


# A class of customized proximal point algorithms for linearly constrained convex optimization

Feng Ma<sup>1</sup>  · Mingfang Ni<sup>2</sup>

Received: 9 July 2015 / Revised: 19 July 2016 / Accepted: 26 July 2016 / Published online: 19 August 2016  
© SBMAC - Sociedade Brasileira de Matemática Aplicada e Computacional 2016

**Abstract** In this paper, we propose a class of customized proximal point algorithms for linearly constrained convex optimization problems. The algorithms are implementable, provided that the proximal operator of the objective function is easy to evaluate. We show that, with special setting of the algorithmic scalar, our algorithms contain the customized proximal point algorithm (He et al., *Optim Appl* 56:559–572, 2013), the linearized augmented Lagrangian method (Yang and Yuan, *Math Comput* 82:301–329, 2013), the Bregman Operator Splitting algorithm (Zhang et al., *SIAM J Imaging Sci* 3:253–276, 2010) as special cases. The global convergence and worst-case convergence rate measured by the iteration complexity are established for the proposed algorithms. Numerical results demonstrate that the algorithms work well for a wide range of the scalar.

**Keywords** Convex optimization · Proximal point algorithm · Linear constraints · Augmented Lagrangian method

**Mathematics Subject Classification** 65K10 · 90C25 · 90C30

## 1 Introduction

In this paper, we consider the following convex minimization problem with linear constraint:

$$\min\{\theta(x) \mid Ax = b, x \in \mathcal{X}\}, \quad (1.1)$$

---

F. Ma was supported by the NSFC Grant 71401176.

---

✉ Feng Ma  
mafengnju@gmail.com

Mingfang Ni  
mingfang\_ni@163.com

<sup>1</sup> High-Tech Institute of Xi'an, Xi'an 710025, Shaanxi, China

<sup>2</sup> College of Communications Engineering, PLA University of Science and Technology, Nanjing 210007, China

where  $A \in \mathfrak{N}^{m \times n}$ ,  $b \in \mathfrak{N}^m$ ,  $\mathcal{X}$  is a closed convex set and  $\theta(x) : \mathfrak{N}^n \rightarrow \mathfrak{N}$  is a proper convex function. Without loss of generality, we assume that the solution set of (1.1), denoted by  $\mathcal{X}^*$ , is nonempty. Problem (1.1) has many applications, such as compressive sensing (Yin et al. 2008; Starck et al. 2010), image processing (Chambolle and Pock 2011; Zhang et al. 2010), machine learning (Cai et al. 2010).

To solve (1.1), a benchmark is the augmented Lagrangian multiplier method (ALM) (Hestenes 1969; Powell 1969). For given  $\lambda^k$ , ALM generates the new iterate via the scheme:

$$\begin{cases} x^{k+1} = \operatorname{argmin}_{x \in \mathcal{X}} \left\{ \theta(x) + \frac{\beta}{2} \left\| Ax - b - \frac{1}{\beta} \lambda^k \right\|^2 \right\}, & (1.2a) \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - b), & (1.2b) \end{cases}$$

where  $\lambda$  is the Lagrange multiplier and  $\beta$  is a penalty parameter. ALM is a fundamental and effective approach in optimization. In particular, it was shown by Rockafellar (1976) that ALM is the proximal point algorithm (PPA) applied to the dual of (1.1).

At each iteration, the computation of ALM for (1.2a) is dominated by solving the following subproblem:

$$x := \operatorname{argmin}\{\theta(x) + \frac{r}{2} \|Ax - a\|^2 \mid x \in \mathcal{X}\}, \tag{1.3}$$

where  $r > 0$ ,  $a \in \mathfrak{N}^m$ . When the problem (1.3) has closed form or easily computable solutions, the implementation of the ALM could be easy. However, in many application cases,  $x^{k+1}$  in (1.3) cannot be computed directly. Here, we list two examples.

- **Basis pursuit:** The basis pursuit problem seeks to recover a sparse vector by finding solutions to underdetermined linear systems. It can be formulated as follows:

$$\min\{\|x\|_1 \mid Ax = b\}, \tag{1.4}$$

where  $A$  is a matrix, and  $b$  is a given vector. When use ALM to solve (1.4), one need to solve the following minimization problem:

$$x := \operatorname{argmin}\{\|x\|_1 + \frac{r}{2} \|Ax - a\|^2\}, \tag{1.5}$$

Note that the operator  $\|\cdot\|_1$  is non-smooth, thus it is not easy to obtain a solution.

- **Matrix completion:** The matrix completion problem consists of reconstructing an unknown low-rank matrix from a given subset of observed entries. Mathematically, the matrix completion problem is

$$\min\{\|X\|_* \mid \mathcal{A}X = b\}, \tag{1.6}$$

where  $\|\cdot\|_*$  denotes the nuclear norm of a matrix and  $\mathcal{A}$  is a linear mapping. When use ALM to solve (1.6), one need to solve the following minimization problem:

$$X := \operatorname{argmin} \left\{ \|X\|_* + \frac{r}{2} \|\mathcal{A}X - \Lambda\|_F^2 \right\}, \tag{1.7}$$

where  $\Lambda$  is a given matrix. The operator  $\|\cdot\|_*$  is also non-smooth, thus it is not easy to obtain a solution.

For both problems (1.5) and (1.7), no exact solutions can be obtained. To guarantee the global convergence of ALM, a number of inner iterations are required to pursuit an approximate solution subject to certain inexactness criteria. Consequently, the efficiency of

ALM becomes entirely dependent upon the inner iterations. Interestingly, for the following  $l_1 + l_2$  problem

$$x := \operatorname{argmin} \left\{ \|x\|_1 + \frac{r}{2} \|x - a\|^2 \right\}, \tag{1.8}$$

one can get the closed form solution directly by the soft thresholding operator given by

$$x = \operatorname{sgn}(a) \max \left( 0; |a| - \frac{1}{r} \right), \tag{1.9}$$

where all the operations are defined elementwise; for the following matrix problem

$$X := \operatorname{argmin} \left\{ \|X\|_* + \frac{r}{2} \|X - \Lambda\|_F^2 \right\}, \tag{1.10}$$

the closed form solution can also be easily obtained using the matrix shrinkage operator (see [Cai et al. 2010](#); [Ma et al. 2011](#))

$$X = U \operatorname{Diag} \left( \max \left( \sigma - \frac{1}{r}, 0 \right) \right) V^T \tag{1.11}$$

where  $U \operatorname{Diag}(\sigma) V^T$  is the singular value decomposition of the matrix  $\Lambda$ .

Accordingly, it is interesting to discuss special strategies to utilize these special computational properties of the objective function. This motivates us to conduct our analysis by assuming the proximal operator

$$x := \left( I + \frac{1}{r} \partial \theta \right)^{-1} (a) = \operatorname{argmin} \left\{ \theta(x) + \frac{r}{2} \|x - a\|^2 \mid x \in \mathcal{X} \right\} \tag{1.12}$$

is easy to evaluate, where  $r$  has the same meaning as (1.3). The assumption associated with the proximal operator is not restricted; details about this kind of assumption can be found in, e.g., [Cai et al. \(2010\)](#) and [Chambolle and Pock \(2011\)](#). We refer the reader to [Parikh and Boyd \(2014\)](#) for additional examples. Note that for the subproblem (1.2a) of ALM, at each iteration, one need to evaluate the operator  $(\beta A^T A + \partial \theta)^{-1}(\cdot)$ , while for the subproblem (1.12), one need to evaluate the operator  $(rI + \partial \theta)^{-1}(\cdot)$ . In numerical analysis, the condition number  $A^T A$  is usually larger than 1. When  $A^T A$  is ill-conditioned, i.e., the condition number is too large (this is usually encountered in image processing); the subproblem (1.2a) is less stable than (1.12). As we know that a smaller condition number leads to a faster convergence, thus the subproblem (1.12) is easier than (1.2a) when  $A^T A$  is ill-conditioned. See [Cai et al. \(2009\)](#) for some numerical demonstrations on these two subproblems. Furthermore, as we already demonstrated for some applications,  $(rI + \partial \theta)^{-1}(\cdot)$  can admit a closed form solution, while  $(\beta A^T A + \partial \theta)^{-1}(\cdot)$  cannot, and need inner iterations to solve. Altogether, the assumption (1.12) is useful for the algorithmic implementation.

[He et al. \(2013\)](#) developed a customized proximal point algorithm for (1.1). They showed that by choosing a suitable proximal regularization parameter, the difficulty of solving the subproblems in ALM becomes alleviated, see also in [Gu et al. \(2014\)](#) and [You et al. \(2013\)](#). In [Yang and Yuan \(2013\)](#), a linearized augmented Lagrangian multiplier method (LALM) has been proposed. The key ideal of LALM is to linearize the quadratic penalties in ALM by adding a proximal term  $\frac{1}{2} \|x - x\|_G^2$  with  $G = rI - \beta A^T A$ . With this choice, the  $x$ -subproblem reduces to estimating the proximal operator of  $\theta(\cdot)$  and, thus, it is suitable to solve the nuclear norm minimization problem. This technique is also used in [Zhang et al. \(2010\)](#), [Ma \(2016\)](#) and [Li and Yuan \(2015\)](#) for tackling imaging processing problems. Note that the problem (1.1) can be reformulated as a saddle point algorithm, then could be solved by primal-dual hybrid gradient method (PDHG) as discussed in [Chambolle and Pock \(2011\)](#) and [He and](#)

Yuan (2012). Alternatively, one can regularize (1.1) as a structured convex problem, thus the proximal gradient method (Beck and Teboulle 2009; Cai et al. 2010; Daubechies et al. 2004), alternating direction methods (Glowinski and Le Tallec 1989; Ma 2016; Li and Yuan 2015), and Bregman methods (Goldstein and Osher 2009; Ma et al. 2011; Huang et al. 2013; Yin et al. 2008) are applicable. We refer to Esser (2009) for the connection of these methods.

In this paper, we focus on designing implementable methods for solving the problem (1.1) under the same assumption. Differ from the PPA in He et al. (2013), we propose a class of PPA. The proposed methods can be applied to solve concrete models with only requiring evaluation of the proximal operator. Moreover, we show that the PPA proposed in He et al. (2013), LALM (Yang and Yuan 2013) and the Bregman Operator Splitting (BOS) proposed in Zhang et al. (2010) are all special cases of our algorithmic framework. Our study can provide a better understanding of the customized version of PPA. The paper is organized as follows. In Sect. 2, we review some preliminaries. In Sect. 3, we present the new method and discuss its connection with other existing methods. Sections 4 and 5 analyze its convergence and convergence rate. In Sect. 6, numerical experiments are conducted to illustrate the performances. Finally, we conclude the paper in Sect. 7.

## 2 Preliminaries

### 2.1 Variational characterization of (1.1)

In this section, we characterize the optimal condition of (1.1) as a variational reformulation, which is easier to expose our motivation.

The Lagrangian function associated with (1.1) is

$$\mathcal{L}(x, y, \lambda) = \theta(x) - \lambda^T(Ax - b), \tag{2.1}$$

where  $\lambda \in \mathfrak{R}^m$  is a Lagrangian multiplier. Solving (1.1) is equivalent to finding a saddle point of  $\mathcal{L}$ . Let  $w^* = (x^*, \lambda^*)$  be a saddle point of  $\mathcal{L}$  (2.1). We have

$$\mathcal{L}_{\lambda \in \mathfrak{R}^m}(x^*, \lambda) \leq \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}_{x \in \mathcal{X}}(x, \lambda^*). \tag{2.2}$$

Then, finding  $w^*$  amounts to solve the following mixed variational inequality:

$$\theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega, \tag{2.3a}$$

with

$$w := \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(w) := \begin{pmatrix} -A^T \lambda \\ Ax - b \end{pmatrix}, \tag{2.3b}$$

and

$$\Omega = \mathcal{X} \times \mathfrak{R}^m. \tag{2.3c}$$

Here, we denote (2.3a–2.3c) by VI( $\Omega, F$ ).

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

$$(F(w) - F(w'))^T(w - w') \geq 0, \quad \forall w, w' \in \Omega. \tag{2.4}$$

Consequently, it can be easily verified that  $F(w)$  (2.3b) is monotone. Under the aforementioned nonempty assumption on the solution set of (1.1), the solution set of VI( $\Omega, F$ ), denoted by  $\Omega^*$ , is also nonempty.

### 2.2 Proximal point algorithmic framework

In this subsection, we briefly review the generalized proximal point algorithm (PPA) for solving VI( $\Omega, F$ ) (2.3).

PPA which was initiated by [Martinet \(1970\)](#) and subsequently generalized by [Rockafellar \(1976\)](#) plays a fundamental role in optimization. For given  $w^k \in \Omega$ , PPA generates the new iterate via

$$\theta(x') - \theta(x^{k+1}) + (w' - w^{k+1})^T(F(w^{k+1}) + G(w^{k+1} - w^k)) \geq 0, \quad \forall w' \in \Omega, \quad (2.5)$$

where  $G \in \mathbb{R}^{(n+m) \times (n+m)}$  is a positive definite matrix, playing the role of proximal regularization. The case of the classical PPA corresponds to taking  $G = \beta \cdot I$ , where  $\beta > 0$  and  $I$  is the identity matrix. This simplest choice of  $G$  regularizes the proximal terms  $w^{k+1} - w^k$  in the uniform way. Although the classical PPA can be used to solve many convex problems, it may encounter unbearable difficulty of computing the subproblem in practice. In fact, the resulting subproblem is, in many cases, as hard as the original problem. To overcome this drawback, some customized PPA schemes with metric proximal regularization are proposed in the literatures, see, e.g., [He et al. \(2013, 2016\)](#), [Gu et al. \(2014\)](#), [Han et al. \(2014\)](#) and [He \(2015\)](#). These customized PPAs choose  $G$  in accordance with the separable structures of their VI reformulations for various problems. As a result, the reduced subproblems are considerably easier compared to the classical PPA.

### 3 Algorithm

In this section, we describe our customized PPA for solving (1.1), and then extend to the case of inequality constraints in Sect. 3.3. To better differentiate our contributions, we make explicit comparisons of our customized PPA with the PPA proposed in [He et al. \(2013\)](#) and [Gu et al. \(2014\)](#), the linearized ALM proposed in [Yang and Yuan \(2013\)](#) and the Bregman operator splitting (BOS) method proposed in [Zhang et al. \(2010\)](#).

#### 3.1 Customized PPA

Recall that the main computational difficulty of the ALM lies in the fact that the evaluation of (1.3) is not necessarily easy to compute. For PPA scheme, if we choose a suitable proximal regularization matrix  $G$  to simplify the  $x$ -subproblem by reducing it to a evaluation of the proximal operator of  $\theta(x)$ , then the encountered difficulty would be particularly alleviated under our assumption (1.12). Here, we judiciously choose  $G$  as

$$G = \begin{pmatrix} rI + \frac{1}{s}(t^2 - 1)A^T A & tA^T \\ tA & sI \end{pmatrix}. \quad (3.1)$$

Obviously, to ensure the positive definite of  $G$ , we need to impose restrictions on the parameters. Here, we restrict  $r > 0, s > 0, rs > \|A^T A\|$  and take  $t$  an arbitrary real scalar, respectively.

To get the scheme of our customized PPA, substitute  $G$  (3.1) into PPA (2.5), we get

$$\forall w' \in \Omega, \quad \theta(x') - \theta(x^{k+1}) + \begin{pmatrix} x' - x^{k+1} \\ \lambda' - \lambda^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T \lambda^{k+1} \\ Ax^{k+1} - b \end{pmatrix} + \begin{pmatrix} rI + \frac{1}{s}(t^2 - 1)A^T A & tA^T \\ tA & sI \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} \right\} \geq 0. \quad (3.2)$$

Note that the above VI is equivalent to

$$Ax^{k+1} - b + tA(x^{k+1} - x^k) + s(\lambda^{k+1} - \lambda^k) = 0, \tag{3.3}$$

and

$$\begin{aligned} &x^{k+1} \in \mathcal{X}, \quad \theta(x') - \theta(x^{k+1}) \\ &+ (x' - x^{k+1})^T \left\{ -A^T\lambda^{k+1} + \left( rI + \frac{1}{s}(t^2 - 1)A^T A \right) (x^{k+1} - x^k) + tA^T(\lambda^{k+1} - \lambda^k) \right\} \geq 0, \quad \forall x' \in \mathcal{X}. \end{aligned} \tag{3.4}$$

From (3.3), we obtain

$$\lambda^{k+1} = \lambda^k - \frac{1}{s}((1 + t)Ax^{k+1} - tAx^k - b), \tag{3.5}$$

The  $x$ -subproblem (3.4) amounts to solving the following minimization:

$$x^{k+1} = \operatorname{argmin}_{x \in \mathcal{X}} \left\{ \theta(x) + \frac{r}{2}\|x - x^k - \frac{1}{r}A^T \left( \lambda^k - \frac{1}{s}(1 - t)(Ax^k - b) \right)\|^2 \right\}. \tag{3.6}$$

Combining (3.5) and (3.6), the customized PPA iterates as follows.

**Algorithm 1: The customized PPA for (1.1)**

**Input:** Choose a scalar  $t \in (-\infty, +\infty)$  and let the parameters  $r > 0, s > 0$  satisfy the condition  $rs > \|A^T A\|$ .

**Initialization:**  $(x^0, \lambda^0) \in \Omega$ .

**Iteration Step:**

$$\begin{cases} x^{k+1} = \operatorname{argmin}_{x \in \mathcal{X}} \{ \theta(x) + \frac{r}{2}\|x - x^k - \frac{1}{r}A^T(\lambda^k - \frac{1}{s}(1 - t)(Ax^k - b))\|^2 \}, & (3.7a) \\ \lambda^{k+1} = \lambda^k - \frac{1}{s}((1 + t)Ax^{k+1} - tAx^k - b). & (3.7b) \end{cases}$$

*Remark 1* Notice that the  $x$ -subproblem of the algorithm amounts to evaluating the resolvent operator of  $\theta(x)$ . Due to the assumption (1.12), such a problem is well solved. This is a computational advantage over ALM.

*Remark 2* When the condition  $rs > \|A^T A\|$  holds, the matrix  $G$  is positive definite for every specified  $t \in (-\infty, +\infty)$ . Then, with different choices of  $t$ , a class of specific PPA for solving (1.1) is proposed.

### 3.2 Connections with existing methods

Recently, several relevant algorithms have been proposed for solving (1.1). We present some of them here and show their connection to Algorithm 1.

#### 3.2.1 The linearized augmented Lagrangian method

In Yang and Yuan (2013), the authors proposed a linearized augmented Lagrangian method (LALM) for the nuclear norm minimization problem. The main idea of LALM is to linearize

the  $x$ -subproblem in ALM. By taking one proximal gradient step to approximate the term  $\frac{\beta}{2} \|Ax - b - \frac{1}{\beta} \lambda^k\|^2$  at  $x^k$ , one obtains

$$\frac{\beta}{2} \left\| Ax - b - \frac{1}{\beta} \lambda^k \right\|^2 \approx \frac{\beta}{2} \|Ax^k - b - \frac{1}{\beta} \lambda^k\|^2 + (x - x^k)^T g^k + \frac{r}{2} \|x - x^k\|^2. \tag{3.8}$$

where  $g^k = \beta A^T(Ax^k - b - \frac{1}{\beta} \lambda^k)$  is the gradient of the quadratic term  $\frac{\beta}{2} \|Ax - b - \frac{1}{\beta} \lambda^k\|^2$ . Substitute (3.8) into (1.2a) and make simple manipulation, the LALM iterates as follows:

$$\begin{cases} x^{k+1} = \operatorname{argmin}_{x \in \mathcal{X}} \{ \theta(x) + \frac{r}{2} \|x - x^k - \frac{1}{r} A^T(\lambda^k - \beta(Ax^k - b))\|^2 \}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - b). \end{cases} \tag{3.9}$$

It is easy to check that the LALM takes a form of our customized PPA with  $t = 0, \beta = 1/s$ .

### 3.2.2 The Bregman Operator Splitting algorithm

In Zhang et al. (2010), A Bregman operator splitting (BOS) is proposed for solving nonlocal TV denoising problems. BOS can be interpreted as an inexact Uzawa method (Glowinski and Le Tallec 1989) applied to the augmented Lagrangian function associated with the original problem (Esser 2009). The main idea of BOS is to adding an additional quadratic proximal term  $\frac{1}{2} \|x - x^k\|_{rI - \beta A^T A}^2$  to the  $x$ -subproblem in ALM, so that the term  $\|Ax\|$  can be canceled out. More precisely, let  $r > \beta \|A^T A\|$ . Then, the  $x$ -subproblem is

$$x^{k+1} = \operatorname{argmin}_{x \in \mathcal{X}} \left\{ \theta(x) + \frac{\beta}{2} \left\| Ax - b - \frac{1}{\beta} \lambda^k \right\|^2 + \frac{1}{2} \|x - x^k\|_{rI - \beta A^T A}^2 \right\}. \tag{3.10}$$

Rewriting the above term yields

$$x^{k+1} = \operatorname{argmin}_{x \in \mathcal{X}} \left\{ \theta(x) + \frac{r}{2} \left\| x - x^k - \frac{1}{r} A^T(\lambda^k - \beta(Ax^k - b)) \right\|^2 \right\}. \tag{3.11}$$

Now, it can easily seen that the linearized ALM is equivalent to BOS with a different motivation. As a result, BOS also takes the form of our customized PPA with  $t = 0, \beta = 1/s$ .

### 3.2.3 The customized proximal point algorithm

He et al. (2013) proposed a customized proximal point algorithm to exploit the simplicity in (1.12). The authors suggested to choose the regularization matrix as:

$$G_1 = \begin{pmatrix} rI & -A^T \\ -A & sI \end{pmatrix}. \tag{3.12}$$

where  $r > 0, s > 0$  and  $rs > \|A^T A\|$ . With this customized choice of  $G$ , the resulting PPA scheme (2.5) can be written as:

$$\begin{cases} \lambda^{k+1} = \lambda^k - \frac{1}{s}(Ax^k - b), \\ x^{k+1} = \operatorname{argmin}_{x \in \mathcal{X}} \left\{ \theta(x) + \frac{r}{2} \left\| x - x^k - \frac{1}{r} A^T(2\lambda^{k+1} - \lambda^k) \right\|^2 \right\}. \end{cases} \tag{3.13}$$

Note that  $x$ -subproblem amounts to evaluating the proximal operator of  $\theta(x)$ . Therefore, the PPA scheme enjoys easily implementable feature based on the assumption (1.12).

The update order of (3.13) is  $\lambda \rightarrow x$  and, thus, it is an dual-primal algorithm (denoted by DP-PPA). Motivated by He et al. (2013), the authors in Gu et al. (2014) also suggested to choose the regularization matrix:

$$G_2 = \begin{pmatrix} rI & A^T \\ A & sI \end{pmatrix}, \tag{3.14}$$

where  $r > 0, s > 0$  and  $rs > \|A^T A\|$ . With this customized choice, the resulting PPA scheme (2.5) can be written as:

$$\begin{cases} x^{k+1} = \operatorname{argmin}_{x \in \mathcal{X}} \{ \theta(x) + \frac{r}{2} \|x - x^k - \frac{1}{r} A^T \lambda^k\|^2 \}, \\ \lambda^{k+1} = \lambda^k - \frac{1}{s} (A(2x^{k+1} - x^k) - b), \end{cases} \tag{3.15}$$

Compared to the iterative schemes of (3.13), (3.15) iterates in a primal-dual order (denoted by PD-PPA). Based on the assumption (1.12), the PPA scheme (3.15) is also easily implementable. We refer to Li et al. (2015) for its multi-block case.

Obviously, we can easily see that the two PPAs are special cases of our customized PPA with  $t = -1, t = 1$ , respectively.

### 3.3 Extension

We now further extend our proposed algorithm to solve the following convex problem with inequality constraints.

$$\min \{ \theta(x) \mid Ax \geq b, x \in \mathcal{X} \}. \tag{3.16}$$

Similarly, the optimality conditions of (3.16) can be formulated as the following VI:

$$\theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega, \tag{3.17a}$$

with

$$w := \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(w) := \begin{pmatrix} -A^T \lambda \\ Ax - b \end{pmatrix}, \tag{3.17b}$$

and

$$\Omega = \mathcal{X} \times \mathfrak{R}_+^m. \tag{3.17c}$$

Let  $P_\Omega$  denote the projection onto  $\Omega$ , the iterative scheme for (3.16) can directly extend as follows.

**Algorithm 2: The customized PPA for (3.16)**

**Input:** Choose scalars  $t \in (-\infty, +\infty)$  and  $r > 0, s > 0$  satisfying the condition  $rs > \|A^T A\|$ .

**Initialization:**  $(x^0, \lambda^0) \in \Omega$ .

**Iteration Step:**

$$\begin{cases} x^{k+1} = \operatorname{argmin}_{x \in \mathcal{X}} \{ \theta(x) + \frac{r}{2} \|x - x^k - \frac{1}{r} A^T (\lambda^k - \frac{1}{s} (1-t)(Ax^k - b))\|^2 \}, \end{cases} \tag{3.18a}$$

$$\begin{cases} \lambda^{k+1} = P_{\mathfrak{R}_+^m} \{ \lambda^k - \frac{1}{s} ((1+t)Ax^{k+1} - tAx^k - b) \}. \end{cases} \tag{3.18b}$$



### 4 Global convergence

In this section, we establish the convergence of Algorithms 1 and 2. To this end, we first prove the following lemma.

**Lemma 4.1** *Suppose the condition*

$$r > 0, s > 0, rs > \|A^T A\| \tag{4.1}$$

*holds. Then, for any  $w^* = (x^*, \lambda^*)^T \in \Omega^*$ , the sequence  $w^{k+1} = (x^{k+1}, \lambda^{k+1})^T$  generated by Algorithm 1 (Algorithm 2) satisfies the following inequality:*

$$\|w^{k+1} - w^*\|_G^2 \leq \|w^k - w^*\|_G^2 - \|w^k - w^{k+1}\|_G^2, \tag{4.2}$$

*where the norm  $\|\cdot\|_G^2$  is defined as  $\|w\|_G^2 = w^T G w$ .*

*Proof* Note that  $w^* \in \Omega$ , it follows from (2.5) that

$$\theta(x^*) - \theta(x^{k+1}) + (w^* - w^{k+1})^T (F(w^{k+1}) + G(w^{k+1} - w^k)) \geq 0. \tag{4.3}$$

It can be further written as:

$$(w^{k+1} - w^*)^T G(w^k - w^{k+1}) \geq \theta(x^{k+1}) - \theta(x^*) + (w^{k+1} - w^*)^T F(w^{k+1}). \tag{4.4}$$

On the other hand, using (2.4), and  $w^* \in \Omega^*$ , we have

$$\theta(x^{k+1}) - \theta(x^*) + (w^{k+1} - w^*)^T F(w^{k+1}) \geq \theta(x^{k+1}) - \theta(x^*) + (w^{k+1} - w^*)^T F(w^*) \geq 0. \tag{4.5}$$

Thus, we get

$$(w^{k+1} - w^*)^T G(w^k - w^{k+1}) \geq 0. \tag{4.6}$$

It follows from the above inequality that

$$\begin{aligned} \|w^k - w^*\|_G^2 &= \|(w^{k+1} - w^*) + (w^k - w^{k+1})\|_G^2 \\ &\geq \|w^{k+1} - w^*\|_G^2 + \|w^k - w^{k+1}\|_G^2. \end{aligned} \tag{4.7}$$

Consequently, we obtain

$$\|w^{k+1} - w^*\|_G^2 \leq \|w^k - w^*\|_G^2 - \|w^k - w^{k+1}\|_G^2. \tag{4.8}$$

This completes the proof. □

**Theorem 4.2** *Suppose the condition*

$$r > 0, s > 0, rs > \|A^T A\| \tag{4.9}$$

*holds. Then, the sequence  $\{w^k\}$  generated by Algorithm 1 (Algorithm 2) converges to a solution point in  $\Omega^*$  globally.*

*Proof* Lemma 4.1 means that the generated iterative sequence  $\{w^k\}$  is Fejér monotone with respect to the solution set. The proof follows immediately from the results in (Bauschke and Combettes 2011, Chapt. 5). □

### 5 A worst-case $O(1/t)$ convergence rate

In this section, we establish the worst-case  $O(1/t)$  convergence rate in a nonergodic sense for Algorithms 1 and 2, where  $t$  denotes the iteration counter. We first prove the following lemma.

**Lemma 5.1** *Let  $\{w^k = (x^k, \lambda^k)\}$  be the sequence generated by Algorithm 1 (Algorithm 2). Then, we have*

$$(w^{k+1} - w^{k+2})^T G \{(w^k - w^{k+1}) - (w^{k+1} - w^{k+2})\} \geq 0, \tag{5.1}$$

where the matrix  $G$  is defined in (3.1).

*Proof* Set  $w = w^{k+2}$  in (2.5), we have

$$\theta(x^{k+2}) - \theta(x^{k+1}) + (w^{k+2} - w^{k+1})^T F(w^{k+1}) \geq (w^{k+2} - w^{k+1})^T G(w^k - w^{k+1}). \tag{5.2}$$

Note that (2.5) is also true for  $k := k + 2$  and thus

$$\theta(x) - \theta(x^{k+2}) + (w - w^{k+2})^T F(w^{k+2}) \geq (w - w^{k+2})^T G(w^{k+1} - w^{k+2}), \quad \forall w \in \Omega. \tag{5.3}$$

Set  $w = w^{k+1}$  in the above inequality, we obtain

$$\theta(x^{k+1}) - \theta(x^{k+2}) + (w^{k+1} - w^{k+2})^T F(w^{k+2}) \geq (w^{k+1} - w^{k+2})^T G(w^{k+1} - w^{k+2}). \tag{5.4}$$

Adding (5.2) and (5.4) and using the monotonicity of  $F$ , we have

$$(w^{k+1} - w^{k+2})^T G \{(w^k - w^{k+1}) - (w^{k+1} - w^{k+2})\} \geq 0. \tag{5.5}$$

The assertion (5.1) is proved. □

**Lemma 5.2** *Let  $\{w^k = (x^k, \lambda^k)\}$  be the sequence generated by Algorithm 1 (Algorithm 2). Then, we have*

$$(w^k - w^{k+1})^T G \{(w^k - w^{k+1}) - (w^{k+1} - w^{k+2})\} \geq \|(w^k - w^{k+1}) - (w^{k+1} - w^{k+2})\|_G^2, \tag{5.6}$$

where the matrix  $G$  is defined in (3.1).

*Proof* Adding  $\|(w^k - w^{k+1}) - (w^{k+1} - w^{k+2})\|_G^2$  to both sides of (5.1), we get

$$(w^k - w^{k+1})^T G \{(w^k - w^{k+1}) - (w^{k+1} - w^{k+2})\} \geq \|(w^k - w^{k+1}) - (w^{k+1} - w^{k+2})\|_G^2. \tag{5.7}$$

The lemma is proved. □

Now, we are ready to show that the sequence  $\{\|w^k - w^{k+1}\|_G^2\}$  is non-increasing.

**Theorem 5.3** *Let  $\{w^k = (x^k, \lambda^k)\}$  be the sequence generated by Algorithm 1 (Algorithm 2). Then, we have*

$$\|w^{k+1} - w^{k+2}\|_G^2 \leq \|w^k - w^{k+1}\|_G^2, \tag{5.8}$$

where the matrix  $G$  is defined in (3.1).

*Proof* First, applying the identity

$$\|a\|_G^2 - \|b\|_G^2 = 2a^T G(a - b) - \|a - b\|_G^2, \tag{5.9}$$

with  $a = w^k - w^{k+1}$  and  $b = w^{k+1} - w^{k+2}$ , we get

$$\begin{aligned} & \|w^k - w^{k+1}\|_G^2 - \|w^{k+1} - w^{k+2}\|_G^2 \\ &= 2(w^k - w^{k+1})^T G \left\{ (w^k - w^{k+1}) - (w^{k+1} - w^{k+2}) \right\} \\ & \quad - \|(w^k - w^{k+1}) - (w^{k+1} - w^{k+2})\|_G^2. \end{aligned} \tag{5.10}$$

Inserting (5.6) into the first term of the right-hand side of the last equality, we obtain

$$\|w^k - w^{k+1}\|_G^2 - \|w^{k+1} - w^{k+2}\|_G^2 \geq \|(w^k - w^{k+1}) - (w^{k+1} - w^{k+2})\|_G^2. \tag{5.11}$$

Now, according to (5.11), we have

$$\|w^{k+1} - w^{k+2}\|_G^2 \leq \|w^k - w^{k+1}\|_G^2. \tag{5.12}$$

That is, the monotonicity of the sequence  $\|w^k - w^{k+1}\|_G^2$  is proved.

Then, we can prove a worst-case  $O(1/t)$  convergence rate in a nonergodic sense for Algorithm 1 (Algorithm 2). We summarize the result in the following theorem.

**Theorem 5.4** *Let  $\{w^k = (x^k, \lambda^k)\}$  be the sequence generated by Algorithm 1 (Algorithm 2). Then, we have*

$$\|w^t - w^{t+1}\|_G^2 \leq \frac{1}{t+1} \|w^0 - w^*\|_G^2, \quad \forall w^* \in \Omega^*, \tag{5.13}$$

where the matrix  $G$  is defined in (3.1).

*Proof* First, it follows from Lemma 4.1 that

$$\sum_{t=0}^{\infty} \|w^t - w^{t+1}\|_G^2 \leq \|w^0 - w^*\|_G^2, \quad \forall w^* \in \Omega^*. \tag{5.14}$$

According to Theorem 5.3, the sequence  $\{\|w^t - w^{t+1}\|_G^2\}$  is monotonically non-increasing. Therefore, we have

$$(t+1)\|w^t - w^{t+1}\|_G^2 \leq \sum_{i=0}^t \|w^i - w^{i+1}\|_G^2. \tag{5.15}$$

The assertion (5.13) follows from (5.14) and (5.15) immediately. □

Notice that  $\Omega^*$  is convex and closed. Let  $d := \inf\{\|w - w^*\|_G \mid w^* \in \Omega^*\}$ . Then, for any given  $\epsilon > 0$ , Theorem 5.4 shows that the PPA scheme (3.7) needs at most  $\lfloor d^2/\epsilon \rfloor$  iterations to ensure that  $\|w^t - w^{t+1}\|_G^2 \leq \epsilon$ . Recall that  $w^{k+1}$  is a solution of VI( $\Omega, F$ ) if  $\|w^k - w^{k+1}\|_G^2 = 0$ . A worst-case  $O(1/t)$  convergence rate in a nonergodic sense is thus established for the PPA in Theorem 5.4. Note that in the paper (Gu et al. 2014), only worst-case  $O(1/t)$  convergence rate in ergodic sense for the PPA schemes (3.13) and (3.15) is established. In general, a worst-case nonergodic convergence rate is stronger than its ergodic counterpart. Thus, our analysis provides an important supplement of the convergence result for the PPA schemes.

## 6 Numerical experiments

In this section, we evaluate the performance of the customized PPA for solving image reconstruction problem and matrix completion problem, and report some preliminary numerical results. The codes of our algorithm were written entirely in Matlab 2015b, and they were executed on a laptop with an Intel quad-core i7 2.4 GHz CPU with 8 GB of RAM.

### 6.1 Image reconstruction

Note that our main motivation of designing customized PPA is to alleviate the  $x$ -subproblem in ALM (1.2) under the simplicity assumption (1.12) holds, here we first focus on comparing the customized PPA with the direct application of ALM by solving the wavelet-based image processing problem.

We first briefly review the background of the wavelet-based image processing. Let  $\mathbf{x} \in \mathfrak{R}^l$  represent a digital image with  $l = l_1 \times l_2$ , and  $W \in \mathfrak{R}^{l \times n}$  be a wavelet dictionary, i.e., each column of  $W$  be the elements of a wavelet frame. The image  $\mathbf{x}$  is usually sparse under wavelet transform  $W$ , thus we have  $\mathbf{x} = Wx$  with  $x$  being a sparse vector, see Starck et al. (2010) for more details. In image reconstruction, one tries to reconstruct the clean image  $x$  based on some observation  $b$ , then the model can be formulated as:

$$\min\{\|\mathbf{x}\|_1 \mid BWx = b\}, \tag{6.1}$$

where  $B \in \mathfrak{R}^{m \times l}$  is a diagonal matrix whose diagonal elements are either 0 or 1. The locations of 0 and 1 in  $B$  correspond to missing and known pixels of the image. The objective function  $\|\mathbf{x}\|_1$  is to deduce a sparse representation under the wavelet dictionary. The model (6.1) is a special case of the abstract model (1.1), thus the proposed method is applicable.

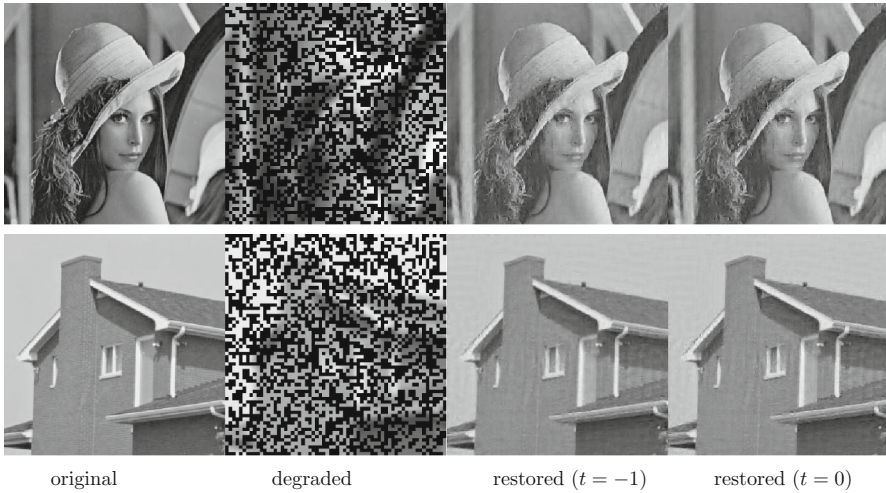
In our experiments, we adopt the reflective boundary condition for the image reconstruction problems. Under this circumstance, the mask  $B$  can be expressed as  $B = SH$ , where  $S \in \mathfrak{R}^{m \times l}$  is a downsampling matrix that satisfies  $\|S\| = 1$  and  $H \in \mathfrak{R}^{l \times l}$  is a blurry matrix that can be diagonalized by the discrete cosine transform (DCT). Note that the dictionary  $W$  has the property  $W^T W = I$ , we have  $\|A^T A\| = \|W^T B^T W B\| = 1$  (where  $A := BW$ ) (He et al. 2013). Therefore, the condition of the parameters  $rs > \|A^T A\|$  in Algorithm 1 reduces to  $rs > 1$  in our experiments.

We test the  $256 \times 256$  images of Lena.jpg and House.png for the image inpainting problem. The dictionary  $W$  is chosen as the inverse discrete Haar wavelet transform with a level of 6. For all the cases, we create the noisy images using the out-of focus kernel with a radius of 7 and the masking operator  $S$  to hide 50 % pixels. The positions of missing pixels are setting randomly. For the implementation, we take  $r = 0.6$ ,  $s = 1.02/r$  to implement Algorithm 1 and  $\beta = 10$  and  $\lambda^0 = 0$  for ALM (1.2). The initial point is set to be  $(x^0, \lambda^0) = (W^T(b), 0)$ . We refer the reader to He and Yuan (2012); He et al. (2013) for more implementation details. For solving the concrete application (6.1) by ALM, since the  $x$ -subproblem does not have closed form solutions, we need to employ certain algorithms to solve it iteratively. We choose the popular solvers ‘‘ISTA’’ in Daubechies et al. (2004) and ‘‘FISTA’’ in Beck and Teboulle (2009) for this purpose, and they both allow for a maximal number of 10 iterations for the inner iteration. The ALM embedded by ISTA and FISTA is denoted by ‘‘ALM-ISTA’’ and ‘‘ALM-FISTA’’, respectively. When measuring the quality of the restored images, we use the value of signal-to-noise ratio (SNR) in decibel (dB)

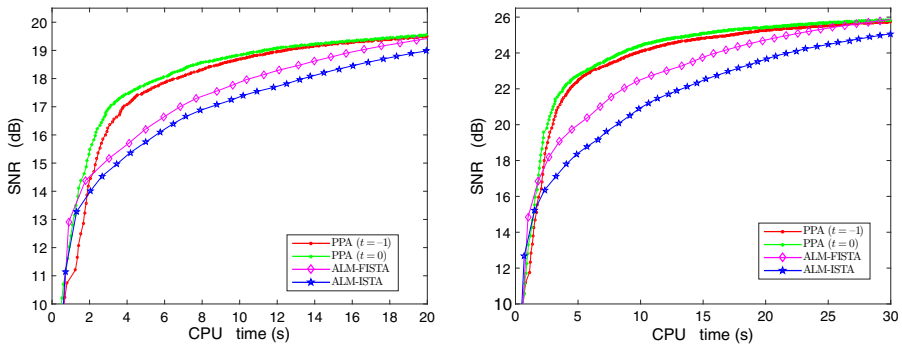
$$\text{SNR} := 20 \log_{10} \frac{\|\mathbf{x}\|}{\|\bar{\mathbf{x}} - \mathbf{x}\|},$$

where  $\bar{\mathbf{x}}$  is a image restored and  $\mathbf{x}$  represents the clean image.

The images restored by Algorithm 1 are shown in Fig. 1 and we also plot the evolutions of SNR with respect to computing time for Algorithm 1 ( $t = -1, t = 0$ ), ALM-ISTA and ALM-FISTA in Fig. 2. It shows clearly that the Algorithm 1 with  $t = -1$  and  $t = 0$  outperforms the ALM in terms of the improvement in signal-to-noise ratio. This is mainly because of the simple evaluation of  $x$ -subproblem.



**Fig. 1** The original images (*first column*), the degraded images (*second column*), the restored images by PPA  $t = -1$  (*third column*) and the restored images by PPA  $t = 0$  (*forth column*)



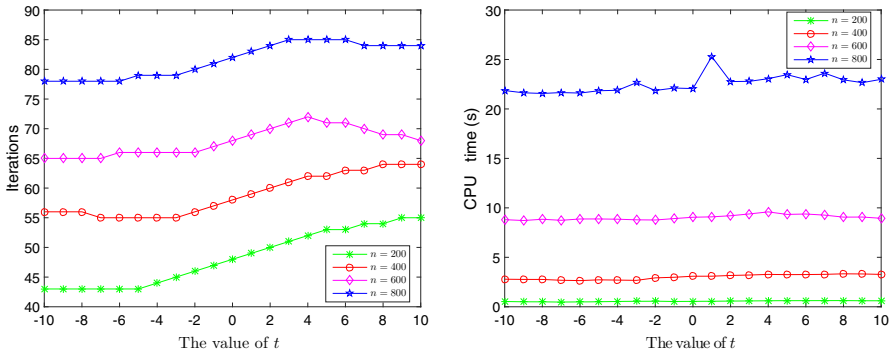
**Fig. 2** Evolutions of SNRs with respect to CPU time (s). *Left* Lena; *right* house

### 6.2 Matrix completion problem

The matrix completion problem consists of reconstructing an unknown low rank matrix from a given subset of observed entries. Specifically, let  $M \in \mathfrak{R}^{m \times n}$  be a low rank matrix,  $\Omega$  be a subset of the indices of entries  $\{1, \dots, l\} \times \{1, \dots, n\}$ , the convex model is

$$\min\{\|X\|_* \mid X_\Omega = M_\Omega\}, \tag{6.2}$$

where  $\|\cdot\|_*$  denotes the nuclear norm of a matrix, and the constraint means that  $X_{ij} = M_{ij}, \forall \{ij\} \in \Omega$ . Note that it can also be viewed as a projection equation  $P_\Omega(X) = P_\Omega(M)$ , where  $P_\Omega(\cdot)$  denotes the orthogonal projection onto  $\Omega$ . i.e.,



**Fig. 3** Performance of Algorithm 1 on four different dimension  $n$ . *Left* the number of iterations with respect to  $t$ ; *right* the computing time with respect to  $t$

$$P_{\Omega} : X_{i,j} \mapsto \begin{cases} X_{i,j}, & \text{if } \{i, j\} \in \Omega, \\ 0, & \text{if } \{i, j\} \notin \Omega, \end{cases} \tag{6.3}$$

thus, we have  $\|A^T A\| = 1$ .

Applying our customized PPA to MCP, we obtain the following updates:

$$\begin{cases} X^{k+1} = \operatorname{argmin} \left\{ \|X\|_* + \frac{r}{2} \|X - X^k - \frac{1}{r} \left( \Lambda^k - \frac{1}{s} (1-t)(X_{\Omega}^k - M_{\Omega}) \right)\|_F^2 \right\}, \\ \Lambda^{k+1} = \Lambda^k - \frac{1}{s} ((1+t)X_{\Omega}^{k+1} - t \cdot X_{\Omega}^k - M_{\Omega}). \end{cases} \tag{6.4}$$

The first term of the above algorithm has a closed form solution that is given by

$$X^{k+1} = \operatorname{Shrink} \left\{ X^k + \frac{1}{r} \left( \Lambda^k - \frac{1}{s} (1-t)(X_{\Omega}^k - M_{\Omega}) \right), \frac{1}{r} \right\} \tag{6.5}$$

where the shrinkage operator is defined by

$$\operatorname{Shrink}(Z, \mu) = U \operatorname{diag}(\max\{\sigma - \mu, 0\}) V^T, \tag{6.6}$$

where  $U \operatorname{diag}(\sigma) V^T$  is the SVD decomposition of  $Z$ , see [Cai et al. \(2010\)](#) for details.

In our experiment, we generate the rank- $r$  matrix  $M$  as a product  $M_L M_R^T$ , where  $M_L$  and  $M_R$  are independent  $n \times r$  matrices who have i.i.d. Gaussian entries. The set of observations  $\Omega$  is sampled uniformly at random among all sets of cardinality  $|\Omega|$ . A useful quantity for reference is  $d_r = r(2m - r)$ , which is the number of the degrees of freedom in a non-symmetric matrix of rank- $r$  and the oversampling factor  $|\Omega|/d_r$  is defined as the ratio of the number of samples to the degrees of freedom.

For all tests, the rank- $r$  is set to be 10 and the oversampling factor  $|\Omega|/d_r$  is 5, and the penalty parameters is set to be  $r = 0.005, s = 1.01/r$ . The primal variable  $X^0$  and the dual variable  $\Lambda^0$  were all initialized as zeros( $n$ ). The stopping criteria is set to be

$$\frac{\|X_{\Omega}^k - M_{\Omega}\|_F}{\|M_{\Omega}\|_F} \leq 10^{-4}. \tag{6.7}$$

To expose the behaviors of Algorithm 1 with different values of the scalar  $t$ , we plot the number of iterations and runtime in seconds to solve Problem (6.2) for different values of  $t \in [-10, 10]$  in Fig. 3. We observe that Algorithm 1 exhibits numerical stability when  $t$  is getting changed. Thus, Algorithm 1 works well for a wide range of  $t$ , not only in the special cases  $\pm 1, 0$ .

## 7 Conclusions

In this paper, we proposed a class of customized proximal point algorithms for solving convex programming problem with linear constraints. Differ from the augmented Lagrangian multiplier method, the new methods can fully exploit the special structures and properties of some concrete models. In this context, they are very suitable for solving large-scale problems. We show that the PPA (He et al. 2013; Gu et al. 2014), LALM (Yang and Yuan 2013) and BOS (Zhang et al. 2010) are three special cases of our algorithms with  $t = \pm 1, 0$ , respectively. The global convergence and a worst-case  $O(1/t)$  convergence rate in nonergodic sense for this series of algorithms are proved in a uniform framework. Numerical performance demonstrated that the algorithms with the scalar  $t$  in a wide range also work well, and are competitive with the three special cases.

## References

- Bauschke HH, Combettes PL (2011) Convex analysis and monotone operator theory in Hilbert spaces. Springer, Berlin
- Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J Imaging Sci* 2:183–202
- Cai JF, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. *SIAM J Opt* 20:1956–1982
- Cai JF, Osher S, Shen Z (2009) Linearized Bregman iterations for compressed sensing. *Math Comput* 78:1515–1536
- Chambolle A, Pock T (2011) A first-order primal-dual algorithm for convex problems with applications to imaging. *J Math Imag Vis* 40:120–145
- Daubechies I, Defrise M, Mol CD (2004) An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun Pure Appl Math* 57:1413–1457
- Esser E (2009) Applications of Lagrangian-based alternating direction methods and connections to split Bregman. CAM report, UCLA, Center for Applied Mathematics
- Glowinski R, Le Tallec P (1989) Augmented Lagrangian and operator-splitting methods in nonlinear mechanics. SIAM, Philadelphia
- Goldstein T, Osher S (2009) The split Bregman method for L1-regularized problems. *SIAM J Imaging Sci* 2:323–343
- Gu G, He B, Yuan X (2014) Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach. *Comput Optim Appl* 59:135–161
- Han D, He H, Yang H, Yuan X (2014) A customized Douglas–Rachford splitting algorithm for separable convex minimization with linear constraints. *Numer Math* 127:167–200
- He B (2015) PPA-like contraction methods for convex optimization: a framework using variational inequality approach. *J Oper Res Soc China* 3:391–420
- He B, Xu H, Yuan X (2016) On the proximal Jacobian decomposition of ALM for multiple-block separable convex minimization problems and its relationship to ADMM. *J Sci Comput* 3:1204–1217
- He B, Yuan X (2012) Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM J Imaging Sci* 5:119–149
- He B, Yuan X, Zhang W (2013) A customized proximal point algorithm for convex minimization with linear constraints. *Comput Optim Appl* 56:559–572
- Hestenes MR (1969) Multiplier and gradient methods. *J Optim Theory Appl* 4:303–320
- Huang B, Ma S, Goldfarb D (2013) Accelerated linearized Bregman method. *J Sci Comput* 54:428–453
- Li M, Jiang Z, Zhou Z (2015) Dual-primal proximal point algorithms for extended convex programming. *Int J Comput Math* 92:1473–1495
- Li X, Yuan X (2015) A proximal strictly contractive Peaceman–Rachford splitting method for convex programming with applications to imaging. *SIAM J Imaging Sci* 8:1332–1365
- Ma S, Goldfarb D, Chen L (2011) Fixed point and Bregman iterative methods for matrix rank minimization. *Math Prog* 128:321–353
- Ma S (2016) Alternating proximal gradient method for convex minimization. *J Sci Comput* 68:546–572

- Martinet B (1970) *Revue Française d'Informatique et de Recherche Oprationnelle*. Rgularisation d'inquations variationnelles par approximations successives 4:154–158
- Parikh N, Boyd SP (2014) Proximal algorithms. *Found Trends Opt* 1:127–239
- Powell MJ (1969) A method for non-linear constraints in minimization problems. In: Fletcher R (ed) *Optimization*. Academic Press, Waltham Massachusetts, pp 283–298
- Rockafellar RT (1976) Monotone operators and the proximal point algorithm. *SIAM J Control Optim* 14:877–898
- Starck JL, Murtagh F, Fadili JM (2010) *Sparse image and signal processing, wavelets, curvelets. Morphological diversity*. Cambridge University Press, Cambridge
- Yang J, Yuan X (2013) Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Math Comput* 82:301–329
- Yin W, Osher S, Goldfarb D, Darbon J (2008) Bregman iterative algorithms for  $l_1$ -minimization with applications to compressed sensing. *SIAM J Imaging Sci* 1:143–168
- You Y, Fu X, He B (2013) Lagrangian PPA-based contraction methods for linearly constrained convex optimization. *Pacific J Optim* 10:201–213
- Zhang X, Burger M, Bresson X, Osher S (2010) Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM J Imaging Sci* 3:253–276