Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 396753, 9 pages
http://dx.doi.org/10.1155/2014/396753

Research Article

An Implementable First-Order Primal-Dual Algorithm for Structured Convex Optimization

Feng Ma, Mingfang Ni, Lei Zhu, and Zhanke Yu

College of Communications Engineering, PLA University of Science and Technology, Nanjing 210007, China

Correspondence should be addressed to Feng Ma; mafengnju@gmail.com

Received 2 December 2013; Accepted 17 February 2014; Published 30 March 2014

Academic Editor: Guanglu Zhou

Many application problems of practical interest can be posed as structured convex optimization models. In this paper, we study a new first-order primaldual algorithm. The method can be easily implementable, provided that the resolvent operators of the component objective functions are simple to evaluate. We show that the proposed method can be interpreted as a proximal point algorithm with a customized metric proximal parameter. Convergence property is established under the analytic contraction framework. Finally, we verify the efficiency of the algorithm by solving the stable principal component pursuit problem.

1. Introduction

In this paper, we consider the following separable optimization problem:

$$\min_{x\in\mathcal{X},y\in\mathcal{Y}} f(x)+g(y) \quad \text{s.t.} \quad Ax+By=b, \tag{1}$$

where $A\in\Re^{m\times n}$ and $B\in\Re^{m\times p}$ are given matrices, $b\in\Re^m$ is a given vector, and $\mathcal{X}\subset\Re^n$ and $\mathcal{Y}\subset\Re^p$ are nonempty closed convex sets. $f:\Re^n\to\Re$ and $g:\Re^p\to\Re$ are convex functions (not necessarily smooth). Throughout this paper, we assume that the solution set of (1) is nonempty. Problem of this type arises in many applications, ranging from machine learning to compressed sensing. We refer to, for example, [1–9], for a few examples of applications.

To solve (1), one can use the classical augmented Lagrangian method (ALM). Starting from any initial iterate (x^0,y^0,λ^0), ALM iterates via the following procedure:

$$\left(x^{k+1},y^{k+1}\right)=\arg\min_{x,y}\mathcal{L}_{\mathcal{A}}\left(x,y;\lambda^k\right), \tag{2a}$$

$$\lambda^{k+1}=\lambda^k-\beta\left(Ax^{k+1}+By^{k+1}-b\right), \tag{2b}$$

where the augmented Lagrangian function $\mathcal{L}_{\mathcal{A}}(x,y;\lambda)$ associated with the problem (1) is given by

$$\mathcal{L}_{\mathcal{A}}(x,y,\lambda)=f(x)+g(y)-\lambda^T(Ax+By-b)+\frac{\beta}{2}\|Ax+By-b\|^2, \tag{3}$$

and $\lambda\in\Re^m$ is the Lagrangian multiplier vector associated with the linear constraint and $\beta>0$ is a penalty parameter for the violation of the linear constraint. ALM enjoys very nice convergence and has been shown to be equivalent to a proximal point algorithm applied to the dual of (1) [10]. A noticeable feature of ALM is that it treats (1) as a generic minimization problem and ignores completely the nice separable structure emerging in the objective function. The minimizations of the two functions f and g in (2a) are strongly coupled because of the quadratic term $(\beta/2)\|Ax+By-b\|^2$. Hence, the implementation of ALM ((2a) and (2b)) can be computationally challenging. To utilize the separable structure of the problem, the well-known alternating direction method (ADM) essentially splits the ALM subproblem into two subproblems with respect to

x and y in Gauss-Seidel manner. More specifically, at each iteration, ADM takes the following form:

$$x^{k+1} = \arg\min_{x\in\mathcal{X}} \mathcal{L}_{\mathcal{A}}\left(x, y^k; \lambda^k\right), \tag{4a}$$

$$y^{k+1} = \arg\min_{y\in\mathcal{Y}} \mathcal{L}_{\mathcal{A}}\left(x^{k+1}, y; \lambda^k\right), \tag{4b}$$

$$\lambda^{k+1} = \lambda^k - \beta\left(Ax^{k+1} + By^{k+1} - b\right). \tag{4c}$$

ADM can be interpreted as Douglas-Rachford splitting method applied to the dual problem [11] and proximal point method [12]. We refer to, for example, [13–17] for recent study of ADM and its variants. Comparing to ALM, ADM minimizes (3) with respect to x and y alternatingly at each iteration, rather than with respect to both x and y simultaneously. This splitting procedure makes it possible to exploit the special separable structure of the objective functions. Thus, ADM appears to be a natural fit for solving very large scale distributed machine learning and big-data related optimization problems.

When A and B in (1) are both identity matrices, ADM can be very efficient. The first two subproblems ((4a) and (4b)) correspond to evaluating the resolvent operators of component functions f and g, respectively. Here, the *proximal operator* of the function $\theta : \mathfrak{R}^n \to \mathfrak{R}$ is defined by

$$u := \mathbf{prox}_{\xi\theta}(a) = \arg\min\left\{\theta(u) + \frac{1}{2\xi}\|u-a\|^2 \mid u \in \mathfrak{R}^n\right\}, \tag{5}$$

where $a \in \mathfrak{R}^n$ and $\xi > 0$. In most popular applications of sparse optimization, the proximal operator can be computed exactly and efficiently (e.g., $\theta(x) = \|x\|_1 := \sum_{i=1}^n |x_i|$ or $\theta(x) = \|x\|_2 := \sqrt{\sum_{i=1}^n |x_i|^2}$). While when A (resp., B) is not an identity matrix, the resulting ADM subproblems may not be easily solvable, since they involve inverting of A (resp., B), and there is no efficient way of doing so directly. This difficulty could result in inefficiency of the ADM greatly. As a result, first-order algorithms that preserves both the alternating computation feature of ADM and the advantages of not involving the inverse of A and B are highly desirable.

In [18, 19], an alternating proximal gradient method (APGM) for solving (1) was given. APGM is based on the framework of ADM, which solves the first two subproblems ((4a) and (4b)) inexactly by taking one proximal gradient step. More specifically, APGM generates the new iterate $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ via the following procedure:

$$x^{k+1} = \arg\min_{x\in\mathcal{X}} f(x) + \frac{r}{2}\left\|x - x^k - \frac{1}{r}A^T\left(\lambda^k - \beta\left(Ax^k + By^k - b\right)\right)\right\|^2, \tag{6a}$$

$$y^{k+1} = \arg\min_{y\in\mathcal{Y}} g(y) + \frac{s}{2}\left\|y - y^k - \frac{1}{s}B^T\left(\lambda^k - \beta\left(Ax^{k+1} + By^k - b\right)\right)\right\|^2, \tag{6b}$$

$$\lambda^{k+1} = \lambda^k - \beta\left(Ax^{k+1} + By^{k+1} - b\right), \tag{6c}$$

where $\beta > 0$, $r > \beta\|A^TA\|$, and $s > \beta\|B^TB\|$.

In this paper, we also focus on the specific scenario of (1), where both A and B are not identity matrices. We propose a new first-order primal-dual algorithm for (1). We show that our algorithm can be viewed as a customized proximal point method with a special proximal regularization parameter, and it is within the framework of the contraction type methods. The proposed algorithm has three main advantages: first, it can be easily implementable, the main computational effort of each iteration is to evaluate the proximal operators of the component objective functions; second, the involved subproblems are solved consecutively in the ADM manner, which makes the algorithm amenable to distributed optimization. Finally, it only uses the first-order information and does not require any matrix inversion or solving linear systems. Hence, the method is well suited for solving large scale distributed optimization problems.

The paper is organized as follows. In Section 2, we review some preliminaries. In Sections 3 and 4, we present the new method and analyze its convergence. In Section 5 we conduct numerical experiments to compare the proposed method with APGM for solving the stable principal component pursuit problem. We conclude the paper in Section 6.

2. Preliminaries

2.1. Variational Characterization of (1). In this section, we reformulate (1) as a variational form, which is useful for succedent algorithmic illustration and convergence analysis.

The Lagrangian function associated with (1) is

$$\mathcal{L}(x, y, \lambda) = f(x) + g(y) - \lambda^T(Ax + By - b), \tag{7}$$

where $\lambda \in \mathfrak{R}^m$ is a Lagrangian multiplier. According to the previous convex assumption of (1), finding optimal solutions of (1) and its dual form is equivalent to finding a saddle point of $\mathcal{L}$. More precisely, let (x^*, y^*, λ^*) be a saddle point of $\mathcal{L}$. We have

$$\mathcal{L}_{\lambda\in\mathfrak{R}^m}(x^*, y^*, \lambda) \le \mathcal{L}(x^*, y^*, \lambda^*) \le \mathcal{L}_{x\in\mathcal{X}, y\in\mathcal{Y}}(x, y, \lambda^*). \tag{8}$$

Then we can directly read off the optimality conditions with variational characterization. More specifically, $w^* = (x^*, y^*, \lambda^*) \in \Omega$ is a saddle point of $\mathcal{L}$ if and only if it satisfies the following mixed variational inequality (VI):

$$h(u) - h(u^*) + \left(w' - w^*\right)^T F(w^*) \ge 0, \quad \forall w' \in \Omega, \tag{9a}$$

with

$$u := \begin{pmatrix} x \\ y \end{pmatrix}, \qquad w := \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \qquad h(u) := f(x) + g(y),$$

$$F(w) := \begin{pmatrix} -A^T\lambda \\ -B^T\lambda \\ Ax + By - b \end{pmatrix}, \tag{9b}$$

$$\Omega = \mathcal{X} \times \mathcal{Y} \times \Re^m. \tag{9c}$$

Hence, a solution of ((9a)–(9c)) yields a solution of (1).

Note that the mapping $F(\cdot)$ is said to be *monotone* with respect to Ω if

$$\left(F(w) - F\left(w'\right)\right)^T \left(w - w'\right) \geq 0, \quad \forall w, w' \in \Omega. \tag{10}$$

Consequently, it can be easily verified that $F(w)$ is monotone. Under the aforementioned nonempty assumption on the solution set of (1), the solution set of ((9a)–(9c)), denoted by $\mathcal{W}^*$, is also nonempty.

2.2. Proximal Point Algorithmic Framework. In this subsection, we review the classical proximal point algorithm (PPA) for solving the VI ((9a)–(9c)).

PPA, which was proposed by Martinet [20] and further studied by Rockafellar [10], plays a fundamental rule in optimization. For given iterate $w^k \in \Omega$, PPA generates the new iterative w^{k+1} via the following procedure:

$$\begin{aligned} &h\left(u'\right) - h\left(u^{k+1}\right) + \left(w' - w^{k+1}\right)^T \\ &\quad \times \left(F\left(w^{k+1}\right) + G\left(w^{k+1} - w^k\right)\right) \geq 0, \quad \forall w' \in \Omega, \end{aligned} \tag{11}$$

where G is a positive definite matrix, playing the role of proximal regularization parameter. A simple choice of G is that $G = \beta \cdot I$ where $\beta > 0$ and I is the identity matrix, regularizing the proximal terms $w^{k+1} - w^k$ in the uniform way. We refer the reader to example [21–24] for some special choices of G in different scenarios.

3. The Main Algorithm

3.1. Assumption. Before we present our new algorithm, we need to make the following assumption:

Assumption 1. For any give $a \in \Re^{n+p}$ and $\xi > 0$, the *proximal operator* of $h(u)$ (see (5)) has a closed-form solution or it can be efficiently solved up to a high precision.

Whenever the assumption holds, we say that the proximal operator of h is "easy" to evaluate. Note that h is separable across two variables, that is, $h = f + g$; according to the definition (5), we have

$$\mathbf{prox}_{\xi h}(b, c) = \left(\mathbf{prox}_{\xi f}(b), \mathbf{prox}_{\xi g}(c)\right), \tag{12}$$

where $b \in \Re^n$ and $c \in \Re^p$. Hence, under the assumption, the proximal operators of the component objective functions f and g are also "easy" to evaluate.

3.2. Motivation. The motivation for our algorithm is directly related to the linearized augmented Lagrangian method proposed in [25] and the customized proximal point algorithm proposed in [21] for convex problems with linear constraints.

In order to obtain a closed-form solution of x, we first add a proximal regularization parameter $G = \begin{pmatrix} rI & 0 & A^T \\ 0 & 0 & 0 \\ A & 0 & (1/\beta)I \end{pmatrix}$ to the variational characterization of the problem ((9a)–(9c)). Then, like the algorithm in [21], we get the following proximal point algorithm:

$$x^{k+1} = \arg\min_{x \in \mathcal{X}} f(x) + \frac{r}{2}\left\| x - x^k - \frac{1}{r}A^T\lambda^k \right\|^2, \tag{13a}$$

$$y^{k+1} = \arg\min_{y \in \mathcal{Y}} g(y) + \frac{\beta}{2}\left\| A\left(2x^{k+1} - x^k\right) + By - b - \frac{\lambda^k}{\beta} \right\|^2, \tag{13b}$$

$$\lambda^{k+1} = \lambda^k - \beta\left(Ax^{k+1} + By^{k+1} - b\right). \tag{13c}$$

However, (13b) is still not implementable. Inspired in [25], in order to alleviate the computation required by y-subproblems, we try to linearize the quadratic term in (13b) by

$$\begin{aligned} &\frac{\beta}{2}\left\| A\left(2x^{k+1} - x^k\right) + By - b - \frac{\lambda^k}{\beta} \right\|^2 \\ &\quad \approx \frac{\beta}{2}\left\| A\left(2x^{k+1} - x^k\right) + By^k - b - \frac{\lambda^k}{\beta} \right\|^2 \\ &\quad + \beta\left(y - y^k\right)^T g^k + \frac{s}{2}\left\| y - y^k \right\|, \end{aligned} \tag{14}$$

where $s > 0$ is a proximal parameter, and $g^k = B^T(A(2x^{k+1} - x^k) + By^k - b - \lambda^k/\beta)$ is the gradient of $(1/2)\|A(2x^{k+1} - x^k) + By - b - \lambda^k/\beta\|^2$ at y^k. With a simple manipulation, we get the following approximation to (13b):

$$\begin{aligned} y^{k+1} = \arg\min_{y \in \mathcal{Y}} &g(y) + \frac{s}{2}\left\| y - y^k - \frac{1}{s}B^T\left(\lambda^k - \beta\right.\right. \\ &\left.\left. \times \left(A\left(2x^{k+1} - x^k\right) + By^k - b\right)\right)\right\|^2. \end{aligned} \tag{15}$$

In this case, the closed-form solutions of the resulting subproblems can be easily obtained.

3.3. Description of the Algorithm. In this section, we formally present Algorithm 1.

Remark 2. From the algorithm, we can see that the minimizations ((∗) and (∗∗)) each requires the evaluation of the proximal operators for the component objective functions f and g. It is clear that the implementation of the proposed method is simple under our assumption.

The proposed algorithm for (1)
Step 0. Input $(x^0, y^0, \lambda^0) \in \mathcal{X} \times \mathcal{Y}$.
Step 1. Set

$$x^{k+1} = \arg\min_{x\in\mathcal{X}} f(x) + \frac{r}{2}\left\|x - x^k - \frac{1}{r}A^T\lambda^k\right\|^2, \quad (*)$$

$$y^{k+1} = \arg\min_{y\in\mathcal{Y}} g(y) + \frac{s}{2}\left\|y - y^k - \frac{1}{s}B^T\left(\lambda^k - \beta\left(A\left(2x^{k+1} - x^k\right) + By^k - b\right)\right)\right\|^2, \quad (**)$$

$$\lambda^{k+1} = \lambda^k - \beta\left(A\left(2x^{k+1} - x^k\right) + By^{k+1} - b\right), \quad (***)$$

where $\beta > 0$, and $r > \beta\left\|A^TA\right\|, s > \beta\left\|B^TB\right\|$.
Step 2. If a termination criterion is not met, go to Step 1.

Algorithm 1

4. Global Convergence

In this section, we show that the proposed method is in some sense equivalent to the proximal point algorithm with a special proximal regularization parameter. Then its convergence can be easily established under the analytic framework of contraction type methods.

Lemma 3. *Let* $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1})^T$ *be generated by* $((*)$–$(***))$ *from a given* $w^k = (x^k, y^k, \lambda^k)^T$ *and* $H = \begin{pmatrix} rI & 0 & A^T \\ 0 & sI-\beta B^TB & 0 \\ A & 0 & (1/\beta)I \end{pmatrix}$. *Then*

$$h(u') - h\left(u^{k+1}\right) + \left(w' - w^{k+1}\right)^T \times \left(F\left(w^{k+1}\right) + H\left(w^{k+1} - w^k\right)\right) \geq 0, \quad \forall w' \in \Omega. \quad (16)$$

Proof. First, by deriving the optimality condition for the subproblem $(*)$, we have

$$f(x) - f\left(x^{k+1}\right) + \left(x' - x^{k+1}\right)^T\left(r\left(x^{k+1} - x^k\right) - A^T\lambda^k\right) \geq 0, \quad \forall x' \in \mathcal{X}. \quad (17)$$

It can be further rewritten as

$$f(x) - f\left(x^{k+1}\right) + \left(x' - x^{k+1}\right)^T \times \left(-A^T\lambda^{k+1} + r\left(x^{k+1} - x^k\right) + A^T\left(\lambda^{k+1} - \lambda^k\right)\right) \geq 0, \quad \forall x' \in \mathcal{X}. \quad (18)$$

Similarly, the optimality condition for the subproblem $(**)$ is

$$g(y) - g\left(y^{k+1}\right) + \left(y' - y^{k+1}\right)^T\left(s\left(y^{k+1} - y^k\right) - B^T\left(\lambda^k - \beta\left(A\left(2x^{k+1} - x^k\right) + By^k - b\right)\right)\right) \geq 0, \quad \forall y' \in \mathcal{Y}. \quad (19)$$

It can be further rewritten as

$$g(y) - g\left(y^{k+1}\right) + \left(y' - y^{k+1}\right)^T\left(\left(s - \beta B^TB\right)\left(y^{k+1} - y^k\right) - B^T\left(\lambda^k - \beta\left(A\left(2x^{k+1} - x^k\right) + By^{k+1} - b\right)\right)\right) \geq 0, \quad \forall y' \in \mathcal{Y}. \quad (20)$$

Substituting $(***)$ into (19), we have

$$g(y) - g\left(y^{k+1}\right) + \left(y' - y^{k+1}\right)^T \times \left(-B^T\lambda^{k+1} + \left(s - \beta B^TB\right)\left(y^{k+1} - y^k\right)\right) \geq 0, \quad \forall y' \in \mathcal{Y}. \quad (21)$$

In addition, it follows from $(***)$ that

$$Ax^{k+1} + By^{k+1} - b + A\left(x^{k+1} - x^k\right) + \frac{1}{\beta}\left(\lambda^{k+1} - \lambda^k\right) = 0. \quad (22)$$

Combining (18), (21), and (22) together, we get

$$h(u) - h\left(u^{k+1}\right) + \begin{pmatrix} x - x^{k+1} \\ y - y^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}\left\{\begin{pmatrix} -A^T\lambda^{k+1} \\ -B^T\lambda^{k+1} \\ Ax^{k+1} + By^{k+1} - b \end{pmatrix} + \begin{pmatrix} r\left(x^{k+1} - x^k\right) + A^T\left(\lambda^{k+1} - \lambda^k\right) \\ \left(s - \beta B^TB\right)\left(y^{k+1} - y^k\right) \\ A\left(x^{k+1} - x^k\right) + \frac{1}{\beta}\left(\lambda^{k+1} - \lambda^k\right) \end{pmatrix}\right\} \geq 0, \quad (23)$$

with $w = (x, y, \lambda)^T \in \Omega$. Using the notation of H, the assertion (16) is proved. □

Note that when $\beta > 0$ and $r > \beta\|A^TA\|$, $s > \beta\|B^TB\|$, H is a positive definite matrix. Lemma 3 implies that the proposed algorithm can be viewed as a customized version of the PPA, where H is the metric proximal parameter. Henceforth, we denote the proposed algorithm as CPPA. Now we state and prove the contractive property of our algorithm.

Lemma 4. *Suppose the condition*

$$\beta > 0, \quad r > \beta\left\|A^TA\right\|, \quad s > \beta\left\|B^TB\right\| \tag{24}$$

holds. Then, for any $w^* = (x^*, y^*, \lambda^*)^T \in \Omega^*$, the sequence $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1})^T$ generated by ((∗)–(∗ ∗ ∗)) satisfies the following inequality:

$$\left\|w^{k+1} - w^*\right\|_H^2 \le \left\|w^k - w^*\right\|_H^2 - \left\|w^{k+1} - w^k\right\|_H^2, \tag{25}$$

where the norm $\|\cdot\|_H^2$ is defined as $\|w\|_H^2 = w^THw$.

Proof. Note that $w^* \in \Omega$; it follows from (16) that

$$\begin{aligned} &h(u^*) - h\left(u^{k+1}\right) + \left(w^* - w^{k+1}\right)^T \\ &\quad \times \left(F\left(w^{k+1}\right) + H\left(w^{k+1} - w^k\right)\right) \ge 0, \end{aligned} \tag{26}$$

which can be further written as

$$\begin{aligned} &\left(w^* - w^{k+1}\right)^T H\left(w^{k+1} - w^k\right) \\ &\quad \ge h\left(u^{k+1}\right) - h(u^*) + \left(w^{k+1} - w^*\right)^T F\left(w^{k+1}\right). \end{aligned} \tag{27}$$

On the other hand, using the fact that $F(\cdot)$ is a monotone operator, we have

$$\left(w^{k+1} - w^*\right)^T \left(F\left(w^{k+1}\right) - F(w^*)\right) \ge 0. \tag{28}$$

Combining (28) and (27), we have

$$\begin{aligned} &\left(w^* - w^{k+1}\right)^T H\left(w^{k+1} - w^k\right) \\ &\quad \ge h\left(u^{k+1}\right) - h(u^*) + \left(w^{k+1} - w^*\right)^T F(w^*). \end{aligned} \tag{29}$$

Since w^* is a solution of (9a) and $w^{k+1} \in \Omega$, we have

$$h\left(u^{k+1}\right) - h(u^*) + \left(w^{k+1} - w^*\right)^T F(w^*) \ge 0. \tag{30}$$

Thus, we get

$$\left(w^* - w^{k+1}\right)^T H\left(w^{k+1} - w^k\right) \ge 0. \tag{31}$$

Since H is positive definite under the requirements of the parameters (24), (31) can be further written as

$$\left(w^* - w^k\right)^T H\left(w^{k+1} - w^k\right) \ge \left\|w^{k+1} - w^k\right\|_H^2. \tag{32}$$

On the other hand,

$$\begin{aligned} \left\|w^{k+1} - w^*\right\|_H^2 &= \left\|w^{k+1} - w^k + w^k - w^*\right\|_H^2 \\ &= \left\|w^k - w^*\right\|_H^2 + 2\left(w^k - w^*\right)^T H\left(w^{k+1} - w^k\right) \\ &\quad + \left\|w^{k+1} - w^k\right\|_H^2. \end{aligned} \tag{33}$$

Inserting (32) into (33), we obtain

$$\left\|w^{k+1} - w^*\right\|_H^2 \le \left\|w^k - w^*\right\|_H^2 - \left\|w^{k+1} - w^k\right\|_H^2. \tag{34}$$

This completes the proof. □

Corollary 5. *Let w^* be an arbitrary point in Ω^*, and let the sequence $\{w^k\}$ be generated by ((∗)–(∗ ∗ ∗)). Then*

(1) *the sequence $\{w^k\}$ is bounded;*

(2) *the sequence $\{\|w^k - w^*\|_H\}$ is nonincreasing;*

(3) $\lim_{k\to\infty}\{\|w^{k+1} - w^k\|_H\} = 0$.

We are now ready to prove the global convergence of the new method.

Theorem 6. *Let the sequence $\{w^k\}$ be generated by the proposed algorithm and the condition (24) holds. Then the sequence $\{w^k\}$ converges to an optimal primal-dual solution for (1) from any starting point.*

Proof. It follows from Corollary 5 that

$$\begin{aligned} &\lim_{k\to\infty}\left\|x^{k+1} - x^k\right\| = 0, \qquad \lim_{k\to\infty}\left\|y^{k+1} - y^k\right\| = 0, \\ &\lim_{k\to\infty}\left\|\lambda^{k+1} - \lambda^k\right\| = 0. \end{aligned} \tag{35}$$

Since $\{w^k\}$ is bounded, it has at least one cluster point. Suppose w^k has a subsequence $\{w^{k_j}\}$ that converges to $w^\infty = \{x^\infty, y^\infty, \lambda^\infty\}$. It follows from (23) that

$$\begin{aligned} &\lim_{j\to\infty} h(u) - h\left(u^{k_j}\right) \\ &\quad + \begin{pmatrix} x - x^{k_j} \\ y - y^{k_j} \\ \lambda - \lambda^{k_j} \end{pmatrix}^T \left\{\begin{pmatrix} -A^T\lambda^{k_j} \\ -B^T\lambda^{k_j} \\ Ax^{k_j} + By^{k_j} - b \end{pmatrix}\right\} \ge 0. \end{aligned} \tag{36}$$

Consequently, we have

$$h(u) - h\left(u^\infty\right) + \begin{pmatrix} x - x^\infty \\ y - y^\infty \\ \lambda - \lambda^\infty \end{pmatrix}^T \left\{\begin{pmatrix} -A^T\lambda^\infty \\ -B^T\lambda^\infty \\ Ax^\infty + By^\infty - b \end{pmatrix}\right\} \ge 0, \tag{37}$$

which implies that $w^\infty \in \Omega^*$.

Because $\lim_{k\to\infty}\{\|w^{k+1} - w^k\|_H\} = 0$, for any given $\epsilon > 0$, there exists $l_0 > 0$ such that

$$\left\|w^{k_{l+1}} - w^{k_l}\right\|_H < \frac{\epsilon}{2}, \quad \forall l > l_0. \tag{38}$$

Since $w^{k_j} \to w^\infty$, for the $\epsilon > 0$ given above, there is an integer $k_l > l_0$, such that,

$$\left\|w^{k_l} - w^\infty\right\|_H < \frac{\epsilon}{2}. \tag{39}$$

Therefore, for any $k > k_l$, it follows from (38) and (39) that

$$\begin{aligned}\left\|w^k - w^\infty\right\|_H &\le \left\|w^{k_l} - w^\infty\right\|_H \\ &\le \left\|w^{k_{l+1}} - w^{k_l}\right\|_H + \left\|w^{k_{l+1}} - w^\infty\right\|_H < \epsilon.\end{aligned} \tag{40}$$

This implies that the sequence $\{w^k\}$ converges to $w^\infty \in \Omega^*$. □

5. Application to the Stable Principal Component Pursuit Problem

In this section, we apply the proposed method to solve the stable principal component pursuit problem with nonnegative constraints (SPCP). The problem tested is from Example 2 of [19]. Our code was written in Matlab R2009b and all experiments were performed on a laptop with Intel Core 2 Duo @ 2.0 GHz CPU and 2 GB of memory.

Let $M = L+S+Z$ be a given observation matrix, where L is a low-rank and non-negative matrix, S is a sparse matrix, Z is a noise matrix. SPCP arising from image processing seeks to recover L and S by solving the following nonsmooth convex optimization problem:

$$\begin{aligned}&\min_{L,S,Z} \|L\|_* + \rho\|S\|_1 + \mathcal{J}\left(\|Z\|_F \le \sigma\right) + \mathcal{J}\left(L \ge 0\right) \\ &\text{s.t.} \quad L + S + Z = M,\end{aligned} \tag{41}$$

where $\|\cdot\|_*$ is the nuclear norm (defined as the sum of all singular values), $\|\cdot\|_1$ and $\|\cdot\|_F$ denote the l_1 norm and the Frobenius norm of a matrix, respectively, and $\mathcal{J}(\cdot)$ is the indicator function for the nonnegative orthant $\Re^{m\times n}$. In the above model, Z denotes the noise matrix and ρ and σ are some fixed parameters.

Following the procedure described in [19], by introducing an auxiliary variable K and grouping L and S as one big block $[L;S]$ and grouping Z and K as another big block $[Z;K]$, model (41) can be easily reformulated as

$$\begin{aligned}&\min_{L,S,Z,K} \|L\|_* + \rho\|S\|_1 + \mathcal{J}\left(\|Z\|_F \le \sigma\right) + \mathcal{J}\left(K \ge 0\right) \\ &\text{s.t.} \quad \begin{pmatrix} I & I \\ I & 0 \end{pmatrix}\begin{pmatrix} L \\ S \end{pmatrix} + \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}\begin{pmatrix} Z \\ K \end{pmatrix} = \begin{pmatrix} M \\ 0 \end{pmatrix}.\end{aligned} \tag{42}$$

which fits the setting of (1). The proposed algorithm ((∗)–(∗ ∗ ∗)) is therefore applicable for (42), and we obtain the following iterative scheme:

$$L^{k+1} = \arg\min \|L\|_* + \frac{r}{2}\left\|L - L^k - \frac{1}{r}\left(\Lambda_1^k + \Lambda_2^k\right)\right\|_F^2, \tag{43}$$

$$S^{k+1} = \arg\min \rho\|S\|_1 + \frac{r}{2}\left\|S - S^k - \frac{1}{r}\Lambda_1^k\right\|_F^2, \tag{44}$$

$$\begin{aligned}Z^{k+1} &= \arg\min \mathcal{J}\left(\|Z\|_F \le \sigma\right) \\ &\quad + \frac{s}{2}\left\|Z - Z^k - \frac{1}{s}\left(\Lambda_1^k - \beta\left(2L^{k+1} - L^k + 2S^{k+1} - S^k + Z^k - K^k - M\right)\right)\right\|_F^2,\end{aligned} \tag{45}$$

$$\begin{aligned}K^{k+1} &= \arg\min \mathcal{J}\left(K \ge 0\right) \\ &\quad + \frac{s}{2}\left\|K - K^k + \frac{1}{s}\left(\Lambda_2^k - \beta\left(2L^{k+1} - L^k - K^k\right)\right)\right\|_F^2,\end{aligned} \tag{46}$$

$$\Lambda_1^{k+1} = \Lambda_1^k - \beta\left(2L^{k+1} - L^k + 2S^{k+1} - S^k + Z^{k+1} - M\right), \tag{47}$$

$$\Lambda_2^{k+1} = \Lambda_2^k - \beta\left(2L^{k+1} - L^k - K^{k+1}\right). \tag{48}$$

The main advantage of CPPA applied to SPCP is that the generated minimizations in ((43)–(46)) are all simple enough to have closed-form solutions. For completeness, we elaborate on the strategy of solving the resulted subproblems at each iteration.

(i) The L-subproblem (43) amounts to evaluate the proximal operator of the nuclear norm function and is given by the matrix shrinkage operation:

$$L^{k+1} := \text{MatShrink}\left(L^k + \frac{1}{r}\left(\Lambda_1^k + \Lambda_2^k\right), \frac{1}{r}\right), \tag{49}$$

where the matrix shrinkage operator $\text{MatShrink}(M,\xi)$ $(\xi > 0)$ is defined as

$$\text{MatShrink}(M,\xi) := U\text{Diag}\left(\max\{\sigma - \xi, 0\}\right)V^T, \tag{50}$$

and $U\text{Diag}(\sigma)V^T$ is the SVD of matrix M.

(ii) The closed-form solution of S-subproblem (44) can be given by the l_1 shrinkage operation:

$$S^{k+1} := \text{Shrink}\left(S^k + \frac{1}{r}\Lambda_1^k, \frac{\rho}{r}\right), \tag{51}$$

where the l_1 shrinkage operator $\text{Shrink}(M,\xi)$ is defined as

$$\left[\text{Shrink}(M,\xi)\right]_{ij} := \begin{cases} M_{ij} - \xi, & \text{if } M_{ij} > \xi \\ M_{ij} + \xi, & \text{if } M_{ij} < -\xi \\ 0, & \text{if } \left|M_{ij}\right| \le \xi. \end{cases} \tag{52}$$

(iii) The Z-subproblem (45) amounts to projecting the matrix $W^k := M + (1/\beta)\Lambda_1^k - (2L^{k+1} - L^k + 2S^{k+1} - S^k)$ onto the Euclidean ball $\|Z\|_F \le \sigma$, whose closed-form solution is given by

$$Z^{k+1} := \frac{W^k}{\max\left\{1, \left\|W^k\right\|_F/\sigma\right\}}. \tag{53}$$

Table 1: Numerical results for stable principal component pursuit problem.

n	APGM				CPPA			
	Iter.	rel_L	rel_S	CPU(s)	Iter.	rel_L	rel_S	CPU(s)
				$\text{Rank}_r = 0.01, \text{Card}_r = 0.01$				
50	84	$9.36e-003$	$2.67e-005$	0.2	88	$9.15e-003$	$2.29e-005$	0.2
100	44	$8.41e-003$	$2.47e-005$	0.6	52	$8.38e-003$	$1.87e-005$	0.6
150	58	$5.00e-003$	$2.16e-005$	2.1	57	$4.89e-003$	$1.87e-005$	2.0
200	52	$4.81e-003$	$2.01e-005$	4.1	45	$4.42e-003$	$2.36e-005$	3.4
250	52	$3.40e-003$	$1.97e-005$	7.4	46	$3.14e-003$	$2.42e-005$	6.4
300	47	$3.30e-003$	$1.90e-005$	11.4	42	$2.53e-003$	$3.58e-005$	9.8
400	43	$2.57e-003$	$1.83e-005$	39.1	42	$1.66e-003$	$4.18e-005$	37.7
500	38	$2.08e-003$	$2.77e-005$	82.1	42	$1.13e-003$	$4.68e-005$	89.2
				$\text{Rank}_r = 0.02, \text{Card}_r = 0.02$				
50	65	$1.38e-002$	$3.24e-005$	0.2	78	$1.32e-002$	$2.68e-005$	0.2
100	72	$7.05e-003$	$2.71e-005$	1.1	68	$6.77e-003$	$2.27e-005$	1.0
150	63	$4.88e-003$	$2.69e-005$	2.8	48	$4.61e-003$	$2.42e-005$	2.1
200	53	$3.66e-003$	$2.52e-005$	7.4	43	$3.24e-003$	$4.50e-005$	5.4
250	47	$2.98e-003$	$2.62e-005$	7.3	43	$2.10e-003$	$4.51e-005$	6.7
300	41	$2.55e-003$	$3.02e-005$	10.4	43	$1.58e-003$	$5.38e-005$	10.8
400	34	$1.70e-003$	$4.95e-005$	32.5	44	$1.14e-003$	$5.74e-005$	41.9
500	34	$7.67e-004$	$4.88e-005$	75.3	45	$9.18e-004$	$5.73e-005$	101.0
				$\text{Rank}_r = 0.03, \text{Card}_r = 0.03$				
50	98	$9.03e-003$	$3.91e-005$	0.4	88	$8.86e-003$	$3.09e-005$	0.3
100	78	$6.41e-003$	$3.18e-005$	1.3	64	$6.13e-003$	$2.62e-005$	1.0
150	56	$3.67e-003$	$3.42e-005$	2.5	51	$3.41e-003$	$2.97e-005$	2.2
200	45	$3.09e-003$	$3.32e-005$	4.2	46	$2.54e-003$	$3.94e-005$	4.0
250	38	$2.16e-003$	$4.35e-005$	6.1	46	$1.67e-003$	$4.94e-005$	7.1
300	35	$1.71e-003$	$4.86e-005$	8.7	46	$1.32e-003$	$5.39e-005$	11.5
400	34	$1.22e-003$	$8.13e-005$	31.9	47	$9.56e-004$	$5.85e-005$	45.4
500	34	$9.29e-004$	$8.18e-005$	74.5	47	$8.43e-004$	$6.44e-005$	106.4

(iv) The K-subproblem (46) amounts to projecting the matrix $2L^{k+1} - L^k - (1/\beta)\Lambda_2^k$ onto the the nonnegative orthant, whose closed-form solution is given by:

$$K^{k+1} := \max\left\{2L^{k+1} - L^k - \frac{1}{\beta}\Lambda_2^k, 0\right\}. \tag{54}$$

For detailed analytical methods of ((43)–(46)), the reader is referred to, for example, [19, 26]. Thus, the simplicity assumption (5) holds for the application (41).

For the numerical experiments, we follow [19] to randomly generate the data of (41). We consider the scenario of $m = n$. For given n, $r < n$, we generate $L^* = R_1 R_2^T$, where R_1 and R_2 are independent $n \times r$ full row rank matrices whose elements are independently and identically (i.i.d.) uniformly distributed in $[0, 1]$. Note that in this experiment, L^* is a component-wise nonnegative and low-rank matrix to be recovered. The support of the sparse matrix S^* was chosen uniformly and randomly, and the nonzero entries in S^* are generated i.i.d. uniformly in $[-500, 500]$. The entries of matrix Z^* for noise were generated as i.i.d. Gaussian with standard deviation 10^{-4}; we set $M := L^* + S^* + Z^*$.

In the following, we compare CPPA with APGM in [18, 19], since they are all designed based on the "simple" assumption and share the same conditions for convergence. For the penalty parameter β, we take $\beta = 0.01$. For other individual parameters required by these methods, we choose $r = 2.618\beta$, $s = \beta$, and $\rho := 1/\sqrt{n}$. To implement all the compared methods, the initial iterate for both CPPA and APGM is chosen $L^0 = K^0 = -M$, $S^0 = Z^0 = 0$, $\Lambda_1^0 = \Lambda_2^0 = 0$. The stopping criterion is set as

$$\text{resid} := \frac{\|L + S + Z - M\|_F}{\|M\|_F} < \epsilon_r, \tag{55}$$

where ϵ_r is the tolerance set as $\epsilon_r = 10^{-4}$. We denoted $\text{Rank}_r := r/n$ so that the rank of L^* is $n * \text{Rank}_r$ and $\text{Card}_r :=$ cardinality $(S^*)/(n^2)$ so that the cardinality of S^* is $n^2 * \text{Card}_r$.

Some preliminary numerical results are reported in Table 1. Since they are synthetic examples with random date, for each scenario we test for 10 times, and the results were averaged over ten runs. Specifically, we reported the number of iteration (Iter.), relative error of the low-rank matrix $L(\text{rel}_L)$, relative error of the sparse matrix $S(\text{rel}_S)$, and CPU

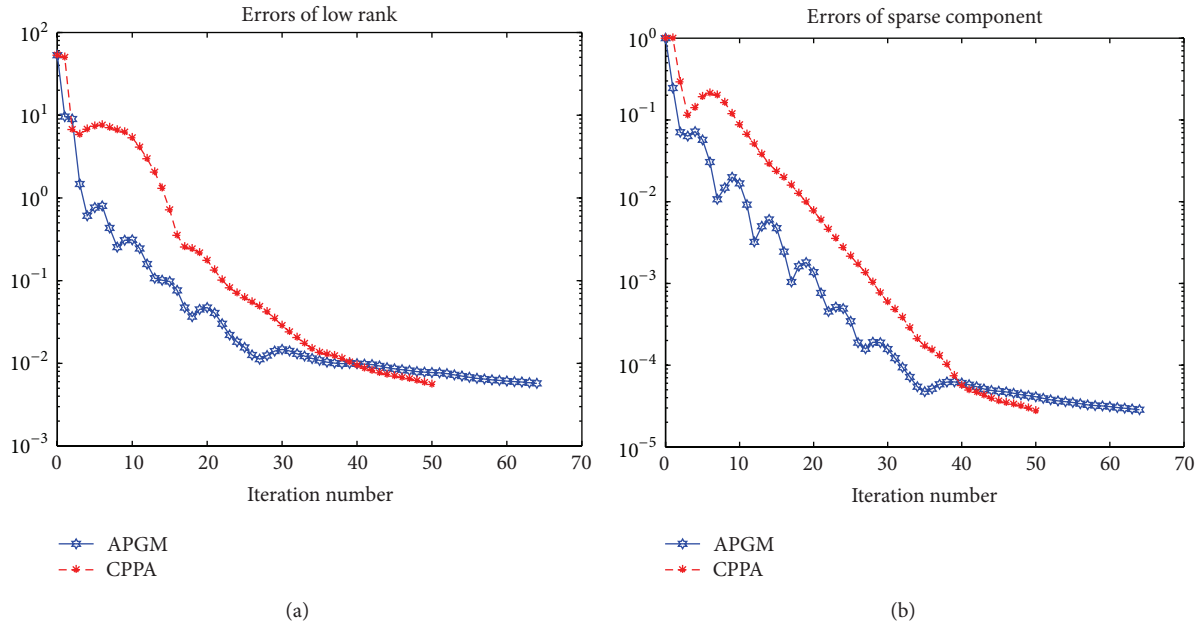

FIGURE 1: (a) Relative errors of low-rank matrix L. (b) Relative errors of sparse matrix S.

times in seconds (CPU(s)), where the relative errors are defined as

$$\text{rel}_L := \frac{\|L - L^*\|_F}{\|L^*\|_F}, \qquad \text{rel}_S := \frac{\|S - S^*\|_F}{\|S^*\|_F}. \tag{56}$$

From Table 1, we observe that both of the methods are efficient. Our method tends to be competitive with APGM and performs reasonably well for moderate dimensions (say, $n \in [100\text{–}300]$). To see the comparison clearly, we focus on the particular case where $n = 150$ and $\text{Rank}_r = 0.02$, $\text{Card}_r = 0.02$; we visualize the iterative processes of different method in Figure 1. More specifically, we plot the evolutions of the relative error rel_L and rel_S, with respect to the iterations. According to the curves in Figure 1, the performance of CPPA is slightly worse than that of APGM, when the iterations are small. However, with the iterations going large, CPPA shows a better performance than APGM.

6. Concluding Remarks

We have proposed a new algorithm for solving (1) which admits easy subproblems assuming the proximal mappings of f and g are easy to compute. Our algorithm can be viewed as a customized proximal point method with a special proximal regularization parameter. We established its global convergence under the analytic framework of contraction type methods. The computational results on solving the stable principal component pursuit problem show that our algorithm works reasonably well on large-scale instances.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The work is supported in part by the Natural Science Foundation of China, Grant NSFC-70971136.

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," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[2] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, Article 11, 2011.

[3] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[4] B. He, M. Xu, and X. Yuan, "Solving large-scale least squares semidefinite programming by alternating direction methods," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 1, pp. 136–152, 2011.

[5] L. Xue, S. Ma, and H. Zou, "Positive-definite ℓ_1-penalized estimation of large covariance matrices," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1480–1491, 2012.

[6] S. Ma, L. Xue, and H. Zou, "Alternating direction methods for latent variable Gaussian graphical model selection," *Neural Computation*, vol. 25, no. 8, pp. 2172–2198, 2013.

[7] Z. Wen, D. Goldfarb, and W. Yin, "Alternating direction augmented Lagrangian methods for semidefinite programming," *Mathematical Programming Computation*, vol. 2, no. 3-4, pp. 203–230, 2010.

[8] J. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-problems in compressive sensing," *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 250–278, 2011.

[9] J. Yang, Y. Zhang, and W. Yin, "A fast alternating direction method for TVL1-L2 signal reconstruction from partial fourier

data," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 288–297, 2010.

[10] R. T. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 97–116, 1976.

[11] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, vol. 9, Society for Industrial and Applied Mathematics, 1989.

[12] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1–3, pp. 293–318, 1992.

[13] B. He and X. Yuan, "On the O(1/n) convergence rate of the Douglas-Rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.

[14] M. Hong and Z. Q. Luo, "On the linear convergence of the alternating direction method of multipliers," In press, http://arxiv.org/abs/1208.3922.

[15] W. Deng and W. Yin, "On the globaland linear convergence of the generalized alternating direction method of multipliers," *Preprint*. In press, http://www.optimization-online.org/DB_FILE/2012/08/3578.pdf.

[16] D. Han and X. Yuan, "A note on the alternating direction method of multipliers," *Journal of Optimization Theory and Applications*, vol. 155, no. 1, pp. 227–238, 2012.

[17] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-blockconvex minimization problems is not necessarily convergent," *Preprint*. In press, http://www.optimization-online.org/DB_FILE/2013/09/4059.pdf.

[18] B. He and X. Yuan, "The unified framework of some proximal-based decomposition methods for monotone variational inequalities with separable structures," *Pacific Journal of Optimization*, vol. 8, no. 4, pp. 817–844, 2012.

[19] S. Ma, "Alternating proximal gradient method for convex minimization.," *Preprint*. In press, http://www.optimization-online.org/DB_FILE/2012/09/3608.pdf.

[20] B. Martinet, "Régularisation d'inéquations variationnelles par approximations successives," *Revue Française d'Informatique et de Recherche Opérationnelle*, vol. 4, no. 3, pp. 154–158, 1970.

[21] B. He, X. Yuan, and W. Zhang, "A customized proximal point algorithm for convex minimization with linear constraints," *Computational Optimization and Applications*, vol. 56, no. 3, pp. 559–572, 2013.

[22] X. Cai, G. Gu, B. He, and X. Yuan, "A relaxed customized proximal point algorithm for separable convex programming," *Preprint*. In press, http://www.optimization-online.org/DB_FILE/2011/08/3141.pdf.

[23] G. Gu, B. He, and X. Yuan, "Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach," *Computational Optimization and Applications*, 2013.

[24] X. Cai, G. Gu, B. He, and X. Yuan, "A proximal point algorithm revisit on the alternating direction method of multipliers," *Science China Mathematics*, vol. 56, no. 10, pp. 2179–2186, 2013.

[25] J. Yang and X. Yuan, "Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization," *Mathematics of Computation*, vol. 82, no. 281, pp. 301–329, 2013.

[26] N. Parikh and S. Boyd, *Proximal Algorithms. Foundations and Trends in Optimization*, 2013.