Alternating Direction Method of Multipliers for Linear Programming

Bingsheng He[1] and Xiaoming Yuan [2]

June 7, 2015

Abstract. Recently the alternating direction method of multipliers (ADMM) has been widely used for various applications arising in scientific computing areas. Most of these application models are, or can be easily reformulated as, linearly constrained convex minimization models with separable nonlinear objective functions. In this note we show that ADMM can also be easily used for the canonical linear programming model; and the resulting complexity is $O(mn)$ where m is the constraint number and n is the variable dimension. Moreover, at each iteration there are m subproblems that are eligible for parallel computation; and each of them only requires $O(n)$ flops. This ADMM application provides a new approach to linear programming, which is completely different from the major simplex and interior point approaches in the literature.

Key Words: Linear programming, Alternating direction method of multipliers, Parallel computation

1 Introduction

We consider the canonical linear programming (LP) model

$$\min\{c^T x \,|\, a_i^T x \geq b_i,\ i = 1, 2, \ldots, m\}, \tag{1.1}$$

where c, a_i, $i = 1, \ldots, m$ are given vectors in $\mathbb{R}^n$ and $b_i \in \mathbb{R}$. Among the most fundamental mathematical problems with various core applications in different areas, Linear programming has been intensively studied in the literature. The simplex method and interior point method represent two most important categories among different methods for LP, we refer to, e.g., [3, 7, 9] for reviews. The purpose of this short note is to show that the alternating direction method of multipliers (ADMM), which was originally proposed in [6] and has recently found many impressive applications in a broad spectrum of areas, can be easily used for the LP model (1.1). We thus aim at proposing the new ADMM approach to LP, which is completely different from the major simplex and interior point approaches in the literature.

To introduce ADMM, we consider a convex minimization problem with linear constraints and its objective function is the sum of two functions without coupled variables:

$$\min\left\{\vartheta_1(\boldsymbol{x}_1) + \vartheta_2(\boldsymbol{x}_2) \,\middle|\, \mathcal{A}_1\boldsymbol{x}_1 + \mathcal{A}_2\boldsymbol{x}_2 = \mathbf{b},\ \boldsymbol{x}_1 \in \mathcal{X}_1, \boldsymbol{x}_2 \in \mathcal{X}_2\right\}, \tag{1.2}$$

where $\mathcal{X}_i \subseteq \mathbb{R}^{s_i}$ are nonempty closed convex sets; $\vartheta_i : \mathbb{R}^{s_i} \to \mathbb{R}$ are closed proper convex functions and $\mathcal{A}_i \in \mathbb{R}^{t \times s_i}$ for $i = 1, 2$; and $\mathbf{b} \in \mathbb{R}^t$. Note that we use bold letters in (1.2) because it may be in block-wise form. For example, $\boldsymbol{x}_i$ may include more than one variable and θ_i may be the sum of more than one function.

Let $\boldsymbol{\lambda}^T \in \mathbb{R}^t$ and $\beta \in \mathbb{R}$ denote the Lagrangian multiplier and penalty parameter of (1.2), respectively. Then, the augmented Lagrangian function of (1.2) is

$$\mathcal{L}_\beta(\boldsymbol{x}_1, \boldsymbol{x}_2, \lambda) = \vartheta_1(\boldsymbol{x}_1) + \vartheta_2(\boldsymbol{x}_2) - \boldsymbol{\lambda}^T(\mathcal{A}_1\boldsymbol{x}_1 + \mathcal{A}_2\boldsymbol{x}_2 - \mathbf{b}) + \frac{\beta}{2}\|\mathcal{A}_1\boldsymbol{x}_1 + \mathcal{A}_2\boldsymbol{x}_2 - \mathbf{b}\|^2. \tag{1.3}$$

[1]Department of Mathematics, Nanjing University, Nanjing, 210093, China. This author was supported by the NSFC Grant 91130007 and 11471156. Email: hebma@nju.edu.cn

[2]Department of Mathematics, Hong Kong Baptist University, Hong Kong, China. This author was supported by a General Research Fund from Hong Kong Research Grants Council. Email: xmyuan@hkbu.edu.hk

When the ADMM proposed in [6] is applied to (1.2), the iterative scheme reads as

$$\begin{cases} \boldsymbol{x}_1^{k+1} = \arg\min\{\mathcal{L}_\beta(\boldsymbol{x}_1, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_1 \in \mathcal{X}_1\}, \\ \boldsymbol{x}_2^{k+1} = \arg\min\{\mathcal{L}_\beta(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_2 \in \mathcal{X}_2\}, \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \beta(\mathcal{A}_1\boldsymbol{x}^{k+1} + \mathcal{A}_2\boldsymbol{x}_2^{k+1} - \mathbf{b}). \end{cases} \tag{1.4}$$

We refer the reader to [1, 4, 5] for some recent review papers on ADMM; and in particular, its applications in various fields such as image processing learning, statistical learning, computer vision, wireless network, cloud computing, and so on. It worths to mention that these application models usually have nonlinear objective functions; but they are in separable form and thus the decomposition over variables often makes the subproblems in (1.4) extremely easy.

Next, we will show that after an appropriate reformulation, the LP model (1.1) can be solved by the ADMM scheme (1.4); and this application results in an iteration complexity of $O(mn)$ where m is the constraint number and n is the variable dimension. Moreover, at each iteration there are m subproblems that are eligible for parallel computation and they all have closed-form solutions. Indeed, each of these subproblems only requires $O(n)$ flops. We refer to [8] for some augmented-Lagrangian-based efforts to LP.

2 Reformulation

In this section, we reformulate the LP model (1.1) as a more favorable form so that the ADMM scheme (1.4) can be used conveniently. Without loss of generality, we assume m is odd, i.e., $m = 2l + 1$ where l is an integer. For the case where m is even, we can add a dummy constraint to make m odd (or vice versa if m is assumed to be even).

Clearly, if we introduce auxiliary variables $x_i \in \mathbb{R}^n$; and define $\theta_i(x_i) = c^T x_i$ and $X_i := \{x \in \mathbb{R}^n \mid a_i^T x \geq b_i\}$ for $i = 1, \cdots, m$, the LP model (1.1) can be written as

$$\begin{array}{ll} \min & \sum_{i=1}^m \theta_i(x_i) \\ \text{s.t.} & \begin{pmatrix} I & -I & & & & \\ & I & -I & & & \\ & & I & -I & & \\ & & & \ddots & \ddots & \\ & & & & \ddots & \ddots \\ & & & & & I \quad -I \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{m-1} \\ x_m \end{pmatrix} = 0, \\ & x_i \in X_i, \quad i = 1, \ldots, m. \end{array} \tag{2.1}$$

Let us denote by A the coefficient matrix in (2.1). Then we have

$$A = (A_1, A_2, \ldots, A_m) = \begin{pmatrix} I & -I & & & & \\ & I & -I & & & \\ & & I & -I & & \\ & & & \ddots & \ddots & \\ & & & & \ddots & \ddots \\ & & & & & I \quad -I \end{pmatrix},$$

where A_i's denote the columns of A. For these columns A_i's, we have

$$A_i^T A_j = \begin{cases} I, & \text{if} \quad i = j = 1 \quad \text{or} \quad i = j = m; \\ 2I, & \text{if} \quad 1 \neq i = j \neq m; \\ -I, & \text{if} \quad |i - j| = 1; \\ 0, & \text{otherwise.} \end{cases}$$

Further, let us regroup the variables, functions and constraint sets in (2.1) as

$$\boldsymbol{x}_1 = \begin{pmatrix} x_1 \\ x_3 \\ \vdots \\ x_m \end{pmatrix}, \quad \boldsymbol{x}_2 = \begin{pmatrix} x_2 \\ x_4 \\ \vdots \\ x_{m-1} \end{pmatrix}, \quad \mathcal{A}_1 = (A_1, A_3, \ldots, A_m), \quad \mathcal{A}_2 = (A_2, A_4, \ldots, A_{m-1}),$$

$$\vartheta_1(\boldsymbol{x}_1) = \theta_1(x_1) + \theta_3(x_3) + \cdots + \theta_m(x_m), \qquad \vartheta_2(\boldsymbol{x}_2) = \theta_2(x_2) + \theta_4(x_4) + \cdots + \theta_{m-1}(x_{m-1}),$$

and

$$\mathcal{X}_1 = X_1 \times X_3 \times \cdots \times X_m, \qquad \mathcal{X}_2 = X_2 \times X_4 \times \cdots \times X_{m-1}.$$

Then, the model (2.1) is a special case of the block-wise model (1.2) with the specifications above and $\mathbf{b} = 0$. Thus, the ADMM scheme (1.4) is applicable. Note that we additionally have the following identities:

$$\mathcal{A}_1^T \mathcal{A}_1 = \begin{pmatrix} I & & & & \\ & 2I & & & \\ & & \ddots & & \\ & & & 2I & \\ & & & & I \end{pmatrix}_{\frac{m+1}{2} \text{ blocks}}, \qquad \mathcal{A}_2^T \mathcal{A}_2 = \begin{pmatrix} 2I & & & \\ & 2I & & \\ & & \ddots & \\ & & & 2I \end{pmatrix}_{\frac{m-1}{2} \text{ blocks}},$$

and

$$A_i^T A_j = \begin{cases} 0, & \text{if} \quad |i-j| > 1; \\ -I, & \text{if} \quad |i-j| = 1. \end{cases} \tag{2.2}$$

which are actually important facts that make the implementation of the ADMM scheme (1.4) extremely easy.

3 Application of ADMM for LP

Recall we assume $m = 2l+1$ in (1.1). Moreover, to implement the ADMM (1.4) to the reformulation (2.1), the Lagrange multiplier $\boldsymbol{\lambda}$ can be denoted as

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{m-1} \end{pmatrix}.$$

3.1 Algorithm

Now, let us elucidate the k-th iteration of the ADMM scheme (1.4) when it is applied to solve the model (2.1). More specifically, starting from given $\boldsymbol{x}_2^k$ and $\boldsymbol{\lambda}^k$, the new iterate $(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2^{k+1}, \boldsymbol{\lambda}^{k+1})$ is generated by the following steps.

Step 1. With given $(\boldsymbol{x}_2^k, \boldsymbol{\lambda}^k)$, obtain

$$\boldsymbol{x}_1^{k+1} = \arg\min\{\mathcal{L}_\beta(\boldsymbol{x}_1, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_1 \in \mathcal{X}_1\}$$

via

$$\left\{\begin{array}{l} x_1^{k+1} = \arg\min\{\theta_1(x_1) + \frac{\beta}{2}\|x_1 - [x_2^k + \frac{1}{\beta}\lambda_1^k]\|^2 | x_1 \in X_1\}, \\ \text{For } p = 1,\ldots,l-1, \text{do} \\ \quad x_{2p+1}^{k+1} = \arg\min\{\theta_{2p+1}(x_{2p+1}) + \beta\|x_{2p+1} - q_p\|^2 | x_{2p+1} \in X_{2p+1}\}, \\ \text{where} \quad q_p = \frac{1}{2}[(x_{2p}^k + x_{2(p+1)}^k) + \frac{1}{\beta}(-\lambda_{2p}^k + \lambda_{2p+1}^k)]. \\ \\ x_{2l+1}^{k+1} = \arg\min\{\theta_{2l+1}(x_{2l+1}) + \frac{\beta}{2}\|x_{2l+1} - [x_{2l}^k - \frac{1}{\beta}\lambda_{2l}^k]\|^2 | x_{2l+1} \in X_{2l+1}\}. \end{array}\right. \tag{3.1a}$$

Step 2. With given $(\boldsymbol{x}_1^{k+1}, \boldsymbol{\lambda}^k)$, obtain

$$\boldsymbol{x}_2^{k+1} = \arg\min\big\{\mathcal{L}_\beta(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2, \boldsymbol{\lambda}^k) \;\big|\; \boldsymbol{x}_2 \in \mathcal{X}_2\big\}$$

via

$$\left\{\begin{array}{l} \text{For } p = 1,\ldots,l, \text{do} \\ \quad x_{2p}^{k+1} = \arg\min\{\theta_{2p}(x_{2p}) + \beta\|x_{2p} - q_p\|^2 | x_{2p} \in X_{2p}\}, \\ \text{where} \quad q_p = \frac{1}{2}[(x_{2p-1}^{k+1} + x_{2p+1}^{k+1}) + \frac{1}{\beta}(-\lambda_{2p-1}^k + \lambda_{2p}^k)]. \end{array}\right. \tag{3.1b}$$

Step 3. Update the Lagrange multipliers $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \beta(\mathcal{A}_1\boldsymbol{x}_1^{k+1} + \mathcal{A}_2\boldsymbol{x}_2^{k+1})$ via

$$\lambda_p^{k+1} = \lambda_p^k - \beta(x_p^{k+1} - x_{p+1}^{k+1}), \quad p = 1,2,\ldots,m-1. \tag{3.1c}$$

3.2 An Illustrative Example

Let us take the particular example of (1.1) with $m = 5$ and see how to implement the ADMM (1.4). That is, we consider the specific case of (2.1):

$$\begin{array}{ll} \min & \theta_1(x_1) + \theta_2(x_2) + \theta_3(x_3) + \theta_4(x_4) + \theta_5(x_5) \\ \text{s.t.} & \begin{pmatrix} I & -I & & & \\ & I & -I & & \\ & & I & -I & \\ & & & I & -I \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = 0 \\ & x_i \in X_i, \quad i = 1,\ldots,5. \end{array} \tag{3.2}$$

For the coefficient matrix A in (3.2), we have

$$A_1 = \begin{pmatrix} I \\ 0 \\ 0 \\ 0 \end{pmatrix}, \; A_2 = \begin{pmatrix} -I \\ I \\ 0 \\ 0 \end{pmatrix}, \; A_3 = \begin{pmatrix} 0 \\ -I \\ I \\ 0 \end{pmatrix}, \; A_4 = \begin{pmatrix} 0 \\ 0 \\ -I \\ I \end{pmatrix} \text{ and } A_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -I \end{pmatrix}. \tag{3.3}$$

We suggest regrouping this model as

$$\begin{array}{llcl} \min & \big(\theta_1(x_1) + \theta_3(x_3) + \theta_5(x_5)\big) & + & \big(\theta_2(x_2) + \theta_4(x_4)\big) \\ \text{s.t.} & \begin{pmatrix} I & & \\ -I & & \\ & I & \\ & & -I \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ x_5 \end{pmatrix} & + & \begin{pmatrix} -I & \\ I & \\ & -I \\ & I \end{pmatrix} \begin{pmatrix} x_2 \\ x_4 \end{pmatrix} \; = \; 0 \\ & x_1 \in X_1,\, x_3 \in X_3,\, x_5 \in X_5; & & x_2 \in X_2,\, x_4 \in X_4. \end{array} \tag{3.4}$$

Then, using these notation:

$$\boldsymbol{x}_1 = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \qquad \boldsymbol{x}_2 = \begin{pmatrix} x_2 \\ x_4 \end{pmatrix}, \quad \mathcal{A}_1 = (A_1, A_3, A_5), \qquad \mathcal{A}_2 = (A_2, A_4), \tag{3.5}$$

$$\vartheta_1(\boldsymbol{x}_1) = \theta_1(x_1) + \theta_3(x_3) + \theta_5(x_5), \qquad \vartheta_2(\boldsymbol{x}_2) = \theta_2(x_2) + \theta_4(x_4),$$

and

$$\mathcal{X}_1 = X_1 \times X_3 \times X_5, \qquad \mathcal{X}_2 = X_2 \times X_4,$$

the model (3.2) is reformulated as a special case of (1.2) with

$$\mathcal{A}_1^T\mathcal{A}_1 = \begin{pmatrix} I & & \\ & 2I & \\ & & I \end{pmatrix}_{\frac{5+1}{2}\text{ blocks}} \qquad \text{and} \quad \mathcal{A}_2^T\mathcal{A}_2 = \begin{pmatrix} 2I & \\ & 2I \end{pmatrix}_{\frac{5-1}{2}\text{ blocks}}.$$

To execute the k-th iteration of the ADMM (1.4) for the problem (3.4), we start with the given iterate

$$\boldsymbol{x}_2^k = \begin{pmatrix} x_2^k \\ x_4^k \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{\lambda}^k = \begin{pmatrix} \lambda_1^k \\ \lambda_2^k \\ \lambda_3^k \\ \lambda_4^k \end{pmatrix}.$$

Below, let us explain how to solve the resulting subproblems (3.1a) and (3.1b).

(1). Obtain $\boldsymbol{x}_1^{k+1} = \arg\min\{\mathcal{L}_\beta(\boldsymbol{x}_1, \boldsymbol{x}_2^k, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_1 \in \mathcal{X}_1\}$.

The first step of (1.4) is

$$\begin{aligned}
\begin{pmatrix} x_1^{k+1} \\ x_3^{k+1} \\ x_5^{k+1} \end{pmatrix} &= \arg\min \left\{ \begin{array}{c} \theta_1(x_1) + \theta_3(x_3) + \theta_5(x_5) + \theta_2(x_2^k) + \theta_4(x_4^k) \\ -(\boldsymbol{\lambda}^k)^T\big(A_1x_1 + A_3x_3 + A_5x_5 + (A_2x_2^k + A_4x_4^k)\big) \\ +\frac{\beta}{2}\|A_1x_1 + A_3x_3 + A_5x_5 + (A_2x_2^k + A_4x_4^k)\|^2 \end{array} \left| \begin{array}{c} x_1 \in X_1, \\ x_3 \in X_3, \\ x_5 \in X_5 \end{array} \right. \right\} \\
&= \arg\min \left\{ \begin{array}{c} \theta_1(x_1) + \theta_3(x_3) + \theta_5(x_5) \\ -(\boldsymbol{\lambda}^k)^T\big(A_1x_1 + A_3x_3 + A_5x_5\big) \\ +\frac{\beta}{2}\|\big(A_1x_1 + A_3x_3 + A_5x_5\big) + (A_2x_2^k + A_4x_4^k)\|^2 \end{array} \left| \begin{array}{c} x_1 \in X_1, \\ x_3 \in X_3, \\ x_5 \in X_5 \end{array} \right. \right\}. \qquad (3.6)
\end{aligned}$$

Note that we get the second equation in (3.6) by ignoring some constant terms in its objective function. Since the first-order optimality condition of the optimization problem (3.6) is

$$\begin{aligned}
&(x_1^{k+1}, x_3^{k+1}, x_5^{k+1}) \in X_1 \times X_3 \times X_5, \quad \begin{pmatrix} \theta_1(x_1) - \theta_1(x_1^{k+1}) \\ +\theta_3(x_3) - \theta_3(x_3^{k+1}) \\ +\theta_5(x_5) - \theta_5(x_5^{k+1}) \end{pmatrix} + \begin{pmatrix} x_1 - x_1^{k+1} \\ x_3 - x_3^{k+1} \\ x_5 - x_5^{k+1} \end{pmatrix}^T \\
&\times \left\{ \begin{pmatrix} -A_1^T\boldsymbol{\lambda}^k \\ -A_3^T\boldsymbol{\lambda}^k \\ -A_5^T\boldsymbol{\lambda}^k \end{pmatrix} + \beta \begin{pmatrix} A_1^T \\ A_3^T \\ A_5^T \end{pmatrix} [(A_1x_1^{k+1} + A_3x_3^{k+1} + A_5x_5^{k+1}) + (A_2x_2^k + A_4x_4^k)] \right\} \\
&\geq 0, \ \forall\, (x_1, x_3, x_5) \in X_1 \times X_3 \times X_5,
\end{aligned}$$

it follows from the orthogonal property (2.2) that $(x_1^{k+1}, x_3^{k+1}, x_5^{k+1}) \in X_1 \times X_3 \times X_5$, and

$$\left\{ \begin{array}{ll} \theta_1(x_1) - \theta_1(x_1^{k+1}) + (x_1 - x_1^{k+1})^T\{-A_1^T\boldsymbol{\lambda}^k + \beta A_1^T[A_1x_1^{k+1} + A_2x_2^k]\} \geq 0, & \forall x_1 \in \mathcal{X}_1; \\ \theta_3(x_3) - \theta_3(x_3^{k+1}) + (x_3 - x_3^{k+1})^T\{-A_3^T\boldsymbol{\lambda}^k + \beta A_3^T[A_3x_3^{k+1} + (A_2x_2^k + A_4x_4^k)]\} \geq 0, & \forall x_3 \in \mathcal{X}_3; \\ \theta_5(x_5) - \theta_5(x_5^{k+1}) + (x_5 - x_5^{k+1})^T\{-A_5^T\boldsymbol{\lambda}^k + \beta A_5^T[A_5x_5^{k+1} + A_4x_4^k]\} \geq 0, & \forall x_5 \in \mathcal{X}_5. \end{array} \right.$$

According to the structure A_j in (3.3), we further have $(x_1^{k+1}, x_3^{k+1}, x_5^{k+1}) \in X_1 \times X_3 \times X_5$ and

$$\left\{ \begin{array}{ll} \theta_1(x_1) - \theta_1(x_1^{k+1}) + (x_1 - x_1^{k+1})^T\{-\lambda_1^k + \beta[x_1^{k+1} - x_2^k]\} \geq 0, & \forall x_1 \in X_1; \\ \theta_3(x_3) - \theta_3(x_3^{k+1}) + (x_3 - x_3^{k+1})^T\{(\lambda_2^k - \lambda_3^k) + \beta[2x_3^{k+1} - (x_2^k + x_4^k)]\} \geq 0, & \forall x_3 \in X_3; \\ \theta_5(x_5) - \theta_5(x_5^{k+1}) + (x_5 - x_5^{k+1})^T\{\lambda_4^k + \beta[x_5^{k+1} - x_4^k]\} \geq 0, & \forall x_5 \in X_5. \end{array} \right.$$

Based on these inequalities, the solution of (3.6) can be obtained via

$$\left\{\begin{array}{lcl} x_1^{k+1} & = & \arg\min\{\theta_1(x_1) + \frac{\beta}{2}\|x_1 - [x_2^k + \frac{1}{\beta}\lambda_1^k]\|^2 | x_1 \in X_1\}, \\ x_3^{k+1} & = & \arg\min\{\theta_3(x_3) + \beta\|x_3 - \frac{1}{2}[(x_2^k + x_4^k) + \frac{1}{\beta}(-\lambda_2^k + \lambda_3^k)]\|^2 | x_3 \in X_3\}, \\ x_5^{k+1} & = & \arg\min\{\theta_5(x_5) + \frac{\beta}{2}\|x_5 - [x_4^k - \frac{1}{\beta}\lambda_4^k]\|^2 | x_5 \in X_5\}. \end{array}\right. \tag{3.7}$$

This is just the concrete form of (3.1a) for the case where $l = 2$.
(2). Obtain $\boldsymbol{x}_2^{k+1} = \arg\min\{\mathcal{L}_\beta(\boldsymbol{x}_1^{k+1}, \boldsymbol{x}_2, \boldsymbol{\lambda}^k) \mid \boldsymbol{x}_2 \in \mathcal{X}_2\}$.
The second step of the k-th iteration of (1.4) is

$$\begin{aligned} \begin{pmatrix} x_2^{k+1} \\ x_4^{k+1} \end{pmatrix} &= \arg\min \left\{ \begin{array}{c} \big(\theta_1(x_1^{k+1}) + \theta_3(x_3^{k+1}) + \theta_5(x_5^{k+1})\big) + \big(\theta_2(x_2) + \theta_4(x_4)\big) \\ -(\boldsymbol{\lambda}^k)^T\big((A_1x_1^{k+1} + A_3x_3^{k+1} + A_5x_5^{k+1} + (A_2x_2 + A_4x_4)\big) \\ +\frac{\beta}{2}\|\big(A_1x_1^{k+1} + A_3x_3^{k+1} + A_5x_5^{k+1}\big) + (A_2x_2 + A_4x_4)\|^2 \end{array} \left| \begin{array}{l} x_2 \in X_2, \\ x_4 \in X_4, \end{array} \right. \right\} \\ &= \arg\min \left\{ \begin{array}{c} \theta_2(x_2) + \theta_4(x_4) - (\boldsymbol{\lambda}^k)^T\big(A_2x_2 + A_4x_4\big) + \\ \frac{\beta}{2}\|\big(A_1x_1^{k+1} + A_3x_3^{k+1} + A_5x_5^{k+1}\big) + \big(A_2x_2 + A_4x_4\big)\|^2 \end{array} \left| \begin{array}{l} x_2 \in X_2, \\ x_4 \in X_4 \end{array} \right. \right\}. \end{aligned} \tag{3.8}$$

Again, we get the second equation in (3.8) by ignoring some constant terms in its objective function. Since the first-order optimality condition of the optimization problem (3.8) is

$$\begin{aligned} &(x_2^{k+1}, x_4^{k+1}) \in X_2 \times X_4, \quad \begin{pmatrix} \theta_2(x_2) - \theta_2(x_2^{k+1}) \\ +\theta_4(x_4) - \theta_4(x_4^{k+1}) \end{pmatrix} + \begin{pmatrix} x_2 - x_2^{k+1} \\ x_4 - x_4^{k+1} \end{pmatrix}^T \\ &\quad \times \left\{ \begin{pmatrix} -A_2^T\boldsymbol{\lambda}^k \\ -A_4^T\boldsymbol{\lambda}^k \end{pmatrix} + \beta \begin{pmatrix} A_2^T \\ A_4^T \end{pmatrix} [(A_2x_2^{k+1} + A_4x_4^{k+1}) + (A_1x_1^{k+1} + A_3x_3^{k+1} + A_5x_5^{k+1})] \right\} \\ &\quad \geq 0, \ \forall\, (x_2, x_4) \in X_2 \times X_4, \end{aligned}$$

it follows from the orthogonal property (2.2)) that $(x_2^{k+1}, x_4^{k+1}) \in X_2 \times X_4$, and

$$\left\{\begin{array}{l} \theta_2(x_2) - \theta_2(x_2^{k+1}) + (x_2 - x_2^{k+1})^T\{-A_2^T\boldsymbol{\lambda}^k + \beta A_2^T\big[A_2x_2^{k+1} + (A_1x_1^{k+1} + A_3x_3^{k+1})\big]\} \geq 0, \quad \forall x_2 \in \mathcal{X}_2; \\ \theta_4(x_4) - \theta_4(x_4^{k+1}) + (x_4 - x_4^{k+1})^T\{-A_4^T\boldsymbol{\lambda}^k + \beta A_4^T\big[A_4x_4^{k+1} + (A_3x_3^{k+1} + A_5x_5^{k+1})\big]\} \geq 0, \quad \forall x_4 \in \mathcal{X}_4. \end{array}\right.$$

According to the structure A_j in (3.3), we further have $(x_2^{k+1}, x_4^{k+1}) \in X_2 \times X_4$, and

$$\left\{\begin{array}{l} \theta_2(x_2) - \theta_2(x_2^{k+1}) \ + (x_2 - x_2^{k+1})^T\{(\lambda_1^k - \lambda_2^k) + \beta[2x_2^{k+1} - (x_1^{k+1} + x_3^{k+1})]\} \geq 0, \quad \forall x_2 \in X_2; \\ \theta_4(x_4) - \theta_4(x_4^{k+1}) \ + (x_4 - x_4^{k+1})^T\{(\lambda_3^k - \lambda_4^k) + \beta[2x_4^{k+1} - (x_3^{k+1} + x_5^{k+1})]\} \geq 0, \quad \forall x_4 \in X_4. \end{array}\right.$$

Thus, the solution of (3.8) can be obtained via

$$\left\{\begin{array}{lcl} x_2^{k+1} & = & \arg\min\{\theta_2(x_2) + \beta\|x_2 - \frac{1}{2}[(x_1^{k+1} + x_3^{k+1}) + \frac{1}{\beta}(-\lambda_1^k + \lambda_2^k)]\|^2 | x_2 \in X_2\}, \\ x_4^{k+1} & = & \arg\min\{\theta_4(x_4) + \beta\|x_4 - \frac{1}{2}[(x_3^{k+1} + x_5^{k+1}) + \frac{1}{\beta}(-\lambda_3^k + \lambda_4^k)]\|^2 | x_4 \in X_4\}. \end{array}\right. \tag{3.9}$$

This is just the concrete form of (3.1b) for the case where $l = 2$.

3.3 Subproblems

Recall that when the LP model (1.1) considered, we have

$$X_i = \{x \in \mathbb{R}^n \mid a_i^T x \geq b_i\}, \quad i = 1, \ldots, m,$$

in (2.1). It is thus clear that when the ADMM scheme (1.4) is used to solve (1.1), the computation at each iteration is dominated by the subproblems in form of

$$\min\{\|x_i - q_i\|^2 \mid x_i \in \mathbb{R}^n, a_i^T x_i \geq b_i\}, \quad i = 1, \ldots, m,$$

where $a_i, q_i \in \mathbb{R}^n$ are given vectors and $b_i \in \mathbb{R}$ is a given scalar. In this subsection, we discuss how to solve these subproblems. The main result is summarized in the following theorem.

Theorem 3.1. *For given $a, q \in \mathbb{R}^n$ and $b \in \mathbb{R}$, we have*

$$\arg\min\{\|x - q\|^2 \mid x \in \mathbb{R}^n, a^T x \geq b\} = q + \Big(\frac{b - a^T q}{a^T a}\Big)_+ a. \tag{3.10}$$

Proof. First, for any $q \in \mathbb{R}^n$, we know that

$$q_1 = p_+, \qquad q_2 = (-p)_+,$$

is the unique decomposition satisfying the following conditions:

$$q = q_1 - q_2, \qquad 0 \leq q_1 \perp q_2 \geq 0. \tag{3.11}$$

Note that the Lagrangian function of the minimization problem in (3.10) is

$$L(x, \lambda) = \|x - q\|^2 - \lambda(a^T x - b).$$

The first-order optimality condition is

$$\begin{cases} 2(x - q) - \lambda a = 0, & (3.12\text{a}) \\ 0 \leq \lambda \perp a^T x - b \geq 0. & (3.12\text{b}) \end{cases}$$

Left multiplying a^T to (3.12a), we get

$$(a^T x - b) - \frac{1}{2}\lambda a^T a = a^T q - b.$$

Using (3.12b) and (3.11), we get

$$a^T x - b = (a^T q - b)_+, \qquad \frac{1}{2}\lambda a^T a = [-(a^T q - b)]_+,$$

and thus

$$\lambda = 2\Big[\frac{b - a^T q}{a^T a}\Big]_+.$$

Substituting it into (3.12a), we obtain the assertion (3.10) immediately. □

This theorem thus shows that when the ADMM scheme (1.4) is applied to (1.1), all the resulting subproblems have closed-form solutions. This ADMM application is thus easy to be implemented. Moreover, as shown, at each iteration there are m subproblems that are eligible for parallel computation; and each of them only requires $O(n)$ flops. We summarize this complexity result in the following theorem.

Theorem 3.2. *When the ADMM scheme (1.4) is applied to (1.1) via the reformulation (2.1), at each iteration there are m subproblems that are eligible for parallel computation; and each of them only requires $O(n)$ flops.*

4 Extension

We can easily extend our previous analysis to the LP model with equality constraints:

$$\min\{c^T x \mid a_i^T x = b_i,\ i = 1, 2, \ldots, m, x \geq 0\}. \tag{4.13}$$

Indeed, we can reformulate (4.13) as

$$\begin{array}{ll} \min & \sum_{i=1}^m \theta_i(x_i) \\ \text{s.t.} & \begin{pmatrix} I & & & & -I \\ & I & & & -I \\ & & \ddots & & \vdots \\ & & & I & -I \\ & & & & I \;\; -I \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \\ y \end{pmatrix} = 0, \\ & x_i \in X_i = \{x \in \mathbb{R}^n \mid a_i^T x = b_i\}, \quad i = 1, \ldots, m; \\ & y \in Y = \{y \in \mathbb{R}^n \mid y \geq 0\}. \end{array} \tag{4.14}$$

with $\theta_i(x_i) = c^T x_i$ for $i = 1, \cdots, m$. Furthermore, (4.14) can be regrouped as

$$\boldsymbol{x}_1 = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \end{pmatrix}, \quad \boldsymbol{x}_2 = y, \qquad \mathcal{A}_1 = \begin{pmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & I \end{pmatrix} \quad \text{and} \quad \mathcal{A}_2 = \begin{pmatrix} -I \\ -I \\ \vdots \\ -I \\ -I \end{pmatrix},$$

which is a special case of (1.2) and thus the ADMM scheme (1.4) can be applied. Note that for the case (4.14), $\mathcal{A}_1$ is an identity matrix and $\mathcal{A}_2$ can be viewed a "one column" matrix. Thus, as the analysis in Section 3, the resulting subproblems are all easy. Indeed, these subproblems have the forms of

$$\min\{\|x - q\|^2 \mid x \in \mathbb{R}^n, a^T x = b\} \qquad \text{or} \qquad \min\{\|x - q\|^2 \mid x \in \mathbb{R}^n_+\},$$

whose closed-form solutions can be given by

$$q + \Big(\frac{b - a^T q}{a^T a}\Big) a \qquad \text{or} \qquad q_+,$$

respectively. Similarly as the analysis in Section 3, the ADMM scheme (1.4) for (4.13) also needs to solve m subproblems that are eligible for parallel computation, and each of them only requires $O(n)$ flops.

Remark 4.1. Note that one can artificially treat each $x_i \in \mathbb{R}$ as one block of variable and each $c_i x_i$ as one function; thus the direct extension of ADMM for a multiple-block (more than two block) can be applied to solve (4.13). However, as proved in [2], the direct extension of ADMM is not necessarily convergent.

Finally, we mention that we can extend our techniques for the LP models (1.1) and (4.13) to a more general model

$$\min\{\theta(x) \mid x \in \prod_{i=1}^m X_i\}, \tag{4.15}$$

where $\theta(x) : \mathbb{R}^n \to \mathbb{R}$ is a convex (not necessarily linear) function and X_i's are closed convex nonempty sets that are not necessarily polyhedrons given by linear equalities or inequalities. This model (4.15) includes more applications such as the feasibility set problems and some image restoration models. We omit the detail for succinctness.

5 Conclusions

We show that the alternating direction method of multipliers (ADMM) can be used to solve the canonical linear programming (LP) model; and the resulting complexity is $O(mn)$ where m is the constraint number and n is the variable dimension. Different from the dominate simplex and interior point approaches, this ADMM application is a new approach to LP and it can be very easily implemented. This is also a new application of ADMM that is different from most of its known applications in the literature. To use the ADMM, we need to introduce auxiliary variables and reformulate the LP model. The storage for variables is thus increased by m times accordingly. But, the decomposed m subproblems at each iteration are eligible for parallel computation and each of them has a closed-form solution which can be obtained by $O(n)$ flops. Thus, this new ADMM approach is particularly suitable for the LP scenario where there are many constraints and enough parallel computing infrastructure is available, while the variable is in small or modest size.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2010), pp. 1-122.

[2] C. H. Chen, B. S. He, Y. Y. Ye and X. M. Yuan, *The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent*, Math. Program., to appear.

[3] G. B. Dantzig and M. N. Thapa, Linear Programming 1: Introduction. Springer-Verlag, 1997.

[4] J. Eckstein and W. Yao, *Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results*, Pacific J. Optim., to appear.

[5] R. Glowinski, *On alternating directon methods of multipliers: a historical perspective*, Springer Proceedings of a Conference Dedicated to J. Periaux, to appear.

[6] R. Glowinski and A. Marrocco, *Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problèmes non linéaires*, R.A.I.R.O., R2 (1975), pp. 41-76.

[7] N. Karmarkar, *A new polynomial time algorithm for linear programming*, Combinatorica, 4 (4), pp.373-395, 1984.

[8] S. Wright, *Implementing proximal point methods for linear programming*, J Optim. Theory Appl., 65(3), pp. 531-554, 1990.

[9] S. Wright, Primal-Dual Interior-Point Methods, Philadelphia, SIAM, 1997.