robust principal component analysis?, Journal of ACM., 58 (1), pp. 1-37, 2011.

[11] E. J. Candès, J. Romberg and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Transactions on Information Theory, 52, pp. 489-509, 2006.

[12] E. J. Candès and T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies, IEEE Transactions on Information Theory, 52, pp. 5406-5425, 2004.

[13] E. J. Candès, and T. Tao, The Dantzig selector: statistical estimation when p is much larger than n, Annals of Statistics, 35, pp. 2313–2351, 2007.

[14] E. J. Candès and T. Tao, The power of convex relaxation: near-optimial matrix completion, IEEE Transaction on Information Theory, 56(5), pp. 2053-2080, 2010.

[15] A. Chambolle and T. Pock, A first-order primal-dual algorithm for convex problem with applications to imaging, Journal of Mathematical Imaging and Vision, 40(1), pp. 120-145, 2011.

[16] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo and A. S. Willskyc, Rank-sparsity incoherence for matrix decomposition, SIAM Journal on Matrix Analysis and Applications, to appear.

[17] T. F. Chan, G. H. Golub and P. Mulet, A nonlinear primal dual method for total variation based image restoration, SIAM Journal on Scientific Computing, 20(6), pp. 1964-1977, 1999.

[18] R. H. Chan, M. Tao and X. M. Yuan, Linearized alternating direction method for constrained linear least-squares problem, submitted, 2011.

[19] R. H. Chan, J. F. Yang and X. M. Yuan, Alternating direction method for image inpainting in wavelet domain, SIAM Journal on Imaging Sciences, 4, pp. 807-826, 2011.

[20] C. H. Chen, B. S. He and X. M. Yuan, Matrix completion via alternating direction methods, IMA Journal of Numerical Analysis, to appear.

[21] S. Chen, D. Donoho, and M. Saunders, Atomic decomposition by basis pursuit, SIAM Journal on Scientific Computing, 20(1), pp. 33-61, 1998.

[22] L. Condat, A generic first-order primal-dualmethod for convex optimization involving Lipschitzian, proximable and linear composite terms, manuscritp, 2011.

[23] A. d'Aspremont, O. Banerjee and L. El Ghaoui, First-order methods for sparse covariance selection, SIAM Journal on Matrix Analysis and Applications, 30(1), pp. 56-66, 2008.

[24] A. Dempster, Covariance selection, Biometrics, 28, pp. 157-175, 1972.

[25] J. Douglas and H.H. Rachford, On the numerical solution of the heat conduction problem in 2 and 3 space variables, Transactions of American Mathematical Society, 82, pp. 421-439, 1956.

[26] J. Eckstein and D. P. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, Mathematical Programming, 55, pp. 293-318, 1992.

[27] E. Esser, Applications of Lagrangian-based alternating direction methods and connections to split Bregman, CAM Report 09-31, UCLA, April 2009.

[28] E. Esser, M. Möller, S. Osher, G. Sapiro and J. Xin, A convex model for non-negative matrix factorization and dimensionality reduction on physical space, manuscript, 2011.

[29] E. Esser, X. Q. Zhang and T. F. Chan, A general framework for a class of first order primal-dual algorithms for convex pptimization in imaging science, SIAM Journal on Imaging Sciences, 3, pp. 1015-1046, 2010.

[30] F. Facchinei and J.S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Vol. I., Springer Series in Operations Research. Springer Verlag, New York, 2003.

[31] M. Fortin and F. Brezzi, Mixed and Hybrid Element Methods, Springer Series in Computational Mathematics, Springer, 1991.

[32] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite-element approximations, Computer and Mathematics with Applications, 2, pp. 17-40, 1976.

[33] D. Gabay, Applications of the method of multipliers to variational inequalities, Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems, M. Fortin and R. Glowinski, eds., North Holland, Amsterdam, The Netherlands, pp. 299-331, 1983.

[34] T. Goldstein and S. Osher, The split Bregman method for l_1 regularized problems, SIAM Journal on Imaging Sciences, 2(2), pp. 323-343, 2008.

[35] D. Goldfarb and W. Yin, Second-order cone programming methods for total variation-based image restoration, SIAM Journal on Scientific Computing, 27, pp. 622-645, 2005.

[36] E. G. Gol'shtein and N. V. Tret'yakov, Modified Lagrangian in convex programming and their generalizations, Mathematical Programming Studies, 10, pp. 86-97, 1979.

[37] O. Güler, On the convergence of the proximal point algorithm for convex minimization, SIAM Journal on Optimization, 29(2), pp. 403-419, 1991.

[38] O. Güler, New proximal point algorithms for convex minimization, SIAM Journal on Optimization, 2, pp. 649-664, 1992.

[39] D. R. Han, X. M. Yuan and W. X. Zhang, An augmented-Lagrangian-based parallel splitting method for separable convex programming with applications to image processing, submitted, 2010.

[40] W. M. Han and B. D. Reddy, On the finite element method for mixed variational inequalities airising in elastoplasticity, SIAM Journal on Numerical Analysis, 32(6), pp. 1778-1807, 1995.

[41] B. S. He, On the $O(1/t)$ convergence rate of the projection and contraction methods for variational inequalities with Lipschitz continuous monotone operators, submitted, 2011.

[42] B. S. He, X. L. Fu and Z. K. Jiang, Proximal point algorithm using a linear proximal term, Journal of Optimization Theory and Applications, 141, pp. 299-319, 2009.

[43] B. S. He, M. Tao, M. H. Xu and X. M. Yuan, Alternating directions based contraction method for generally separable linearly constrained convex programming problems, Optimization, to appear.

[44] B. S. He, M. Tao and X. M. Yuan, A splitting method for separable convex programming, IMA Journal of Numerical Analysis, under revision.

[45] B. S. He, M. Tao and X. M. Yuan, Alternating direction method with Gaussian back substitution for separable convex programming, SIAM Journal on Optimization, under revision.

[46] B.S. He, M. H. Xu and X. M. Yuan, Solving large-scale least squares semidefinite programming by alternating direction methods, SIAM Journal on Matrix Analisis and Applications, 32, pp. 136-152, 2011.

[47] B. S. He and H. Yang, Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities, Operations Research Letters, 23, pp. 151-161, 1998.

[48] B. S. He and X. M. Yuan, A contraction method with implementable proximal regularization for linearly constrained convex programming, Computational Optimization and Applications, under revision.

[49] B. S. He and X. M. Yuan, Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective, SIAM Journal on Imaging Sciences, under minor revision.

[50] B. S. He and X. M. Yuan, On the $O(1/t)$ convergence rate of the alternating direction method, submitted, 2011.

[51] M. R. Hestenes, Multiplier and gradient methods, Journal of Optimization Theory and Applications, 4, pp. 302-320, 1969.

[52] Y. M. Huang, M. K. Ng and Y. W. Wen, Fast image restoration methods for impulse and Gaussian noise removal, IEEE Signal Processing Letters, 16, pp.457-460, 2009.

[53] I. V. Konnov, Combined Relaxation Methods for Variational Inequalities, Lecture Notes in Economics and Mathematical Systems, Springer, 2001.

[54] S. Kontogiorgis and R. R. Meyer, A variable-penalty alternating directions method for convex optimization, Mathematical Programming, 83, pp. 29-53, 1998.

[55] B. Lemaire, Saddle point problems in partial differential equations and applications to linear quadratic differential games, Annali della Scuola Normale Superiore Di Pisa, Class di Scienze 3^0 série, tome 27, $N^0$1, pp. 105-160, 1973.

[56] B. Martinet, Regularisation, d'inéquations variationelles par approximations succesives, Rev. Francaise d'Inform. Recherche Oper., 4, pp. 154-159, 1970.

[57] J. J. Moreau, Proximité et dualité dans un espace hilbertien, Bulletin de la Société Mathématique de France, 93, pp. 273-299, 1965.

[58] B. Morini, M. Porcelli and R. H. Chan, A reduced Newton method for constrained linear least-squares problems, Journal of Computational and Applied Mathematics, 233, pp. 2200-2212, 2010.

[59] M. K. Ng, F. Wang and X. M. Yuan, Inexact alternating direction methods for image recovery, SIAM Journal on Scientific Computing, 33(4), pp. 1643-1668, 2011.

[60] M. K. Ng, P.A. Weiss and X.M. Yuan, Solving constrained total-variation problems via alternating direction methods, SIAM Journal on Scientific Computing, 32, pp. 2710-2736, 2010.

[61] J. S. Pang and M. Fukushima, Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games, Computational Management Science, 2, pp. 21-56, 2005.

[62] Y. G. Peng, A. Ganesh, J. Wright, W. L. Xu and Y. Ma, Robust alignment by sparse and low-rank decomposition for linearly correlated images, IEEE Transactions on Pattern Analysis and Machine Intellegence, to appear.

[63] T. Pock and A. Chambolle, Diagonal preconditioning for first order primal-dual algorithms in convex optimization, manuscript, 2011.

[64] T. Pock, D. Cremers, H. Bischof and A. Chambolle, An algorithm for minimizing the Mumford-Shah functional, ICCV Proceedings, Lecture Note in Computer Science, 2009.

[65] M. J. D. Powell, A method for nonlinear constraints in minimization problems, In R. Fletcher, editor, Optimization. Academic Press, 1969.

[66] B. D. Reddy, Mixed variational inequalities arising in elastoplasticity, Nonlinear Anal., 19, pp. 1071-1089, 1992.

[67] R. T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, NJ, 1970.

[68] R.T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, Mathematics of Operations Research, 1, pp. 97-116, 1976.

[69] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM Journal on Control and Optimization, 14, pp. 877-898, 1976.

[70] L. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D, 60, pp. 227-238, 1992.

[71] A. Ruszczyński, Parallel decomposition of multistage stochastic programming problems, Mathematical Programming, 58, pp. 201-228, 1993.

[72] S. Setzer, Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage, Lecture Notes in Computer Science, 5567, pp. 464-476, 2009.

[73] J. Sun and S. Zhang, A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs, European Journal of Operational Research, 207, pp. 1210-1220, 2010.

[74] M. Tao and X. M. Yuan, Recovering low-rank and sparse components of matrices from incomplete and noisy observations, SIAM Journal on Optimization, 21, pp. 57-81, 2011.

[75] M. Tao and X. M. Yuan, A customized proximal point algorithm for separable convex minimization, manuscript, 2011.

[76] R. Tibshirani, Regression shrinkage and selection via the Lasso, Journal of the Royal Statistical Society Ser. B, 58, pp. 267-288, 1996.

[77] R. Tibshirani, M. Saunders, S. Rosset, J.Zhu and K. Knight, Sparsity and smoothness via the fused lasso, Journal of the Royal Statistical Society Ser. B, 67, pp. 91-108, 2005.

[78] A. N. Tikhonov and V.Y. Arsenin, Solutions of Ill-Posed Problems, Winston, New York, 1977.

[79] F. Tisseur and K. Meerbergen, The quadratic eigenvalue problem, SIAM Review, 43, pp. 235-286, 2011.

[80] X. F. Wang and X. M. Yuan, The linearized alternating direction method for Dantzig Selector, SIAM Journal on Scientific Computing, under revision.

[81] B. C. V$\tilde{u}$, A splitting algorithm for dual monotone inclusions involving cocoercive operators, manuscript, 2011.

[82] P. Weiss, L. Blanc-Féraud and G. Aubert, Efficient schemes for total variation minimization under constraints in image processing, SIAM Journal on Scientific Computing, 31, pp. 2047-2080, 2009.

[83] Z. Wen, W. Yin and D. Goldfarb, Alternating direction augmented Lagrangian methods for semideffinite programming, Mathematical Programming Computation, 2, pp. 203-230, 2010.

[84] Y. Xiao, J. Yang and X. M. Yuan, Fast algorithms for total variation image reconstruction from random projections, submitted, 2011.

[85] J. F. Yang and X. M. Yuan, Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization, Mathematics of Computation, to appear.

[86] X. M. Yuan, Alternating direction method of multipliers for covariance selection models, Journal of Scientific Computing, to apper.

[87] X. M. Yuan and J. Yang, Sparse and low rank sparse matrix decomposition via alternating directions method, Pacific Journal of Optimziation, to appear.

[88] S. Zhang, J. Ang and J. Sun, An alternating direction method for solving convex nonlinear semidefinite programming problem, Optimization, to appear.

[89] X. Q. Zhang, M. Burger, X. Bresson and S. Osher, Bregmanized nonlocal regularization for deconvolution and sparse reconstruction, SIAM Journal on Imaging Sciences, 3(3), pp. 253-276, 2010.

[90] X. Q. Zhang, M. Burger and S. Osher, A unified primal-dual algorithm framework based on Bregman iteration, Journal of Scientific Computing, 46(1), pp. 20-46, 2010.

[91] M. Zhu and T. Chan, An efficient primal-dual hybrid gradient algorithm for total variation image restoration. CAM Reports 08-34, UCLA, Center for Applied Math., 2008.