

Research Article

An Asymmetric Proximal Decomposition Method for Convex Programming with Linearly Coupling Constraints

Xiaoling Fu,¹ Xiangfeng Wang,² Haiyan Wang,¹ and Ying Zhai³

¹ Institute of System Engineering, Southeast University, Nanjing 210096, China

² Department of Mathematics, Nanjing University, Nanjing 210093, China

³ Department of Mathematics, Guangxi Normal University, Guilin 541004, China

Correspondence should be addressed to Xiaoling Fu, fuxlnju@hotmail.com

Received 17 November 2011; Accepted 10 January 2012

Academic Editor: Abdellah Bnouhachem

Copyright © 2012 Xiaoling Fu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The problems studied are the separable variational inequalities with linearly coupling constraints. Some existing decomposition methods are very problem specific, and the computation load is quite costly. Combining the ideas of proximal point algorithm (PPA) and augmented Lagrangian method (ALM), we propose an asymmetric proximal decomposition method (AsPDM) to solve a wide variety separable problems. By adding an auxiliary quadratic term to the general Lagrangian function, our method can take advantage of the separable feature. We also present an inexact version of AsPDM to reduce the computation load of each iteration. In the computation process, the inexact version only uses the function values. Moreover, the inexact criterion and the step size can be implemented in parallel. The convergence of the proposed method is proved, and numerical experiments are employed to show the advantage of AsPDM.

1. Introduction

The original model considered here is the convex minimization problem with linearly coupling constraints:

$$\begin{aligned} & \text{minimize } \sum_{j=1}^N \theta_j(x_j) \\ & \sum_{j=1}^N A_j x_j - b_j \geq 0 \left(\text{or } \sum_{j=1}^N A_j x_j - b_j = 0 \right), \\ & x_i \in \mathcal{X}_i, \quad i = 1, \dots, N, \end{aligned} \tag{1.1}$$

where $\mathcal{X}_i \subset \mathcal{R}^{n_i}$, A_i are given $m \times n_i$ matrixes, b_i are given m -vector, and $\theta_i : \mathcal{R}^{n_i} \rightarrow \mathcal{R}$ are the i th block convex differentiable functions for each $i = 1, \dots, N$. This special problem is called convex separable problem. Problems possessing such separable structure arise in discrete-time deterministic optimal control and in the scheduling of hydroelectric power generation [1]. Note that θ_i are differentiable, setting $\nabla\theta_i(x_i) = f_i(x_i)$; by the well-known minimum principle in nonlinear programming, it is easy to get an equivalent form of problem (1.1): find $x^* = (x_1^*, \dots, x_N^*) \in \Omega$ such that

$$(x'_i - x_i^*)^T f_i(x_i^*) \geq 0, \quad i = 1, \dots, N, \quad \forall x' \in \Omega, \quad (1.2)$$

where

$$\Omega = \left\{ (x_1, \dots, x_N) \mid \sum_{j=1}^N A_j x_j \geq b \left(\text{or } \sum_{j=1}^N A_j x_j = b \right), x_i \in \mathcal{X}_i, i = 1, \dots, N \right\}, \quad b = \sum_{j=1}^N b_j. \quad (1.3)$$

Problems of this type are called separable variational inequalities (VIs). We will utilize this equivalent formulation and provide method for solution of separable VI.

One of the best-known algorithms for solving convex programming or equivalent VI is the *proximal point algorithm* (PPA) first proposed by Martinet (see [2]) and had been studied well by Rockafellar [3, 4]. PPA and its dual version, the method of multipliers, draw on a large volume of prior work by various authors [5–9]. However, classical PPA and most of its subsequent papers cannot take advantage of the separability of the original problem, and this makes them inefficient in solving separable structure problems. One major direction of PPA's study is to develop decomposition methods for separable convex programming and VI. The motivations for decomposition techniques are splitting the problem into isolate smaller or easier subproblems and parallelizing computations on specific parallel computing device. Decomposition-type methods [10–14] for large-scale problems have been widely studied in optimization as well as in variational problems and are explicitly or implicitly derived from PPA. However, most of those methods only can solve separable problems with special equality constraints:

$$\begin{aligned} \text{Minimize} \quad & \theta(x) + \psi(y), \\ & Ax + y = 0, \\ & x \in \mathcal{X}, y \in \mathcal{Y}. \end{aligned} \quad (1.4)$$

Two very well-known methods for solving equality constrained convex problems and VI are the *augmented Lagrangian method* [15, 16] (ALM) and the *alternating direction method* (ADM) [17]. The classic ALM has been deeply studied and has many advantages over the general Lagrange methods; see [18] for more detail. However, it can not preserve separability. ADM is a different method but closely related to ALM, which essentially can preserve separability for problems with two operators ($N = 2$). Recently, *separable augmented Lagrangian method* (SALM) [19, 20] overcomes the nonseparability of ALM. For example, for solving problem

(1.1) with equality constraints, Hamdi and Mahey [19] allocated a resource quantity y_i to each block $A_i x_i - b_i$ leading (1.1) to an enlarged problem in $\prod_{j=1}^N \mathcal{X}_j \times \mathcal{R}^{mN}$

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^N \theta_j(x_j) \\ & \sum_{j=1}^N y_j = 0 \\ & A_i x_i - b_i + y_i = 0, \quad i = 1, \dots, N, \\ & x_i \in \mathcal{X}_i, \quad y_i \in \mathcal{R}^m, \quad i = 1, \dots, N. \end{aligned} \quad (1.5)$$

It is worth mentioning that (1.5) is only equivalent to problem (1.1) with equality constraints. The expression of the augmented lagrangian function of (1.5) is:

$$L(x, y, u, \tau) = \sum_{j=1}^N L_j(x_j, y_j, u_j, \tau), \quad (1.6)$$

with

$$L_j(x_j, y_j, u_j, \tau) = \theta_j(x_j) - \langle u_j, A_j x_j - b_j + y_j \rangle + \frac{\tau}{2} \|A_j x_j - b_j + y_j\|^2. \quad (1.7)$$

SALM finds a saddle point of problem (1.5) by the following stages:

- (i) $x_i^{k+1} = \arg \min_{x_i \in \mathcal{X}_i} L_i(x_i, y_i^k, u_i^k, \tau)$;
- (ii) $y_i^{k+1} = \arg \min_{\sum_{j=1}^N y_j = 0} L_i(x_i^{k+1}, y_i, u_i^k, \tau)$;
- (iii) $u_i^{k+1} = u_i^k + \tau(A_i x_i^{k+1} - b_i + y_i^{k+1})$.

Note that the process in SALM for x^{k+1} allows one to solve N subproblems in parallel. This has great practical importance from the computation point of view. In fact, SALM belongs to the family of splitting algorithms and ADM for solving special convex problem (1.4) with $\psi(y) = 0$ and

$$A = \begin{pmatrix} A_1 & & \\ & \ddots & \\ & & A_N \end{pmatrix}; \quad (1.8)$$

SALM has to introduce an additive variable y to exploit the inner separable structure of the problem, which makes the problem larger. Moreover, SALM is suitable to solve equality constraints problems and fraught with difficulties in solving inequality constraints problems.

To our best knowledge, there are few dedicated methods for solving inequality constraints problems (1.1) or VI(1.2)-(1.3), except the decomposition method proposed by Tseng [21] and the PPA-based contraction method by He et al. [22]. The decomposition method in [21] decomposes the computation of x^{k+1} at a fine level without introducing

additive variable y . But, in each iteration of this method, the minimization subproblems for x^{k+1} are dependent on the step size of multiplier, which greatly restricts the computation of subproblems. The PPA-based contraction method in [22] has a nice decomposable structure; however, it has to solve the subproblem exactly. To solve (1.1) or VI(1.2)-(1.3), motivated by PPA-based contraction method and SLAM, we propose an asymmetric proximal decomposition method (AsPDM) which can well conserve the separability feature of the problem. Besides, it does not need to introduce the resource variables y like SALM and the subproblems do not depend on the step size of multiplier. In the following, we briefly describe our method for (1.1): we add an auxiliary quadratic term to the general Lagrangian function:

$$L(x, x_i^k, \lambda) = \sum_{j=1}^N L_j(x_j, x_j^k, \lambda), \quad (1.9)$$

with

$$L_j(x_j, x_j^k, \lambda) = \theta_j(x_j) - \langle \lambda, A_j x_j - b_j \rangle + \frac{\beta_j}{2} \|x_j - x_j^k\|^2. \quad (1.10)$$

The general framework of AsPDM is as follows:

Phase I

$$\tilde{x}_i^k = \underset{x_i \in \mathcal{X}_i}{\operatorname{argmin}} L_i(x_i, x_i^k, \lambda^k); \quad (1.11)$$

Phase II

$$\begin{aligned} \tilde{\lambda}^k &= P_\Lambda \left[\lambda^k - \mu^{-1} \sum_{j=1}^N (A_j \tilde{x}_j^k - b_j) \right], \quad \Lambda = \mathcal{R}_+^m \text{ (or } \mathcal{R}^m), \\ w^{k+1} &= w^k - \alpha_k G(w^k - \tilde{w}^k), \quad w = (x, \lambda). \end{aligned} \quad (1.12)$$

Here, $\beta_i > 0$, $\mu > 0$, $\alpha_k > 0$, and G are proper chosen which will be detailed in the later sections. Note that the first phase consists of N isolate subproblems, and each involves x_i , $i = 1, \dots, N$ only, namely; it can be partitioned into N independent lower-dimension subproblems. Hence, this method can take advantage of operators' separability. Since we mainly focus on solving equivalent separable VI, hence, we present this method under VI framework and analyze its convergence in the following sections.

2. The Asymmetric Proximal Decomposition Method

2.1. Structured VI

The separable VI(1.2)-(1.3) consists of N partitioned sub-VIs. Introducing a Lagrange multiplier vector $\lambda \in \Lambda$ ($\Lambda = \mathcal{R}_+^m$ or \mathcal{R}^m) associated with the linearly coupling constraint

$\sum_{j=1}^N A_j x_j \geq b$ (or $\sum_{j=1}^N A_j x_j = b$), we equivalently formulate the separable VI(1.2)-(1.3) as an enlarged VI: find $w^* = (x^*, \lambda^*) \in \mathcal{W}$ such that

$$\begin{aligned} (x'_i - x_i^*)^T \{f_i(x_i^*) - A_i^T \lambda^*\} &\geq 0, \quad i = 1, \dots, N, \\ (\lambda' - \lambda^*)^T \left(\sum_{j=1}^N A_j x_j^* - b \right) &\geq 0, \quad \forall w' \in \mathcal{W}, \end{aligned} \quad (2.1)$$

where

$$\mathcal{W} = \mathcal{X} \times \Lambda, \quad \mathcal{X} = \prod_{j=1}^N \mathcal{X}_j. \quad (2.2)$$

VI(2.1)-(2.2) is referred as *structured variational inequality* (SVI), denoted as $\text{SVI}(\mathcal{W}, Q)$. Here,

$$Q(w) = \begin{pmatrix} f_1(x_1) - A_1^T \lambda \\ \vdots \\ f_N(x_N) - A_N^T \lambda \\ \sum_{j=1}^N A_j x_j - b \end{pmatrix}. \quad (2.3)$$

2.2. Preliminaries

We summarize some basic properties and related definitions which will be used in the following discussions.

Definition 2.1. (i) The mapping f is said to be monotone if and only if

$$(w - \tilde{w})^T (f(w) - f(\tilde{w})) \geq 0, \quad \forall w, \tilde{w}. \quad (2.4)$$

(ii) A function f is said to be Lipschitz continuous if there is a constant $L > 0$ such that

$$\|f(w) - f(\tilde{w})\| \leq L \|w - \tilde{w}\|, \quad \forall w, \tilde{w}. \quad (2.5)$$

The projection onto a closed convex set is a basic concept in this paper. Let $\mathcal{W} \subset \mathcal{R}^n$ be any closed convex set. We use $P_{\mathcal{W}}(w)$ to denote the projection of w onto \mathcal{W} under the Euclidean norm; that is,

$$P_{\mathcal{W}}(w) = \arg \min \{ \|w - w'\| \mid w' \in \mathcal{W} \}. \quad (2.6)$$

The following lemmas are useful for the convergence analysis in this paper.

Lemma 2.2. Let \mathcal{W} be a closed convex set in \mathcal{R}^n , then one has

(1)

$$(w - P_{\mathcal{W}}(w))^T (w' - P_{\mathcal{W}}(w)) \leq 0 \quad \forall w \in \mathcal{R}^n, \forall w' \in \mathcal{W}, \quad (2.7)$$

(2)

$$\|P_{\mathcal{W}}(w) - w'\|^2 \leq \|w - w'\|^2 - \|w - P_{\mathcal{W}}(w)\|^2, \quad \forall w \in \mathcal{R}^n, w' \in \mathcal{W}. \quad (2.8)$$

Proof. See [23]. □

Lemma 2.3. Let \mathcal{W} be a closed convex set in \mathcal{R}^n , then w^* is a solution of $\text{VI}(\mathcal{W}, Q)$ if and only if

$$w^* = P_{\mathcal{W}}[w^* - \beta Q(w^*)], \quad \forall \beta > 0. \quad (2.9)$$

Proof. See [10, page 267]. □

Hence, solving $\text{VI}(\mathcal{W}, Q)$ is equivalent to finding a zero point of the residue function

$$e(w, \beta) = w - P_{\mathcal{W}}[w - \beta Q(w)], \quad \beta > 0. \quad (2.10)$$

Generally, the term $\|e(w)\|$ (denotes $\|e(w, 1)\|$) is referred to as the error bound of $\text{VI}(\mathcal{W}, Q)$, since it measures the distance of u from the solution set.

2.3. The Presentation of the Exact AsPDM

In each iteration, by our proper construction, our method solves N independent sub-VIs involving each individual variable x_i only so that x_i can be obtained in parallel. In what follows, to illustrate our method's practical significance, we interpret our algorithm process as a system which has a central authority and N local administrators; each administrator attempts to unilaterally solve a certain problem under the presumption that the instructions given by the authority are parametric inputs and the responses of other administrations' actions are not available; namely, the N local administrators acts synchronously and independently once they receive the information given by the central authority. We briefly describe our method which consists of two main phases.

Phase I: For arbitrary $(x_1^k, \dots, x_N^k, \lambda^k)$ given by the central authority, each local administrator uses his own way to offer the solution (denoted as \tilde{x}_i^k) of his individual problem: find $\tilde{x}_i^k \in \mathcal{X}_i$, such that

$$(x'_i - \tilde{x}_i^k)^T \{f_i(\tilde{x}_i^k) + \beta_i(\tilde{x}_i^k - x_i^k) - A_i^T \lambda^k\} \geq 0, \quad \forall x'_i \in \mathcal{X}_i. \quad (2.11)$$

Phase II: After the N local administrators accomplish their tasks, the central authority collects the resulting $(\tilde{x}_1^k, \dots, \tilde{x}_N^k)$, moreover, corresponding $(f_1(\tilde{x}_1^k), \dots, f_N(\tilde{x}_N^k))$ which can be viewed as the feedback information from the N local administrators and sets:

$$\tilde{\lambda}^k = P_\Lambda \left[\lambda^k - \mu^{-1} \left(\sum_{j=1}^N A_j \tilde{x}_j^k - b \right) \right]. \quad (2.12)$$

Here, μ is suitably chosen by the central authority. So the central authority aims to employ this feedback information effectively to provide $(x_1^{k+1}, \dots, x_N^{k+1}, \lambda^{k+1})$ which will be beneficial for the next iteration loop. In this paper, our proposed methods will update the new iterate by the following two forms:

$$w_1^{k+1} = w^k - \alpha_k G(w^k - \tilde{w}^k), \quad (2.13)$$

or

$$w_2^{k+1} = P_{\mathcal{W}}[w^k - \alpha_k Q(\tilde{w}^k)], \quad (2.14)$$

where $\alpha_k > 0$ is a specific step size and

$$G = \begin{pmatrix} \beta_1 & & A_1^T \\ & \ddots & \vdots \\ & & \beta_N & A_N^T \\ & & & \mu I \end{pmatrix}. \quad (2.15)$$

We make the standard assumptions to guarantee that the problem under consideration is solvable and the proposed methods are well defined.

Assumption A. $f_i(x_i)$ is monotone and Lipschitz continuous, $i = 1, \dots, N$.

By this assumption, it is easy to get that $Q(w)$ is monotone.

2.4. Presentation of the Inexact AsPDM

In this section, the inexact version of the AsPDM method is present, and some remarks are briefly made.

For later analysis convenience, we denote

$$D = \begin{pmatrix} \frac{\beta_1}{2} & & & \\ & \ddots & & \\ & & \frac{\beta_N}{2} & \\ & & & \eta I \end{pmatrix}. \quad (2.16)$$

At each iteration, we solve N sub-VIs (see (2.11)) independently. No doubt, the computation load for an exact solution of (2.11) is usually expensive. Hence, it is desirable for us to consider solving (2.11) inexactly under a relative relaxed inexact criterion. We now describe and analyze our inexact method. Each iteration consists of two main phases, one of which provides an inexact solution of (2.11) and the other of which employs the inexact solution to offer a new iterate for the next iteration.

The first phase of our method works as follows. At the beginning of k th iteration, an iterate $w^k = (x^k, \lambda^k)$ is given. If $x_i^k = P_{\mathcal{X}_i}[x_i^k - (f_i(x_i^k) - A_i^T \lambda^k)]$, then x_i^k is the exact solution of i th sub-VI; there is nothing we need to do with the i th sub-VI. Otherwise, we should find $\tilde{x}_i^k \in \mathcal{X}_i$ such that

$$(x_i' - \tilde{x}_i^k)^T \left\{ f_i(\tilde{x}_i^k) + \beta_i(\tilde{x}_i^k - x_i^k) - A_i^T \lambda^k + \xi_{x_i}^k \right\} \geq 0, \quad \forall x_i' \in \mathcal{X}_i, \quad (2.17)$$

with

$$\xi_{x_i}^k = f_i(x_i^k) - f_i(\tilde{x}_i^k). \quad (2.18)$$

Here, the obtained β_i should satisfy following two inexact criteria:

$$(x_i^k - \tilde{x}_i^k)^T \xi_{x_i}^k \leq \frac{\nu \beta_i}{2} \|x_i^k - \tilde{x}_i^k\|^2, \quad \nu \in (0, 1), \quad (2.18^*a)$$

$$\|\xi_{x_i}^k\| \leq \frac{\beta_i}{\sqrt{2}} \|x_i^k - \tilde{x}_i^k\|. \quad (2.18^*b)$$

Once one of the above criteria fails to be satisfied, we will increase β_i by $\beta_i = \beta_i * 1.8$ and turn back to solve the i th sub-VI of (2.17) with this updated β_i . It should be noted that both inexact criteria are quite easy to check since they do not contain any unknown variables. In addition, another favorable characterization of these criteria is that they are independent; namely, they only involve x_i^k, \tilde{x}_i^k , irrelevant to $x_j^k, \tilde{x}_j^k (j \neq i)$.

In what follows, let us describe the second phase. We require

$$\tilde{\lambda}^k = P_{\Lambda} \left[\lambda^k - \mu^{-1} \left(\sum_{j=1}^N A_j \tilde{x}_j^k - b \right) \right], \quad (2.19)$$

where $\mu \in (0, \sum_{j=1}^N (\|A_j A_j^T\| / 2\beta_j) + \eta]$ (here, $\eta > 0$) is suitably chosen to satisfy

$$(w^k - \tilde{w}^k)^T G(w^k - \tilde{w}^k) \geq \|w^k - \tilde{w}^k\|_D^2. \quad (2.20)$$

Now we use this $\tilde{w}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ (or $Q(\tilde{w}^k)$) to construct the new iteration. Here, we provide two simple forms for the new iteration:

$$w_1^{k+1} = w^k - \alpha_k \left[G(w^k - \tilde{w}^k) - \xi^k \right], \quad \xi^k = (\xi_x^k, 0), \quad \xi_x^k = (\xi_{x_1}^k, \dots, \xi_{x_N}^k), \quad (2.21^*a)$$

or

$$w_2^{k+1} = P_{\mathcal{W}} \left[w^k - \alpha_k Q(\tilde{w}^k) \right], \quad (2.21^*b)$$

where

$$\alpha_k = \gamma \alpha_k^*, \quad \alpha_k^* = \frac{(w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k)}{\|G(w^k - \tilde{w}^k) - \xi^k\|^2}, \quad \gamma \in (0, 2). \quad (2.21)$$

In fact, each iteration of the proposed method consists of two main phases. Using the point of view that the problem is a system with a central authority and N administrators, the first phase is accomplished by N administrators based on the instruction given by the authority. That is, i th sub-VI only involves i th administrator's activities. On the other hand, the second phase is implemented by the central authority to give new instruction for the next iteration.

Remark 2.4. In the inexact AsPDM, the main task of Phase I is to find a solution for (2.17). From (2.17), it is easy to get that

$$\tilde{x}_i^k = P_{\mathcal{X}_i} \left[\tilde{x}_i^k - \left(f_i(x_i^k) + \beta_i (\tilde{x}_i^k - x_i^k) - A_i^T \lambda^k \right) \right]. \quad (2.22)$$

It seems that equality (2.22) is an implicit form since both sides of (2.22) contain \tilde{x}_i^k . In fact, we can transform equality (2.22) to an explicit form. Using the property of the projection, we have

$$\begin{aligned} \tilde{x}_i^k &= P_{\mathcal{X}_i} \left\{ \tilde{x}_i^k - \beta_i \left[(\tilde{x}_i^k - x_i^k) + \beta_i^{-1} \left(f_i(x_i^k) - A_i^T \lambda^k \right) \right] \right\} \\ &= P_{\mathcal{X}_i} \left[x_i^k - \frac{1}{\beta_i} \left(f_i(x_i^k) - A_i^T \lambda^k \right) \right]. \end{aligned} \quad (2.23)$$

Consequently, using the above formula, we can compute \tilde{x}_i^k quite easily.

Remark 2.5. Combining (2.22) and (2.19), we then find that

$$\tilde{w}^k = P_{\mathcal{W}} \left[\tilde{w}^k - \left(Q(\tilde{w}^k) + G(\tilde{w}^k - w^k) + \xi^k \right) \right]. \quad (2.24)$$

If $\xi^k = 0$, it yields an exact version. In this special case, it is clear that

$$\tilde{w}^k = P_{\mathcal{W}} \left[\tilde{w}^k - \left(Q(\tilde{w}^k) + G(\tilde{w}^k - w^k) \right) \right]. \quad (2.25)$$

We find that this formula is quite similar to the iterates produced by the classic PPA [3], which employs

$$w^{k+1} = P_{\mathcal{W}} \left[w^{k+1} - \left(Q(w^{k+1}) + S(w^{k+1} - w^k) \right) \right] \quad (2.26)$$

as the new iterate; here, S is a positive symmetry definite matrix. For deeper insight, our method does not appear fit into any of the known PPA frameworks. It is virtually not equivalent to PPA even if G is positive definite. The reason why our method can not be viewed as PPA lies in the fact that G is asymmetry, moreover, may be not positive definite. This lack of symmetry makes it fail to introduce an inner product as S . Consequently, if one sets \tilde{w}^k as the new iterate, one may fail to obtain the convergence. Due to the asymmetric feature of G , we call our method asymmetric proximal decomposition method.

Remark 2.6. Recalling that $\tilde{\lambda}^k$ is obtained by (2.19), it is easy to get that

$$\left(\lambda' - \tilde{\lambda}^k\right)^T \left\{ \tilde{\lambda}^k - \lambda^k + \mu^{-1} \left(\sum_{j=1}^N A_j \tilde{x}_j - b \right) \right\} \geq 0, \quad \forall \lambda' \in \mathcal{R}_+^m. \quad (2.27)$$

Combining (2.17) and (2.27), we have

$$\left(w' - \tilde{w}^k\right)^T \left(Q\left(\tilde{w}^k\right) + G\left(\tilde{w}^k - w^k\right) + \xi^k\right) \geq 0, \quad \forall w' \in \mathcal{W}. \quad (2.28)$$

Since $\tilde{w}^k = (\tilde{x}_1^k, \dots, \tilde{x}_N^k, \tilde{\lambda}^k) \in \mathcal{W}$ is generated by (2.17)–(2.20) from a given $(x_1^k, \dots, x_N^k, \lambda^k)$, we have that $\|w^k - \tilde{w}^k\| = 0$ implies $\|\xi^k\| = 0$ and $\|G(\tilde{w}^k - w^k) + \xi^k\| = 0$. According to (2.28), we have

$$\left(w' - \tilde{w}^k\right)^T Q\left(\tilde{w}^k\right) \geq 0 \quad \forall w' \in \mathcal{W}. \quad (2.29)$$

In other words, \tilde{w}^k is a solution of Problem (2.1)–(2.2) if $x_i^k = \tilde{x}_i^k$ ($i = 1, \dots, N$) and $\lambda^k = \tilde{\lambda}^k$. Hence, we use $\|w^k - \tilde{w}^k\| \leq \varepsilon$ as stopping criterion in the proposed method.

Remark 2.7. The update form (2.21*a) is based on the fact that $G(\tilde{w}^k - w^k) + \xi^k$ is a descent direction of the unknown distance function $(1/2)\|w - w^*\|^2$ at point w^k . This property will be proved in Section 3.1. α_k^* in (2.21) is the “optimal” step length, which will be detailed in Section 3.2. We can also use (2.21*b) to update the new iterate. For fast convergence, the practical step length should be multiplied by a relaxed factor $\gamma \in [1, 2)$.

Remark 2.8. Note that $\|w^k - \tilde{w}^k\| = 0$ if and only if $\|w^k - \tilde{w}^k\|_D = 0$. In the case $\|w^k - \tilde{w}^k\|_D \neq 0$, by choosing a suitable $\mu \in (0, \sum_{j=1}^N (\|A_j A_j^T\| / 2\beta_j) + \eta]$, (2.20) will be satisfied. We state this fact in the following lemma.

Lemma 2.9. Let G and D be defined in (2.15) and (2.16), respectively. If $\mu = \sum_{j=1}^N (\|A_j A_j^T\| / 2\beta_j) + \eta$, for all $w = (x_1, \dots, x_N, \lambda) \in \mathcal{R}^{\sum_{j=1}^N n_j + m}$, one has

$$w^T G w \geq \|w\|_D^2. \quad (2.30)$$

Proof. According to Definition (2.16), we have

$$\begin{aligned}
w^T G w &= \sum_{j=1}^N \beta_j \|x_j\|^2 + \left(\sum_{j=1}^N \frac{\|A_j A_j^T\|}{2\beta_j} + \eta \right) \|\lambda\|^2 + \sum_{j=1}^N x_j^T A_j^T \lambda \\
&= \sum_{j=1}^N \frac{\beta_j}{2} \|x_j\|^2 + \eta \|\lambda\|^2 + \frac{1}{2} \sum_{j=1}^N \left\| \sqrt{\beta_j} x_j + \frac{A_j^T \lambda}{\sqrt{\beta_j}} \right\|^2 + \sum_{j=1}^N \left(\frac{\|A_j A_j^T\| \|\lambda\|^2 - \|A_j^T \lambda\|^2}{2\beta_j} \right) \\
&\geq \|w\|_D^2,
\end{aligned} \tag{2.31}$$

and the assertion is obtained. \square

Set $w = w^k - \tilde{w}^k$ in the above lemma, we get

$$(w^k - \tilde{w}^k)^T G (w^k - \tilde{w}^k) \geq \|w^k - \tilde{w}^k\|_D^2. \tag{2.32}$$

If one chooses $\mu = \sum_{j=1}^N (\|A_j A_j^T\|/2\beta_j) + \eta$, then the Condition (2.20) is always satisfied; hence, $\sum_{j=1}^N (\|A_j A_j^T\|/2\beta_j) + \eta$ can be regarded as a safe upper bound for this condition. Note that we use an inequality in the proof of Lemma 2.9; it seems that there exists some relaxations. As a result, rather than fix $\mu = \sum_{j=1}^N (\|A_j A_j^T\|/2\beta_j) + \eta$, let μ be a smaller value, and check if Condition (2.20) is satisfied. If not, increase μ by $\mu = \min\{\sum_{j=1}^N (\|A_j A_j^T\|/2\beta_j) + \eta, 4\mu\}$ and try again. This process enables us to reach a suitable $\mu \in (0, (\sum_{j=1}^N \|A_j A_j^T\| + 1)/2]$ to meet (2.20).

Note that, in our proposed method, problems VI (2.17) produce \tilde{x}_i^k in a parallel wise. In addition, instead of taking the solution of the subproblems, the new iterate in the proposed methods is updated by a simple manipulation, for example, (2.18*a)-(2.18*b).

3. Convergence of AsPDM

In the proposed methods, the first phase (accomplished by the local administrators) offers a descent direction of the unknown distance function, and the second phase (accomplished by the central authority) determines the “optimal” step length along this direction. This section gives more theory analysis.

3.1. The Descent Direction in the Proposed AsPDM

For any $w^* \in \mathcal{W}^*$, $(w^k - w^*)$ is the gradient of the unknown distance function $(1/2)\|w - w^*\|^2$ at point $w^k \notin \mathcal{W}^*$. A direction d is called a descent direction of $(1/2)\|w - w^*\|^2$ at point w^k if and only if the inner product $\langle w^k - w^*, d \rangle < 0$. Let $\tilde{w}^k = (\tilde{x}_1^k, \dots, \tilde{x}_N^k, \tilde{\lambda}^k)$ be generated by

(2.17)–(2.20) from a given $w^k = (x_1^k, \dots, x_N^k, \lambda^k)$. A goal of this subsection is to elucidate that, for any $w^* \in \mathcal{W}^*$,

$$(w^k - w^*)^T (G(w^k - \tilde{w}^k) - \xi^k) \geq (1 - \nu) \|w^k - \tilde{w}^k\|_D^2. \quad (3.1)$$

It guarantees that $G(\tilde{w}^k - w^k) + \xi^k$ is a descent direction of $(1/2)\|w - w^*\|^2$ at point $w^k \notin \mathcal{W}^*$. The above inequality plays an important role in the convergence analysis.

Lemma 3.1. *Assume that $\tilde{w}^k = (\tilde{x}_1^k, \dots, \tilde{x}_N^k, \tilde{\lambda}^k)$ is generated by (2.17)–(2.20) from a given $w^k = (x_1^k, \dots, x_N^k, \lambda^k)$, then for any $w^* = (x_1^*, \dots, x_N^*, \lambda^*) \in \mathcal{W}^*$ one has*

$$(w^k - w^*)^T (G(w^k - \tilde{w}^k) - \xi^k) \geq (1 - \nu) \|w^k - \tilde{w}^k\|_D^2. \quad (3.2)$$

Proof. Since $w^* \in \mathcal{W}$, substituting $w' = w^*$ in (2.28), we obtain

$$(w^* - \tilde{w}^k)^T (Q(\tilde{w}^k) + G(\tilde{w}^k - w^k) + \xi^k) \geq 0. \quad (3.3)$$

Using the monotonicity of $Q(w)$ and applying with $w' = \tilde{w}^k$ in (2.1), it is easy to get

$$(\tilde{w}^k - w^*)^T Q(\tilde{w}^k) \geq (\tilde{w}^k - w^*)^T Q(w^*) \geq 0. \quad (3.4)$$

Combining (3.3) and (3.4), we then find

$$(w^k - w^*)^T (G(w^k - \tilde{w}^k) - \xi^k) \geq (w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k). \quad (3.5)$$

Note that Criterion (2.18*a) holds; we have

$$\begin{aligned} (w^k - \tilde{w}^k)^T \xi^k &\leq \nu \sum_{j=1}^N \frac{\beta_j}{2} \|x_j^k - \tilde{x}_j^k\|^2 \\ &\leq \nu \|w^k - \tilde{w}^k\|_D^2 \\ &\leq \nu (w^k - \tilde{w}^k)^T G(w^k - \tilde{w}^k). \end{aligned} \quad (3.6)$$

The last inequality follows directly from the result of Lemma 2.9. Consequently,

$$(w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k) \geq (1 - \nu) (w^k - \tilde{w}^k)^T G(w^k - \tilde{w}^k). \quad (3.7)$$

Using the preceding inequality and (2.32) in (3.5) yields

$$\left(w^k - w^*\right)^T \left(G\left(w^k - \tilde{w}^k\right) - \xi^k\right) \geq (1 - \nu) \left\|w^k - \tilde{w}^k\right\|_D^2, \quad (3.8)$$

completing the proof. \square

Now, we state the main properties of $Q(\tilde{w}^k)$ in the lemma below.

Lemma 3.2. *Let $\tilde{w}^k = (\tilde{x}_1^k, \dots, \tilde{x}_N^k, \tilde{\lambda}^k)$ be generated by (2.17)–(2.20) from given $w^k = (x_1^k, \dots, x_N^k, \lambda^k)$. Then for any $w' \in \mathcal{W}$ one has*

$$\left(w' - \tilde{w}^k\right)^T Q\left(\tilde{w}^k\right) \geq \left(w' - \tilde{w}^k\right)^T \left[G\left(w^k - \tilde{w}^k\right) - \xi^k\right]. \quad (3.9)$$

Proof. Recalling that $\tilde{w}^k = P_{\mathcal{W}}[\tilde{w}^k - (Q(\tilde{w}^k) + G(\tilde{w}^k - w^k) + \xi^k)] \in \mathcal{W}$, substituting $w = \tilde{w}^k - (Q(\tilde{w}^k) + G(\tilde{w}^k - w^k) + \xi^k)$ in inequality (2.1), we have, immediately,

$$\left(w' - \tilde{w}^k\right)^T \left\{ \tilde{w}^k - \left[\tilde{w}^k - \left(Q\left(\tilde{w}^k\right) + G\left(\tilde{w}^k - w^k\right) + \xi^k \right) \right] \right\} \geq 0. \quad (3.10)$$

By some manipulations, our assertion holds immediately. \square

3.2. The Step Size and the New Iterate

Since $G(\tilde{w}^k - w^k) + \xi^k$ is a descent direction of $(1/2)\|w - w^*\|^2$ at point w^k , the new iterate will be determined along this direction by choosing a suitable step length. In order to explain why we have the “optimal” step α_k^* as defined in (2.21), we let

$$\begin{aligned} w_1^{k+1}(\alpha) &= w^k - \alpha \left[G\left(w^k - \tilde{w}^k\right) - \xi^k \right], \\ w_2^{k+1}(\alpha) &= P_{\mathcal{W}} \left[w^k - \alpha Q\left(\tilde{w}^k\right) \right] \end{aligned} \quad (3.11)$$

be the step-size-dependent new iterate, and let

$$\Theta_{k,i}(\alpha) = \left\|w^k - w^*\right\|^2 - \left\|w_i^{k+1}(\alpha) - w^*\right\|^2, \quad i = 1, 2, \quad (3.12)$$

be the profit function of the k th iteration. Because $\Theta_{k,i}(\alpha)$ includes the unknown vector w^* , it can not be maximized directly. The following lemma offers us a lower bound of $\Theta_{k,i}(\alpha)$ which is a quadratic function of α .

Lemma 3.3. *Let $\tilde{w}^k = (\tilde{x}_1^k, \dots, \tilde{x}_N^k, \tilde{\lambda}^k)$ be generated by (2.17)–(2.20) from a given $w^k = (x_1^k, \dots, x_N^k, \lambda^k)$. Then one has*

$$\Theta_{k,i}(\alpha) \geq q_k(\alpha), \quad \forall \alpha > 0, \quad i = 1, 2, \quad (3.13)$$

where

$$q_k(\alpha) = 2\alpha(w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k) - \alpha^2 \|G(w^k - \tilde{w}^k) - \xi^k\|^2. \quad (3.14)$$

Proof. It follows from Definition (3.12) and inequality (3.5) that

$$\begin{aligned} \Theta_{k,1}(\alpha) &= \|w^k - w^*\|^2 - \|w^k - w^* - \alpha[G(w^k - \tilde{w}^k) - \xi^k]\|^2 \\ &= 2\alpha(w^k - w^*)^T [G(w^k - \tilde{w}^k) - \xi^k] - \alpha^2 \|G(w^k - \tilde{w}^k) - \xi^k\|^2 \\ &\stackrel{(3.5)}{\geq} q_k(\alpha). \end{aligned} \quad (3.15)$$

Let us deal with $\Theta_{k,2}(\alpha)$ which seems more complicated:

$$\begin{aligned} \Theta_{k,2}(\alpha) &\stackrel{(2.8)}{\geq} \|w^k - w^*\|^2 - \|w^k - w^* - \alpha_k Q(\tilde{w}^k)\|^2 + \|w^k - w^{k+1} - \alpha_k Q(\tilde{w}^k)\|^2 \\ &= 2\alpha(w^{k+1} - w^*)^T Q(\tilde{w}^k) + \|w^k - w^{k+1}\|^2 \\ &\geq 2\alpha(w^{k+1} - \tilde{w}^k)^T Q(\tilde{w}^k) + \|w^k - w^{k+1}\|^2 \\ &\stackrel{(3.9)}{\geq} \|w^k - w^{k+1}\|^2 + 2\alpha(w^{k+1} - \tilde{w}^k)^T [G(w^k - \tilde{w}^k) - \xi^k] \\ &= \|w^k - w^{k+1} - \alpha[G(w^k - \tilde{w}^k) - \xi^k]\|^2 + 2\alpha(w^k - \tilde{w}^k)^T [G(w^k - \tilde{w}^k) - \xi^k] \\ &\quad - \alpha^2 \|G(w^k - \tilde{w}^k) - \xi^k\|^2 \\ &\geq 2\alpha(w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k) - \alpha^2 \|G(w^k - \tilde{w}^k) - \xi^k\|^2. \end{aligned} \quad (3.16)$$

Since $q_k(\alpha)$ is a quadratic function of α , it reaches its maximum at

$$\alpha_k^* = \frac{(w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k)}{\|G(w^k - \tilde{w}^k) - \xi^k\|^2}; \quad (3.17)$$

this is just the same defined in (2.21). In practical computation, taking a relaxed factor γ is wise for fast convergence. Note that for any $\alpha_k = \gamma \alpha_k^*$, it follows from (3.13), (3.14), and (2.21) that

$$\begin{aligned} \Theta_{k,i}(\gamma \alpha_k^*) &\geq q_k(\gamma \alpha_k^*) \\ &= 2\gamma \alpha_k^* (w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k) - \gamma^2 (\alpha_k^*)^2 (\alpha_k^* \|G(w^k - \tilde{w}^k) - \xi^k\|^2) \\ &= \gamma(2 - \gamma) \alpha_k^* (w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k). \end{aligned} \quad (3.18)$$

In order to guarantee that the right-hand side of (3.18) is positive, we take $\gamma \in (0, 2)$. \square

In fact, α_k^* is bounded below by a positive amount which is the subject of the following lemma.

Lemma 3.4. *Assume that $\tilde{w}^k = (\tilde{x}_1^k, \dots, \tilde{x}_N^k, \tilde{\lambda}^k)$ is generated by (2.17)–(2.20) from a given $w^k = (x_1^k, \dots, x_N^k, \lambda^k)$, then one has*

$$\alpha_k^* \geq \frac{1 - \nu}{(\|GD^{-1/2}\| + 1)^2}. \quad (3.19)$$

Proof. Using the fact that square matrix D is positive symmetric definite, we have

$$\|G(w^k - \tilde{w}^k)\| = \|GD^{-1/2}D^{1/2}(w^k - \tilde{w}^k)\| \leq \|GD^{-1/2}\| \|D^{1/2}(w^k - \tilde{w}^k)\|. \quad (3.20)$$

Moreover, Note that Criterion (2.18*b) implies

$$\begin{aligned} \|\xi^k\|^2 &\leq \sum_{j=1}^N \frac{\beta_j}{2} \|x_j^k - \tilde{x}_j^k\|^2 + \eta \|\lambda^k - \tilde{\lambda}^k\|^2 \\ &= \|w^k - \tilde{w}^k\|_D^2. \end{aligned} \quad (3.21)$$

Hence, applying $\|w^k - \tilde{w}^k\|_D^2 = \|D^{1/2}(w^k - \tilde{w}^k)\|^2$ in the above inequality, we get

$$\|\xi^k\| \leq \|D^{1/2}(w^k - \tilde{w}^k)\|. \quad (3.22)$$

Combining (3.20) and (3.22), we have

$$\|G(w^k - \tilde{w}^k) - \xi^k\| \leq (\|GD^{-1/2}\| + 1) \|D^{1/2}(w^k - \tilde{w}^k)\|. \quad (3.23)$$

Consequently, applying the above inequality, (3.7), and (2.32) to α_k^* yields

$$\begin{aligned} \alpha_k^* &= \frac{(w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k)}{\|G(w^k - \tilde{w}^k) - \xi^k\|^2} \\ &\geq \frac{(1 - \nu) \|w^k - \tilde{w}^k\|_D^2}{(\|GD^{-1/2}\| + 1)^2 \|D^{1/2}(w^k - \tilde{w}^k)\|^2} \\ &\geq \frac{1 - \nu}{(\|GD^{-1/2}\| + 1)^2}, \end{aligned} \quad (3.24)$$

and thus the assertion is proved. □

Now, we are in the stage to prove the main convergence theorem of this paper.

Theorem 3.5. For any $w^* = (x^*, \lambda^*) \in \mathcal{W}^*$, the sequence $\{w^k = (x^k, \lambda^k)\}$ generated by the proposed method satisfies

$$\|w^{k+1} - w^*\|^2 \leq \|w^k - w^*\|^2 - \frac{\gamma(2-\gamma)(1-\nu)^2}{(\|GD^{-1/2}\| + 1)^2} \|w^k - \tilde{w}^k\|_D^2. \quad (3.25)$$

Thus we have

$$\lim_{k \rightarrow \infty} \|w^k - \tilde{w}^k\|^2 = 0, \quad (3.26)$$

and the iteration of the proposed method will terminate in finite loops.

Proof. First, it follows from (3.12) and (3.18) that

$$\|w^{k+1} - w^*\|^2 \leq \|w^k - w^*\|^2 - \gamma(2-\gamma)\alpha_k^* (w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k). \quad (3.27)$$

Using (2.32), (3.7), and (3.19), we have

$$\alpha_k^* (w^k - \tilde{w}^k)^T (G(w^k - \tilde{w}^k) - \xi^k) \geq \frac{(1-\nu)^2}{(\|GD^{-1/2}\| + 1)^2} \|w^k - \tilde{w}^k\|_D^2. \quad (3.28)$$

Substituting (3.28) in (3.27), Assertion (3.25) is proved. Therefore, we have

$$\sum_{k=0}^{\infty} \frac{\gamma(2-\gamma)(1-\nu)^2}{(\|GD^{-1/2}\| + 1)^2} \|w^k - \tilde{w}^k\|_D^2 \leq \|w^0 - w^*\|^2, \quad (3.29)$$

and Assertion (3.26) follows immediately.

Since we use

$$\max \left\{ \|x_1^k - \tilde{x}_1^k\|_{\infty}, \dots, \|x_N^k - \tilde{x}_N^k\|_{\infty}, \|\lambda^k - \tilde{\lambda}^k\|_{\infty} \right\} < \epsilon, \quad (3.30)$$

as the stopping criterium, it follows from (3.26) that the iteration will terminate in finite loops for any given $\epsilon > 0$. \square

4. Numerical Examples

This section describes experiments testifying to the good performance of proposed method. The algorithms were written in Matlab (version 7.0) and compiled on a notebook with CPU of Intel Core 2 Duo (2.01 GHz and RAM of 0.98 GB).

To evaluate the behavior of the proposed method, we construct examples about convex separable quadratic programming (CSQP) with linearly coupling constraints. The convex separable quadratic programming was generated as follows:

$$\min \left\{ \frac{1}{2} \sum_{i=1}^N x_i^T H_i x_i - c_i^T x_i \mid x \in \Omega \right\}, \quad \Omega = \left\{ x \geq 0 \mid \sum_{i=1}^N A_i x_i \leq b \right\}, \quad (4.1)$$

where $H_i \in R^{n_i \times n_i}$ is a symmetric positive definite matrix, $A_i \in R^{m \times n_i}$, $b \in R^m$, and $c_i \in R^{n_i}$. We construct matrices A_i and H_i in the test examples as follows. The elements of A_i are randomly given in $(-5, 5)$, and the matrices H_i are defined by setting

$$H_i := V_i \Sigma_i V_i^T, \quad (4.2)$$

where

$$V_i = I_{n_i} - \frac{2v_i v_i^T}{v_i^T v_i}, \quad \Sigma_i = \text{diag}(\sigma_{k,i}), \quad \sigma_{k,i} = \cos \frac{k\pi}{n_i + 1} + 1. \quad (4.3)$$

In this way, H_i is positive definite and has prescribed eigenvalues between $(0, 2)$. If x^* is the solution of Problem (4.1), according to the KKT principle, there is a $0 \leq y^* \in R^m$ such that

$$\begin{aligned} H_i x_i^* + A_i^T y^* - c_i &\geq 0, & x^{*T} (H_i x_i^* + A_i^T y^* - c_i) &= 0, & x_i^* &\geq 0, \\ \sum_{i=1}^N A_i x_i^* &\leq b, & y^{*T} \left(\sum_{i=1}^N A_i x_i^* - b \right) &= 0, & y^* &\geq 0. \end{aligned} \quad (4.4)$$

Let $\xi_i \in R^{n_i}$ and $z \in R^m$ be random vectors whose elements are between $(-1, 1)$. We set

$$\begin{aligned} x_i^* &= \max(\xi_i, 0) * \tau_1, & \xi_i^* &= \max(-\xi_i, 0) * \tau_2, \\ y^* &= \max(z, 0) * \tau_3, & z^* &= \max(-z, 0) * \tau_4, \end{aligned} \quad (4.5)$$

where τ_i , $i = 1, 2, 3, 4$, are positive parameters. By setting

$$c_i = H_i x_i^* + A_i^T y^* - \xi_i^*, \quad b = \sum_{i=1}^N A_i x_i^* + z^*, \quad (4.6)$$

we constructed a test problem of (4.1) which has the known solution point x^* and the optimal Lagrangian multipliers y^* . We tested such problems with $\tau_1 = 0.5$, $\tau_2 = 10$, $\tau_3 = 0.5$, $\tau_4 = 10$. Here, two example sets were considered. The problems in the first set have 3 separable operators ($N = 3$), and the second have 2 ($N = 2$).

In the first experiment, we employ AsPDM with update w_2^{k+1} to solve CSQP with 3 separable operators. (The reason why we choose w_2^{k+1} here is that it usually performs better than w_1^{k+1} .) The stopping criterion was chosen as $\|e(w)\|_\infty < 10^{-3}$; the parameters were set as $\nu = 0.2$, $\eta = 0.5$, and $\gamma = 1.8$. Table 1 reports the number iterations (denoted as Its.),

Table 1: AsPDM for CSQP with 3 separable operators.

m	n_1	n_2	n_3	Its.	Num _{f}
100	50	50	50	674	4066
100	100	100	100	1581	9508
150	150	150	150	1775	10672
200	200	200	200	2108	12670

Table 2: Its. and function eval. for different problem sizes.

m	n_1	n_2	AsPDM		PCM	
			Its.	Num _{f}	Its.	Num _{f}
100	100	100	1089	4371	1343	5393
200	200	200	1265	5075	1649	6621
300	300	300	1655	6635	1765	7087
400	400	400	1573	6307	1834	7365
500	500	500	2299	9211	2218	8901
600	600	600	2267	9083	2289	9187
100	80	80	568	2287	616	2485
100	50	150	1192	4787	1231	4945
200	150	100	567	2283	572	2311
200	100	200	744	2991	846	3407

the total number of function evaluations (denoted as Num _{f}) for different problem-sizes. Here, Num _{f} = $\sum_{i=1}^3$ Num _{f_i} . Observed from Table 1, the solutions are obtained in a moderate number of iterations; thus the proposed method is effectively applicable. In addition, the evaluations of f_i per iteration are approximately equal to 2. AsPDM is well suited to solve separable problems.

Next, we compared the computational efficiency of AsPDM against the method in [7] (denoted as PCM), regarded as a highly efficient PPA-based method that can be well suited to solve VI. Iterations were terminated when the criterion $\|e(w)\|_\infty < 10^{-5}$ was met. Table 2 reports the iterations, the total number of function evaluation for both methods. We observe that both methods are acceptable to for us to find a solution. Concerning computational efficiency, we can observe that AsPDM is comparable and clearly faster than PCM; moreover, function evaluations are also less, except in the case of $m = 500, n_1 = 500, n_2 = 500$. In some cases, AsPDM can reduce about 20% computation cost than PCM. For $m = 100, n_1 = 100, n_2 = 100$, we plot the error versus iteration number for both AsPDM and PCM in Figure 1. We have found that both methods converge quickly for the first hundred iterations but slow down as the exact solution is reached. The speed of AsPDM is better than PCM.

In addition to being fast, AsPDM can solve the problem separately; that is the most significant advantage over other methods. Hence, AsPDM is more suitable to solve the real-life separable problems.

5. Conclusions

We have proposed AsPDM for solving separable problems. It decomposes the original problem to independent low-dimension subproblems and solves those subproblems in parallel. Only the function values is required in the process, and the total computational

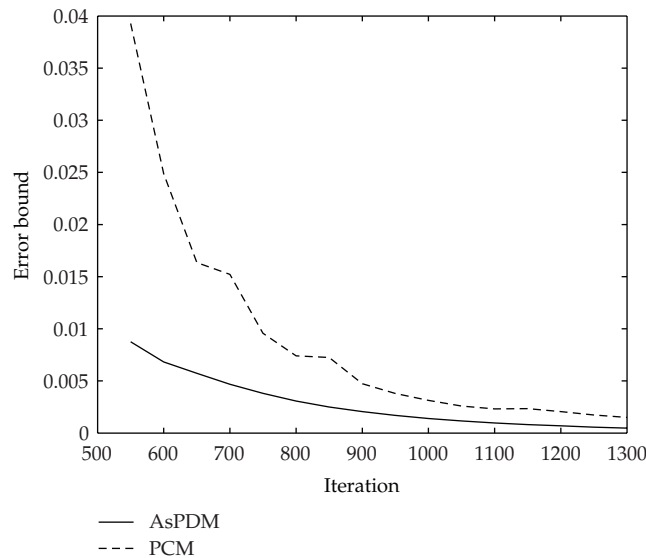


Figure 1: Error versus iteration number for the method and for with $m = 100$, $n_1 = 100$, $n_2 = 100$.

cost is very small. AsPDM is easy to implement and does not appear to require application-specific tuning. The numerical results also evidenced the efficiency of our method. Thus, the new method is applicable and recommended in practice.

Acknowledgment

The author was supported by the NSFC Grant 70901018.

References

- [1] R. T. Rockafellar and R. J.-B. Wets, "Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time," *SIAM Journal on Control and Optimization*, vol. 28, no. 4, pp. 810–822, 1990.
- [2] B. Martinet, "Régularisation d'inéquations variationnelles par approximations successives," *Revue Française d'Informatique et de Recherche Opérationnelle*, vol. 4, pp. 154–158, 1970.
- [3] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877–898, 1976.
- [4] R. T. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 97–116, 1976.
- [5] A. Auslender, M. Teboulle, and S. Ben-Tiba, "A logarithmic-quadratic proximal method for variational inequalities," *Computational Optimization and Applications*, vol. 12, no. 1-3, pp. 31–40, 1999.
- [6] Y. Censor and S. A. Zenios, "Proximal minimization algorithm with D -functions," *Journal of Optimization Theory and Applications*, vol. 73, no. 3, pp. 451–464, 1992.
- [7] B. He, X. Yuan, and J. J. Z. Zhang, "Comparison of two kinds of prediction-correction methods for monotone variational inequalities," *Computational Optimization and Applications*, vol. 27, no. 3, pp. 247–267, 2004.
- [8] A. Nemirovsky, Prox-method with rate of convergence $0(1/k)$ for smooth variational inequalities and saddle point problem, Draft of 30/10/2003.
- [9] M. Teboulle, "Convergence of proximal-like algorithms," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 1069–1083, 1997.

- [10] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation, Numerical Methods*, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.
- [11] G. Chen and M. Teboulle, "A proximal-based decomposition method for convex minimization problems," *Mathematical Programming*, vol. 64, no. 1, Ser. A, pp. 81–101, 1994.
- [12] B. He, L.-Z. Liao, and S. Wang, "Self-adaptive operator splitting methods for monotone variational inequalities," *Numerische Mathematik*, vol. 94, no. 4, pp. 715–737, 2003.
- [13] P. Mahey, S. Oualibouch, and D. T. Pham, "Proximal decomposition on the graph of a maximal monotone operator," *SIAM Journal on Optimization*, vol. 5, no. 2, pp. 454–466, 1995.
- [14] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM Journal on Control and Optimization*, vol. 29, no. 1, pp. 119–138, 1991.
- [15] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, vol. 9 of *SIAM Studies in Applied Mathematics*, SIAM, Philadelphia, Pa, USA, 1989.
- [16] B.-S. He, H. Yang, and C.-S. Zhang, "A modified augmented Lagrangian method for a class of monotone variational inequalities," *European Journal of Operational Research*, vol. 159, no. 1, pp. 35–51, 2004.
- [17] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems," *Computational Optimization and Applications*, vol. 1, no. 1, pp. 93–111, 1992.
- [18] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, NY, USA, 1999.
- [19] A. Hamdi and P. Mahey, "Separable diagonalized multiplier method for decomposing nonlinear programs," *Computational & Applied Mathematics*, vol. 19, no. 1, p. 1–29, 125, 2000.
- [20] A. Hamdi, P. Mahey, and J. P. Dussault, "A new decomposition method in nonconvex programming via a separable augmented Lagrangian," in *Recent advances in Optimization*, vol. 452 of *Lecture Notes in Economics and Mathematical Systems*, pp. 90–104, Springer, Berlin, Germany, 1997.
- [21] P. Tseng, "Alternating projection-proximal methods for convex programming and variational inequalities," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 951–965, 1997.
- [22] B. S. He, X. L. Fu, and Z. K. Jiang, "Proximal-point algorithm using a linear proximal term," *Journal of Optimization Theory and Applications*, vol. 141, no. 2, pp. 299–319, 2009.
- [23] E. H. Zarantonello, "Projections on convex sets in Hilbert space and spectral theory," in *Contributions to Nonlinear Functional Analysis*, E. H. Zarantonello, Ed., Academic Press, New York, NY, USA, 1971.