Analysis in Theory and Applications
Anal. Theory Appl., Vol. **36**, No. 3 (2020), pp. 262-282
DOI: 10.4208/ata.OA-SU13

Study on the Splitting Methods for Separable Convex Optimization in a Unified Algorithmic Framework

Bingsheng He[1,2,*]

[1] *Department of Mathematics, Southern University of Science and Technology, Shenzhen, Guangdong 510085, China*
[2] *Department of Mathematics, Nanjing University, Nanjing, Jiangsu 210093, China*

Received 5 July 2020; Accepted (in revised version) 11 July 2020

Dedicated to Professor Weiyi Su on the occasion of her 80th birthday

Abstract. It is well recognized the convenience of converting the linearly constrained convex optimization problems to a monotone variational inequality. Recently, we have proposed a unified algorithmic framework which can guide us to construct the solution methods for solving these monotone variational inequalities. In this work, we revisit two full Jacobian decomposition of the augmented Lagrangian methods for separable convex programming which we have studied a few years ago. In particular, exploiting this framework, we are able to give a very clear and elementary proof of the convergence of these solution methods.

Key Words: Convex programming, augmented Lagrangian method, multi-block separable model, Jacobian splitting, unified algorithmic framework.

AMS Subject Classifications: 90C25, 90C30, 90C33

1 Introduction

In this paper, we consider the generic convex minimization model with linear constraints:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \sum\limits_{i=1}^{m} \theta_i(x_i) \\ \text{s.t.} & \sum\limits_{i=1}^{m} A_i x_i = b; \\ & x_i \in \mathcal{X}_i, \quad i = 1, \cdots, m, \end{array} \tag{1.1}$$

*Corresponding author. *Email address:* hebma@nju.edu.cn (B. S. He)

where $\theta_i : \Re^{n_i} \to \Re$ $(i = 1, \cdots, m)$ are closed proper convex functions and they are not necessarily smooth; $\mathcal{X}_i \subseteq \Re^{n_i}$ $(i = 1, \cdots, m)$ are closed convex sets; $A_i \in \Re^{l\times n_i}$ $(i = 1, \cdots, m)$ are given matrices; $b \in \Re^l$ is a given vector; and $\sum_{i=1}^m n_i = n$. The solution set of (1.1) is assumed to be nonempty throughout our discussion. The Lagrangian function of the problem (1.1) is

$$L(x_1, x_2, \cdots, x_m, \lambda) = \sum_{i=1}^m \theta_i(x_i) - \lambda^T \Big(\sum_{i=1}^m A_i x_i - b \Big), \tag{1.2}$$

in which $\lambda \in \Re^\ell$ is the Lagrange multiplier. By adding a penalty term to the Lagrangian function (1.2), we obtain its augmented Lagrangian function

$$\mathcal{L}_\beta(x_1, \cdots, x_m, \lambda) = \sum_{i=1}^m \theta_i(x_i) - \lambda^T \Big(\sum_{i=1}^m A_i x_i - b \Big) + \frac{\beta}{2} \Big\| \sum_{i=1}^m A_i x_i - b \Big\|^2, \tag{1.3}$$

where $\beta > 0$ is the penalty parameter for the linear constraints of (1.1). The augmented Lagrangian method (ALM) originally proposed in [11,13] for the problem (1.1) reads as

$$\begin{cases} (x_1^{k+1}, \cdots, x_m^{k+1}) = \arg\min \{ \mathcal{L}_\beta(x_1, \cdots, x_m, \lambda^k) \mid x_i \in X_i,\ i = 1, \cdots, m \}, \\ \lambda^{k+1} = \lambda^k - \beta \Big(\sum_{i=1}^m A_i x_i^{k+1} - b \Big). \end{cases} \tag{1.4}$$

The ALM plays a significant role in both theoretical study and algorithmic design for various convex programming models. ALM scheme (1.4) is indeed an application of the well-known proximal point algorithm (PPA) that can date back to the seminal work [12, 14, 15] to the dual problem of (1.1). Throughout, we call $(x_1, \cdots, x_m)$ and λ the primal and dual variables, respectively.

It is well known that ADMM [3] is powerful for the problem (1.1) when $m = 2$. In order to use the separability of the problem, one considers to use the direct extension of ADMM [3] to solve (1.1) for $m \geq 3$. It leads to the following recursion:

1.1 The direct extension of ADMM

The k-th iteration begins with a given $(x_2^k, \cdots, x_m^k, \lambda^k)$, then

$$\begin{cases} x_1^{k+1} \in \arg\min \{ \mathcal{L}_\beta(x_1, x_2^k, \cdots, x_m^k, \lambda^k) \mid x_1 \in \mathcal{X}_1 \}, \\ x_2^{k+1} \in \arg\min \{ \mathcal{L}_\beta(x_1^{k+1}, x_2, \cdots, x_m^k, \lambda^k) \mid x_2 \in \mathcal{X}_2 \}, \\ \qquad \vdots \\ x_i^{k+1} \in \arg\min \{ \mathcal{L}_\beta(x_1^{k+1}, \cdots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \cdots, x_m^k, \lambda^k) \mid x_i \in \mathcal{X}_i \}, \\ \qquad \vdots \\ x_m^{k+1} \in \arg\min \{ \mathcal{L}_\beta(x_1^{k+1}, \cdots, x_{m-1}^{k+1}, x_m, \lambda^k) \mid x_m \in \mathcal{X}_m \}, \\ \lambda^{k+1} = \lambda^k - \beta \Big(\sum_{i=1}^m A_i x_i^{k+1} - b \Big). \end{cases} \tag{1.5}$$