An Implementable Proximal Point Algorithmic Framework for Nuclear Norm Minimization

Defeng Sun

Department of Mathematics

National University of Singapore

July 3, 2009

Joint work with Yong-Jin Liu and Kim-Chuan Toh

Let $\Re^{n_1 \times n_2}$ be the linear space of all $n_1 \times n_2$ real matrices, not necessarily square or symmetric.

- Trace product:

$$\langle X, Y \rangle = \sum_{i,j} X_{ij} Y_{ji} = \mathrm{Trace}(X^T Y).$$

- The Frobenius norm: $\|X\| = \sqrt{\langle X, X \rangle}$.
- The nuclear norm: $\|X\|_* = \sum_{i=1}^{\min\{n_1,n_2\}} \sigma_i$, where $\sigma_i, 1 \le i \le \min\{n_1, n_2\}$ are the singular values of X.

Given a linear mapping $\mathcal{A} : \Re^{n_1 \times n_2} \rightarrow \Re^m$ and $b \in \Re^m$.

Consider the following nuclear norm minimization problem with linear equality and second order cone (SOC) constraints:

$$
\begin{array}{lll}
\text{(P)} & \min & \|X\|_* \\
& \text{s.t.} & \mathcal{A}(X) \in b + \mathcal{Q},
\end{array}
$$

where $\mathcal{Q} := \{0\}^{m_1} \times \mathcal{K}^{m_2}$ and $\mathcal{K}^{m_2}$ denotes the SOC of dimension m_2. Here, $m = m_1 + m_2$.

The problem (P) is the convex relaxation[a] of the rank minimization problem with/without noise arising in many fields of engineering and science.

The rank minimization problem is:

$$\begin{aligned} \min \quad & \mathrm{Rank}(X) \\ \text{s.t.} \quad & \mathcal{A}(X) = b. \end{aligned}$$

[a] Recht, B., Fazel, M. and Parrilo, P.A., *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, preprint.

A special case is the matrix completion problem, which has the form:

$$\begin{array}{ll} \min & \mathrm{rank}(X) \\[1em] \text{s.t.} & X_{ij} = M_{ij}, \ (i,j) \in \Omega, \end{array}$$

where M is the unknown matrix with m available sampled entries, and Ω is a set of index pairs (i,j) of cardinality m.

• In this case, $\mathcal{A}(X) = X_\Omega$, where $X_\Omega \in \Re^{|\Omega|}$ is the vector consisting of elements selected from X whose indices are in Ω.

• The rank minimization problem is NP-hard in general and computationally hard to solve.

Question: How can we get the matrix with the minimum rank of the rank minimization problem?

• A novel and tractable approach[a] is to consider solving its convex relaxation problem:

$$\text{(CRP)} \quad \min \ \|X\|_* \quad \text{s.t.} \ \|b - \mathcal{A}(X)\| \leq \delta,$$

where $\delta \geq 0$ estimates the uncertainty about the observation b.

[a] Candès, E.J. and Recht, B., *Exact matrix completion via convex optimization*, preprint.

Some works[a,b] consider solving the Lagrangian version of (CRP):

$$\min_{X \in \Re^{n_1 \times n_2}} \ \frac{1}{2}\|\mathcal{A}(X) - b\|^2 + \mu \|X\|_*.$$

- Although (CRP) and its Lagrangian version are equivalent whenever δ and μ obey some special relationship, it is generally hard to find the relationship.

---

[a] Ma, S.Q., Goldfarb, D. and Chen, L.F., *Fixed point and Bregman iterative methods for matrix rank minimization*, preprint, 2008.

[b] Toh, K.C. and Yun, S.W., *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, preprint, 2009.

One obvious approach for (P) is to consider solving its semidefinite programming (SDP) reformulation:

$$\begin{array}{ll} \min & (\mathrm{Tr}(W_1) + \mathrm{Tr}(W_2))/2 \\ \text{s.t.} & \mathcal{A}(X) \in b + \mathcal{Q} \\ & \begin{bmatrix} W_1 & X \\ X^T & W_2 \end{bmatrix} \succeq 0. \end{array}$$

But, the problem is:

The above approach makes the SDP more difficult since it greatly enlarges the dimension of problem.

This suggests that other methods are needed to solve (P) directly.

Related Algorithms:

- The projected subgradient method [Recht, Fazel and Parrilo]: Convergence?
- The singular value thresholding (SVT) algorithm [Cai, Candès and Shen]: It is in the class of penalty methods so that it requires large parameters.
- An accelerated proximal gradient (APG) algorithm [Toh and Yun]: It is applied to the Lagrangian version of (CRP).
- Fixed point method and Bregman iterative method [Ma, Goldfarb and Chen]: They are also applied to the Lagrangian version of (CRP).

Our work: Three classes of proximal point algorithms (PPA) in the primal, dual and primal-dual forms.

- The Lagrangian function of (P) in the extended form:

$$l(X, y) := \begin{cases} \|X\|_* + \langle y, b - \mathcal{A}(X) \rangle & \text{if } \; y \in \mathcal{Q}^*, \\ -\infty & \text{if } \; y \notin \mathcal{Q}^*. \end{cases}$$

where $\mathcal{Q}^*$ is the dual cone of $\mathcal{Q}$.

- The dual problem of (P):

$$\boxed{\text{(D)} \quad \max_{y \in \Re^m} \{ g(y) := \inf_{X \in \Re^{n_1 \times n_2}} l(X, y) \}.}$$

- The primal form.

The primal PPA is the application of the general proximal point method to (P). That is,

$$X^{k+1} \approx \arg \min_{X \in \Re^{n_1 \times n_2}} \{f(X) + \frac{1}{2\lambda_k}\|X - X^k\|^2\},$$

where f is the convex function defined by

$$f(X) = \sup_{y \in \Re^m} l(X, y).$$

- How to approximately get X^{k+1}?

For given $\lambda > 0$, let $\Theta_\lambda(y; X)$ be defined by

$$\boxed{\Theta_\lambda(y; X) = \langle b, y \rangle - \frac{1}{2\lambda}\Big(\|\mathcal{P}_\lambda[X + \lambda \mathcal{A}^*(y)]\|^2 - \|X\|^2\Big),}$$

where $(y, X) \in \Re^m \times \Re^{n_1 \times n_2}$ and for any $\lambda > 0$ and $X \in \Re^{n_1 \times n_2}$, $\mathcal{P}_\lambda(X)$ is the unique optimal solution to

$$\min_{Y \in \Re^{n_1 \times n_2}} \lambda \|Y\|_* + \frac{1}{2}\|Y - X\|^2.$$

Let X be of rank r and have the following singular value decomposition:

$$X = U\Sigma V^T, \quad \Sigma = \mathrm{diag}(\{\sigma_i\}_{1\le i\le r}),$$

where $U \in \Re^{n_1\times r}$ and $V \in \Re^{n_2\times r}$ have orthonormal columns, respectively, and the positive singular values σ_i are arranged in descending order. Then,

$$\mathcal{P}_\lambda(X) = U\mathrm{diag}(\max\{\sigma_i - \lambda, 0\})V^T.$$

• The function $\Theta_\lambda(y; X)$ is continuously differentiable. For any given $X \in \Re^{n_1 \times n_2}$, we have

$$\boxed{\nabla_y \Theta_\lambda(y; X) = b - \mathcal{A}\mathcal{P}_\lambda(X + \lambda \mathcal{A}^*(y)).}$$

• For any given $X \in \Re^{n_1 \times n_2}$, $\nabla_y \Theta_\lambda(\cdot, X)$ is globally Lipschitz continuous.

- The primal PPA

For given $X^0 \in \Re^{n_1 \times n_2}$, $\lambda_0 > 0$, and $\rho > 1$, the primal PPA for solving problem (P) generates sequences $\{y^k\} \subset \Re^m$ and $\{X^k\} \subset \Re^{n_1 \times n_2}$ for $k = 0, 1, 2, \ldots$

$$\left\{ \begin{array}{l} y^{k+1} \approx \arg\max_{y \in \mathcal{Q}^*} \Theta_{\lambda_k}(y; X^k), \\ X^{k+1} = \mathcal{P}_{\lambda_k}[X^k + \lambda_k \mathcal{A}^*(y^{k+1})], \\ \lambda_{k+1} = \rho \lambda_k \text{ or } \lambda_{k+1} = \lambda_k. \end{array} \right.$$

- Connection to the SVT algorithm

For a special case of (P) without the SOC constraints, from the primal PPA, if $X^0 = 0$ and $\lambda_0 = \lambda^{-1} > 0$, then X^1 is exactly the solution to the following regularized problem:

$$\begin{array}{ll} \min & \lambda\|X\|_* + \dfrac{1}{2}\|X\|^2 \\ \text{s.t.} & \mathcal{A}(X) = b\,. \end{array}$$

The SVT algorithm by [Cai, Candès and Shen] solves it by applying the gradient method to its dual problem and it is just one-step of the primal PPA.

- The dual form.

The dual PPA is the application of the general proximal point method to the dual problem (D), instead of (P).

The sequence $\{y^k\} \subset \mathcal{Q}^*$ is generated by the dual PPA as follows:

$$\boxed{y^{k+1} \approx \operatorname*{argmax}_{y \in \Re^m} \{g(y) - \frac{1}{2\lambda_k}\|y - y^k\|^2\}.}$$

Recall that $g(y) = \inf_{X \in \Re^{n_1 \times n_2}} l(X, y)$.

- How can we approximately obtain y^{k+1}?

For given $\lambda > 0$, let $\Psi_\lambda(X; y)$ be defined by

$$\boxed{\Psi_\lambda(X; y) = \|X\|_* + \frac{1}{2\lambda}\left[\|\Pi_{\mathcal{Q}^*}[y + \lambda(b - \mathcal{A}(X))]\|^2 - \|y\|^2\right],}$$

where $(X, y) \in \Re^{n_1 \times n_2} \times \Re^m$ and for any $\lambda > 0$ and $x \in \Re^m$, $\Pi_{\mathcal{Q}^*}(x)$ is the unique optimal solution to

$$\min_{z \in \mathcal{Q}^*} \frac{1}{2}\|z - x\|^2.$$

For any $x = (x^1; x^2) \in \Re^{m_1} \times \Re^{m_2}$, one has

$$\boxed{\Pi_{\mathcal{Q}^*}(x) = (x^1; \Pi_{\mathcal{K}^{m_2}}(x^2)).}$$

where $\Pi_{\mathcal{K}^{m_2}}(x^2)$ is given by

$$\Pi_{\mathcal{K}^{m_2}}(x^2) = \begin{cases} \frac{1}{2}\Big(1 + \frac{x_0^2}{\|\bar{x}^2\|}\Big)(\|\bar{x}^2\|; \bar{x}^2) & \text{if} \;\; |x_0^2| < \|\bar{x}^2\|, \\ (x_0^2; \bar{x}^2) & \text{if} \;\; \|\bar{x}^2\| \leq x_0^2, \\ 0 & \text{if} \;\; \|\bar{x}^2\| \leq -x_0^2, \end{cases}$$

here, $x^2 = (x_0^2, \bar{x}^2) \in \Re \times \Re^{m_2 - 1}$.

- The dual PPA

For given $y^0 \in \Re^m$, $\lambda_0 > 0$, and $\rho > 1$, the dual PPA for solving problem (P) and (D) generates sequences $\{y^k\} \subset \Re^m$ and $\{X^k\} \subset \Re^{n_1 \times n_2}$ for $k = 0, 1, 2, \ldots$

$$\left\{ \begin{array}{l} X^{k+1} \approx \arg \min\limits_{X \in \Re^{n_1 \times n_2}} \Psi_{\lambda_k}(X; y^k), \\ y^{k+1} = \Pi_{\mathcal{Q}^*}[y^k + \lambda_k(b - \mathcal{A}(X^{k+1}))], \\ \lambda_{k+1} = \rho\lambda_k \text{ or } \lambda_{k+1} = \lambda_k. \end{array} \right.$$

For the special case of (P) with equality constraints only, then with $y^0 = 0$,

$$y^1 = \lambda_0(b - \mathcal{A}(X^1)),$$

where X^1 (approximately) solves the following penalized problem:

$$\min_{X \in \Re^{n_1 \times n_2}} \left\{ \frac{1}{2} \|\mathcal{A}(X) - b\|^2 + \lambda_0^{-1} \|X\|_* \right\}.$$

Again, this says that y^1 is the result for one-step of the dual PPA.

- Connection to the Bregman iterative method

The Bregman iterative method by [Ma, Goldfarb and Chen] for solving the special case of (P) without noise can be described as:

$$\begin{cases} b^{k+1} = b^k + (b - \mathcal{A}(X^k)) \\ X^{k+1} = \arg \min_{X \in \Re^{n_1 \times n_2}} \{\frac{1}{2}\|\mathcal{A}(X) - b^{k+1}\|^2 + \mu \|X\|_*\} \end{cases}$$

for some fixed $\mu > 0$. By noting that $b^{k+1} = \mu y^{k+1}$ with $\mu = {\lambda_k}^{-1}$, we know that the Bregman iterative method is actually a special case of the dual PPA with $\lambda_k \equiv \mu^{-1}$.

- The primal-dual form.

The primal-dual PPA is the application of the general proximal point method to the monotone operator corresponding to the convex-concave Lagrangian function l, i.e.,

$$
\begin{aligned}
(X^{k+1}, y^{k+1}) \approx \min_{X \in \Re^{n_1 \times n_2}} \max_{y \in \Re^m} \{ l(X, y) + \\
+ \frac{1}{2\lambda_k} \|X - X^k\|^2 - \frac{1}{2\lambda_k} \|y - y^k\|^2 \}.
\end{aligned}
$$

- How to get an approximation solution (X^{k+1}, y^{k+1})?

- The primal-dual PPA-I

For given $(X^0, y^0) \in \Re^{n_1\times n_2} \times \Re^m$, $\lambda_0 > 0$, and $\rho > 1$, the primal-dual PPA-I for solving problem (P) and (D) generates sequences $\{(X^k, y^k)\} \subset \Re^{n_1\times n_2} \times \Re^m$ for $k = 0, 1, 2, \ldots$

$$\left\{\begin{array}{l} y^{k+1} \approx \arg\max\limits_{y\in\mathcal{Q}^*}\{\Theta_{\lambda_k}(y; X^k) + \frac{1}{2\lambda_k}\|y - y^k\|^2\}, \\ X^{k+1} = \mathcal{P}_{\lambda_k}[X^k + \lambda_k \mathcal{A}^*(y^{k+1})], \\ \lambda_{k+1} = \rho\lambda_k \text{ or } \lambda_{k+1} = \lambda_k. \end{array}\right.$$

- The primal-dual PPA-II

For given $(X^0, y^0) \in \Re^{n_1 \times n_2} \times \Re^m$, $\lambda_0 > 0$, and $\rho > 1$, the primal-dual PPA-II for solving problem (P) and (D) generates sequences $\{(X^k, y^k)\} \subset \Re^{n_1 \times n_2} \times \Re^m$ for $k = 0, 1, 2, \ldots$

$$\left\{ \begin{array}{l} X^{k+1} \approx \arg \min\limits_{X \in \Re^{n_1 \times n_2}} \{ \Psi_{\lambda_k}(X; y^k) + \frac{1}{2\lambda_k} \|X - X^k\|^2 \}, \\ y^{k+1} = \Pi_{\mathcal{Q}^*}[y^k + \lambda_k (b - \mathcal{A}(X^{k+1}))], \\ \lambda_{k+1} = \rho \lambda_k \text{ or } \lambda_{k+1} = \lambda_k. \end{array} \right.$$

- First-order methods for the inner problems

We consider the gradient projection method (GPM) to solve the inner problems of the primal PPA and the primal-dual PPA-I, which have the form:

$$\text{(SP1)} \quad \min\{h(y) : y \in \mathcal{Q}^*\},$$

where h is continuously differentiable and its gradient is Lipshitz continuous with modulus $L_h > 0$.

For given $y^0 \in \mathcal{Q}^*$, the GPM for solving (SP1) is:

$$y^{k+1} = \Pi_{\mathcal{Q}^*}[y^k - \alpha_k \nabla h(y^k)],$$

where $\alpha_k > 0$ is the steplength to be decided in various rules, e.g., the Armijo line search rule or the constant steplength rule:

$$\alpha_k = s \quad \text{with } s \in (0, 2/L_h).$$

• The GPM has an iteration complexity of $O(L_h/\varepsilon)$ for achieving ε-optimality for any $\varepsilon > 0$.

- An accelerated proximal gradient method

We consider an accelerated proximal gradient (APG) method to solve the inner problems of the primal PPA and the primal-dual PPA-II, which have the form:

$$\text{(SP2)} \qquad \min_{X \in \Re^{n_1 \times n_2}} \{H(X) := \|X\|_* + h(X)\},$$

where h are proper, convex, continuously differentiable on $\Re^{n_1 \times n_2}$, and ∇h are globally Lipschitz continuous with modulus $L_h > 0$.

- The APG algorithm

For given $\tau_0 = \tau_{-1} = 1$ and $X^0 = X^{-1} \in \Re^{n_1 \times n_2}$, the APG algorithm applied to solving (SP2) can be expressed as:

$$\begin{cases} \widehat{X}^k &= X^k + \tau_k^{-1}(\tau_{k-1} - 1)(X^k - X^{k-1}), \\ X^{k+1} &= \mathcal{P}_{L_h^{-1}}[\widehat{X}^k - L_h^{-1}\nabla h(\widehat{X}^k)], \\ \tau_{k+1} &= (\sqrt{1 + 4\tau_k^2} + 1)/2, \end{cases}$$

where L_h is the Lipschitz modulus of h.

• The APG algorithm has an attractive iteration complexity of $O(\sqrt{L_h/\varepsilon})$ for achieving ε-optimality for any $\varepsilon > 0$.

♠ Why we do not apply the APG algorithm to the inner problems of the primal PPA and the primal-dual PPA-I?

The reason is that it requires two SVDs in each iteration. Two SVDs per iteration is too expensive.

Numerical results

- Stopping Criteria:

$$\frac{\|b - \mathcal{A}(X^k)\|}{\max\{1, \|b\|\}} < 5 \times 10^{-5}$$

or

$$\frac{\big|\|b - \mathcal{A}(X^k)\| - \|b - \mathcal{A}(X^{k-1})\|\big|}{\max\{1, \|b\|\}} < 5 \times 10^{-5}.$$

- PC: Intel Xeon 3.20GHz with 4GB, Linux and MATLAB (Version 7.6).

- PROPACK package to compute partial SVDs.

Table 1: Numerical results for the primal PPA versus the SVT algorithm.

$\mathbf{n/r/(}m/d_r\mathbf{)}$	method	$\lambda = n/2$	$\lambda = n$	$\lambda = 5n$	$\lambda = 10n$
1000/50/4	PPA	64	60	88	169
	SVT ($\delta = 1.0/p$)	fail	fail	135	250
	SVT ($\delta = 1.2/p$)	fail	fail	112	208
	SVT ($\delta = 1.5/p$)	fail	fail	89	165
5000/50/5	PPA	70	72	86	141
	SVT ($\delta = 1.0/p$)	fail	fail	129	239
	SVT ($\delta = 1.2/p$)	fail	fail	108	199
	SVT ($\delta = 1.5/p$)	fail	fail	86	159

Table 2: Numerical results for **the primal PPA** on random matrix completion problems without noise.

Unknown M					**Results**			
n	p	r	p/d_r	μ	iter	#sv	time	error
1000	119560	10	6	1.00e-03	54	10	5.77e+00	7.02e-05
	389638	50	4	1.00e-03	61	50	3.19e+01	7.42e-05
	569896	100	3	1.00e-03	77	100	1.03e+02	5.34e-05
5000	599936	10	6	2.00e-04	57	10	2.23e+01	6.11e-05
	2487739	50	5	2.00e-04	72	50	2.13e+02	4.12e-05
	3960882	100	4	2.00e-04	83	100	6.94e+02	1.04e-04
10000	1200730	10	6	1.00e-04	52	10	4.40e+01	1.43e-04
	4985869	50	5	1.00e-04	81	50	5.61e+02	3.05e-05
	7959722	100	4	1.00e-04	82	100	1.48e+03	8.35e-05
20000	2400447	10	6	5.00e-05	66	10	1.07e+02	1.20e-04
30000	3599590	10	6	3.33e-05	72	10	1.86e+02	5.90e-05
50000	5995467	10	6	2.00e-05	70	10	3.49e+02	5.59e-04
100000	11994813	10	6	1.00e-05	99	10	9.85e+02	8.58e-05

Table 3: Numerical results for **the primal PPA** on random matrix completion problems with noise. The noise factor is set to 0.1.

Unknown M					**Results**			
n	p	r	p/d_r	μ	iter	#sv	time	error
1000 /0.10	119560	10	6	1.00e-03	39	10	5.30e+00	5.62e-02
	389638	50	4	1.00e-03	47	51	3.06e+01	7.74e-02
	569896	100	3	1.00e-03	48	100	6.23e+01	7.94e-02
5000 /0.10	599936	10	6	2.00e-04	45	10	2.31e+01	5.02e-02
	2487739	50	5	2.00e-04	53	50	2.20e+02	5.93e-02
	3960882	100	4	2.00e-04	47	100	4.07e+02	7.72e-02
10000 /0.10	1200730	10	6	1.00e-04	45	10	4.78e+01	4.89e-02
	4985869	50	5	1.00e-04	36	50	2.89e+02	5.84e-02
	7959722	100	4	1.00e-04	57	100	1.23e+03	6.82e-02
20000 /0.10	2400447	10	6	5.00e-05	47	10	9.26e+01	5.60e-02
30000 /0.10	3599590	10	6	3.33e-05	53	10	1.69e+02	4.80e-02
50000 /0.10	5995467	10	6	2.00e-05	58	10	3.28e+02	5.24e-02
100000 /0.10	11994813	10	6	1.00e-05	67	10	7.52e+02	5.42e-02

Table 4: Numerical results for **the dual PPA** on random matrix completion problems without noise.

Unknown M					Results			
n	p	r	p/d_r	μ	iter	#sv	time	error
1000	119560	10	6	1.44e-02	35	10	3.90e+00	1.05e-04
	389638	50	4	5.37e-02	51	50	2.95e+01	6.21e-05
	569896	100	3	8.66e-02	56	100	7.78e+01	2.41e-05
5000	599936	10	6	1.38e-02	42	10	1.71e+01	7.34e-05
	2487739	50	5	6.08e-02	50	50	1.47e+02	6.50e-05
	3960882	100	4	1.02e-01	56	100	4.32e+02	9.68e-05
10000	1200730	10	6	1.37e-02	40	10	2.96e+01	1.40e-04
	4985869	50	5	5.93e-02	51	50	3.19e+02	6.54e-05
	7959722	100	4	9.88e-02	56	100	9.05e+02	1.04e-04
20000	2400447	10	6	1.35e-02	45	10	6.72e+01	1.50e-04
30000	3599590	10	6	1.35e-02	54	10	1.21e+02	1.41e-04
50000	5995467	10	6	1.34e-02	58	10	2.46e+02	4.83e-05
100000	11994813	10	6	1.34e-02	55	10	5.19e+02	1.04e-04

Table 5: Numerical results for **the dual PPA** on random matrix completion problems with noise. The noise factor is set to 0.1.

Unknown M					Results			
n	p	r	p/d_r	μ	iter	#sv	time	error
1000 /0.10	119560	10	6	1.44e-02	29	10	3.95e+00	4.49e-02
	389638	50	4	5.37e-02	31	50	1.52e+01	5.49e-02
	569896	100	3	8.67e-02	39	100	4.36e+01	6.39e-02
5000 /0.10	599936	10	6	1.38e-02	39	10	2.20e+01	4.51e-02
	2487739	50	5	6.08e-02	39	50	1.09e+02	4.96e-02
	3960882	100	4	1.02e-01	41	100	2.71e+02	5.67e-02
10000 /0.10	1200730	10	6	1.37e-02	44	10	4.73e+01	4.53e-02
	4985869	50	5	5.93e-02	39	50	2.26e+02	4.99e-02
	7959722	100	4	9.89e-02	47	100	6.92e+02	5.73e-02
20000 /0.10	2400447	10	6	1.35e-02	44	10	9.65e+01	4.52e-02
30000 /0.10	3599590	10	6	1.35e-02	45	10	1.45e+02	4.53e-02
50000 /0.10	5995467	10	6	1.34e-02	47	10	2.70e+02	4.53e-02
100000 /0.10	11994813	10	6	1.34e-02	43	10	5.42e+02	4.53e-02

Remarks:

• Comparing the performances of the primal PPA and the dual PPA for random matrix completion problems without and with noisy data, we observe that the dual PPA outperforms the primal PPA.

★ In our experiments, we observe that the performance of the primal-dual PPA-I is similar to that of the primal PPA, and the performance of the primal-dual PPA-II is also similar to that of the dual PPA.