Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2018.m1621

Vol. **11**, No. 1, pp. 187-210
February 2018

Adaptive Parallel Primal-Dual Method for Saddle Point Problems

Xiayang Zhang*

Department of Mathematics, Nanjing University, Nanjing210093, China

Received 22 June 2016; Accepted (in revised version) 24 March 2017

Abstract. The primal-dual hybrid gradient method is a classic way to tackle saddle-point problems. However, its convergence is not guaranteed in general. Some restrictions on the step size parameters, e.g., $\tau\sigma \leq 1/\|A^TA\|$, are imposed to guarantee the convergence. In this paper, a new convergent method with no restriction on parameters is proposed. Hence the expensive calculation of $\|A^TA\|$ is avoided. This method produces a predictor like other primal-dual methods but in a parallel fashion, which has the potential to speed up the method. This new iterate is then updated by a simple correction to guarantee the convergence. Moreover, the parameters are adjusted dynamically to enhance the efficiency as well as the robustness of the method. The generated sequence monotonically converges to the solution set. A worst-case $\mathcal{O}(1/t)$ convergence rate in ergodic sense is also established under mild assumptions. The numerical efficiency of the proposed method is verified by applications in LASSO problem and Steiner tree problem.

AMS subject classifications: 49K35, 49M27, 90C25, 65K10

Key words: Adaptive, Parallel, Primal-dual method, Saddle-point problem, LASSO.

1. Introduction

This paper is concerned about solving the following saddle-point problem:

$$\min_{x\in X}\max_{y\in Y}\ g(x)+y^TAx-f^*(y), \tag{1.1}$$

where $A\in\Re^{m\times n}$, $X\subset\Re^n$, $Y\subset\Re^m$ are closed convex sets, g, f^* are convex functions and f^* is the conjugate function of a convex function f, $f^*(x)=\sup_{x\in domf}\left(y^Tx-f(x)\right)$. Note that this saddle-point problem is a primal-dual formulation of the nonlinear primal problem

$$\min_{x\in X}\ g(x)+f(Ax). \tag{1.2}$$

*Corresponding author. *Email address:* 369318324@qq.com (X. Y. Zhang)

The formulation (1.1) has a wide range of applications including image denoising [6,24], statistical learning [21], compressive sensing [9] etc.

In many problems of practical interest, g and f^* do not share common properties, making it difficult to derive numerical schemes for (1.1) that address both terms simultaneously. Fortunately, it frequently occurs in practice that efficient algorithms exist for minimizing g and f^* separately. The primal-dual hybrid gradient (PDHG) method was first mentioned in [24] to tackle total variation (TV) minimization problems. This method removes the coupling between g and f^*, enabling each term to be addressed separately. Because it decouples g and f^*, the steps of PDHG can often be written explicitly, as opposed to other splitting methods that require expensive minimization sub-problems. As a result PDHG shows high numerical efficiency when applied to total variation (TV) minimization problems. However the convergence of PDHG is highly dependent on the choice of parameters. In [4], Chambolle and Pock's (CP) method improved PDHG method by a change on dual variable update. Their method is convergent and numerically competitive. CP method was further studied by He and Yuan in [12] . They explained the method from the aspect of Proximal Point Algorithm (PPA) and a relaxation factor was also introduced to PPA scheme to accelerate the convergence. More recently, Goldstein et al. introduced the Adaptive Primal-Dual Splitting (APD)method in [9] which tunes the step size parameters automatically for the CP method. The primal-dual decomposition method was proposed by O'Connor and Vandenberghe in [18] which applied the Douglas-Rachford splitting method to various splitting of the primal-dual optimality conditions.

More specifically, a general framework of some existing primal-dual methods solve the saddle-point problems (1.1) by the following procedures:

$$\begin{cases} x^{k+1} = \arg\min_{x\in X} \{g(x) + x^T A^T y^k + \frac{1}{2\tau}\|x - x^k\|^2\}, \\ \bar{x}^k = x^{k+1} + \theta(x^{k+1} - x^k), \\ y^{k+1} = \arg\min_{y\in Y} \{f^*(y) - y^T A\bar{x}^k + \frac{1}{2\sigma}\|y - y^k\|^2\}. \end{cases} \tag{1.3}$$

In (1.3), θ is called the combination parameter, $\sigma > 0$ and $\tau > 0$ are proximal parameters of the regularization terms, also referred as step size parameters in e.g. [9]. In [4], it was shown that the primal-dual procedure (1.3) is closely related to the extrapolational gradient methods in [15, 20], the Douglas-Rachford splitting method in [8, 16] and the alternating direction method of multipliers (ADMM) in [5]. With specific choice of parameters in (1.3), some existing primal-dual algorithms for (1.1) are recovered, and their convergence can be guaranteed when certain restriction are imposed on these parameters. Below are some examples.

- When $\theta = 0$, the primal-dual procedure in (1.3) reduces to the PDHG scheme in [24] which is indeed the Arrow-Hurwicz algorithm in [1]. This scheme has shown numerical efficiencies in [24] for TV image restoration problems. In [6], the convergence of the PDHG method has been studied insightfully by imposing additional restrictions ensuring that the parameters $\sigma > 0$ and $\tau > 0$ are small. However, a counter example has been given in [11] to show that PDHG method could be divergent even if $\sigma > 0$ and $\tau > 0$ are fixed at very small values.

- When $\theta \in [0,1]$, the CP algorithm proposed in [4] is recovered. Note that it could be numerically beneficial to tune the parameters σ and τ as shown in, e.g., [12,23]; and it is still possible to investigate the convergence of the primal-dual scheme (1.3) with adaptively-adjusting proximal parameters, see, e.g., [2,6,9].

- When $\theta = 1$, by the analysis in [4], the convergence of (1.3) can be guaranteed under the condition

$$\tau\sigma \leq 1/\|A^T A\|. \tag{1.4}$$

  In [12], a primal-dual scheme (1.3) with $\theta = 1$ is proved to be an application of the the proximal point algorithm (PPA) in [17], and thus the acceleration scheme in [10] can be immediately used to accelerate the primal-dual procedure (see Algorithm 4 in [12]). Its numerical efficiency has also been verified therein. This PPA revisit has been further studied in [19], in which a preconditioning version of the primal-dual procedure (1.3) was proposed.

- When $\theta \in [-1,1]$, it is shown in [12] (see also Lemma 3.1) that the matrix associated with the proximal regularization terms in (1.3) is not symmetric and thus the scheme (1.3) cannot be categorized as an application of the PPA. Nevertheless, the convergence can be guaranteed if the output of the primal-dual subroutine (1.3) is further corrected by some correction steps (see Algorithms 1 and 2 in [12]).

At the Discussion part of [4], some remaining challenges for the existing primal-dual methods are posed. The first challenge is how to deal with the linear operator A with a large (or unknown) norm. Since most primal-dual methods e.g. PDHG method in [24], CP method in [4] and RPPA in [12], require (1.4) be satisfied to guarantee the convergence, the calculation of the norm must take place before initializing the parameters. This computational load can be huge as the dimension of A grows. Another challenge mentioned is how to automatically determine the smoothness parameters or to locally adapt to the regularity of the objective. It is thus natural to ask whether we can make some improvements on the primal-dual procedure (1.3) so that there are no restrictions on parameter σ and τ? In addition, can the parameters be adjusted automatically during the iteration progress?

It is worth mentioning that the primal and dual variables are updated in turns when $\theta \in (-1,1]$ in (1.3); On the other hand the primal and dual variables are updated in parallel when $\theta = -1$. So one more question is whether we can find a method based on the parallel version of the primal-dual subroutine (1.3).

The Parallel Primal-Dual(PPD) Algorithm studied in this paper provides an affirmative answer to the above questions. Our method comprises of two stages: a parallel version of primal-dual subroutine (1.3) as prediction stage and a simple correction stage. At correction stage, step size α_k is calculated to guarantee the convergence of PPD method. At the end of each iteration, parameters τ_k and σ_k are adjusted by the calculation of residuals (see also [3,9,14]). Both size and ratio of τ_k and σ_k are tuned quantitatively. Parameter adjustment by calculation during the iteration makes the method less sensitive to initial parameter choices. In practice it is quite helpful since the optimal parameters are difficult

to determine. The PPD method adjust the parameters differently from the APD method. In PPD method the aggressiveness of adjustment is based on the ratio of residuals while in APD method, the aggressiveness decreases geometrically with iteration increases. The initial value $\tau_0\sigma_0$ in PPD method can take any value while they are set to be very large in APD method. The quantitative approach of parameter adjustment in our method is more efficient than APD in [9] by the experiments in later sections.

The major contributions of this paper are listed below.

- The proposed methods compute the primal and dual variables in a parallel fashion, while the other primal-dual methods update the primal and dual variables alternatively. This approach can reduce the computational time greatly if parallel computing is performed.
- The proposed algorithm does not require any prior information about the linear operator A. The parameter σ and τ can take any positive values. In fact, as the dimension of the problem grows, it becomes much more expensive to evaluate the spectral norm of A which is essential for the implementation of most primal-dual methods.
- The parameters σ and τ are adjusted dynamically during the iteration progress. This not only speed up the convergence but also make the method more robust to different problem settings. Unlike the non-adaptive primal-dual methods e.g. PDHG method, CP method and RPPA whose convergence speed rely heavily on the precise choice of parameters σ and τ, our method performs consistently well for different initial parameters.

The contents of this paper are arranged as follows. The new algorithm is proposed in Section 2 and its convergence and computational complexity are analyzed in Sections 3 and 4 respectively. In Section 5 the new algorithm is tested in some applications and the numerical results are reported. Finally the conclusion of this paper and future works are discussed in Section 6.

2. The parallel primal-dual algorithm

For simplicity primal and dual variables are grouped together and defined as follows:

$$u = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \theta(u) = g(x) + f^*(y), \quad F(u) = \begin{pmatrix} A^T y \\ -Ax \end{pmatrix}, \quad \Omega = X \times Y.$$

The following notations are also used:

$$Q_k = \begin{pmatrix} \tau_k^{-1} I_n & -A^T \\ A & \sigma_k^{-1} I_m \end{pmatrix}, \quad M_k = \begin{pmatrix} I_n & -\tau_k A^T \\ \sigma_k A & I_m \end{pmatrix}, \quad H_k = \begin{pmatrix} \tau_k^{-1} I_n & 0 \\ 0 & \sigma_k^{-1} I_m \end{pmatrix},$$

where $\tau_k, \sigma_k > 0$ are positive. It is clear that

$$M_k = H_k^{-1} Q, \qquad Q_k^T + Q_k = 2H_k.$$

Algorithm 2.1 Parallel Primal-Dual Algorithm.

1. Initialize $\tau_0 > 0$, $\sigma_0 > 0$, $\overline{\tau} > 0$, $\overline{\sigma} > 0$, $\eta \in (0,1)$, $x^0 \in X$, $y^0 \in Y$.

2. While p_k, $d_k >$ tolerance do:

   (a) Prediction

$$\begin{cases} \tilde{x}^k = \arg\min_{x\in X} \ \{g(x) + x^T A^T y^k + \frac{1}{2\tau_k}\|x - x^k\|^2\}, \\ \tilde{y}^k = \arg\min_{y\in Y} \ \{f^*(y) - y^T A x^k + \frac{1}{2\sigma_k}\|y - y^k\|^2\}. \end{cases} \tag{2.1}$$

   (b) Compute residual norms

$$\begin{cases} p_k = \tau_k^{-1}\left\|(x^k - \tilde{x}^k - \tau_k A^T(y^k - \tilde{y}^k))\right\|, \\ d_k = \sigma_k^{-1}\left\|(y^k - \tilde{y}^k + \sigma_k A(x^k - \tilde{x}^k))\right\|. \end{cases}$$

   (c) Calculate step size

$$\alpha_k = \left(\|x^k - \tilde{x}^k\|^2/\tau_k + \|y^k - \tilde{y}^k\|^2/\sigma_k\right)/\left(\tau_k p_k^2 + \sigma_k d_k^2\right).$$

   (d) Correction

$$\begin{cases} x^{k+1} = x^k - \alpha_k\left(x^k - \tilde{x}^k - \tau_k A^T(x^k - \tilde{x}^k)\right), \\ y^{k+1} = y^k - \alpha_k\left(y^k - \tilde{y}^k + \sigma_k A(y^k - \tilde{y}^k)\right). \end{cases} \tag{2.2}$$

   (e) Update Parameters (If $p_k \geq 2d_k$ or $p_k \leq \frac{1}{2}d_k$)

$$\begin{cases} \tau_{k+1} = \min\left(\max(\sqrt{\alpha_k p_k/(1-\alpha_k)d_k},\ 1-(\eta)^k)\tau_k,\ \overline{\tau}\right), \\ \sigma_{k+1} = \min(\max\left(\sqrt{\alpha_k d_k/(1-\alpha_k)p_k},\ 1-(\eta)^k)\sigma_k,\ \overline{\sigma}\right). \end{cases} \tag{2.3}$$

3. End while.

Moreover,

$$Q_k^T + Q_k \quad \text{is positive definite.}$$

The Parallel Primal-Dual(PPD) Algorithm is presented in Algorithm 2.1. The loop in Algorithm 2.1 begins by performing the general framework of primal-dual methods (1.3) with $\theta = -1$ in step 3.

In step 4 we compute the primal and dual residuals and store their norms in p_k and d_k respectively. Since the convergence of the primal-dual methods can be measured by the norm of the residuals (see also [3,9,14]), we can use them as the stopping criteria.

Most primal-dual methods require the parameter $\tau\sigma \leq 1/\|A\|_2^2$ to guarantee the con-

vergence. In this paper we try to deal with issue of not knowing the spectral norm of A. When the condition $\tau\sigma \leq 1/\|A\|_2^2$ is not met, a variate step-size is chosen in step 5 to guarantee the convergence. This step size is used in step 6 for correction.

Step 1 to 6 alone is still sensitive to the initial parameter values according to some preliminary numerical experiments. There has to be a way to tune the parameters automatically so that the algorithm fit different type of problems. In addition the tuning should not be intuitive but rather be quantitative. In fact, a simple modification allows the method to be applied when the optimal value of parameters is unavailable.

Without any prior knowledge of the linear operator A, assume $\tau_k\sigma_k = c^{-1}/\|A\|_2^2$ for an unknown constant c. Lemma 3.2 indicate that α_k is a rough approximation of $c/(1+c)$ and $\alpha_k \geq c/(1+c)$, hence $c^{-1} \geq (1-\alpha_k)/\alpha_k$. In practice we let

$$\tau_{k+1}\sigma_{k+1} = \frac{\alpha_k}{1-\alpha_k}\tau_k\sigma_k,$$

the value of $\tau_{k+1}\sigma_{k+1}$ will be a bit larger than $1/\|A\|_2^2$, the optimal value for most primal-dual methods.

Moreover, the primal and dual residuals should be of the same scale so that the convergence of primal and dual variables are balanced. The purpose of balancing parameters is also explained in [9] and [3]. In our approach, we set the aggressiveness of balancing direct proportional to the residual ratio, i.e., $\tau_{k+1}/\sigma_{k+1} = p_k/d_k$.

Combining the two goals above, our quantitative parameter upstate scheme is

$$\begin{cases} \tau_{k+1} = \max\left(\sqrt{\alpha_k p_k/(1-\alpha_k)d_k},\ 1-(\eta)^k\right)\tau_k, \\ \sigma_{k+1} = \max\left(\sqrt{\alpha_k d_k/(1-\alpha_k)p_k},\ 1-(\eta)^k\right)\sigma_k. \end{cases}$$

Take note of that η measures the aggressiveness bound for parameter adjustment. In practice η can be close to 1 so that it does not affect the parameter tuning yet could guarantee the convergence In addition, we set ceilings $\overline{\tau}$ and $\overline{\sigma}$ for the parameters for the convergence purposes. In practice, the value of τ and σ approach to their optimal value fast and vibrate around that value. Therefore we can set the ceilings to be very large so that it hardly has any effect on parameter tuning.

The PPD method and APD method adjust the parameters different in the following ways. In PPD method the aggressiveness of adjustment is based on the ratio of residuals, the large value of p_k/d_k, the more aggressive $\tau_k\sigma_k$ changes. In APD method, the aggressiveness decreases geometrically with iteration increases. The initial value $\tau_0\sigma_0$ in PPD method can take any value while they are set to be very large in APD method. The quantitative approach of parameter adjustment in our method is more efficient than APD in [9] by the experiments in later sections.

3. Convergence analysis

The following assumptions are imposed to guarantee the convergence of the PPD algorithm.

1. The sequences $\{\tau_k\}$ and $\{\sigma_k\}$ are positive and bounded.
2. The sequences $\{\phi_k\}$ is summable, where $\phi_k = \max\{\frac{\tau_k-\tau_{k+1}}{\tau_k}, \frac{\sigma_k-\sigma_{k+1}}{\sigma_k}, 0\}$.

Apparently Algorithm 2.1 satisfies Assumption 1 because of the ceilings $\overline{\tau}$ and $\overline{\sigma}$. The geometric series $\{\eta^k\} = \{(\eta)^k\}$ in Algorithm 2.1 makes Assumption 2 satisfied as $\eta < 1$. If the ceilings $\overline{\tau}$ and $\overline{\sigma}$ are not reached, (2.3) is equivalent to

$$\begin{cases} \tau_{k+1} = \max\left(\sqrt{\alpha_k p_k/(1-\alpha_k)d_k},\, 1-(\eta)^k\right)\tau_k, \\ \sigma_{k+1} = \max\left(\sqrt{\alpha_k d_k/(1-\alpha_k)p_k},\, 1-(\eta)^k\right)\sigma_k. \end{cases}$$

Thus $\tau_{k+1}/\tau_k \geq 1-(\eta)^k$ and $(\tau_k - \tau_{k+1})/\tau_k \leq (\eta)^k$. For the same reason, $(\sigma_k - \sigma_{k+1})/\sigma_k \leq (\eta)^k$ hence $\phi_k \leq (\eta)^k$. If the ceiling is reached, $\tau_{k+1} = \overline{\tau} \geq \tau_k$, $\phi_k = 0$. Therefore the sequences $\{\phi_k\}$ is summable.

We first present the following VI reformulation of (1.1): Find $u^* = (x^*, y^*)$ such that

$$\mathrm{VI}(\Omega, F, \theta): \quad u^* \in \Omega, \quad \theta(u) - \theta(u^*) + (u-u^*)^T F(u^*) \geq 0, \quad \forall\, u \in \Omega = X \times Y.$$

Obviously, the mapping $F(u)$ is affine with a skew-symmetric matrix, and it is thus monotone. We denote the solution set of VI (Ω, F, θ) by Ω^*, and assume it is nonempty.

Lemma 3.1. *For given $u^k = \begin{pmatrix} x^k \\ y^k \end{pmatrix}$, let $\tilde{u}^k$ be defined by* (2.1). *Then*

$$\tilde{u}^k \in \Omega, \quad \theta(u) - \theta(\tilde{u}^k) + (u-\tilde{u}^k)^T F(\tilde{u}^k) \geq (u-\tilde{u}^k)^T Q_k(u^k - \tilde{u}^k), \quad \forall\, u \in \Omega.$$

Proof. Using the optimal condition of (2.1), we obtain

$$\begin{aligned} &g(x) - g(\tilde{x}^k) + (x-\tilde{x}^k)^T\{A^T y^k + \tau_k^{-1}(\tilde{x}^k - x^k)\} \geq 0, && \forall x \in X, \\ &f^*(y) - f^*(\tilde{y}^k) + (y-\tilde{y}^k)^T\{-Ax^k + \sigma_k^{-1}(\tilde{y}^k - y^k)\} \geq 0, && \forall y \in Y. \end{aligned}$$

Combining the above two inequalities yields $\tilde{u}^k = (\tilde{x}^k, \tilde{y}^k)^T \in \Omega$, and

$$\begin{pmatrix} (g(x) - g(\tilde{x}^k)) + \\ (f^*(y) - f^*(\tilde{y}^k)) \end{pmatrix} + \begin{pmatrix} x-\tilde{x}^k \\ y-\tilde{y}^k \end{pmatrix}^T \cdot \left\{ \begin{pmatrix} A^T\tilde{y}^k \\ -A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} \tau_k^{-1}I & -A^T \\ A & \sigma_k^{-1}I \end{pmatrix} \begin{pmatrix} \tilde{x}^k - x^k \\ \tilde{y}^k - y^k \end{pmatrix} \right\} \geq 0$$

for all $u \in \Omega$. The assertion is proved using the notations in previous section. □

Lemma 3.2. *There exists a constant $c > 0$ such that*

$$\tau_k^{-1}\sigma_k^{-1} \geq c\|A\|_2^2, \qquad \forall k > 0,$$

and the corresponding step size $\alpha_k \geq \frac{c}{1+c}$ for all $k > 0$.

Proof. Such c can be easily found since τ and σ are bounded above. Steps 5 and 6 in Algorithm 2.1 are equivalent to

$$\alpha_k = \|u^k - \tilde{u}^k\|_{H_k}^2 / \|M_k(u^k - \tilde{u}^k)\|_{H_k}^2, \tag{3.1a}$$

$$u^{k+1} = u^k - \alpha_k M_k(u^k - \tilde{u}^k). \tag{3.1b}$$

On the other hand, we have

$$\begin{aligned}
&\left\|M_k(u^k - \tilde{u}^k)\right\|_{H_k}^2 \\
=&\tau_k^{-1}\left\|x^k - \tilde{x}^k\right\|^2 + \sigma_k^{-1}\left\|y^k - \tilde{y}^k\right\|^2 + \sigma_k\left\|A(x^k - \tilde{x}^k)\right\|^2 + \tau_k\left\|A^T(y^k - \tilde{y}^k)\right\|^2 \\
\le&\tau_k^{-1}\left\|x^k - \tilde{x}^k\right\|^2 + \sigma_k^{-1}\left\|y^k - \tilde{y}^k\right\|^2 + c^{-1}\tau_k^{-1}\left\|x^k - \tilde{x}^k\right\|^2 + c^{-1}\sigma_k^{-1}\left\|y^k - \tilde{y}^k\right\|^2 \\
=&\left(1 + c^{-1}\right)\left(\tau_k^{-1}\left\|x^k - \tilde{x}^k\right\|^2 + \sigma_k^{-1}\left\|y^k - \tilde{y}^k\right\|^2\right) \\
=&\frac{1+c}{c}\left\|u^k - \tilde{u}^k\right\|_{H_k}^2.
\end{aligned}$$

Substituting it in (3.1a), the assertion is proved. □

Lemma 3.3. *For given* $u^k = \begin{pmatrix} x^k \\ y^k \end{pmatrix}$, *let* $\tilde{u}^k$ *and* u^{k+1} *be defined by* (2.1) *and* (2.2). *If* $\tau_k^{-1}\sigma_k^{-1} \ge c\|A\|_2^2$, *then*

$$\left\|u^{k+1} - u^*\right\|_{H_k}^2 \le \left\|u^k - u^*\right\|_{H_k}^2 - \frac{c}{1+c}\left\|u^k - \tilde{u}^k\right\|_{H_k}^2, \quad \forall u^* \in \Omega^*. \tag{3.2}$$

Proof. Substituting $u = u^*$ in (3.1), it follows that

$$\left(\tilde{u}^k - u^*\right)^T Q_k\left(u^k - \tilde{u}^k\right) \ge \theta\left(\tilde{u}^k\right) - \theta(u^*) + \left(\tilde{u}^k - u^*\right)^T F\left(\tilde{u}^k\right), \quad \forall\, u^* \in \Omega^*.$$

Note that

$$\left(\tilde{u}^k - u^*\right)^T F\left(\tilde{u}^k\right) = \left(\tilde{u}^k - u^*\right)^T F(u^*), \qquad \theta\left(\tilde{u}^k\right) - \theta(u^*) + \left(\tilde{u}^k - u^*\right)^T F(u^*) \ge 0.$$

Combining the above inequalities yields

$$\left(u^k - u^*\right)^T Q_k\left(u^k - \tilde{u}^k\right) \ge \left(u^k - \tilde{u}^k\right)^T Q_k\left(u^k - \tilde{u}^k\right) = \left\|u^k - \tilde{u}^k\right\|_{H_k}^2.$$

Using (3.1a) and (3.1b), the inequality below is established.

$$\begin{aligned}
&\left\|u^k - u^*\right\|_{H_k}^2 - \left\|u^{k+1} - u^*\right\|_{H_k}^2 \\
=&\left\|u^k - u^*\right\|_{H_k}^2 - \left\|(u^k - u^*) - (u^k - u^{k+1})\right\|_{H_k}^2 \\
=&\left\|u^k - u^*\right\|_{H_k}^2 - \left\|(u^k - u^*) - \alpha_k M_k(u^k - \tilde{u}^k)\right\|_{H_k}^2 \\
=&2\alpha_k\left(u^k - u^*\right)^T Q_k\left(u^k - \tilde{u}^k\right) - \left(\alpha_k\right)^2\left(u^k - \tilde{u}^k\right)M_k^T H_k M_k\left(u^k - \tilde{u}^k\right) \\
\ge&2\alpha_k\left\|u^k - \tilde{u}^k\right\|_{H_k}^2 - \alpha_k\left(\alpha_k\left(u^k - \tilde{u}^k\right)M_k^T H_k M_k\left(u^k - \tilde{u}^k\right)\right) \\
=&\alpha_k\left\|u^k - \tilde{u}^k\right\|_{H_k}^2.
\end{aligned}$$

Using Lemma 3.2, the assertion follows immediately. □

Lemma 3.4. *For all initial $u_0 \in \Omega$, let the sequence $\{u_k\}$ be defined by* (2.1). *Then*

$$\begin{aligned}&\sum_{k=1}^{\infty}\left(\left\|u^k-u\right\|_{H_k}^2-\left\|u^k-u\right\|_{H_{k-1}}^2\right)\\ \le& 2C_\phi C_H\|u-u^*\|^2+2C_\phi C_U\left\|u^0-u^*\right\|_{H_0}^2,\end{aligned}\tag{3.3}$$

where $C_U=\prod_{i=1}^{\infty}(1-\phi_i)^{-1}$, $C_\phi=\sum_{k=0}^{\infty}\phi_k$ and C_H is a constant such that

$$\|u-u^*\|_{H_k}^2\le C_H\|u-u^*\|^2,\quad \forall k>0.$$

Proof. We observe that

$$H_{k+1}=H_k\begin{pmatrix}\tau_k/\tau_{k+1} & 0\\ 0 & \sigma_k/\sigma_{k+1}\end{pmatrix}\le H_k(1-\phi_k)^{-1},\tag{3.4}$$

Therefore

$$\begin{aligned}&\left\|u^{k+1}-u^*\right\|_{H_{k+1}}^2\le\left\|u^{k+1}-u^*\right\|_{H_k}^2(1-\phi_k)^{-1}\le\left\|u^k-u^*\right\|_{H_k}^2(1-\phi_k)^{-1}\\ \le&\left\|u^0-u^*\right\|_{H_0}^2\prod_{i=1}^{k}(1-\phi_i)^{-1}\le\left\|u^0-u^*\right\|_{H_0}^2\prod_{i=1}^{\infty}(1-\phi_i)^{-1}.\end{aligned}\tag{3.5}$$

Since $\{\phi_k\}$ is summable, the product $C_U=\prod_{i=1}^{\infty}(1-\phi_i)^{-1}$ is finite, then

$$\|u^k-u^*\|_{H_k}^2\le C_U\|u^0-u^*\|_{H_0}^2.\tag{3.6}$$

We use (3.6) to derive the following inequality.

$$\begin{aligned}&\sum_{k=1}^{\infty}\left(\left\|u^k-u\right\|_{H_k}^2-\left\|u^k-u\right\|_{H_{k-1}}^2\right)\\ \le&\sum_{k=1}^{\infty}\left(\left\|u^k-u\right\|_{H_k}^2-\left\|u^k-u\right\|_{H_k}^2(1-\phi_{k-1})\right)\\ =&\sum_{k=1}^{\infty}\phi_{k-1}\left\|u^k-u\right\|_{H_k}^2\\ \le&\sum_{k=1}^{\infty}\phi_{k-1}\left(2\left\|u-u^*\right\|_{H_k}^2+2\left\|u^k-u^*\right\|_{H_k}^2\right)\\ \le&\sum_{k=1}^{\infty}2\phi_{k-1}\left(C_H\left\|u-u^*\right\|^2+C_U\left\|u^0-u^*\right\|_{H_0}^2\right)\\ \le&2C_\phi C_H\left\|u-u^*\right\|^2+2C_\phi C_U\left\|u^0-u^*\right\|_{H_0}^2.\end{aligned}$$

The proof is completed. □

Theorem 3.1. *The sequence* $\{u^k\}$ *generated by Algorithm* 2.1 *converges to a solution of* $VI(\Omega, F, \theta)$ (3).

Proof. Summing (3.2) for $1 \le k \le n$ leads to

$$\begin{aligned}&\sum_{k=1}^{n} \frac{c}{1+c}\left\|u^k - \tilde{u}^k\right\|_{H_k}^2 \\ \le &\sum_{k=1}^{n}\left(\left\|u^k - u^*\right\|_{H_k}^2 - \left\|u^k - u^*\right\|_{H_{k-1}}^2\right) + \left\|u^0 - u^*\right\|_{H_0}^2 - \left\|u^{n+1} - u^*\right\|_{H_n}^2.\end{aligned}$$

Letting $n \to \infty$ and applying (3.3), we obtain

$$\sum_{k=1}^{\infty}\left\|u^k - \tilde{u}^k\right\|_{H_k}^2 < +\infty.$$

It follows that $\lim_{k\to\infty}\left\|u^k - \tilde{u}^k\right\|_{H_k}^2 = 0$. Since

$$\overline{\tau} \ge \tau_{k+1} = \tau_0 \prod_{i=0}^{k}(\tau_{i+1}/\tau_i) \ge \tau_0 \prod_{i=0}^{\infty}(1-\phi_i) \quad \forall k,$$

the sequences $\{\tau_k\}$ is bounded. So is $\{\sigma_k\}$. Therefore $\|H_k\|$ and $\|Q_k\|$ are also bounded. Hence

$$\lim_{k\to\infty}\left\|u^k - \tilde{u}^k\right\| = 0. \tag{3.7}$$

Since $\{u^k\}$ is bounded by (3.6), $\{\tilde{u}^k\}$ is also bounded.

Let u^∞ be a cluster point of $\{\tilde{u}^k\}$ and $\{\tilde{u}^{k_j}\}$ is a subsequence which converges to u^∞. Let $\{u^k\}$ and $\{u^{k_j}\}$ be the induced sequences by $\{\tilde{u}^k\}$ and $\{\tilde{u}^{k_j}\}$, respectively. It follows from lemma 3.1 that

$$\tilde{u}^{k_j} \in \Omega, \quad \theta(u) - \theta\left(\tilde{u}^{k_j}\right) + \left(u - \tilde{u}^{k_j}\right)^T F\left(\tilde{u}^{k_j}\right) \ge \left(u - u^{k_j}\right)^T Q_k\left(u^{k_j} - \tilde{u}^{k_j}\right), \quad \forall\, u \in \Omega.$$

Since $\|Q_k\|$ is bounded, it follows from the continuity of $\theta(u)$ and $F(u)$ that

$$u^\infty \in \Omega, \quad \theta(u) - \theta(u^\infty) + (u - u^\infty)^T F(u^\infty) \ge 0, \quad \forall\, u \in \Omega.$$

The above variational inequality indicates that u^∞ is a solution of VI (Ω, F). By using (3.7) and $\lim_{j\to\infty} u^{k_j} = u^\infty$, the subsequence $\{u^{k_j}\}$ converges to u^∞.

Similar to (3.5), we have

$$\left\|u^{n+1} - u^\infty\right\|_{H_{n+1}} \le \left\|u^{k_j} - u^\infty\right\|_{H_{k_j}} \prod_{i=k_j}^{n}(1-\phi_i)^{-1},$$

where $k_j \le n < k_j + 1$. Let n approaches infinity, $\|H_n\|$ converges to a positive value,

$$\lim_{n\to\infty} \left\|u^{n+1} - u^{\infty}\right\|_{H_{n+1}} \le \lim_{j\to\infty} \left\|u^{k_j} - u^{\infty}\right\|_{H_{k_j}} \cdot 1 = 0.$$

Consequently,

$$\lim_{n\to\infty} \left\|u^{n+1} - u^{\infty}\right\| = 0,$$

i.e., $\{u^k\}$ converges to u^{∞}. The proof is completed. □

4. Computational complexity

Theorem 4.1. *The solution set of VI (Ω, F, θ) is convex and it can be characterized as*

$$\Omega^* = \bigcap_{u\in\Omega} \{\tilde{u} \in \Omega : \big(\theta(u) - \theta(\tilde{u})\big) + (u - \tilde{u})^T F(u) \ge 0\}.$$

Proof. The proof is an incremental extension of Theorem 2.3.5 in [7], or see the proof of Theorem 2.1 in [13]. □

Theorem 4.1 implies that $\tilde{u} \in \Omega$ is an approximate solution of VI (Ω, F, θ) with the accuracy $\epsilon > 0$ if it satisfies

$$\theta(u) - \theta(\tilde{u}) + (u - \tilde{u})^T F(u) \ge -\epsilon, \quad \forall u \in \Omega \cap \mathscr{D}(\tilde{u}), \tag{4.1}$$

where $\mathscr{D}(\tilde{u}) = \{u \,|\, \|u - \tilde{u}\| \le 1\}$ is a neighborhood of $\tilde{u}$.

Theorem 4.2. *The Algorithm* 2.1 *converges at rate of $\mathscr{O}(1/t)$ in ergodic sense. More specifically let the sequence $\{u^k\}$ be generated by Algorithm* 2.1 *with arbitrary initial input u_0, for the sequence $\{\tilde{u}_t\}$ defined by*

$$\tilde{u}_t = \frac{1}{\sum_{k=0}^{t} \alpha_k} \sum_{k=0}^{t} \alpha_k \tilde{u}^k,$$

the convergence bound below is satisfied:

$$\begin{aligned} &\theta(u) - \theta(\tilde{u}_t) + (u - \tilde{u}_t)^T F(u) \\ \ge &\frac{(1+c)/2c}{t}\big(- \|u - u_0\|^2_{H_0} - 2C_\phi C_U \|u_0 - u^*\|^2_{H_0} - 2C_\phi C_H \|u - u^*\|^2\big). \end{aligned} \tag{4.2}$$

Proof. It follows from (3.1a) and (3.1b) that

$$\begin{aligned} &\theta(u) - \theta(\tilde{u}) + (u - \tilde{u})^T F(u) \\ \ge &\big(u - \tilde{u}^k\big)^T Q_k \big(u^k - \tilde{u}^k\big) \\ = &\frac{1}{\alpha_k}\big(u - \tilde{u}^k\big)^T H_k \big(u^k - u^{k+1}\big) \\ = &\frac{1}{2\alpha_k}\left(\|u - u^{k+1}\|^2_{H_k} - \|u - u^k\|^2_{H_k} + \|u^k - \tilde{u}^k\|^2_{H_k} + \|u^{k+1} - \tilde{u}^k\|^2_{H_k}\right), \end{aligned} \tag{4.3}$$

where the last equality uses the polar identity for normed vector spaces, i.e.,

$$(a-b)^T H(c-d) = \frac{1}{2}\left\{\|a-d\|_H^2 - \|a-c\|_H^2\right\} + \frac{1}{2}\left\{\|c-b\|_H^2 - \|d-b\|_H^2\right\}.$$

Consider the last term of (4.3) and take note that $Q_k^T + Q_k = 2H_k$. We obtain

$$\begin{aligned}
&\left\|u^{k+1} - \tilde{u}^k\right\|_{H_k}^2 \\
=&\left\|(u^k - \tilde{u}^k) - \alpha_k M_k(u^k - \tilde{u}^k)\right\|_{H_k}^2 \\
=&\left\|u^k - \tilde{u}^k\right\|_{H_k}^2 - \alpha_k\left\|u^k - \tilde{u}^k\right\|_{Q+Q^T}^2 + \alpha_k^2\left\|M_k(u^k - \tilde{u}^k)\right\|_{H_k}^2 \\
=&\left\|u^k - \tilde{u}^k\right\|_{H_k}^2 - 2\alpha_k\left\|u^k - \tilde{u}^k\right\|_{H_k}^2 + \alpha_k\left\|u^k - \tilde{u}^k\right\|_{H_k}^2 \\
=&\left\|u^k - \tilde{u}^k\right\|_{H_k}^2 - \alpha_k\left\|u^k - \tilde{u}^k\right\|_{H_k}^2. \qquad (4.4)
\end{aligned}$$

Substitute (4.4) into (4.3) leads to

$$\begin{aligned}
&\theta(u) - \theta(\tilde{u}) + (u-\tilde{u})^T F(u) \\
\geq&\frac{1}{2\alpha_k}\left(\left\|u - u^{k+1}\right\|_{H_k}^2 - \left\|u - u^k\right\|_{H_k}^2 + \frac{1}{2}\left\|u^k - \tilde{u}^k\right\|_{H_k}^2\right).
\end{aligned}$$

Due to the monotonicity of $F(\cdot)$, we have

$$\begin{aligned}
&\alpha_k\left[\theta(u) - \theta(\tilde{u}) + (u-\tilde{u})^T F(u)\right] \\
\geq&\alpha_k\left[\theta(u) - \theta(\tilde{u}) + (u-\tilde{u})^T F(\tilde{u})\right] \\
\geq&\frac{1}{2}\left(\left\|u - u^{k+1}\right\|_{H_k}^2 - \left\|u - u^k\right\|_{H_k}^2\right). \qquad (4.5)
\end{aligned}$$

Summing (4.5) for $0 \leq k \leq t-1$ and using Lemma 3.3, we obtain

$$\begin{aligned}
&\sum_{k=0}^{t-1}\alpha_k\left[\theta(u) - \theta(\tilde{u}) + (u-\tilde{u})^T F(u)\right] \\
\geq&\frac{1}{2}\sum_{k=1}^{t-1}\left(\left\|u - u^{k+1}\right\|_{H_k}^2 - \left\|u - u^k\right\|_{H_k}^2\right) \\
\geq&\frac{1}{2}\left(\left\|u - u^t\right\|_{H_{t-1}}^2 - \left\|u - u^0\right\|_{H_0}^2\right) - \frac{1}{2}\sum_{k=1}^{t-1}\left(\left\|u^k - u\right\|_{H_k}^2 - \left\|u^k - u\right\|_{H_{k-1}}^2\right) \\
\geq&\frac{1}{2}\left(\left\|u - u^t\right\|_{H_{t-1}}^2 - \left\|u - u^0\right\|_{H_0}^2\right) - C_\phi C_H\left\|u - u^*\right\|^2 - C_\phi C_U\left\|u^0 - u^*\right\|_{H_0}^2. \qquad (4.6)
\end{aligned}$$

Let $\sum_{k=0}^{t-1}\alpha_k = \alpha$. It is easy to verify that $\alpha \geq \frac{c}{1+c}t$. By the definition of $\tilde{u}_t$ and the convexity of $\theta(u)$, we can get

$$\alpha\theta(\tilde{u}_t) \leq \sum_{k=0}^{t-1}\alpha_k\theta(\tilde{u}^k).$$

Consequently,

$$\alpha\left[\theta(u)-\theta(\tilde{u}_t)+(u-\tilde{u})^T F(u)\right] \geq \sum_{k=0}^{t-1}\alpha_k\left[\theta(u)-\theta(\tilde{u}^k)+(u-\tilde{u})^T F(u)\right]. \tag{4.7}$$

Combining (4.6) and (4.7) yields the bound

$$\begin{aligned}
&\theta(u)-\theta(\tilde{u}_t)+(u-\tilde{u}_t)^T F(u)\\
\geq&\frac{1}{2\alpha}\left(\left\|u-u^t\right\|^2_{H_{t-1}}-\left\|u-u^0\right\|^2_{H_0}-2C_\phi C_H\left\|u-u^*\right\|^2-2C_\phi C_U\left\|u^0-u^*\right\|^2_{H_0}\right)\\
\geq&\frac{(1+c)/2c}{t}\left(-\|u-u_0\|^2_{H_0}-2C_\phi C_U\left\|u_0-u^*\right\|^2_{H_0}-2C_\phi C_H\left\|u-u^*\right\|^2\right).
\end{aligned}$$

By (4.1), Algorithm 2.1 produce an $\mathcal{O}(1/t)$ approximate solution after t iterations. □

5. Numerical experiments

In the literature, the PDHG method in [24] as well as the other primal-dual methods, see, e.g., [4, 9, 12] have exhibited good numerical performance. The aim of this section is to verify the acceleration of the proposed parallel primal-dual algorithm over other primal-dual methods. Moreover different parameters and problem settings are chosen to prove the robustness of the new methods.

Algorithm 2.1 is applied to LASSO model and Steiner tree problem to show its numerical advantage over other primal-dual methods. We compares its performances with other four primal-dual methods mentioned in this paper, namely the primal-dual hybrid gradient method (PDHG) in [24], the first-order primal-dual method (CP) in [4], the relaxed proximal-point method (RPPA) in [12] and the adaptive primal-dual splitting method (APD) in [9]. The relaxation factor in RPPA is set to be 1.5 which is an estimation of optimal value because the true optimal value varies from case to case. As stated in the conclusion in [9], the backtracking APD method shows no consistently better performance over the non-backtracking version if the ideal step size parameters are known, we omit the backtracking step in the algorithmic comparisons. The APD method is thus simplified by taking the ideal parameter magnitude $\tau_0\sigma_0 = 1/\|A\|_2^2$.

All codes were written and implemented in Matlab 2014a, and all experiments were carried out on a computer with a 2.21 GHz AMD Athlon Dual Core processor and 2 GB of memory.

5.1. LASSO

In this subsection, the parallel primal-dual algorithm are applied to solve the LASSO model. The LASSO model was first mentioned in [21] to solve variable selection regression problems. Given a sample matrix A and the response b, the LASSO model learns the linear regression coefficient x by solving

$$\min_{x\in X}\ \beta\|x\|_1+\frac{1}{2}\|Ax-b\|_2^2 \tag{5.1}$$

for some scaling parameter β. By the definition of conjugate function, (5.1) can be reformulated as the following minmax problem:

$$\min_{x\in X}\max_{y\in Y}\ \beta\|x\|_1 + y^T Ax - \frac{1}{2}\|y+b\|_2^2. \tag{5.2}$$

The data for our test is generated in the same fashion as that in Section 11.1 in [3]. Each element of the sample matrix A is drawn from an $N(0,1)$ distribution. A true value x^{true} is generated with a certain number of non-zero entries, each sampled from an $N(0,1)$ distribution. The label b are computed as $b = Ax^{true} + v$ where v is a gaussian noise. The columns of sample matrices A are not normalized as in [3]. This modification is reasonable because in some applications, such as portfolio optimization, the magnitude of each column stands for the stock prize hence should not be normalized. The default setting of LASSO problem in this subsection use $\tau_0 = \sigma_0 = 1/\|A\|_2$(except for PPD method), sample size is 1000, feature number is 10000, $\beta = 0.1\beta_{max}$, number of non-zero element in x^{true} is 100. Here $\beta_{max} = \|A^T b\|_\infty$ is the minimum value of β that any value leads to a trifle solution zero.

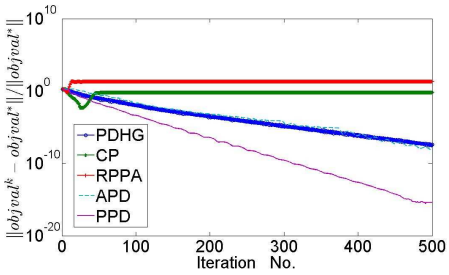
It is important to point out that $\tau_0\sigma_0$ is not known for PPD method. We draw a random number from interval $(0, 10/\|A\|_2)$ for both τ_0 and σ_0 in PPD for all the LASSO problem unless the initial value is specified otherwise.

To show the numerical results of LASSO problems in this subsection, we run all the methods for 500 iterations and use both figure and table to illustrate the error against iteration number and runtime. Here we define error in the following way: We run both CP method and RPPA method for 2000 iterations to ensure the objective value of both methods differs by less than 10^{-10}. We use this value as the approximate optimal value of the objective function value $objval^*$. Then the error is calculated by $\|objval^k - objval^*\|/\|objval^*\|$, where $objval^k$ is the current objective function value. This error is plotted against both iteration number and runtime in all figures and is listed in all tables in this subsection. In the tables, we use NA to show a result if the value of error is not reached within 500 iterations. The runtime is presented in the parenthesis with unit in second.

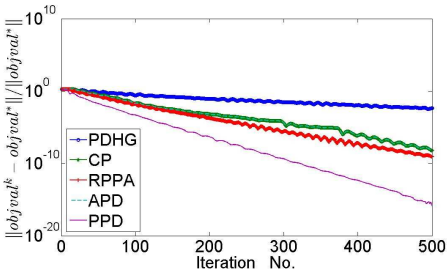
The PDHG, CP and RPPA methods all require prior knowledge of $\|A\|_2$ which is difficult to calculate. In the test, the function "normest" in Matlab is used to estimate $\|A\|_2$. It is the reason that the plot of the above methods does not start from 0 second.

Fig. 1 and Table 5.1 illustrate the performances of all five methods with equal parameters. However the value of $\tau_0\sigma_0$ takes various magnitude. Apparently CP and RPPA do not guarantee the convergence when the magnitude of $\tau_0\sigma_0$ exceed its upper bound $1/\|A\|_2^2$ (see also (1.4)). The same occurs to PDHG if the magnitude of $\tau_0\sigma_0$ further increases. On the other hand all methods suffer from the magnitude decrease of parameters except PPD and APD. This phenomenon was also mentioned and verified in [9].

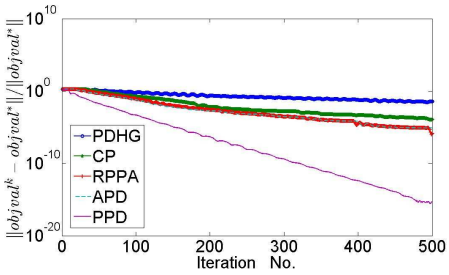
Fig. 2 and Table 5.1 show the performance of the tested methods with parameter τ and σ of optimal magnitude but different ratio. Two examples are chosen as the ratio between parameters are 0.25 and 4. The impact of different ratio on the algorithmic performance is obvious. This is possibly because the optimal choice of ratio is indeed close to $\tau/\sigma = 0.25$.



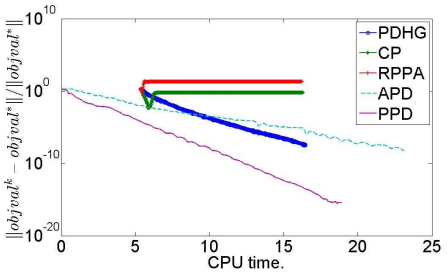
(a) $\tau_0 = \sigma_0 = 4/\|A\|_2$.



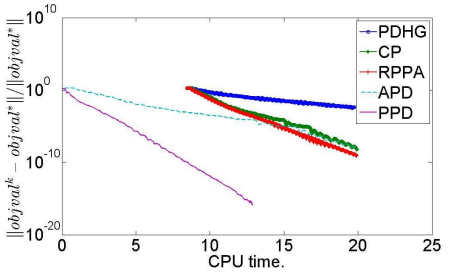
(b) $\tau_0 = \sigma_0 = 1/\|A\|_2$.



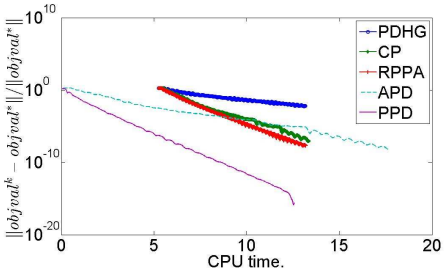
(c) $\tau_0 = \sigma_0 = 0.9/\|A\|_2$.



(d) $\tau_0 = \sigma_0 = 4/\|A\|_2$.



(e) $\tau_0 = \sigma_0 = 1/\|A\|_2$.



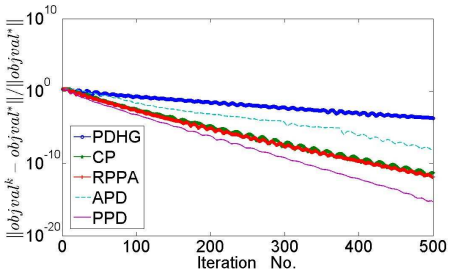
(f) $\tau_0 = \sigma_0 = 0.9/\|A\|_2$.

Figure 1: LASSO problem with different $\tau_0\sigma_0$ size.

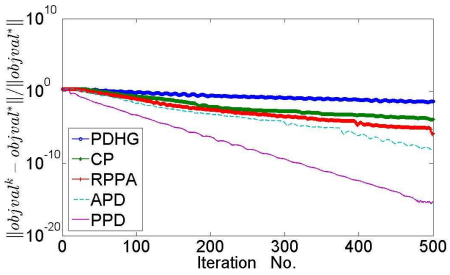
Table 1: Iteration counts and runtime for LASSO problem with different $\tau_0\sigma_0$ size.

	$\frac{objval^k - objval^*}{obj^*}$	PDHG	CP	RPPA	APD	PPD
$\tau_0 = \sigma_0 = \frac{4}{\|A\|_2}$	$< 10^{-1}$	48(6.36)	NA	NA	77(3.59)	31(1.16)
	$< 10^{-3}$	156(8.71)	NA	NA	178(7.93)	88(4.01)
	$< 10^{-5}$	311(12.0)	NA	NA	349(15.3)	150(6.21)
$\tau_0 = \sigma_0 = \frac{1}{\|A\|_2}$	$< 10^{-1}$	180(8.32)	77(6.50)	63(6.37)	77(2.95)	31(0.92)
	$< 10^{-3}$	NA	178(8.29)	160(8.10)	178(6.56)	89(2.58)
	$< 10^{-5}$	NA	349(11.3)	264(10.2)	349(12.8)	152(4.72)
$\tau_0 = \sigma_0 = \frac{0.9}{\|A\|_2}$	$< 10^{-1}$	200(9.98)	87(7.85)	71(7.54)	77(2.88)	31(0.84)
	$< 10^{-3}$	NA	210(10.0)	188(10.0)	178(6.35)	88(2.26)
	$< 10^{-5}$	NA	389(13.5)	306(12.2)	349(12.5)	151(3.89)

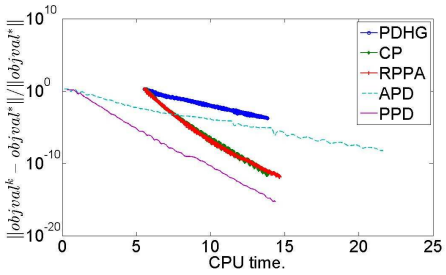
When the ratio between parameters are closer to optimal, PDHG, CP and RPPA methods are much more efficient. However when the ratio is far away from optimal setting, APD and PPD work better since the ratio is balanced during iteration progress. This result is not



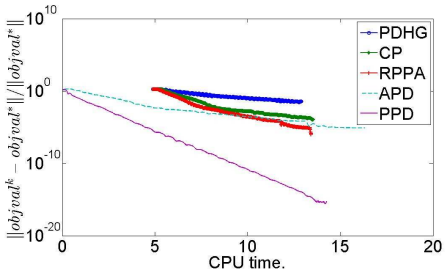
(a) $\tau_0 = 0.5/\|A\|_2, \sigma_0 = 2/\|A\|_2$.



(b) $\tau_0 = 2/\|A\|_2, \sigma_0 = 0.5/\|A\|_2$.



(c) $\tau_0 = 0.5/\|A\|_2, \sigma_0 = 2/\|A\|_2$.



(d) $\tau_0 = 2/\|A\|_2, \sigma_0 = 0.5/\|A\|_2$.

Figure 2: LASSO problem with different $\tau_0\sigma_0$ ratio.

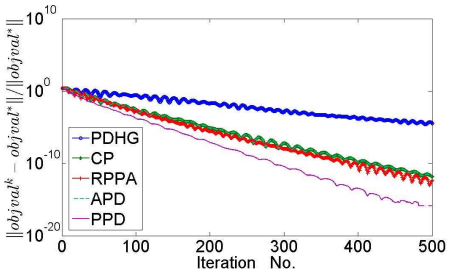
Table 2: Iteration counts and runtime for LASSO problem with different $\tau_0\sigma_0$ ratio.

	$\frac{objval^k - objval^*}{obj^*}$	PDHG	CP	RPPA	APD	PPD
$\tau_0 = \frac{1}{2\|A\|_2}, \sigma_0 = \frac{2}{\|A\|_2}$	$< 10^{-1}$	120(7.58)	48(6.28)	41(6.21)	77(2.80)	31(1.71)
	$< 10^{-3}$	369(11.8)	119(7.41)	111(7.36)	178(6.25)	89(3.18)
	$< 10^{-5}$	NA	205(8.84)	183(8.54)	349(11.9)	158(4.89)
$\tau_0 = \frac{2}{\|A\|_2}, \sigma_0 = \frac{1}{2\|A\|_2}$	$< 10^{-1}$	312(9.90)	128(6.91)	96(6.40)	77(2.79)	31(0.908)
	$< 10^{-3}$	NA	324(10.4)	233(8.79)	178(6.70)	88(2.71)
	$< 10^{-5}$	NA	NA	459(12.8)	349(12.5)	151(4.46)

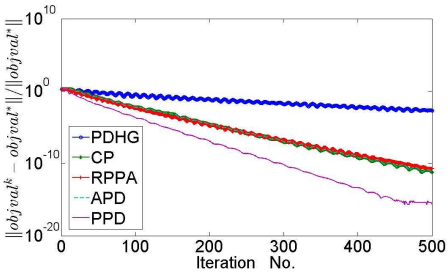
surprising since PDHG, CP and RPPA have been proved to be very sensitive to parameters (see [9]).

Fig. 3 and Table 3 illustrate the performances of all five methods with different problem scales, i.e., all other parameters take the default setting except the scale of A. We can see that our method is the fastest for all problem scales especially the large ones. The advantage of PPD method become more and more obvious as the problem scale increases, since the computation of $\|A\|_2$ is getting too expensive.

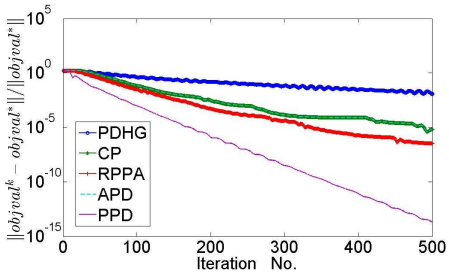
Fig. 4 and Table 4 demonstrate the performance of the test methods with different type of data input. As the ratio between feature number and sample size grows, PPD method and AP method remains the best ones but the other three methods become more and more efficient. The performance of methods are similar to that Fig. 2 and Table 5.1. This could be explained as different data leads to different optimal parameter $\sigma_0\tau_0$ ratio. When the ratio between feature number and sample size grows, the optimal $\tau_0\sigma_0$ ratio approaches 1 in this case.



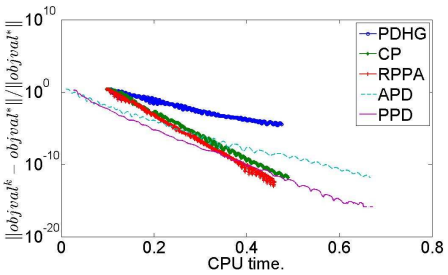
(a) dimension of A is 200×2000.



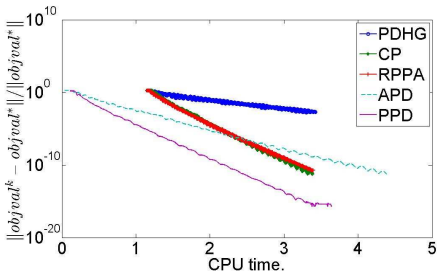
(b) dimension of A is 500×5000.



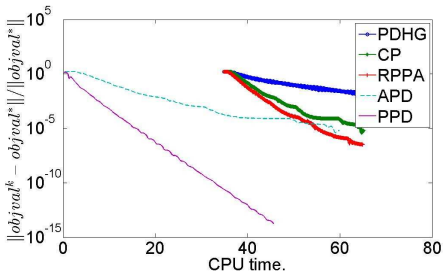
(c) dimension of A is 2000×20000.



(d) dimension of A is 200×2000.



(e) dimension of A is 500×5000.



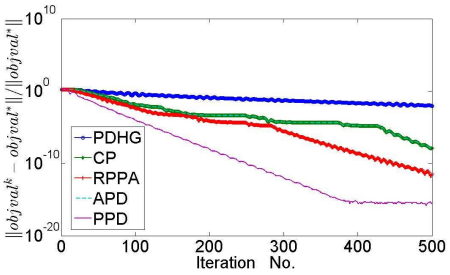
(f) dimension of A is 2000×20000.

Figure 3: LASSO problem with different size.

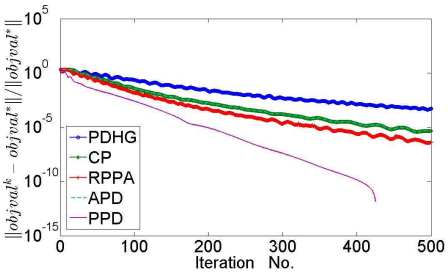
Table 3: Iteration counts and runtime for LASSO problem with different size.

	$\frac{objval^k - objval^*}{obj^*}$	PDHG	CP	RPPA	APD	PPD
$A_{200\times2000}$	$< 10^{-1}$	118(0.173)	38(0.113)	26(0.105)	38(0.0525)	28(0.0503)
	$< 10^{-3}$	308(0.309)	105(0.158)	106(0.164)	105(0.139)	80(0.107)
	$< 10^{-5}$	NA	197(0.221)	178(0.218)	197(0.259)	137(0.179)
$A_{500\times5000}$	$< 10^{-1}$	128(1.76)	54(1.39)	47(1.36)	54(0.469)	29(0.332)
	$< 10^{-3}$	458(2.66)	137(1.76)	125(1.72)	137(1.20)	81(0.685)
	$< 10^{-5}$	NA	215(2.12)	210(2.09)	215(1.89)	140(1.12)
$A_{2000\times20000}$	$< 10^{-1}$	238(49.1)	90(41.5)	74(39.3)	90(10.8)	33(3.21)
	$< 10^{-3}$	NA	245(49.9)	185(46.0)	245(29.4)	99(9.20)
	$< 10^{-5}$	NA	489(64.5)	328(54.6)	489(58.5)	172(15.9)

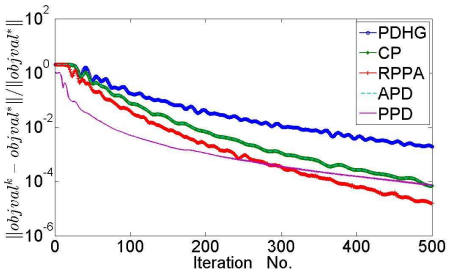
All the above experiments proved that our PPD method works the best for various problem settings. To further show the robustness of our method, we conduct the LASSO experiment for different balancing parameter β and sparsity level of x^{true}. The result are



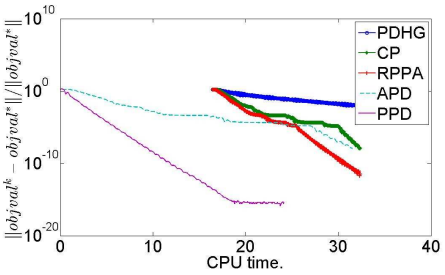
(a) dimension of A is 2000×10000.



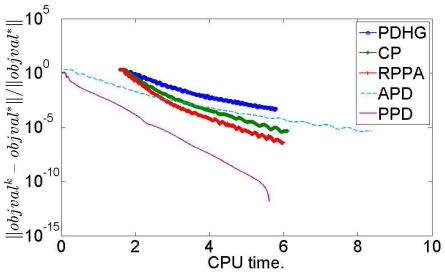
(b) dimension of A is 500×10000.



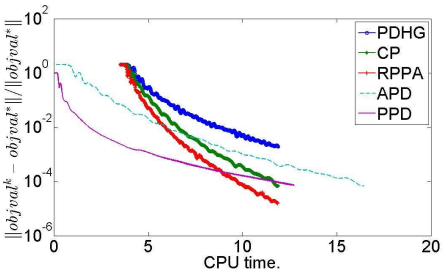
(c) dimension of A is 500×20000.



(d) dimension of A is 2000×10000.



(e) dimension of A is 500×10000.



(f) dimension of A is 500×20000.

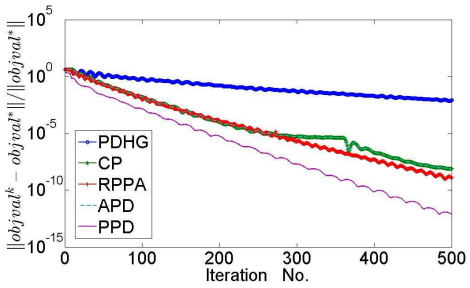
Figure 4: LASSO problem with different sample size to feature number ratio.

Table 4: Iteration counts and runtime for LASSO problem with different sample size to feature number ratio.

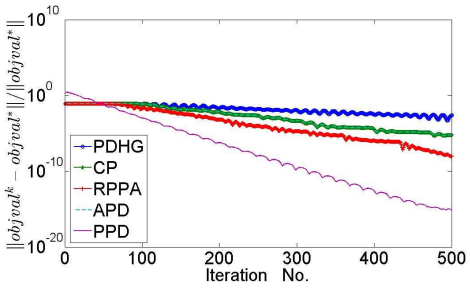
	$\frac{objval^k - objval^*}{obj^*}$	PDHG	CP	RPPA	APD	PPD
$A_{2000\times10000}$	$< 10^{-1}$	192(22.4)	72(18.6)	54(18.1)	72(4.55)	30(1.46)
	$< 10^{-3}$	NA	162(21.6)	123(20.3)	162(10.4)	75(3.57)
	$< 10^{-5}$	NA	430(30.1)	287(25.6)	430(27.2)	124(5.90)
$A_{500\times10000}$	$< 10^{-1}$	133(2.69)	80(2.29)	66(2.16)	80(1.38)	28(0.372)
	$< 10^{-3}$	432(5.16)	219(3.58)	176(3.10)	219(3.69)	118(1.58)
	$< 10^{-5}$	NA	439(5.60)	436(4.69)	439(7.35)	197(2.61)
$A_{500\times20000}$	$< 10^{-1}$	131(5.74)	95(5.14)	74(4.74)	95(3.11)	21(0.535)
	$< 10^{-3}$	458(2.66)	305(8.65)	239(7.52)	305(10.0)	206(5.26)
	$< 10^{-5}$	NA	NA	NA	NA	NA

shown in the following figures. PPD method is robust to various problem settings.

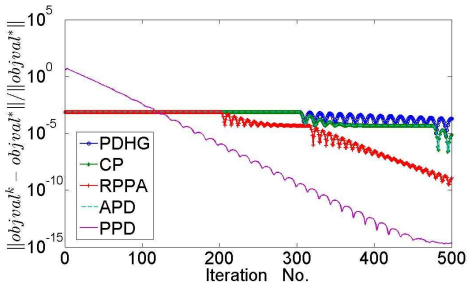
To show the important role of tuning parameters, we use PPDn to denote the PPD method without tuning parameters. We compare the APD, PPDn and PPD methods for



(a) $\beta = 0.05\beta_{max}$.



(b) $\beta = 0.5\beta_{max}$.



(c) $\beta = 0.9\beta_{max}$.

Figure 5: LASSO problem with different balancing parameter β.

$A_{500\times5000}$, the initial proximal parameter value are $\tau_0 = 0.5/\|A\|_2, \sigma_0 = 2/\|A\|_2$ and $\tau_0 = 2/\|A\|_2, \sigma_0 = 0.5/\|A\|_2$ respectively. Fig. 7 illustrate the trend of proximal parameter τ^k and σ^k. Fig. 8 shows the convergence of the objective function value and x^k for APD, PPDn and PPD method. We observe that x^k converges to the solution x^{true} as $objval^k$ converge to $objval^*$, and their speed is positively related. The role of parameter tuning is significant as PPD method is much faster than PPDn method in Fig. 8. In addition our quantitative way of tuning parameters performs better than the empirical way of APD method from the results of all the experiments above.

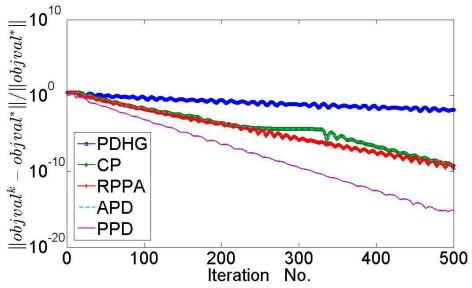
5.2. Steiner tree

The primal-dual methods are further tested on Steiner tree problem. The Steiner tree problem requires to find the shortest interconnection for a given set of objects. A typical example is

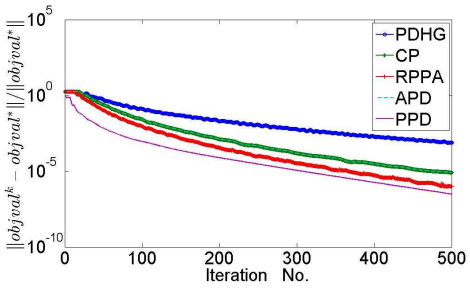
$$\min_{x_j\in X_j} \begin{cases} c\|x_1-b_1\|_2 + \|x_1-b_2\|_2 + \|x_2-b_3\|_2 + \|x_3-b_4\|_2 \\ \quad +\|x_3-b_5\|_2 + \|x_1-x_2\|_2 + \|x_2-x_3\|_2. \end{cases} \tag{5.3}$$

Apparently (5.3) is equivalent to

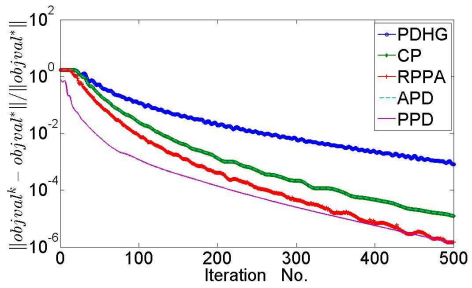
$$\min_{x_j\in X_j}\max_{y_j\in B_2} \begin{cases} y_1^T(x_1-b_1) + y_2^T(x_1-b_2) + y_3^T(x_2-b_3) + y_4^T(x_3-b_4) \\ \quad +y_5^T(x_3-b_5) + y_6^T(x_1-x_2) + y_7^T(x_2-x_3), \end{cases} \tag{5.4}$$



(a) $\frac{1}{200}$ non-zero elements in x^{true}.

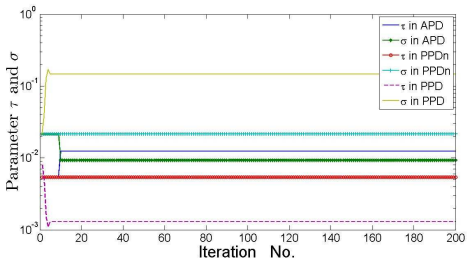


(b) $\frac{1}{10}$ non-zero elements in x^{true}.

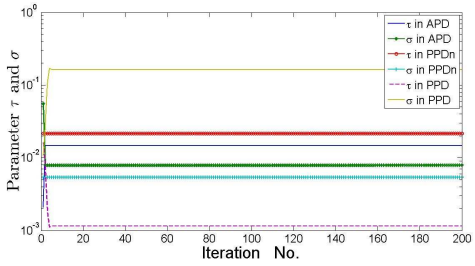


(c) $\frac{1}{2}$ non-zero elements in x^{true}.

Figure 6: LASSO problem with different true sparsity.



(a) $\tau_0 = 0.5/\|A\|_2, \sigma_0 = 2/\|A\|_2$.



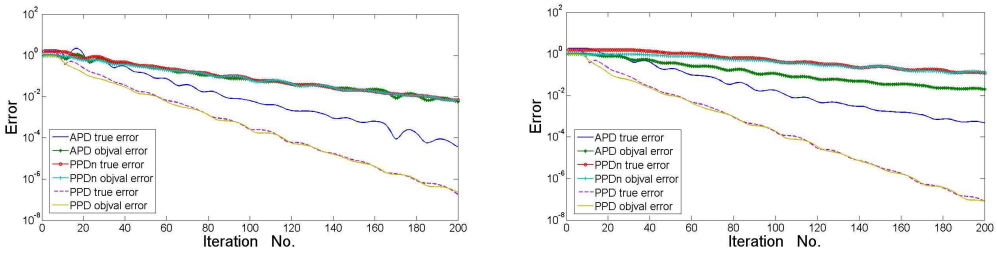
(b) $\tau_0 = 2/\|A\|_2, \sigma_0 = 0.5/\|A\|_2$.

Figure 7: Trend of parameters in LASSO problem.

which is indeed a saddle-point problem of form (1.1).

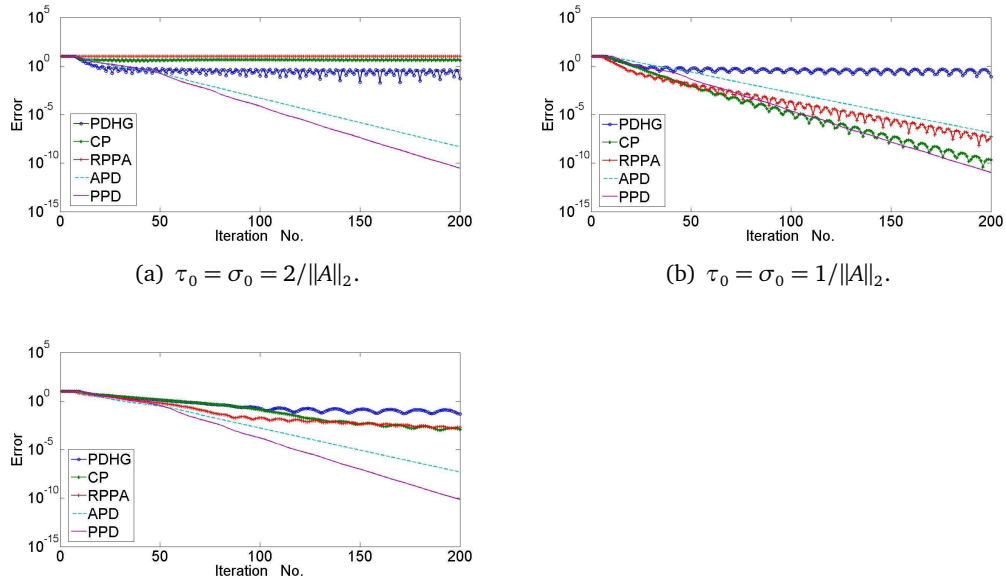
The tested problem is from example 1 of [22]. The coordinates of 10 regular points and their tree topology can be found in Tables 7.1 and 7.2 in [22]. The approximate optimal location was obtained by running the CP method and PPD method until both the primal and the dual variable are differed by less than 10^{-10} in all elements. This approximation is used as x^* The error in our experiment is calculated by $\|x^k - x^*\|$ and plotted against the iteration number.

Fig. 9 illustrates the performances of primal-dual methods when τ_0 and σ_0 take equal value but different magnitude. Fig. 10 shows the performance of all methods when τ_0 and



(a) $\tau_0 = 0.5/\|A\|_2, \sigma_0 = 2/\|A\|_2$. (b) $\tau_0 = 2/\|A\|_2, \sigma_0 = 0.5/\|A\|_2$.

Figure 8: Convergence of objecvtive value and x^k in LASSO problem.



(a) $\tau_0 = \sigma_0 = 2/\|A\|_2$. (b) $\tau_0 = \sigma_0 = 1/\|A\|_2$.

(c) $\tau_0 = \sigma_0 = 0.5/\|A\|_2$.

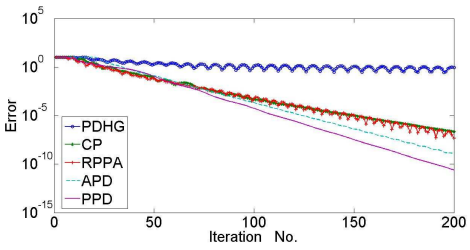
Figure 9: Steiner tree with different $\tau_0\sigma_0$ size.

Table 5: Iteration counts and runtime for Steiner tree problem.

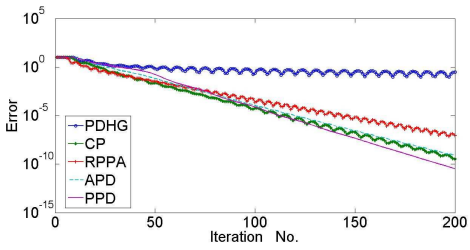
error	PDHG	CP	RPPA	APD	PPD
$< 10^{-5}$	NA	98(0.0426)	120(0.0538)	125(0.0411)	97(0.0302)

σ_0 take optimal magnitude but different ratio. In order to present the numerical result clearly, the error is bounded by the ceiling of 10^1 hence a greater error is not shown in figures.

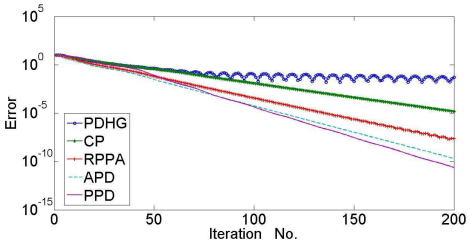
The trend of proximal parameters τ^k and σ^k is similar with that in last subsection. We only show one of the results for $\tau = 1/\|A\|_2, \sigma = 1/\|A\|_2$ in Fig. 11. The iteration count and runtime for $\tau = 1/\|A\|_2, \sigma = 1/\|A\|_2$ is listed in Table 5.



(a) $\tau_0 = 4/\|A\|_2, \sigma_0 = 0.25/\|A\|_2$



(b) $\tau_0 = 2/\|A\|_2, \sigma_0 = 0.5/\|A\|_2$



(c) $\tau_0 = 0.5/\|A\|_2, \sigma_0 = 2/\|A\|_2$

Figure 10: Steiner tree with different $\tau_0\sigma_0$ ratio.

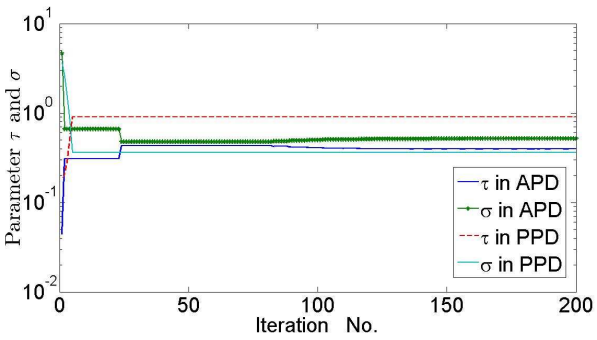


Figure 11: Trend of parameters in Stein tree problem.

The results are consistent with that in the previous subsection. PPD method is fastest and most robust among all five methods. Take note that the performance of PPD and APD can be matched by other three methods if the parameters are close to optimal value.

In general, APD and PPD are more robust than the other three methods. The average convergence speed is ranked as PPD > APD > RPPA > CP > PDHG with various problem settings. Two factors contribute the most for the high speed of PPD method. First, it avoid the calculation of $\|A\|_2$ which is compulsory for all primal-dual methods expect APD and PPD methods. Second, it tune the parameters dynamically and quantitatively. The parallel property of PPD method makes it potential to be even more efficient, especially in multi-core environment. It remains a topic to be further studied.

6. Conclusions

In this paper, the parallel primal-dual (PPD) algorithm are proposed to solve the saddle-point problem. This method update the primal and dual variable at the same time to speed up the convergence. Unlike most other primal-dual methods, it does not require the estimate of $\|A\|$ to determine the parameters. In addition the parameters and step size are adjusted during each iteration.

In the numerical experiments, the new method is compared with primal-dual hybrid gradient method (PDHG), chambolle and Pock's primal-dual method (CP), relaxed proximal point algorithm (RPPA) and adaptive primal-dual method (APD). Our methods perform consistently the best in all applications. One advantage of PPD method over the other methods is that it does not calculate the norm $\|A\|$. This is particularly important in large-scale applications where the norm of the linear operators is hard to estimate. Moreover, the automatic tuning of parameters and step size also make the parallel primal-dual method not only faster but also more robust than other primal-dual methods.

It should be mentioned that the calculation to tune the parameters and step size can be non-negligible if the subproblems are relatively easy to solve. In the future it is possible to reduce the frequency of tuning yet keep the method convergent.

Acknowledgments The author would like to express sincere gratitude to Professor Bingsheng He, Dr. Caihua Chen, Dr. Yuan Shen and Dr. Wenxing Zhang for their valuable advices. The author thanks the editor and referees for constructive comments that led to this improved version of the paper. This research is supported by National Natural Science Foundation of China (Nos. 71201080, 71571096), Social Science Foundation of Jiangsu Province (No. 14GLC001), Fundamental Research Funds for the Central Universities (No. 020314380016).

References

[1] K. J. Arrow, L. Hurwicz and H. Uzawa, Studies in Linear and Non-Linear Programming, Stanford University Press, 1958.

[2] S. Bonettini and V. Ruggiero, *On the convergence of primal-dual hybrid gradient algorithms for total variation image restoration*, J. Math. Imaging Vision, 44 (2012), pp. 236–253.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, *Distributed optimization and statistical learning via admm*, Foundations and Trends in Machine Learning, 3 (2010), pp. 1–122.

[4] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.

[5] J. Douglas and H. H. Rachford, *On the numerical solution of the heat conduction problem in 2 and 3 space variables*, Trans. Amer. Math. Soc., 82 (1956), pp. 421–439.

[6] E. Esser, X. Zhang and T. F. Chan, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM J. Imaging Sci., 3 (2010), pp. 1015–1046.

[7] F. Facchinei and J. S. Pang, *Finite-dimensional variational inequalities and complementarity problems*, Springer Series in Operations Research, Springer-Verlag, 2003, pp. 625–1234.

[8] 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.
[9] T. Goldstein, M. Li, X. Yuan, E. Esser and R. Baraniuk, Adaptive Primal-Dual Hybrid Gradient Methods for Saddle-Point Problems, arXiv:1305.0546v2 [math.NA], 2013.
[10] E. G. Gol'shteĭn and N. V. Tret'yakov, *Modified Lagrangians in convex programming and their generalizations*, Math. Programming Stud., 1979, pp. 86–97.
[11] B. He, Y. You, and X. Yuan, *On the convergence of primal-dual hybrid gradient algorithm*, SIAM J. Imaging Sci., 7 (2014), pp. 2526–2537.
[12] B. He and X. Yuan, *Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective*, SIAM J. Imaging Sci., 5 (2012), pp. 119–149.
[13] B. He and X. Yuan, *On the $o(1/n)$ convergence rate of the douglas-rachford alternating direction method*, SIAM J. Numer. Anal., 50 (2012), pp. 700–709.
[14] B. He, X. Yuan, and J. J. Z. Zhang, *Comparison of two kinds of prediction-correction methods for monotone variational inequalities*, Comput. Optim. Appl., 27 (2004), pp. 247–267.
[15] G. M. Korpelevič, *An extragradient method for finding saddle points and for other problems*, Èkonom. i Mat. Metody, 12 (1976), pp. 747–756.
[16] P. L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979.
[17] B. Martine, *Regularization d'inequations variationelles par approximations successives*, Revue Francaise d'Informatique et de Recherche Opérationelle, 4 (1970), pp. 154–159.
[18] D. O'Connor and L. Vandenberghe, *Primal-dual decomposition by operator splitting and applications to image deblurring*, SIAM J. Imaging Sci., 7 (2014), pp. 1724–1754.
[19] T. Pock and A. Chambolle, *Diagonal preconditioning for first order primal-dual algorithms in convex optimization*, IEEE I. Conf. Comp. Vis., (2011), pp. 1762–1769.
[20] L. D. Popov, *A modification of the Arrow-Hurwitz method of search for saddle points*, Mat. Zametki, 28 (1980), pp. 777–784, 803.
[21] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B, 58(1) (1996), pp. 267–288.
[22] G. Xue and Y. Ye, *An efficient algorithm for minimizing a sum of Euclidean norms with applications*, SIAM J. Optim., 7 (1997), pp. 1017–1036.
[23] X. Zhang, M. Burger and S. Osher, *A unified primal-dual algorithm framework based on Bregman iteration*, J. Sci. Comput., 46 (2011), pp. 20–46.
[24] M. Zhu and T. F. Chan, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration*, UCLA CAM Report, 2008.