Alternating directions based contraction method for generally separable linearly constrained convex programming problems

Bingsheng He[1] **Min Tao** [2] **Ming-Hua Xu**[3] and **Xiao-Ming Yuan**[4]

Abstract. The classical alternating direction method (ADM) has been well studied in the context of linearly constrained convex programming problems and variational inequalities where both the involved operators and constraints are separable into two parts. In particular, recentness has witnessed a number of novel applications arising in diversified areas (e.g. Image Processing and Statistics), for which the ADM is surprisingly efficient. Despite the apparent significance, it is still not clear whether the ADM can be extended to the case where the number of separable parts emerging in the involved operators and constraints is three, no speaking of the generally separable case with finitely many of separable parts. This paper is to extend the spirit of ADM to solve such generally separable linearly constrained convex programming problems that both the involved operators and constraints are separable into finitely many of parts. As a result, the first implementable algorithm—-an ADM based contraction type algorithm, is developed for solving such generally separable linearly constrained convex programming problems. The realization of tackling this class of problems substantially broadens the applicable scope of ADM in more areas.

Keywords: Alternating direction method, convex programming, linear constraint, separable structure, contraction method

---

1 Introduction

For solving the linearly constrained convex programming problem:

$$\min\{\theta(u) \mid Qu = b,\ u \in \mathcal{U}\}, \tag{1.1}$$

where $\theta(u) : \Re^n \to \Re$ is a closed proper convex function (not necessarily smooth); $Q \in \Re^{l\times n}$; $b \in \Re^l$ and $\mathcal{U} \subseteq \Re^n$ is a closed convex set, the classical augmented Lagrangian method (ALM) generates the sequence of iterates via the following scheme:

$$\begin{cases} u^{k+1} = \text{Argmin}\{\mathcal{L}_A(u, \lambda^k) := \theta(u) - (\lambda^k)^T(Qu - b) + \frac{\beta}{2}\|Qu - b\|^2 \mid u \in \mathcal{U}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Qu^{k+1} - b), \end{cases}$$

where $\beta > 0$ is the penalty parameter for the violation of the linear constraint and $\lambda^k \in \Re^l$ is the associated Lagrange multiplier. We refer to, e.g. [1, 23], for the intensive study of ALM.

An important and self-interested scenario of (1.1), which captures concrete applications in many fields, is the following case where both the objective function and the constraint in (1.1) are separable into two parts:

$$\min\{\theta_1(x) + \theta_2(y) \mid Ax + By = b,\ x \in \mathcal{X},\ y \in \mathcal{Y}\}, \tag{1.2}$$

---

[1]Department of Mathematics, Nanjing University, Nanjing, 210093, China. This author was supported by the NSFC Grant 10971095 and the NSF of Province Jiangsu grant BK2008255. Email: hebma@nju.edu.cn
[2]Department of Mathematics, Nanjing University, Nanjing, 210093, China.
[3]School of Mathematics and Physics, Jiangsu Polytechnic University, Changzhou, Jiangsu Province, 213164, China. Email: xuminghua@jpu.edu.cn
[4]Corresponding author. Department of Mathematics, Hong Kong Baptist University, Hong Kong, China. This author was supported in part by the RGC grant 203009 and the NSFC grant 10701055. Email: xmyuan@hkbu.edu.hk

where $\theta_1 : \Re^{n_1} \to \Re$ and $\theta_2 : \Re^{n_2} \to \Re$ are closed proper convex functions (not necessarily smooth); $A \in \Re^{l\times n_1}$ and $B \in \Re^{l\times n_2}$; $\mathcal{X} \subseteq \Re^{n_1}$ and $\mathcal{Y} \subseteq \Re^{n_2}$ are closed convex sets; and $n_1 + n_2 = n$. Applying directly the ALM, we easily have the ALM-specified iterative scheme for (1.2):

$$\begin{cases} (x^{k+1}, y^{k+1}) = \text{Argmin}\left\{\theta_1(x) + \theta_2(y) - (\lambda^k)^T(Ax + By - b) + \frac{\beta}{2}\|Ax + By - b\|^2 \;\middle|\; x \in \mathcal{X}, y \in \mathcal{Y}\right\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \end{cases} \tag{1.3}$$

The direct implementation of ALM, however, treats the well-structured problem (1.2) as a generic convex programming and ignores completely the nice separable structure emerging in both the objective function and constraint of (1.2). Hence, not structure-exploited.

The alternating direction method (ADM), which dates back to [11, 12] and is closely related to the Douglas-Rachford operator splitting method [5, 24], is perhaps the most popular method for solving (1.2). More specifically, the iterative scheme of ADM for (1.2) is:

$$\begin{cases} x^{k+1} = \text{Argmin}\left\{\theta_1(x) - (\lambda^k)^T(Ax + By^k - b) + \frac{\beta}{2}\|Ax + By^k - b\|^2 \;\middle|\; x \in \mathcal{X}\right\}; \\ y^{k+1} = \text{Argmin}\left\{\theta_2(y) - (\lambda^k)^T(Ax^{k+1} + By - b) + \frac{\beta}{2}\|Ax^{k+1} + By - b\|^2 \;\middle|\; y \in \mathcal{Y}\right\}; \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \end{cases}$$

Therefore, ADM can be viewed as a practical and structured-exploited variant (split form or relaxed form) of ALM for solving the separable problem (1.2), with the adaption of minimizing the involved separable variables x and y separably in an alternating order. Because of its significant efficiency and easy implementation, ADM has attracted wide attention of many authors in various areas, see e.g. [2, 4, 6, 8, 10, 13, 14, 16, 18, 19, 26]. In particular, some novel and attractive applications of ADM have been discovered very recently, e.g. the total-variation problem in Image Processing [7, 22, 28], the covariance selection problem and semidefinite least square problem in Statistics [17, 30], the semidefinite programming problems [25, 27], the sparse and low-rank recovery problem in Engineering [21, 31].

For obvious reasons, we are interested in the extension of ADM to the generally separable case of (1.1) in the sense that both the objective function and the constraints are separable into more than two parts—more specifically, finitely many of parts:

$$\min\{\theta_1(x_1) + \theta_2(x_2) + \cdots + \theta_m(x_m) \mid \sum_{i=1}^{m} A_i x_i = b, \quad x_i \in \mathcal{X}_i, \; i = 1, 2, \ldots, m\}, \tag{1.4}$$

where $\theta_i : \Re^{n_i} \to \Re$ $(i = 1, 2, \cdots, m)$ are closed proper convex functions (not necessarily smooth); $A_i \in \Re^{l\times n_i}$ $(i = 1, 2, \cdots, m)$; $\mathcal{X}_i \subseteq \Re^{n_i}$ $(i = 1, 2, \cdots, m)$ are closed convex sets; and $\sum_{i=1}^m n_i = n$. Throughout, we assume that the solution set of (1.4) is not empty. It is then natural to manage to extend ADM to solve the general case (1.4), resulting in the following scheme:

$$\begin{cases} x_1^{k+1} = \text{Argmin}\left\{\theta_1(x_1) - (\lambda^k)^T p_1(x_1) + \frac{\beta}{2}\|p_1(x_1)\|^2 \;\middle|\; x_1 \in \mathcal{X}_1\right\}; \\ \cdots\cdots \\ x_i^{k+1} = \text{Argmin}\left\{\theta_i(x_i) - (\lambda^k)^T p_i(x_i) + \frac{\beta}{2}\|p_i(x_i)\|^2 \;\middle|\; x_i \in \mathcal{X}_i\right\}; \\ \cdots\cdots \\ x_m^{k+1} = \text{Argmin}\left\{\theta_m(x_m) - (\lambda^k)^T p_m(x_m) + \frac{\beta}{2}\|p_m(x_m)\|^2 \;\middle|\; x_m \in \mathcal{X}_m\right\}; \\ \lambda^{k+1} = \lambda^k - \beta(\sum_{j=1}^m A_j x_j^{k+1} - b), \end{cases} \tag{1.5}$$

where

$$p_i(x_i) = \sum_{j=1}^{i-1} A_j x_j^{k+1} + A_i x_i + \sum_{j=i+1}^{m} A_j x_j^k - b, \quad i = 1, \ldots, m,$$

and the involved subproblems of (1.5) are solved consecutively in the ADM manner. Unfortunately, the validity of this extension (1.5) is so far not clear. In fact, even for the special case of (1.4) where $m = 3$, the convergence of the extended ADM (1.5) is still open.

In this paper, we show that although the convergence of the extended ADM (1.5) is ambiguous, the iterate generated by (1.5) is readily useful for constructing a beneficial descent direction, along which a closer iterate to the solution set of (1.4) can be easily obtained via identifying an appropriate step size with economic load of computation. In this sense, we extend the spirit of the ADM from the eligible case where $m = 2$ to the general case with arbitrary m, and derive an implementable ADM-based algorithm for solving the linearly constrained convex programming problem (1.4) with full extent of separable structure. To our best knowledge, it is the first time to develop trackable algorithms for (1.4) based on the full utilization of its sparable structure. As we will show in detail, the resulted method falls into the frameworks of both the descent-like methods in the sense that the iterates generated by the extended ADM scheme (1.5) can be used to construct descent directions; and the contraction-type methods (according to the definition in [3]) in the sense that the sequence of iterates is Fejér monotone with respect to the solution set of (1.4). Hence, we name the new method to be proposed as the Alternating Directions Based Contraction (ADBC) method.

The rest of the paper is organized as follows. We first propose the ADBC method for the special case of (1.4) where $m = 3$ in Section 2, with the aim of exposing the main idea more clearly. Convergence for this case will also be provided in this section. In Section 3, we complete the delineation of the ADBC method for the general case of (1.4). Some discussions on the potential improvements of the ADBC method are provided in Section 4, and thus some more advanced and practical versions of the ADBC method are proposed conceptually. Finally, some conclusions are made in Section 5.

2 The case with three separable parts

In this section, we focus on the special case of (1.4) where $m = 3$ in order to expose the main idea and avoid complicated notation resulted by the general case of (1.4). More specifically, we rewrite this case in the following form:

$$\min\{\theta_1(x) + \theta_2(y) + \theta_3(z) | Ax + By + Cz = b,\ x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}\}, \tag{2.1}$$

where $\theta_i : \Re^{n_i} \to \Re$ $(i = 1, 2, 3)$ are closed proper convex functions (not necessarily smooth); $A \in \Re^{l \times n_1}$, $B \in \Re^{l \times n_2}$ and $C \in \Re^{l \times n_3}$; $\mathcal{X} \subseteq \Re^{n_1}$, $\mathcal{Y} \subseteq \Re^{n_2}$ and $\mathcal{Z} \subseteq \Re^{n_3}$ are closed convex sets; and $n_1 + n_2 + n_3 = n$.

2.1 Algorithm

Let $H \in \Re^{l \times l}$ and $G \in \Re^{(n_2+n_3+l)\times(n_2+n_3+l)}$ be given positive definite matrices; $\|\cdot\|_H$ and $\|\cdot\|_G$ denote the $H-$norm and $G-$norm, respectively; $\gamma \in (0, 2)$. For convenience, we use the notations $w = (x, y, z, \lambda)$ and $v = (y, z, \lambda)$; $w^i = (x^i, y^i, z^i, \lambda^i)$, $\tilde{w}^i = (\tilde{x}^i, \tilde{y}^i, \tilde{z}^i, \tilde{\lambda}^i)$, $v^i = (y^i, z^i, \lambda^i)$ and

$\tilde{v}^i = (\tilde{y}^i, \tilde{z}^i, \tilde{\lambda}^i)$ for any positive integer i. Let

$$M = \begin{pmatrix} B^T HB & 0 & 0 \\ C^T HB & C^T HC & 0 \\ 0 & 0 & H^{-1} \end{pmatrix} \quad \text{and} \quad d(u,v) = M(u-v). \tag{2.2}$$

For the given $v^k = (y^k, z^k, \lambda^k)$, the ADBC method for (2.1) generates the new iterative $w^k = (x^{k+1}, y^{k+1}, z^{k+1}, \lambda^{k+1})$ via the following steps.

The k-th iteration of the ADBC method for (2.1):
Step 1. The ADM step:

$$\tilde{x}^k = \text{Argmin}\big\{\theta_1(x) - (\lambda^k)^T Ax + \frac{1}{2}\|Ax + By^k + Cz^k - b\|_H^2 \;\big|\; x \in \mathcal{X}\big\}, \tag{2.3a}$$

$$\tilde{y}^k = \text{Argmin}\big\{\theta_2(y) - (\lambda^k)^T By + \frac{1}{2}\|A\tilde{x}^k + By + Cz^k - b\|_H^2 \;\big|\; y \in \mathcal{Y}\big\}, \tag{2.3b}$$

$$\tilde{z}^k = \text{Argmin}\big\{\theta_3(z) - (\lambda^k)^T Cz + \frac{1}{2}\|A\tilde{x}^k + B\tilde{y}^k + Cz - b\|_H^2 \;\big|\; z \in \mathcal{Z}\big\}, \tag{2.3c}$$

$$\tilde{\lambda}^k = \lambda^k - H(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b). \tag{2.3d}$$

Step 2. The contractive step:

$$\begin{cases} x^{k+1} = \tilde{x}^k, \\ v^{k+1} = v^k - \alpha_k G^{-1} d(v^k, \tilde{v}^k), \end{cases} \tag{2.4}$$

where

$$\alpha_k = \gamma \alpha_k^*, \qquad \alpha_k^* = \frac{\varphi(v^k, \tilde{v}^k)}{\|G^{-1} d(v^k, \tilde{v}^k)\|_G^2} \tag{2.5}$$

with

$$\varphi(v^k, \tilde{v}^k) := (v^k - \tilde{v}^k)^T d(v^k, \tilde{v}^k) + (\lambda^k - \tilde{\lambda}^k)^T \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big). \tag{2.6}$$

Remark 2.1 *As shown above, the proposed ADBC method falls into the algorithmic framework of prediction-correction methods, where the ADM step generates the predictor* $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$, *and the new iterate* $(x^{k+1}, y^{k+1}, z^{k+1}, \lambda^{k+1})$ *is obtained by further correcting the predictor via the contractive step.*

Remark 2.2 *It is easy to see that the ADM step (2.3) dominates the computation of each iteration of the ADBC method, while the contractive step (2.4) is much cheaper numerically.*

Remark 2.3 *In some practical applications, B and C in* (2.1) *are identity matrices. In this case, if we take $H = \beta I$, it follows that*

$$M = \begin{pmatrix} \beta I & 0 & 0 \\ \beta I & \beta I & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix}$$

and

$$\begin{aligned} \varphi(v^k, \tilde{v}^k) &= \beta\big\{\|y^k - \tilde{y}^k\|^2 + \|z^k - \tilde{z}^k\|^2 + (y^k - \tilde{y}^k)^T (z^k - \tilde{z}^k)\big\} + \frac{1}{\beta}\|\lambda^k - \tilde{\lambda}^k\|^2 \\ &\quad + (\lambda^k - \tilde{\lambda}^k)^T \big((y^k - \tilde{y}^k) + (z^k - \tilde{z}^k)\big). \end{aligned}$$

Remark 2.4 *There are different choices for selecting the positive definite matrix G. In the case when M is non-singular, by setting $G = MM^T$, the update form* (2.4) *becomes*

$$v^{k+1} = v^k - \gamma\alpha_k^* M^{-T}(v^k - \tilde{v}^k) \qquad \textit{and} \qquad \alpha_k^* = \frac{\varphi(v^k, \tilde{v}^k)}{\|v^k - \tilde{v}^k\|^2}. \tag{2.7}$$

For the special case described in Remark 2.3, (2.7) *is particularly recommended.*

Remark 2.5 *When $\theta_3(z) = 0$ and $C = 0$, the ADM step* (2.3a)- (2.3d) *reduces to the classical ADM for the special case of (1.4) with two separable parts. In this case,*

$$M = \begin{pmatrix} B^T HB & 0 \\ 0 & H^{-1} \end{pmatrix} \tag{2.8}$$

is symmetric and positive semi-definite. Even if M is semi-definite, in (2.4) *we can set*

$$G^{-1}d(v^k, \tilde{v}^k) = v^k - \tilde{v}^k.$$

Then, using the strategy of determining step sizes in (2.5)*, it leads to the following scheme:*

$$v^{k+1} = v^k - \gamma\alpha_k^*(v^k - \tilde{v}^k) \qquad \textit{and} \qquad \alpha_k^* = \frac{\varphi(v^k, \tilde{v}^k)}{\|v^k - \tilde{v}^k\|_M^2},$$

which recovers exactly the ADM-based descent method in [29]. However, we note that the above strategy of linear combination does not work for the general case of (1.4) where $m > 2$, and the mainly reason is that the corresponding matrix M (as defined in (2.2)*) is no longer symmetric.*

2.2 The variational inequality characterization

Let $\partial(\cdot)$ denote the subgradient operator of a convex function; let $f(x) \in \partial(\theta_1(x))$, $g(y) \in \partial(\theta_2(y))$ and $h(z) \in \partial(\theta_3(z))$, respectively. Then, it is easy to verify that the convex programming problem (2.1) is characterized by the following variational inequality: Find $w^* = (x^*, y^*, z^*, \lambda^*) \in \mathcal{W} := \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \times \Re^l$ such that

$$\begin{cases} (x' - x^*)^T\{f(x^*) - A^T\lambda^*\} \geq 0, \\ (y' - y^*)^T\{g(y^*) - B^T\lambda^*\} \geq 0, \\ (z' - z^*)^T\{h(z^*) - C^T\lambda^*\} \geq 0, \\ (\lambda' - \lambda^*)^T(Ax^* + By^* + Cz^* - b) \geq 0, \end{cases} \qquad \forall\, w' = (x', y', z', \lambda') \in \mathcal{W}, \tag{2.9}$$

or in the more compact form:

$$(w' - w^*)^T F(w^*) \geq 0, \quad \forall\, w' \in \mathcal{W}, \tag{2.10}$$

where

$$w = \begin{pmatrix} x \\ y \\ z \\ \lambda \end{pmatrix} \qquad \text{and} \qquad F(w) = \begin{pmatrix} f(x) - A^T\lambda \\ g(y) - B^T\lambda \\ h(z) - C^T\lambda \\ Ax + By + Cz - b \end{pmatrix}. \tag{2.11}$$

Note that $F(w)$ defined in (2.11) is monotone whenever f, g and h are monotone. Under the nonempty assumption on the solution set of (1.4), the solution set of (2.10)-(2.11), denoted by $\mathcal{W}^*$, is also nonempty. In fact, according to Theorem 2.3.5 in [9], $\mathcal{W}^*$ is convex. With the notation of $v = (y, z, \lambda)$, we also define

$$\mathcal{V}^* := \{(y^*, z^*, \lambda^*) \,|\, (x^*, y^*, z^*, \lambda^*) \in \mathcal{W}^*\}.$$

On the other hand, note that the main subproblems (2.3a)-(2.3c) dominating the computation of each iteration of the proposed method can be rewritten into the variational inequality forms:

$$\tilde{x}^k \in \mathcal{X}, \quad (x' - \tilde{x}^k)^T \{f(\tilde{x}^k) - A^T[\lambda^k - H(A\tilde{x}^k + By^k + Cz^k - b)]\} \geq 0, \quad \forall x' \in \mathcal{X}; \tag{2.12a}$$

$$\tilde{y}^k \in \mathcal{Y}, \quad (y' - \tilde{y}^k)^T \{g(\tilde{y}^k) - B^T[\lambda^k - H(A\tilde{x}^k + B\tilde{y}^k + Cz^k - b)]\} \geq 0, \quad \forall y' \in \mathcal{Y}; \tag{2.12b}$$

$$\tilde{z}^k \in \mathcal{Z}, \quad (z' - \tilde{z}^k)^T \{h(\tilde{z}^k) - C^T[\lambda^k - H(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b)]\} \geq 0, \quad \forall z' \in \mathcal{Z}. \tag{2.12c}$$

2.3 Rationale of ADM based descent directions

The purpose of this section is to illustrate that $-d(v^k, \tilde{v}^k)$ (defined in (2.2)) is a descent direction of the unknown distance function $\frac{1}{2}\|v - v^*\|^2$ at $v = v^k$. Hence, the rationale of developing the descent step (2.4) is justified.

Theorem 2.1 *Let $\tilde{v}^k = (\tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$ be generated by the ADM step (2.3) from the given vector $v^k = (y^k, z^k, \lambda^k)$. Then, we have*

$$\varphi(v^k, \tilde{v}^k) = \frac{1}{2}\Big(\|B(y^k - \tilde{y}^k)\|_H^2 + \|C(z^k - \tilde{z}^k)\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2\Big) + \frac{1}{2}\|A\tilde{x}^k + By^k + Cz^k - b\|_H^2, \tag{2.13}$$

where $\varphi(v^k, \tilde{v}^k)$ is defined in (2.6).

Proof. Recall that (2.2) implies that

$$d(v^k, \tilde{v}^k) = \begin{pmatrix} B^T H B & & \\ C^T H B & C^T H C & \\ 0 & 0 & H^{-1} \end{pmatrix} \begin{pmatrix} y^k - \tilde{y}^k \\ z^k - \tilde{z}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

Thus, we have

$$\begin{aligned}
&(v^k - \tilde{v}^k)^T d(v^k, \tilde{v}^k) \\
&= \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} H & & \\ H & H & \\ 0 & 0 & H^{-1} \end{pmatrix} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \\
&= \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} H & \frac{1}{2}H & 0 \\ \frac{1}{2}H & H & 0 \\ 0 & 0 & H^{-1} \end{pmatrix} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \\
&= \frac{1}{2} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} H & H & 0 \\ H & H & 0 \\ 0 & 0 & H^{-1} \end{pmatrix} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \\
&\quad + \frac{1}{2}\Big(\|B(y^k - \tilde{y}^k)\|_H^2 + \|C(z^k - \tilde{z}^k)\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2\Big).
\end{aligned} \tag{2.14}$$

Now, we turn to deal with the second term of the right-hand-side of (2.6). By manipulations, we have

$$
\begin{aligned}
& (\lambda^k - \tilde{\lambda}^k)^T \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big) \\
= & \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ I & I & 0 \end{pmatrix} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \\
= & \frac{1}{2} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 0 & 0 & I \\ 0 & 0 & I \\ I & I & 0 \end{pmatrix} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \qquad (2.15)
\end{aligned}
$$

Adding (2.14) and (2.15), it follows that

$$
\begin{aligned}
\varphi(v^k, \tilde{v}^k) = & \frac{1}{2} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} H & H & I \\ H & H & I \\ I & I & H^{-1} \end{pmatrix} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \\
& + \frac{1}{2}\Big(\|B(y^k - \tilde{y}^k)\|_H^2 + \|C(z^k - \tilde{z}^k)\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2 \Big). \qquad (2.16)
\end{aligned}
$$

Because

$$
\begin{pmatrix} H & H & I \\ H & H & I \\ I & I & H^{-1} \end{pmatrix} = \begin{pmatrix} I \\ I \\ H^{-1} \end{pmatrix} H(I, I, H^{-1}),
$$

we have

$$
\begin{aligned}
& \frac{1}{2} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} H & H & I \\ H & H & I \\ I & I & H^{-1} \end{pmatrix} \begin{pmatrix} B(y^k - \tilde{y}^k) \\ C(z^k - \tilde{z}^k) \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \\
= & \frac{1}{2} \big\| B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k) + H^{-1}(\lambda^k - \tilde{\lambda}^k) \big\|_H^2 \\
= & \frac{1}{2} \big\| A\tilde{x}^k + By^k + Cz^k - b \big\|_H^2. \qquad (2.17)
\end{aligned}
$$

Assertion (2.13) follows from (2.16) and (2.17) directly. □

Recall (2.12). For any $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, we have

$$
\begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \\ z - \tilde{z}^k \end{pmatrix}^T \begin{pmatrix} f(\tilde{x}^k) - A^T[\lambda^k - H(A\tilde{x}^k + By^k + Cz^k - b)] \\ g(\tilde{y}^k) - B^T[\lambda^k - H(A\tilde{x}^k + B\tilde{y}^k + Cz^k - b)] \\ h(\tilde{z}^k) - C^T[\lambda^k - H(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b)] \end{pmatrix} \geq 0. \qquad (2.18)
$$

If $\varphi(v^k, \tilde{v}^k) = 0$, it follows from (2.13) that

$$
B(y^k - \tilde{y}^k) = 0, \qquad C(z^k - \tilde{z}^k) = 0
$$

and

$$
A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b = H^{-1}(\lambda^k - \tilde{\lambda}^k) = 0.
$$

Substituting it in (2.18), we get

$$\begin{pmatrix} x-\tilde{x}^k \\ y-\tilde{y}^k \\ z-\tilde{z}^k \end{pmatrix}^T \begin{pmatrix} f(\tilde{x}^k)-A^T\tilde{\lambda}^k \\ g(\tilde{y}^k)-B^T\tilde{\lambda}^k \\ h(\tilde{z}^k)-C^T\tilde{\lambda}^k \end{pmatrix} \geq 0, \quad \forall u \in \mathcal{U}, \tag{2.19}$$

which implies that $\tilde{v}^k \in \mathcal{V}^*$.

Lemma 2.1 *Let $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$ be generated by the ADM step (2.3) from the given vector $v^k = (y^k, z^k, \lambda^k)$. Then, we have*

$$\tilde{w}^k \in \mathcal{W}, \qquad (w'-\tilde{w}^k)^T d_2(v^k, \tilde{w}^k) \geq (w'-\tilde{w}^k)^T d_1(v^k, \tilde{v}^k), \qquad \forall\, w' \in \mathcal{W}, \tag{2.20}$$

where

$$d_1(v^k, \tilde{v}^k) = \begin{pmatrix} 0 & 0 & 0 \\ B^THB & 0 & 0 \\ C^THB & C^THC & 0 \\ 0 & 0 & H^{-1} \end{pmatrix} \begin{pmatrix} y^k-\tilde{y}^k \\ z^k-\tilde{z}^k \\ \lambda^k-\tilde{\lambda}^k \end{pmatrix} \tag{2.21}$$

and

$$d_2(v^k, \tilde{w}^k) = F(\tilde{w}^k) + \begin{pmatrix} A^T \\ B^T \\ C^T \\ 0 \end{pmatrix} H\big(B(y^k-\tilde{y}^k)+C(z^k-\tilde{z}^k)\big). \tag{2.22}$$

Proof. The proof consists of some manipulations. Using (2.3d), it follows from (2.18) that for any $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, we have

$$\begin{pmatrix} x'-\tilde{x}^k \\ y'-\tilde{y}^k \\ z'-\tilde{z}^k \end{pmatrix}^T \left\{ \begin{pmatrix} f(\tilde{x}^k)-A^T\tilde{\lambda}^k \\ g(\tilde{y}^k)-B^T\tilde{\lambda}^k \\ h(\tilde{z}^k)-C^T\tilde{\lambda}^k \end{pmatrix} + \begin{pmatrix} A^THB(y^k-\tilde{y}^k)+A^THC(z^k-\tilde{z}^k) \\ B^THC(z^k-\tilde{z}^k) \\ 0 \end{pmatrix} \right\} \geq 0. \tag{2.23}$$

Adding

$$\begin{pmatrix} x'-\tilde{x}^k \\ y'-\tilde{y}^k \\ z'-\tilde{z}^k \\ \lambda'-\tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 0 \\ B^THB(y^k-\tilde{y}^k) \\ C^THB(y^k-\tilde{y}^k)+C^THC(z^k-\tilde{z}^k) \\ H^{-1}(\lambda^k-\tilde{\lambda}^k) \end{pmatrix}$$

to both sides of (2.23) and using

$$A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b = H^{-1}(\lambda^k - \tilde{\lambda}^k),$$

we get $\tilde{w}^k \in \mathcal{W}$ and

$$\begin{aligned} &\begin{pmatrix} x'-\tilde{x}^k \\ y'-\tilde{y}^k \\ z'-\tilde{z}^k \\ \lambda'-\tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} f(\tilde{x}^k)-A^T\tilde{\lambda}^k \\ g(\tilde{y}^k)-B^T\tilde{\lambda}^k \\ h(\tilde{z}^k)-C^T\tilde{\lambda}^k \\ A\tilde{x}^k+B\tilde{y}^k+C\tilde{z}^k-b \end{pmatrix} + \begin{pmatrix} A^TH\big(B(y^k-\tilde{y}^k)+C(z^k-\tilde{z}^k)\big) \\ B^TH\big(B(y^k-\tilde{y}^k)+C(z^k-\tilde{z}^k)\big) \\ C^TH\big(B(y^k-\tilde{y}^k)+C(z^k-\tilde{z}^k)\big) \\ 0 \end{pmatrix} \right\} \\ &\geq \begin{pmatrix} x'-\tilde{x}^k \\ y'-\tilde{y}^k \\ z'-\tilde{z}^k \\ \lambda'-\tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 0 \\ B^THB(y^k-\tilde{y}^k) \\ C^THB(y^k-\tilde{y}^k)+C^THC(z^k-\tilde{z}^k) \\ H^{-1}(\lambda^k-\tilde{\lambda}^k) \end{pmatrix}, \quad \forall\, w' \in \mathcal{W}. \end{aligned}$$

Using the notations $F(w)$, $d_1(v^k, \tilde{v}^k)$ and $d_2(v^k, \tilde{w}^k)$, this lemma follows immediately. $\square$

Since (see (2.2) and (2.21))

$$d_1(v^k, \tilde{v}^k) = \begin{pmatrix} 0 \\ d(v^k, \tilde{v}^k) \end{pmatrix}, \tag{2.24}$$

we have

$$(\tilde{v}^k - v^*)^T d(v^k, \tilde{v}^k) = (\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k). \tag{2.25}$$

In order to show the descent property of $-d(v^k, \tilde{v}^k)$, we investigate the right-hand-side of (2.25).

Lemma 2.2 *Let $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$ be generated by the ADM step (2.3) from the given vector $v^k = (y^k, z^k, \lambda^k)$. Then, we have*

$$(\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big), \quad \forall\, w^* \in \mathcal{W}^*, \tag{2.26}$$

where $d_1(v^k, \tilde{v}^k)$ is defined in (2.21).

Proof. Since $w^* \in \mathcal{W}$, it follows from (2.20) that

$$(\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k) \geq (\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k). \tag{2.27}$$

We consider the right-hand-side of (2.27) and use the notation (2.22). Because F is monotone and $\tilde{w}^k \in \mathcal{W}$, we have

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

Therefore, by using (2.22) and the above inequality, we get

$$(\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k) \geq \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big)^T H\big(A(\tilde{x}^k - x^*) + B(\tilde{y}^k - y^*) + C(\tilde{z}^k - z^*)\big). \tag{2.28}$$

Since

$$Ax^* + By^* + Cz^* = b \qquad \text{and} \qquad H(A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b) = \lambda^k - \tilde{\lambda}^k,$$

from (2.28) we obtain

$$(\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k) \ \geq \ (\lambda^k - \tilde{\lambda}^k)^T \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big). \tag{2.29}$$

Substituting (2.29) into (2.27), Assertion (2.26) follows immediately. $\square$

From (2.25) and (2.26) follows that

$$(\tilde{v}^k - v^*)^T d(v^k, \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \big(B(y^k - \tilde{y}^k) + C(z^k - \tilde{z}^k)\big), \quad \forall\, v^* \in \mathcal{V}^*. \tag{2.30}$$

Theorem 2.2 *Let $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k, \tilde{\lambda}^k)$ be generated by the ADM step (2.3) from the given vector $v^k = (y^k, z^k, \lambda^k)$. Then, we have*

$$(v^k - v^*)^T d(v^k, \tilde{v}^k) \geq \varphi(v^k, \tilde{v}^k), \quad \forall\, v^* \in \mathcal{V}^*, \tag{2.31}$$

where $\varphi(v^k, \tilde{v}^k)$ and $d(v^k, \tilde{v}^k)$ are defined in (2.6) *and* (2.2)*, respectively.*

Proof. It directly follows from (2.30) and the definition of $\varphi(v^k, \tilde{v}^k)$. $\square$

2.4 Contractive properties

Using the direction $d(v^k, \tilde{v}^k)$ offered by the ADM step (2.3), the new iterate v^{k+1} is determined by the chosen positive definite matrix G and the step size α_k in the contractive step (2.4). In order to explain how to determine the step size α_k in (2.4), we define the step-size-dependent new iterate by

$$v^{k+1}(\alpha) = v^k - \alpha G^{-1} d(v^k, \tilde{v}^k). \tag{2.32}$$

In this way,

$$\vartheta(\alpha) = \|v^k - v^*\|_G^2 - \|v^{k+1}(\alpha) - v^*\|_G^2 \tag{2.33}$$

is the distance decrease function in the k-th iteration by using updating form (2.32). Since $v^* \in \mathcal{V}^*$ is unknown, we cannot maximize $\vartheta(\alpha)$ directly. The following theorem introduces a tight lower bound of $\vartheta(\alpha)$, namely $q(\alpha)$, which does not include the unknown vector v^*.

Lemma 2.3 *For any $v^* \in \mathcal{V}^*$ and $\alpha \geq 0$, we have*

$$\vartheta(\alpha) \geq q(\alpha), \tag{2.34}$$

where

$$q(\alpha) = 2\alpha\varphi(v^k, \tilde{v}^k) - \alpha^2 \|G^{-1} d(v^k, \tilde{v}^k)\|_G^2. \tag{2.35}$$

Proof. From (2.32) and (2.33) we have

$$\begin{aligned} \vartheta(\alpha) &= \|v^k - v^*\|_G^2 - \|v^k - v^* - \alpha G^{-1} d(v^k, \tilde{v}^k)\|_G^2 \\ &= 2\alpha(v^k - v^*)^T d(v^k, \tilde{v}^k) - \alpha^2 \|G^{-1} d(v^k, \tilde{v}^k)\|_G^2 \\ &\geq 2\alpha\varphi(v^k, \tilde{v}^k) - \alpha^2 \|G^{-1} d(v^k, \tilde{v}^k)\|_G^2. \end{aligned}$$

The last inequality follows from (2.31) and the assertion of this lemma is proved. □

Note that $q(\alpha)$ is a quadratic function of α, it reaches its maximum at

$$\alpha_k^* = \frac{\varphi(v^k, \tilde{v}^k)}{\|G^{-1} d(v^k, \tilde{v}^k)\|_G^2}, \tag{2.36}$$

and this is just the same as defined in (2.5). In the following we show that the lower value of $\alpha_k^* > 0$ is bounded away from zero.

Lemma 2.4 *For given positive definite matrix G, there is a constant $c_0 > 0$ such that*

$$\alpha_k^* \geq c_0, \quad \forall\, k \geq 0. \tag{2.37}$$

Proof. Use the notation of $d(v^k, \tilde{v}^k)$ (see (2.2)), we have

$$\|G^{-1} d(v^k, \tilde{v}^k)\|_G^2 = \left\| G^{-1/2} \begin{pmatrix} B^T H^{1/2} & & \\ C^T H^{1/2} & C^T H^{1/2} & \\ 0 & 0 & H^{-1/2} \end{pmatrix} \begin{pmatrix} H^{1/2} B(y^k - \tilde{y}^k) \\ H^{1/2} C(z^k - \tilde{z}^k) \\ H^{-1/2}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix} \right\|^2.$$

On the other hand (see (2.13)),

$$\varphi(v^k, \tilde{v}^k) \geq \frac{1}{2}\Big(\|B(y^k - \tilde{y}^k)\|_H^2 + \|C(z^k - \tilde{z}^k)\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2 \Big).$$

By denoting a constant

$$K = \left\| \begin{pmatrix} H^{1/2}B & H^{1/2}C & \\ 0 & H^{1/2}C & \\ 0 & 0 & H^{-1/2} \end{pmatrix} G^{-1} \begin{pmatrix} B^T H^{1/2} & & \\ C^T H^{1/2} & C^T H^{1/2} & \\ 0 & 0 & H^{-1/2} \end{pmatrix} \right\|,$$

we have

$$\alpha_k^* = \frac{\varphi(v^k, \tilde{v}^k)}{\|G^{-1}d(v^k, \tilde{v}^k)\|_G^2} \geq \frac{1}{2K} := c_0,$$

and the lemma is proved. $\square$

From (2.34), (2.35) and (2.36) follows that

$$\vartheta(\alpha_k^*) \geq q(\alpha_k^*) = \alpha_k^* \varphi(v^k, \tilde{v}^k).$$

Because Inequality (2.31) is used in the proof of (2.34), in practical computation, taking a relaxed factor $\gamma > 1$ is useful for fast convergence. By using (2.35) and (2.36), we have

$$\begin{aligned} q(\gamma\alpha_k^*) &= 2\gamma\alpha_k^*\varphi(v^k, \tilde{v}^k) - (\gamma\alpha_k^*)^2 \|G^{-1}d(v^k, \tilde{v}^k)\|_G^2 \\ &= \gamma(2-\gamma)\alpha_k^*\varphi(v^k, \tilde{v}^k). \end{aligned} \tag{2.38}$$

In order to guarantee that the right hand side of (2.38) is positive, we take $\gamma \in [1, 2)$. The following theorem points out that the sequence $\{v^k\}$ generated by the proposed method is Fejèr monotone with respect to $\mathcal{V}^*$.

Theorem 2.3 *For any $w^* \in \mathcal{W}^*$, the sequence $\{v^k\}$ generated by the proposed method satisfies*

$$\|v^{k+1} - v^*\|_G^2 \leq \|v^k - v^*\|_G^2 - \gamma(2-\gamma)\alpha_k^*\varphi(v^k, \tilde{v}^k), \quad \forall\, v^* \in \mathcal{V}^*. \tag{2.39}$$

Proof. Because the correction form is (see (2.4))

$$v^{k+1} = v^k - \gamma\alpha_k^* G^{-1} d(v^k, \tilde{v}^k),$$

it follows from (2.32), (2.33) and (2.34) that

$$\|v^{k+1} - v^*\|_G^2 \leq \|v^k - v^*\|_G^2 - q(\gamma\alpha_k^*). \tag{2.40}$$

The result of this theorem follows from (2.38) directly. $\square$

2.5 Convergence

We are now in the stage to prove the convergence of the proposed ADBC method for (2.1).

Theorem 2.4 *Let $\{v^k\}$ and $\{\tilde{w}^k\}$ be the sequences generated by the proposed method ADBC method for (2.1). Then we have*

1. *The sequence $\{v^k\}$ is bounded.*
2. $\lim_{k\to\infty}\{\|B(y^k - \tilde{y}^k)\|^2 + \|C(z^k - \tilde{z}^k)\|^2 + \|\lambda^k - \tilde{\lambda}^k\|^2\} = 0.$
3. *Any cluster point of $\{\tilde{w}^k\}$ is a solution point of (2.10)-(2.11).*

4. The sequence $\{\tilde{v}^k\}$ converges to some $v^\infty \in \mathcal{V}^*$ when B and C are column full rank matrices.

Proof. The first assertion follows from (2.39) directly. In addition, since $\alpha_k^* \geq c_0$, from (2.39) we get

$$\sum_{k=0}^{\infty} \gamma(2-\gamma)c_0\varphi(v^k, \tilde{v}^k) \leq \|v^0 - v^*\|_G^2$$

and thus

$$\lim_{k\to\infty} \varphi(v^k, \tilde{v}^k) = 0.$$

Consequently, it follows from (2.13) (see the definition of $\varphi(v^k, \tilde{v}^k)$) that

$$\lim_{k\to\infty} \|B(y^k - \tilde{y}^k)\| = 0, \quad \lim_{k\to\infty} \|C(z^k - \tilde{z}^k)\| = 0 \quad \text{and} \quad \lim_{k\to\infty} \|\lambda^k - \tilde{\lambda}^k\| = 0, \tag{2.41}$$

and the second assertion is proved. Substituting (2.41) in (2.23), we get

$$\begin{cases} \lim_{k\to\infty}(x - \tilde{x}^k)^T\{f(\tilde{x}^k) - A^T\tilde{\lambda}^k\} \geq 0, & \forall x \in \mathcal{X}, \\ \lim_{k\to\infty}(y - \tilde{y}^k)^T\{g(\tilde{y}^k) - B^T\tilde{\lambda}^k\} \geq 0, & \forall y \in \mathcal{Y}, \\ \lim_{k\to\infty}(z - \tilde{z}^k)^T\{h(\tilde{z}^k) - C^T\tilde{\lambda}^k\} \geq 0, & \forall z \in \mathcal{Z}. \end{cases} \tag{2.42}$$

It follows from (2.3d) and $\lim_{k\to\infty} \|\lambda^k - \tilde{\lambda}^k\| = 0$ that

$$\lim_{k\to\infty} (A\tilde{x}^k + B\tilde{y}^k + C\tilde{z}^k - b) = 0. \tag{2.43}$$

Combining (2.42) and (2.43) we get

$$\tilde{w}^k \in \mathcal{W}, \quad \lim_{k\to\infty} (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq 0, \quad \forall\, w \in \mathcal{W} \tag{2.44}$$

and thus any cluster point of $\{\tilde{w}^k\}$ is a solution point of (2.10)-(2.11).

If B and C are column full rank matrices, it follows from the first assertion and (2.41) that $\{\tilde{v}^k\}$ is also bounded. Let v^∞ be a cluster point of $\{\tilde{v}^k\}$ and the subsequence $\{\tilde{v}^{k_j}\}$ converges to v^∞. It follows from (2.44) that

$$\tilde{w}^{k_j} \in \mathcal{W}, \quad \lim_{k\to\infty} (w - \tilde{w}^{k_j})^T F(\tilde{w}^{k_j}) \geq 0, \quad \forall\, w \in \mathcal{W} \tag{2.45}$$

and consequently

$$\begin{cases} (x - x^\infty)^T\{f(x^\infty) - A^T\lambda^\infty\} \geq 0, & \forall x \in \mathcal{X}, \\ (y - y^\infty)^T\{g(y^\infty) - B^T\lambda^\infty\} \geq 0, & \forall y \in \mathcal{Y}, \\ (z - z^\infty)^T\{h(z^\infty) - C^T\lambda^\infty\} \geq 0, & \forall z \in \mathcal{Z}, \\ Ax^\infty + By^\infty + Cz^\infty - b = 0. \end{cases}$$

This means that $v^\infty \in \mathcal{V}^*$. Since $\{v^k\}$ is Fejér monotone and $\lim_{k\to\infty} \|v^k - \tilde{v}^k\| = 0$, the sequence $\{\tilde{v}^k\}$ cannot have other cluster point and $\{\tilde{v}^k\}$ converges to $v^\infty \in \mathcal{V}^*$. □

3 The general case

In this section, we propose the ADBC method for the general case of (1.4), i.e., the generally separable linearly constrained convex programming problem with finitely many separable parts.

3.1 Algorithm

Let $H \in \Re^{l\times l}$ and $G \in \Re^{(n_2+\cdots+n_m+l)\times(n_2+\cdots+n_m+l)}$ be given positive definite matrices; $\|\cdot\|_H$ and $\|\cdot\|_G$ denote the H-norm and G-norm, respectively; $\gamma \in (0,2)$. Let $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2 \cdots \times \mathcal{X}_m$. In this section, we abuse the notations that have been used in Section 2 without ambiguity. More specifically, let $w = (x_1, x_2, \cdots, x_m, \lambda)$ and $v = (x_2, x_3, \cdots, x_m, \lambda)$; $w^i = (x_1^i, x_2^i, \cdots, x_m^i, \lambda^i)$, $\tilde{w}^i = (\tilde{x}_1^i, \tilde{x}_2^i, \cdots, \tilde{x}_m^i, \tilde{\lambda}^i)$, $v^i = (x_2^i, x_3^i, \cdots, x_m^i, \lambda^i)$ and $\tilde{v}^i = (\tilde{x}_2^i, \tilde{x}_3^i, \cdots, \tilde{x}_m^i, \tilde{\lambda}^i)$ for any positive integer i.

Let

$$M = \begin{pmatrix} A_2^T H A_2 & 0 & \cdots & \cdots & 0 \\ A_3^T H A_2 & A_3^T H A_3 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A_m^T H A_2 & A_m^T H A_3 & \cdots & A_m^T H A_m & 0 \\ 0 & 0 & \cdots & 0 & H^{-1} \end{pmatrix} \quad \text{and } d(v^k, \tilde{v}^k) = M(v^k - \tilde{v}^k), \tag{3.1}$$

For the given $v^k = (x_2^k, \cdots, x_m^k, \lambda^k)$, the ADBC method for (1.4) generates the new iterative $w^{k+1} = (x_1^{k+1}, x_2^{k+1}, \cdots, x_m^{k+1}, \lambda^{k+1})$ via the following steps.

The k-th iteration of the ADBC method for (1.4):

Step 1. The ADM step:

$$\begin{cases} \tilde{x}_1^k = \mathrm{Argmin}\left\{\theta_1(x_1) - (\lambda^k)^T p_1(x_1) + \frac{1}{2}\|p_1(x_1)\|_H^2 \;\middle|\; x_1 \in \mathcal{X}_1\right\}; \\ \cdots\cdots \\ \tilde{x}_i^k = \mathrm{Argmin}\left\{\theta_i(x_i) - (\lambda^k)^T p_i(x_i) + \frac{1}{2}\|p_i(x_i)\|_H^2 \;\middle|\; x_i \in \mathcal{X}_i\right\}; \\ \cdots\cdots \\ \tilde{x}_m^k = \mathrm{Argmin}\left\{\theta_m(x_m) - (\lambda^k)^T p_m(x_m) + \frac{1}{2}\|p_m(x_m)\|_H^2 \;\middle|\; x_m \in \mathcal{X}_m\right\}; \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_j \tilde{x}_j^k - b), \end{cases} \tag{3.2}$$

where

$$p_i(x_i) := \sum_{j=1}^{i-1} A_j \tilde{x}_j^k + A_i x_i + \sum_{j=i+1}^{m} A_j x_j^k - b, \quad i = 1, \ldots, m.$$

Step 2. The contractive step:

$$\begin{cases} x^{k+1} = \tilde{x}^k, \\ v^{k+1} = v^k - \alpha_k G^{-1} d(v^k, \tilde{v}^k), \end{cases} \tag{3.3}$$

where

$$\alpha_k = \gamma \alpha_k^*, \qquad \alpha_k^* = \frac{\varphi(v^k, \tilde{v}^k)}{\|G^{-1} d(v^k, \tilde{v}^k)\|_G^2} \tag{3.4}$$

with

$$\varphi(v^k, \tilde{v}^k) = (v^k - \tilde{v}^k)^T d(v^k, \tilde{v}^k) + (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=2}^m A_j (x_j^k - \tilde{x}_j^k)\big). \tag{3.5}$$

3.2 The variational inequality characterization

Let $f_i(x) \in \partial(\theta_i(x))$ for $i = 1, 2, \cdots, m$. Then, it is evident that the generally separable linearly constrained convex programming problem (1.4) is characterized by the following variational inequality:

Find $w^* = (x_1^*, x_2^*, ..., x_m^*, \lambda^*) \in \mathcal{W} := \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m \times \Re^l$ such that

$$\begin{cases} (x_1' - x_1^*)^T \{f_1(x_1^*) - A_1^T \lambda^*\} \geq 0, \\ (x_2' - x_2^*)^T \{f_2(x_2^*) - A_2^T \lambda^*\} \geq 0, \\ \vdots \\ (x_m' - x_m^*)^T \{f_m(x_m^*) - A_m^T \lambda^*\} \geq 0, \\ (\lambda' - \lambda^*)^T (\sum_{i=1}^m A_i x_i^* - b) \geq 0, \end{cases} \quad \forall\, w = (x_1', x_2', \cdots, x_m', \lambda') \in \mathcal{W}, \tag{3.6}$$

or in the more compact form:

$$(w' - w^*)^T F(w^*) \geq 0, \quad \forall\, w' \in \mathcal{W}, \tag{3.7}$$

where

$$w = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ \lambda \end{pmatrix} \quad \text{and} \quad F(w) = \begin{pmatrix} f_1(x_1) - A_1^T \lambda \\ f_2(x_2) - A_2^T \lambda \\ \vdots \\ f_m(x_m) - A_m^T \lambda \\ \sum_{i=1}^m A_i x_i - b \end{pmatrix}. \tag{3.8}$$

Note that $F(w)$ defined in (3.8) is monotone whenever $f_i's$ are monotone. Again, under the nonempty assumption on the solution set of (1.4), the solution set of (3.7)-(3.8), denoted by $\mathcal{W}^*$, is a nonempty and convex set. With the notation of $v = (x_2, \cdots, x_m, \lambda)$, we also define

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

On the other hand, note that the subproblems (3.2) requiring the main computation at each iteration of the ADBC method for (1.4) are characterized by the following variational inequality forms:

$$\begin{cases} (x_1' - \tilde{x}_1^k)^T \big\{f_i(\tilde{x}_1^k) - A_1^T[\lambda^k - H(A_1 \tilde{x}_1^k + \sum_{j=2}^m A_j x_j^k - b)]\big\} \geq 0, \ \ \forall\, x_1' \in \mathcal{X}_i; \\ \cdots\cdots \\ (x_i' - \tilde{x}_i^k)^T \big\{f_i(\tilde{x}_i^k) - A_i^T[\lambda^k - H(\sum_{j=1}^i A_j \tilde{x}_j^k + \sum_{j=i+1}^m A_j x_j^k - b)]\big\} \geq 0, \ \ \forall\, x_i' \in \mathcal{X}_i; \\ \cdots\cdots \\ (x_m' - \tilde{x}_m^k)^T \big\{f_i(\tilde{x}_m^k) - A_m^T[\lambda^k - H(\sum_{j=1}^m A_j \tilde{x}_j^k - b)]\big\} \geq 0, \ \ \forall\, x_m' \in \mathcal{X}_m. \end{cases}$$

3.3 Rationale of ADM based descent directions

Similarly as Section 2.3, we show that $-d(v^k, \tilde{v}^k)$ defined by (3.1) is a descent direction of the unkown distance function $\frac{1}{2}\|v - v^*\|^2$ at $v = v^k$. Thus, the rationale of using the ADM based descent direction is justified.

Theorem 3.1 *Let $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ be generated by the ADM step (3.2) from the given vector $v^k = (x_2^k, \ldots, x_m^k, \lambda^k)$. Then, we have*

$$\varphi(v^k, \tilde{v}^k) = \frac{1}{2}\Big(\sum_{j=2}^m \|A_j(x_j^k - \tilde{x}_j^k)\|_H^2 + \|\lambda^k - \tilde{\lambda}^k\|_{H^{-1}}^2\Big) + \frac{1}{2}\|A_1 \tilde{x}_1^k + \sum_{j=2}^m A_j x_j^k - b\|_H^2, \tag{3.9}$$

where $\varphi(v^k, \tilde{v}^k)$ is defined by (3.5).

Proof. Recall that (3.1) implies that

$$d(v^k,\tilde v^k)=\begin{pmatrix} A_2^THA_2 & & & & \\ A_3^THA_2 & A_3^THA_3 & & & \\ \vdots & \vdots & \ddots & & \\ A_m^THA_2 & A_m^THA_3 & \cdots & A_m^THA_m & \\ 0 & 0 & \cdots & 0 & H^{-1}\end{pmatrix}\begin{pmatrix} x_2^k-\tilde x_2^k\\ x_3^k-\tilde x_3^k\\ \vdots\\ x_m^k-\tilde x_m^k\\ \lambda^k-\tilde\lambda^k\end{pmatrix}.$$

Thus, we have

$$\begin{aligned}&(v^k-\tilde v^k)^Td(v^k,\tilde v^k)\\ =\;& \frac12\begin{pmatrix} A_2(x_2^k-\tilde x_2^k)\\ A_3(x_3^k-\tilde x_3^k)\\ \vdots\\ A_m(x_m^k-\tilde x_m^k)\\ \lambda^k-\tilde\lambda^k\end{pmatrix}^T\begin{pmatrix} H & H & \cdots & H & 0\\ H & H & \cdots & H & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ H & H & \cdots & H & 0\\ 0 & 0 & \cdots & 0 & H^{-1}\end{pmatrix}\begin{pmatrix} A_2(x_2^k-\tilde x_2^k)\\ A_3(x_3^k-\tilde x_3^k)\\ \vdots\\ A_m(x_m^k-\tilde x_m^k)\\ \lambda^k-\tilde\lambda^k\end{pmatrix}\\ &+\frac12\Big(\sum_{j=2}^m\|A_j(x_j^k-\tilde x_j^k)\|_H^2+\|\lambda^k-\tilde\lambda^k\|_{H^{-1}}^2\Big).\end{aligned}\tag{3.10}$$

Now, we turn to deal with the second term of the right-hand-side of (3.5). By manipulations, we have

$$\begin{aligned}&(\lambda^k-\tilde\lambda^k)^T\Big(\sum_{j=2}^m A_j(x_j^k-\tilde x_j^k)\Big)\\ =\;& \frac12\begin{pmatrix} A_2(x_2^k-\tilde x_2^k)\\ A_3(x_3^k-\tilde x_3^k)\\ \vdots\\ A_m(x_m^k-\tilde x_m^k)\\ \lambda^k-\tilde\lambda^k\end{pmatrix}^T\begin{pmatrix} 0 & 0 & \dots & 0 & I\\ 0 & 0 & \dots & 0 & I\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \dots & 0 & I\\ I & I & \dots & I & 0\end{pmatrix}\begin{pmatrix} A_2(x_2^k-\tilde x_2^k)\\ A_3(x_3^k-\tilde x_3^k)\\ \vdots\\ A_m(x_m^k-\tilde x_m^k)\\ \lambda^k-\tilde\lambda^k\end{pmatrix}.\end{aligned}\tag{3.11}$$

Adding (3.10) and (3.11), it follows that

$$\begin{aligned}\varphi(v^k,\tilde v^k)\;=\;& \frac12\begin{pmatrix} A_2(x_2^k-\tilde x_2^k)\\ A_3(x_3^k-\tilde x_3^k)\\ \vdots\\ A_m(x_m^k-\tilde x_m^k)\\ \lambda^k-\tilde\lambda^k\end{pmatrix}^T\begin{pmatrix} H & H & \dots & H & I\\ H & H & \dots & H & I\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ H & H & \dots & H & I\\ I & I & \dots & I & H^{-1}\end{pmatrix}\begin{pmatrix} A_2(x_2^k-\tilde x_2^k)\\ A_3(x_3^k-\tilde x_3^k)\\ \vdots\\ A_m(x_m^k-\tilde x_m^k)\\ \lambda^k-\tilde\lambda^k\end{pmatrix}\\ &+\frac12\Big(\sum_{j=2}^m\|A_j(x_j^k-\tilde x_j^k)\|_H^2+\|\lambda^k-\tilde\lambda^k\|_{H^{-1}}^2\Big)\\ =\;& \frac12\Big\|\sum_{j=2}^m A_j(x_j^k-\tilde x_j^k)+H^{-1}(\lambda^k-\tilde\lambda^k)\Big\|_H^2+\frac12\Big(\sum_{j=2}^m\|A_j(x_j^k-\tilde x_j^k)\|_H^2+\|\lambda^k-\tilde\lambda^k\|_{H^{-1}}^2\Big)\\ =\;& \frac12\Big\|A_1\tilde x_1^k+\sum_{j=2}^m A_jx_j^k-b\Big\|_H^2+\frac12\Big(\sum_{j=2}^m\|A_j(x_j^k-\tilde x_j^k)\|_H^2+\|\lambda^k-\tilde\lambda^k\|_{H^{-1}}^2\Big).\end{aligned}\tag{3.12}$$

Assertion (3.9) follows from (3.12) directly. □

Since $\tilde{x}_i^k$ is the solution of (3.2), we have $\tilde{x}_i^k \in \mathcal{X}_i$ and

$$(x_i' - \tilde{x}_i^k)^T \big\{ f_i(\tilde{x}_i^k) - A_i^T [\lambda^k - H(\sum_{j=1}^{i} A\tilde{x}_j^k + \sum_{j=i+1}^{m} A_j x_j^k - b)] \big\} \geq 0, \ \ \forall \, x_i' \in \mathcal{X}_i.$$

By using (3.2), the above inequality can be written as

$$(x_i' - \tilde{x}_i^k)^T \big\{ f_i(\tilde{x}_i^k) - A_i^T \tilde{\lambda}^k + A_i^T H\big(\sum_{j=i+1}^{m} A_j (x_j^k - \tilde{x}_j^k) \big) \big\} \geq 0, \ \ \forall \, x_i' \in \mathcal{X}_i. \qquad (3.13)$$

The following lemma is the extension of Lemma 2.1 for (3.7)-(3.8).

Lemma 3.1 *Let $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ be generated by the ADM step (3.2) from the given vector $v^k = (x_2^k, \ldots, x_m^k, \lambda^k)$. Then, we have*

$$\tilde{w}^k \in \mathcal{W}, \qquad (w' - \tilde{w}^k)^T \big(d_2(v^k, \tilde{w}^k) - d_1(v^k, \tilde{v}^k) \big) \geq 0, \qquad \forall \, w' \in \mathcal{W}, \qquad (3.14)$$

where

$$d_1(v^k, \tilde{v}^k) = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ A_2^T H A_2 & 0 & \cdots & \cdots & 0 \\ A_3^T H A_2 & A_3^T H A_3 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A_m^T H A_2 & A_m^T H A_3 & \cdots & A_m^T H A_m & 0 \\ 0 & 0 & \cdots & 0 & H^{-1} \end{pmatrix} \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \qquad (3.15)$$

and

$$d_2(v^k, \tilde{w}^k) = F(\tilde{w}^k) + \begin{pmatrix} A_1^T \\ A_2^T \\ A_3^T \\ \vdots \\ A_m^T \\ 0 \end{pmatrix} H\big(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \big). \qquad (3.16)$$

Proof. It follows from (3.13) that $\tilde{x}^k \in \mathcal{X}$ and

$$\begin{pmatrix} x_1' - \tilde{x}_1^k \\ x_2' - \tilde{x}_2^k \\ x_3' - \tilde{x}_3^k \\ \vdots \\ x_m' - \tilde{x}_m^k \end{pmatrix}^T \left\{ \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T \tilde{\lambda}^k \\ f_2(\tilde{x}_2^k) - A_2^T \tilde{\lambda}^k \\ f_3(\tilde{x}_3^k) - A_3^T \tilde{\lambda}^k \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T \tilde{\lambda}^k \end{pmatrix} + \begin{pmatrix} A_1^T H\big(\sum_{j=2}^m A_j (x_j^k - \tilde{x}_j^k)\big) \\ A_2^T H\big(\sum_{j=3}^m A_j (x_j^k - \tilde{x}_j^k)\big) \\ A_3^T H\big(\sum_{j=4}^m A_j (x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ 0 \end{pmatrix} \right\} \geq 0, \ \ \forall \, x' \in \mathcal{X}. \qquad (3.17)$$

Adding

$$\begin{pmatrix} x_1' - \tilde{x}_1^k \\ x_2' - \tilde{x}_2^k \\ x_3' - \tilde{x}_3^k \\ \vdots \\ x_m' - \tilde{x}_m^k \\ \lambda' - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 0 \\ A_2^T H\big(\sum_{j=2}^2 A_j (x_j^k - \tilde{x}_j^k)\big) \\ A_3^T H\big(\sum_{j=2}^3 A_j (x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ A_m^T H\big(\sum_{j=2}^m A_j (x_j^k - \tilde{x}_j^k)\big) \\ H^{-1}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}$$

to both sides of (3.17) and using

$$\sum_{i=1}^{m} A_i\tilde{x}_i^k - b = H^{-1}(\lambda^k - \tilde{\lambda}^k),$$

we get $\tilde{w}^k \in \mathcal{W}$ and

$$\begin{pmatrix} x_1' - \tilde{x}_1^k \\ x_2' - \tilde{x}_2^k \\ x_3' - \tilde{x}_3^k \\ \vdots \\ x_m' - \tilde{x}_m^k \\ \lambda' - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T\tilde{\lambda}^k + A_1^T H\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ f_2(\tilde{x}_2^k) - A_2^T\tilde{\lambda}^k + A_2^T H\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ f_3(\tilde{x}_3^k) - A_3^T\tilde{\lambda}^k + A_3^T H\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T\tilde{\lambda}^k + A_m^T H\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ \sum_{i=1}^m A_i\tilde{x}_i^k - b \end{pmatrix}$$

$$\geq \begin{pmatrix} x_1' - \tilde{x}_1^k \\ x_2' - \tilde{x}_2^k \\ x_3' - \tilde{x}_3^k \\ \vdots \\ x_m' - \tilde{x}_m^k \\ \lambda' - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 0 \\ A_2^T H\big(\sum_{j=2}^2 A_j(x_j^k - \tilde{x}_j^k)\big) \\ A_3^T H\big(\sum_{j=2}^3 A_j(x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ A_m^T H\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ H^{-1}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}, \quad \forall\, w' \in \mathcal{W}.$$

The assertion of this lemma is proved. $\square$

Analogously as Lemma 2.2, we have the following result.

Lemma 3.2 *Let $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ be generated by the ADM step (3.2) from the given vector $v^k = (x_2^k, \ldots, x_m^k, \lambda^k)$. Then, we have*

$$(\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=2}^{m} A_j(x_j^k - \tilde{x}_j^k)\big), \quad \forall\, w^* \in \mathcal{W}^*, \tag{3.18}$$

where $d_1(v^k, \tilde{v}^k)$ is defined in (3.15).

Proof. Since $w^* \in \mathcal{W}$, it follows from (3.14) that

$$(\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k) \geq (\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k). \tag{3.19}$$

We consider the right-hand-side of (3.19) and use the notation (3.16). Because F is monotone and $\tilde{w}^k \in \mathcal{W}$, we have

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

Therefore, by using (3.16) and the above inequality, we get

$$(\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k) \geq \big(\sum_{j=2}^{m} A_j(x_j^k - \tilde{x}_j^k)\big)^T \big(H \sum_{j=1}^{m} A_j(\tilde{x}_j^k - x_j^*)\big). \tag{3.20}$$

Since

$$\sum_{j=1}^{m} A_j x_j^* = b \qquad \text{and} \qquad H(\sum_{j=1}^{m} A_j\tilde{x}_j^k - b) = \lambda^k - \tilde{\lambda}^k,$$

from (3.20) we obtain

$$(\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k) \quad \geq \quad (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=2}^{m} A_j(x_j^k - \tilde{x}_j^k)\big). \tag{3.21}$$

Substituting (3.21) into (3.19), Assertion (3.18) follows immediately. □

Since (see (3.1) and (3.15))

$$d_1(v^k, \tilde{v}^k) = \begin{pmatrix} 0 \\ d(v^k, \tilde{v}^k) \end{pmatrix}, \tag{3.22}$$

we have

$$(\tilde{v}^k - v^*)^T d(v^k, \tilde{v}^k) = (\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k)$$

and consequently from (3.18) it follows that

$$(\tilde{v}^k - v^*)^T d(v^k, \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \big(\sum_{j=2}^{m} A_j(x_j^k - \tilde{x}_j^k)\big), \quad \forall\, v^* \in \mathcal{V}^*. \tag{3.23}$$

Theorem 3.2 *Let $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ be generated by the ADM step (3.2) from the given vector $v^k = (x_2^k, \ldots, x_m^k, \lambda^k)$. Then, we have*

$$(v^k - v^*)^T d(v^k, \tilde{v}^k) \geq \varphi(v^k, \tilde{v}^k), \quad \forall\, v^* \in \mathcal{V}^*, \tag{3.24}$$

where $d(v^k, \tilde{v}^k)$ is defined in (3.1) *and $\varphi(v^k, \tilde{v}^k)$ is defined in (3.5).*

Proof. It follows directly from (3.23). □

3.4 Convergence

Similarly as in Section 2.4 and based on Theorems 3.1 and 3.2, the convergence follows in a straightforward manner. Therefore, we list the following theorems and the proofs are omitted.

Theorem 3.3 *For any $v^* \in \mathcal{V}^*$, the sequence $\{v^k\}$ generated by the proposed ADBC method for (1.4) satisfies*

$$\|v^{k+1} - v^*\|_G^2 \leq \|v^k - v^*\|_G^2 - \gamma(2-\gamma)\alpha_k^* \varphi(v^k, \tilde{v}^k), \quad \forall\, v^* \in \mathcal{V}^*.$$

Theorem 3.4 *Let $\{v^k\}$ and $\{\tilde{w}^k\}$ be the sequences generated by the proposed ADBC method for (1.4). Then, we have:*

1. *The sequence $\{v^k\}$ is bounded.*

2. $\lim_{k\to\infty}\{\|A_2(x_2^k - \tilde{x}_2^k)\|^2 + \cdots + \|A_m(x_m^k - \tilde{x}_m^k)\|^2 + \|\lambda^k - \tilde{\lambda}^k\|^2\} = 0.$

3. *Any cluster point of $\{\tilde{w}^k\}$ is a solution point of (3.7)-(3.8).*

4. *The sequence $\{\tilde{v}^k\}$ converges to some $v^\infty \in \mathcal{V}^*$ when $A_i, i = 2, \ldots, m$ are column full rank matrices.*

4 Discussions

In this section, we discuss some topics that are closely related to what we have done, including the ADBC method's comparison with some existing augmented-Lagrangian-based methods suitable for parallel computation, and its eligibility for inexact modifications.

4.1 Parallel modification

As shown in previous sections, ADM-oriented methods including the proposed ADBC method essentially split the augmented Lagrangian method with the aim of exploiting fully the favorable separable structure of the problem under consideration. To implement the proposed ADBC method, some subproblems are required to solve at each iteration, and they have to be solved in the alternative order—-not eligible for parallel computation. For the possible benefits under such circumstance that parallel computation facilities are available, it is natural to come up with the desire to solve these subproblems in parallel. Taking (2.1) as the illustrative example, then the corresponding subproblem at the $k+1$th iteration for the advantage of parallel computation will be

$$\tilde{x}^k = \text{Argmin}\{\theta_1(x) - (\lambda^k)^T Ax + \frac{1}{2}\|Ax + By^k + Cz^k - b\|_H^2 \mid x \in \mathcal{X}\}, \tag{4.1a}$$

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) - (\lambda^k)^T By + \frac{1}{2}\|Ax^k + By + Cz^k - b\|_H^2 \mid y \in \mathcal{Y}\}, \tag{4.1b}$$

$$\tilde{z}^k = \text{Argmin}\{\theta_3(z) - (\lambda^k)^T Cz + \frac{1}{2}\|Ax^k + By^k + Cz - b\|_H^2 \mid z \in \mathcal{Z}\}. \tag{4.1c}$$

Compared to the alternating subproblems (2.3a)-(2.3c), the subproblems (4.1a)-(4.1c) are ready for parallel computation once such facilities are available. In fact, this idea has inspired the so-called parallel splitting augmented Lagrangian method (PSALM) in [15] and its improvement in [20].

The difference between the PSALM and the proposed ADBC method is mainly the ways of generating the predictor $(\tilde{x}^k, \tilde{y}^k, \tilde{z}^k)$, and it can be summarized in the following table.

Difference between PSALM and ADBC

Objective	PSALM	ADBC
obtaining $\tilde{x}^k \in \mathcal{X}$	from given (y^k, z^k, λ^k)	from given (y^k, z^k, λ^k)
obtaining $\tilde{y}^k \in \mathcal{Y}$	from given (x^k, z^k, λ^k)	from available $\tilde{x}^k$ and given (z^k, λ^k)
obtaining $\tilde{z}^k \in \mathcal{Z}$	from given (x^k, y^k, λ^k)	from available $(\tilde{x}^k, \tilde{y}^k)$ and given λ^k

In terms of the terminology of numerical analysis, PSALM generates the predictor in the Jacobi fashion, while the proposed ADBC method achieves this task in the Gauss-Seidel manner by using the latest iterative information whenever possible. Another significant difference between these two types of methods is that the proposed ADBC method is eligible for being extended to the general case of (1.4) where m is an arbitrary positive integer, while PSALM cannot be extended to the case where $m > 3$, as analyzed in [15].

4.2 Inexact version

Except for some very special cases (e.g. where $H = \beta I$, $A_i^T A_i = r_i I$ and $\theta_i(x_i)'s$ have some special structure), it is not possible to solve the explicit solutions of the ADM subproblems generated by the proposed ADBC method. This difficulty urges intensive investigation on the inexact versions of the ADBC which solve the involved subproblems approximately under certain inexact criteria. More specifically, the inexact version modifies the exact ADM step (3.2) into the following inexact step:

$$\begin{cases} \tilde{x}_1^k \approx \text{Argmin}\left\{\theta_1(x_1) - (\lambda^k)^T p_1(x_1) + \frac{1}{2}\|p_1(x_1)\|_H^2 \;\middle|\; x_1 \in \mathcal{X}_1\right\}; \\ \cdots\cdots \\ \tilde{x}_i^k \approx \text{Argmin}\left\{\theta_i(x_i) - (\lambda^k)^T p_i(x_i) + \frac{1}{2}\|p_i(x_i)\|_H^2 \;\middle|\; x_i \in \mathcal{X}_i\right\}; \\ \cdots\cdots \\ \tilde{x}_m^k \approx \text{Argmin}\left\{\theta_m(x_m) - (\lambda^k)^T p_m(x_m) + \frac{1}{2}\|p_m(x_m)\|_H^2 \;\middle|\; x_m \in \mathcal{X}_m\right\}; \\ \lambda^{k+1} = \lambda^k - H(\sum_{j=1}^m A_j \tilde{x}_j^k - b), \end{cases} \tag{4.2}$$

where

$$p_i(x_i) := \sum_{j=1}^{i-1} A_j \tilde{x}_j^k + A_i x_i + \sum_{j=i+1}^{m} A_j x_j^k - b, \quad i = 1, \ldots, m.$$

Hence, judicious inexact criteria making the subproblems in (4.2) easier to deal with are expected to result in some practical inexact versions of the proposed ADBC method. According to our experience in developing inexact methods for convex programming and variational inequalities, those resulted subproblems in (4.2) usually can be solved approximately under very relaxed criteria at the expense of performing some additional correction computation (hence, prediction-correction type methods can be developed), linearizing locally the operators of $A_i's$, or regularizing the subproblems with proximal terms. We here just present the conceptual framework of the inexact version of the ADBC method without detailed analysis of convergence, for it goes beyond the scope of this paper and it deserves the length of independent papers.

5 Conclusions

Because of the attractive efficiency of the well-known alternating direction method (ADM), it is of strong desire to extend the ADM to the generally separable linearly constrained convex programming problem where both the objective function and the constraint are separable into finitely many of parts, thus broaden the applicable range of ADM significantly. The direct extension of the ADM to the general separable case, however, is generally not convergent, which we believe is mainly due to the fact that the matrix M in (3.1) is no longer symmetric.

In this paper, we make a concrete contribution along this direction by using the ADM output to the full extent. We use what the ADM outputs as the footstone to construct a contractive method which ensures the sequence of iterates to approach to the solution set monotonically in the Fejér sense. The ADM method is thus somehow extended to the desired generally separable case, and its beneficial advantages are completely preserved. Note that the compulsory additional computation to achieve the extension is relatively low. The theoretical framework of the meaningful extension is thus developed.

Our ongoing research in this regard will focus on some concrete applications of the model (1.4), where the intrinsic structure may make even practical the solution of the involved subproblems.

References

[1] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier methods*, Academic Press, 1982.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[3] E. Blum and W. Oettli, *Mathematische Optimierung, Econometrics and Operations Research XX*, Springer Verlag, 1975.

[4] G. Chen, and M. Teboulle, A proximal-based decomposition method for convex minimization problems, *Mathematical Programming*, 64, pp. 81-101, 1994.

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

[6] J. Eckstein, Some saddle-function splitting methods for convex programming, *Optimization Methods and Software*, 4, pp.75-83, 1994.

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

[8] J. Eckstein and M. Fukushima, Some reformulation and applications of the alternating direction method of multipliers, *Large Scale Optimization: State of the Art*, W. W. Hager *et al* eds., Kluwer Academic Publishers, pp. 115-134, 1994.

[9] F. Facchinei and J.-S. Pang, Finite-Dimensional Variational Inequalities and Complementarity problems, Volume I, Springer Series in Operations Research, Springer-Verlag, 2003.

[10] M. Fukushima, Application of the alternating direction method of multipliers to separable convex programming problems, *Computational Optimization and Applications*, 2, pp. 93-111, 1992.

[11] 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.

[12] 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.

[13] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

[14] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM Studies in Applied Mathematics, Philadelphia, PA, 1989.

[15] B. S. He, Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities, *Computational Optimization and Applications*, 42, 195-212, 2009.

[16] B. S. He, L. Z. Liao, D. Han and H. Yang, A new inexact alternating directions method for monontone variational inequalities, *Mathematical Programming*, 92, pp. 103-118, 2002.

[17] B. S. He, M. H. Xu and X. M. Yuan, Solving large-scale least squares covariance matrix problems by alternating direction methods, manuscript, 2009.

[18] 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.

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

[20] Z. K. Jiang and X. M. Yuan, A new parallel descent-like method for solving a class of variational inequalities, *Journal of Optimization Theory and Applications*, to appear.

[21] Z.-C. Lin, M.-M. Chen, L.-Q. Wu, Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices, manuscript, 2009.

[22] M. Ng, P. A. Weiss and X. M. Yuan, Solving constrained total-variation problems via alternating direction methods, Manuscript, 2009, available on Optimization Online.

[23] J. Nocedal and S. J. Wright, *Numerical Optimization.* Second Edition, Spriger Verlag, 2006.

[24] S. Setzer, Split Bregman algorithm, Douglas-Rachford splitting, and frame shrinkage, Report, University of Mannheim, A5, 68131 Mannheim, Germany.

[25] J. Sun and S. Zhang, A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs, manuscript, 2009.

[26] P. Tseng, Alternating projection-proximal methods for convex programming and variational inequalities, *SIAM Journal on Optimization*, 7, pp. 951-965, 1997.

[27] Z. Wen, D. Goldfarb and W. Yin, Alternating direction augmented Lagrangian methods for semidefinite programming, manuscript, 2009.

[28] J. Yang and Y. Zhang, Alternating direction algorithms for l1-problems in compressive sensing, TR09-37, CAAM, Rice University.

[29] C. H. Ye and X. M. Yuan, A descent method for structured monotone variational inequalities, *Optimization Methods and Software*, 22, pp. 329-338, 2007.

[30] X. M. Yuan, Alternating direction methods for sparse covariance selection, manuscript, 2009, available on Optimizaton Online.

[31] X. M. Yuan and J. Yang, Sparse and low rank sparse matrix decomposition via alternating directions method, manuscript, 2009, available on Optimization Online.