Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 613205, 9 pages
http://dx.doi.org/10.1155/2014/613205

Research Article

A Novel Approach for Solving Semidefinite Programs

Hong-Wei Jiao,[1] Ya-Kui Huang,[2] and Jing Chen[3]

[1] *Department of Mathematics, Henan Institute of Science and Technology, Xinxiang 453003, China*
[2] *School of Mathematics and Statistics, Xidian University, Xi'an 710071, China*
[3] *Department of Basic Science, Henan Mechanical and Electrical Engineering College, Xinxiang 453002, China*

Correspondence should be addressed to Hong-Wei Jiao; jhwxd2014@126.com and Ya-Kui Huang; huangykxd@126.com

Received 2 April 2014; Accepted 3 August 2014; Published 18 August 2014

Academic Editor: Ram N. Mohapatra

A novel linearizing alternating direction augmented Lagrangian approach is proposed for effectively solving semidefinite programs (SDP). For every iteration, by fixing the other variables, the proposed approach alternatively optimizes the dual variables and the dual slack variables; then the primal variables, that is, Lagrange multipliers, are updated. In addition, the proposed approach renews all the variables in closed forms without solving any system of linear equations. Global convergence of the proposed approach is proved under mild conditions, and two numerical problems are given to demonstrate the effectiveness of the presented approach.

1. Introduction

Minimizing a linear function of a symmetric positive semidefinite matrix subject to linear equality constraints is called semidefinite programs (SDP), whose form can be given as follows:

$$\begin{aligned} \min \quad & \langle C, X\rangle \\ \text{s.t.} \quad & \mathscr{A}(X) = b, \\ & X \succeq 0, \end{aligned} \tag{1}$$

where $\mathscr{A} : \mathscr{S}^n \to \mathscr{R}^m$ is a linear operator, which can be expressed as

$$\mathscr{A}(X) := \left(\langle A_1, X\rangle, \dots, \langle A_m, X\rangle\right)^T. \tag{2}$$

$C, A_i \in \mathscr{S}^n$, $i = 1, \dots, m$, are all matrices, and $b \in \mathscr{R}^m$ is a vector, and $X \succeq 0$ stands for the fact that X is a symmetric positive semidefinite matrix. Here, $\mathscr{S}^n$ stands for the space of $n \times n$ symmetric matrices and $\langle X, Y\rangle = \operatorname{trace}(XY)$ stands for the standard inner product in $\mathscr{S}^n$. $\mathscr{A}^*(y) := \sum_{i=1}^m y_i A_i$ stands for the adjoint operator $\mathscr{A}^* : \mathscr{R}^m \to \mathscr{S}^n$ of $\mathscr{A}$. The dual problem of (1) is

$$\begin{aligned} \min \quad & -b^T y \\ \text{s.t.} \quad & \mathscr{A}^*(y) + S = C, \\ & S \succeq 0, \end{aligned} \tag{3}$$

where $y \in \mathscr{R}^m$ and $S \in \mathscr{S}^n$.

SDP problem has been always a very active area in optimization research for many years. It has broad applications in many areas, for example, system and control theory [1], combinatorial optimization [2], nonconvex quadratic programs [3], and matrix completion problems [4]. We refer to the reference book [5] for theory and applications of SDP. Interior point approaches (IPMs) have been very successful for solving SDP in polynomial time [6–9]. For small and medium sized SDP problems such as $n \le 1,000$ and $m \le 10,000$, IPMs are generally efficient and robust. However, for large-scale SDP problems with large m and moderate n, IPMs become very slow due to the need of computing and factorizing the $m \times m$ Schur complement matrix. In order to improve this shortcoming, by using an iterative solver to compute a search direction at each iteration, [10, 11] proposed inexact IPMs which manage to solve certain types of SDP problems with m up to 125,000. Based on the augmented Lagrangian approach, many variants for SDP were proposed. For example, [12] introduced the so-called boundary point approach; using an eigenvalue decomposition to maintain complementarity, [13] presented a dual augmented Lagrangian approach. More recently, Huang and Xu [14] proposed a trust region algorithm for SDP problems by performing a number of conjugate gradient iterations to solve the subproblems. Zhao et al. [15] designed a Newton-CG augmented Lagrangian approach for solving SDP problems from the perspective of approximate semismooth Newton methods. Wen et al. [16] presented an alternating direction dual augmented Lagrangian approach for SDP. In [17], Wen et al. proposed a row-by-row approach for solving SDP problems based on solving a sequence of problems obtained by restricting the n-dimensional positive semidefinite constraint on the matrix X. In addition to the former reviewed methods, some related research works on the subject are given as follows. Xu et al. [18] presented a new algorithm for the box-constrained SDP based on the feasible direction method. Zhadan and Orlov [19] presented a dual interior point method for linear SDP problem. Lin [20] proposed an inexact spectral bundle method for convex quadratic SDP. Sun and Zhang [21] proposed a modified alternating direction method for solving convex quadratically constrained quadratic SDP, which requires much less computational effort per iteration than the second-order approaches. In [22], the authors presented penalty and barrier methods for convex SDP. In [23], an alternating direction method is proposed for solving convex SDP problems by Zhang et al., which only computes several metric projections at each iteration. In [24], Huang et al. presented a lower-order penalization approach to solve nonlinear SDP. In [25], Aroztegui et al. presented a feasible direction interior point algorithm for solving nonlinear SDP. Yang and Yu [26] proposed a homotopy method for nonlinear SDP. Kanzow et al. [27] presented successive linearization methods for solving nonlinear SDP. Yamashita et al. [28] presented a primal-dual interior point method for nonlinear SDP. Lu et al. [29] presented a saddle point mirror-prox algorithm for solving a large-scale SDP. In [30], Monteiro et al. presented a first-order block-decomposition method for solving two-easy-block structured SDP. In [31], an efficient low-rank stochastic gradient descent method is proposed for solving a class of SDP problems, which has clear computational advantages over the standard stochastic gradient descent method. Based on a new technique for finding the search direction and the strategy of the central path, Wang and Bai [32] presented a new primal-dual path-following interior-point algorithm for solving SDP problem. By reformulating the complementary conditions in the primal-dual optimality conditions as a projection equation, Yu [33] presented an alternating direction algorithm for the solution of SDP problems. However, most of these existing methods need to solve a system of linear equations for updating the variables which is time consuming especially for the large-scale case.

In this paper, we present a novel linearizing alternating direction dual augmented Lagrangian approach for computing SDP problems. For every iteration, the proposed algorithm works on the augmented Lagrangian function for the dual SDP problem. Specially, for every iteration, by fixing the other variables the proposed algorithm alternatively optimizes the dual variables and the dual slack variables; then the primal variables, that is, Lagrange multipliers, are updated. The proposed algorithm is closely related to the alternating direction augmented Lagrangian approach in [16] but for updating the dual variables. In particular, the proposed algorithm renews the dual variables without solving any system of linear equations. Moreover, the proposed algorithm renews all the variables in closed forms. Numerical experimental results demonstrate that the performance of the proposed approach can be significantly better than that reported in [16].

The remaining section of this paper is described as follows. In Section 2 a novel linearizing alternating direction augmented Lagrangian approach is proposed for solving SDP problems. The convergence of the proposed approach is proved in Section 3. In Section 4, some implementation issues of the proposed approach are discussed. Two numerical examples for frequency assignment problem and binary integer quadratic programs problems are used to demonstrate the performance of the proposed approach in Section 5.

Some notations: $\mathcal{S}_{+}^{n}$ represents the set of $n \times n$ symmetric positive semidefinite matrices. $X \succ 0$ represents the fact that X is positive definite. The notation $\|\cdot\|$ stands for the Euclidean norm and $\|\cdot\|_F$ stands for the Frobenius norm. $\mathrm{vec}(X)$ denotes a vector obtained by stacking X's columns one by one. I denotes the identity matrix in proper order.

2. Linearizing Alternating Direction Augmented Lagrangian Approach

In this section, a linearizing alternating direction augmented Lagrangian approach is proposed for solving (1) and (3).

Let $A := (\mathrm{vec}(A_1), \dots, \mathrm{vec}(A_m))^T \in \mathcal{R}^{m \times n^2}$. The expression $\mathcal{A}(X) = b$ is equal to $A\,\mathrm{vec}(X) = b$. $\mathcal{A}(\mathcal{A}^*(y)) = AA^T y$ is called an operator $\mathcal{A}\mathcal{A}^* : \mathcal{R}^m \to \mathcal{R}^m$.

Without loss of generality, we assume that matrix A is a full row rank and there exists a matrix $\widetilde{X} \succ 0$ such that $\mathcal{A}(\widetilde{X}) = b$. It is well known that, with the above assumption, a point (X, y, S) is optimal for SDP problems (1) and (3) if and only if

$$\mathcal{A}(X) = b, \quad \mathcal{A}^*(y) + S = C, \quad XS = 0, \quad X \succeq 0, \ S \succeq 0. \tag{4}$$

Given a penalty parameter $\mu > 0$, the augmented Lagrangian function for the dual SDP (3) is defined as

$$L_\mu(X, y, S) := -b^T y + \langle X, \mathcal{A}^*(y) + S - C \rangle + \frac{1}{2\mu} \left\| \mathcal{A}^*(y) + S - C \right\|_F^2, \tag{5}$$

where $X \in \mathcal{S}^n$. For given $X^0, S^0 \in \mathcal{S}^n$, the alternating direction augmented Lagrangian approach for solving problems (1) and (3) generates sequences $\{y^k\} \subset \mathcal{R}^m$, $\{S^k\} \subset \mathcal{S}^n$, and $\{X^k\} \subset \mathcal{S}^n$ as follows:

$$y^{k+1} = \arg\min_{y \in \mathcal{R}^m} L_\mu(X^k, y, S^k), \tag{6}$$

$$S^{k+1} = \arg\min_{S \in \mathcal{S}^n} L_\mu(X^k, y^{k+1}, S), \quad S \succeq 0, \tag{7}$$

$$X^{k+1} = X^k + \frac{\mathcal{A}^*(y^{k+1}) + S^{k+1} - C}{\mu}. \tag{8}$$

Apparently, we can obtain y^{k+1} by solving the first-order optimality conditions for (6), which is a system of linear equations associated with AA^T. Since AA^T is a $m \times m$ matrix, it is difficult to get y^{k+1} exactly when m is large. In order to alleviate this difficulty, we use the quadratic approximation of $L_\mu(X^k, y, S^k)$ in (6) around y^kas follows:

$$L_\mu\left(X^k, y, S^k\right) \approx L_\mu\left(X^k, y^k, S^k\right) + \left\langle g^k, y - y^k \right\rangle + \frac{\lambda_k}{2\mu}\left\| y - y^k \right\|^2, \quad (9)$$

where $\lambda_k > 0$ and

$$\begin{aligned} g^k &= \nabla_y L_\mu\left(X^k, y^k, S^k\right) \\ &= \mathscr{A}\left(X^k\right) - b + \frac{1}{\mu}\mathscr{A}\left(\mathscr{A}^*\left(y^k\right) + S^k - C\right). \end{aligned} \quad (10)$$

We replace step (6) by

$$y^{k+1} = \arg\min_{y \in \mathscr{R}^m} \left\langle g^k, y - y^k \right\rangle + \frac{\lambda_k}{2\mu}\left\| y - y^k \right\|^2. \quad (11)$$

Then, we have

$$y^{k+1} = y^k - \frac{\mu}{\lambda_k} g^k. \quad (12)$$

As pointed out in [16], problem (7) is equivalent to

$$\min_{S \in \mathcal{S}^n} \left\| S - V^{k+1} \right\|_F^2, \quad S \succeq 0, \quad (13)$$

where $V^{k+1} = C - \mathscr{A}^*(y^{k+1}) - \mu X^k$. Denote the spectral decomposition of the matrix V^{k+1} by

$$Q \Sigma Q^T = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} \Sigma_+ & 0 \\ 0 & \Sigma_- \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix}, \quad (14)$$

where Σ_+ and Σ_- are the nonnegative and negative eigenvalues of V^{k+1}. We then obtain the fact that $S^{k+1} = V_+^{k+1} = Q_1 \Sigma_+ Q_1$. It follows from (8) that

$$\begin{aligned} X^{k+1} &= X^k + \frac{\mathscr{A}^*\left(y^{k+1}\right) + S^{k+1} - C}{\mu} \\ &= \frac{1}{\mu}\left(S^{k+1} - V^{k+1}\right) = \frac{1}{\mu} V_-^{k+1}, \end{aligned} \quad (15)$$

where $V_-^{k+1} = -Q_2 \Sigma_- Q_2$. Now we present the linearizing alternating direction augmented Lagrangian approach in Algorithm 1.

Remark 1. We can choose $\lambda_{k+1} = \|AA^T\|_F$ to satisfy the condition of Algorithm 1. If $\mathscr{A}\mathscr{A}^* = I$ and $\lambda_k = 1$ for all $k \geq 0$, then Algorithm 1 is same as the approach proposed in [16].

3. The Convergence of the Proposed Approach

In this section, we prove the convergence of Algorithm 1 using the argument similar to the one in [34]. Let $R_d^k = \mathscr{A}^*(y^k) + S^k - C$; then we have the following proposition.

Lemma 2. *Let $w^k = (X^k, y^k, S^k)$ be generated by Algorithm 1 and let $w^* = (X^*, y^*, S^*)$ be an optimal solution of* (1) *and* (3)*; then one has*

$$\left\| w^k - w^* \right\|_{H_\mu}^2 - \left\| w^{k+1} - w^* \right\|_{H_\mu}^2 \geq \left\| w^k - w^{k+1} \right\|_{H_\mu}^2, \quad (16)$$

where

$$H_\mu = \begin{pmatrix} \mu I & 0 & 0 \\ 0 & \frac{1}{\mu}\left(\lambda_k I - AA^T\right) & 0 \\ 0 & 0 & \frac{1}{\mu} I \end{pmatrix}, \qquad \|w\|_H^2 = \langle w, Hw \rangle. \quad (17)$$

Proof. From (8), there holds

$$\left\langle X^{k+1} - X^*, \mu\left(X^k - X^{k+1}\right)\right\rangle = \left\langle X^* - X^{k+1}, R_d^{k+1} \right\rangle. \quad (18)$$

By (12), we know that

$$\left\langle y^* - y^{k+1}, g^k + \frac{\lambda_k}{\mu}\left(y^{k+1} - y^k\right)\right\rangle = 0. \quad (19)$$

That is,

$$\left\langle y^* - y^{k+1}, \mathscr{A}\left(X^k\right) - b + \frac{1}{\mu}\mathscr{A}\left(R_d^k\right) + \frac{\lambda_k}{\mu}\left(y^{k+1} - y^k\right)\right\rangle = 0. \quad (20)$$

By substituting (8) into the above equality, using the fact $\mathscr{A}(X^*) = b$, and rearranging the terms, one has

$$\begin{aligned} &\left\langle y^{k+1} - y^*, \frac{1}{\mu}\left(\lambda_k I - \mathscr{A}\mathscr{A}^*\right)\left(y^k - y^{k+1}\right)\right\rangle \\ &= \left\langle \mathscr{A}^*\left(y^{k+1} - y^*\right), X^{k+1} - X^* \right\rangle \\ &\quad + \left\langle \mathscr{A}^*\left(y^{k+1} - y^*\right), \frac{1}{\mu}\left(S^k - S^{k+1}\right)\right\rangle. \end{aligned} \quad (21)$$

Since $X^{k+1} S^{k+1} = 0$, we have

$$\left\langle S - S^{k+1}, X^{k+1} \right\rangle \geq 0, \quad \forall S \in \mathcal{S}_+^n. \quad (22)$$

By substituting $S = S^*$ into (22), we get

$$\begin{aligned} &\left\langle S^{k+1} - S^*, \frac{1}{\mu}\left(S^k - S^{k+1}\right)\right\rangle \\ &\geq \left\langle S^{k+1} - S^*, X^{k+1} + \frac{1}{\mu}\left(S^k - S^{k+1}\right)\right\rangle. \end{aligned} \quad (23)$$

Initialize $y^0 \in \mathscr{R}^m$, $X^0 \succeq 0$, and $S^0 \succeq 0$. Choose initial step size λ_0 greater than the maximum eigenvalue of the matrix AA^T.
For $k = 0, 1, \ldots$ **do**
Compute g^k and $y^{k+1} = y^k - \mu/\lambda\_k \ g^k$.
Compute V^{k+1} and its eigenvalue decomposition, and set $S^{k+1} = V_+^{k+1}$.
Compute $X^{k+1} = 1/\mu\left(S^{k+1} - V^{k+1}\right)$.
Choose λ_{k+1} greater than the maximum eigenvalue of the matrix AA^T.
end

Algorithm 1: Linearizing alternating direction augmented Lagrangian algorithm for SDP.

By adding (18), (21), and (23) together, we obtain

$$\begin{aligned}
&\left\langle w^{k+1} - w^*, H_\mu\left(w^k - w^{k+1}\right)\right\rangle \\
&\quad \geq \left\langle X^* - X^{k+1}, R_d^{k+1}\right\rangle + \left\langle \mathscr{A}^*\left(y^{k+1} - y^*\right), X^{k+1} - X^*\right\rangle \\
&\qquad + \left\langle \mathscr{A}^*\left(y^{k+1} - y^*\right), \frac{1}{\mu}\left(S^k - S^{k+1}\right)\right\rangle \\
&\qquad + \left\langle S^{k+1} - S^*, X^{k+1}\right\rangle + \left\langle S^{k+1} - S^*, \frac{1}{\mu}\left(S^k - S^{k+1}\right)\right\rangle \\
&\quad = \left\langle X^* - X^{k+1}, S^{k+1} - S^*\right\rangle + \left\langle S^k - S^{k+1}, \frac{1}{\mu}R_d^{k+1}\right\rangle \\
&\qquad + \left\langle S^{k+1} - S^*, X^{k+1}\right\rangle \\
&\quad = \left\langle X^*, S^{k+1} - S^*\right\rangle + \left\langle S^k - S^{k+1}, \frac{1}{\mu}R_d^{k+1}\right\rangle \\
&\quad \geq \left\langle X^*, S^{k+1}\right\rangle - \left\langle S^k - S^{k+1}, X^k\right\rangle \\
&\quad = \left\langle X^*, S^{k+1}\right\rangle + \left\langle S^{k+1}, X^k\right\rangle,
\end{aligned} \tag{24}$$

where the last inequality comes from (8) and (22). Note that X^k, S^{k+1}, and X^* are semidefinite positive matrices; then

$$\left\langle w^{k+1} - w^*, H_\mu\left(w^k - w^{k+1}\right)\right\rangle \geq 0. \tag{25}$$

It follows (25) that

$$\begin{aligned}
&\left\langle w^k - w^*, H_\mu\left(w^k - w^{k+1}\right)\right\rangle \\
&\quad = \left\langle w^{k+1} - w^*, H_\mu\left(w^k - w^{k+1}\right)\right\rangle + \left\|w^k - w^{k+1}\right\|_{H_\mu}^2 \\
&\quad \geq \left\|w^k - w^{k+1}\right\|_{H_\mu}^2.
\end{aligned} \tag{26}$$

By (26) and the fact that

$$\begin{aligned}
&\left\|w^k - w^*\right\|_{H_\mu}^2 - \left\|w^{k+1} - w^*\right\|_{H_\mu}^2 \\
&\qquad = 2\left\langle w^k - w^*, w^k - w^{k+1}\right\rangle - \left\|w^k - w^{k+1}\right\|_{H_\mu}^2,
\end{aligned} \tag{27}$$

we have

$$\left\|w^k - w^*\right\|_{H_\mu}^2 - \left\|w^{k+1} - w^*\right\|_{H_\mu}^2 \geq \left\|w^k - w^{k+1}\right\|_{H_\mu}^2, \tag{28}$$

which completes the proof. □

Theorem 3. *Let* $\{(X^k, y^k, S^k)\}$ *be generated by Algorithm 1; then it converges to a solution of problems* (1) *and* (3).

Proof. By (16), we know that the sequence $\{w^k\}$ is bounded and the sequence $\{\|w^k - w^*\|_{H_\mu}^2\}$ is monotonically nonincreasing. Therefore,

$$\lim_{k\to\infty}\left\|w^k - w^*\right\|_{H_\mu}^2 = \left\|\overline{w} - w^*\right\|_{H_\mu}^2, \tag{29}$$

where $\overline{w} = (\overline{X}, \overline{y}, \overline{S})$ can be any limit point of $\{w^k\}$. It follows that

$$\lim_{k\to\infty}\left\|w^k - w^{k+1}\right\|_{H_\mu}^2 = 0. \tag{30}$$

Since λ_k are greater than the maximum eigenvalue of matrix AA^T, then the matrix H_μ is positive definite. By the definition of w^k, we obtain

$$\begin{gathered}
\lim_{k\to\infty}\left\|X^k - X^{k+1}\right\|_F = 0, \qquad \lim_{k\to\infty}\left\|y^k - y^{k+1}\right\| = 0, \\
\lim_{k\to\infty}\left\|S^k - S^{k+1}\right\|_F = 0.
\end{gathered} \tag{31}$$

From the update formula (8), we have

$$\lim_{k\to\infty}\left\|\mathscr{A}^*\left(y^k\right) + S^k - C\right\|_F = 0. \tag{32}$$

By (12) and the definition of g^k, one has

$$\lim_{k\to\infty}\left\|g^k\right\| = \left\|\mathscr{A}\left(X^k\right) - b + \frac{1}{\mu}\mathscr{A}\left(\mathscr{A}^*\left(y^k\right) + S^k - C\right)\right\| = 0, \tag{33}$$

which together with (32) imply that

$$\lim_{k\to\infty}\left\|\mathscr{A}\left(X^k\right) - b\right\| = 0. \tag{34}$$

By combining (32), (34), $X^kS^k = 0$, and $X^k, S^k \geq 0$ for all $k \geq 1$, we know that any limit point of $\{w^k\}$, say $\overline{w} = (\overline{X}, \overline{y}, \overline{S})$, satisfies

$$\mathscr{A}(\overline{X}) = b, \quad \mathscr{A}^*(\overline{y}) + \overline{S} - C = 0, \quad \overline{X}\,\overline{S} = 0, \quad \overline{X} \geq 0, \quad \overline{S} \geq 0, \tag{35}$$

which means $\overline{w}$ is a solution of problems (1) and (3). By Lemma 2, $\{(X^k, y^k, S^k)\}$ converges to a solution of problems (1) and (3). □

4. Implementation Issues

The proposed algorithm is carried out by modifying the code of the alternating direction approach in [16] which is referred to as SDPAD. Before presenting the numerical results, we discuss some implementation issues of Algorithm 1 in this section.

In order to improve the computational performance of Algorithm 1, using the similar method as many alternating direction approaches [35–37], we replace step (8) by

$$\begin{aligned} X^{k+1} &:= X^k + \rho \frac{\mathscr{A}^*(y^{k+1}) + S^{k+1} - C}{\mu} \\ &:= (1-\rho) X^k + \frac{\rho}{\mu}\left(S^{k+1} - V^{k+1}\right). \end{aligned} \tag{36}$$

We can use an argument similar method to the one in [34] to prove the following theorem.

Theorem 4. *Let $A_{\lambda_k} = \lambda_k I - AA^T$ (X^*, y^*, S^*) be an optimal solution of* (1) *and* (3). *For $\rho \in (0,1]$, it holds that*

$$\begin{aligned} &\left\|X^k - X^*\right\|_F^2 + \frac{\rho}{\mu}\left\|y^k - y^*\right\|_{A_{\lambda_k}}^2 + \frac{\rho}{\mu^2}\left\|S^k - S^*\right\|_F^2 \\ &\quad + \frac{\rho(1-\rho)}{\mu^2}\left\|R_d^k\right\|_F^2 \\ &\geq \left(\left\|X^{k+1} - X^*\right\|_F^2 + \frac{\rho}{\mu}\left\|y^{k+1} - y^*\right\|_{A_{\lambda_k}}^2\right. \\ &\quad \left. + \frac{\rho}{\mu^2}\left\|S^{k+1} - S^*\right\|_F^2 + \frac{\rho(1-\rho)}{\mu^2}\left\|R_d^{k+1}\right\|_F^2\right) \\ &\quad + \frac{\rho}{\mu}\left\|y^k - y^{k+1}\right\|_{A_{\lambda_k}}^2 + \frac{\rho}{\mu^2}\left\|R_d^{k+1}\right\|_F^2 + \frac{\rho^2}{\mu^2}\left\|S^k - S^{k+1}\right\|_F^2. \end{aligned} \tag{37}$$

For $\rho \in (1, (1+\sqrt{5})/2)$, it holds that

$$\begin{aligned} &\left\|X^k - X^*\right\|_F^2 + \frac{\rho}{\mu}\left\|y^k - y^*\right\|_{A_{\lambda_k}}^2 + \frac{\rho}{\mu^2}\left\|S^k - S^*\right\|_F^2 + \frac{\rho - 1}{\mu^2}\left\|R_d^k\right\|_F^2 \\ &\geq \left(\left\|X^{k+1} - X^*\right\|_F^2 + \frac{\rho}{\mu}\left\|y^{k+1} - y^*\right\|_{A_{\lambda_k}}^2 + \frac{\rho}{\mu^2}\left\|S^{k+1} - S^*\right\|_F^2\right. \\ &\quad \left. + \frac{\rho - 1}{\mu^2}\left\|R_d^{k+1}\right\|_F^2\right) + \frac{\rho}{\mu}\left\|y^k - y^{k+1}\right\|_{A_{\lambda_k}}^2 \\ &\quad + \frac{1+\rho-\rho^2}{\rho}\left(\frac{\rho}{\mu^2}\left\|R_d^{k+1}\right\|_F^2 + \frac{\rho^2}{\mu^2}\left\|S^k - S^{k+1}\right\|_F^2\right). \end{aligned} \tag{38}$$

Based on Theorem 4, it is not difficult to show that

$$\begin{aligned} &\lim_{k\to\infty}\left\|\mathscr{A}^*(y^k) + S^k - C\right\|_F = 0, \\ &\lim_{k\to\infty}\left\|\mathscr{A}(X^k) - b\right\| = 0, \qquad \lim_{k\to\infty}\left\|X^kS^k\right\|_F = 0. \end{aligned} \tag{39}$$

In our numerical experiments, we stop the algorithm when

$$\max\{\text{pinf}, \text{dinf}, \text{gap}\} \leq 10^{-6}, \tag{40}$$

where

$$\begin{aligned} \text{pinf} &= \frac{\mathscr{A}(X) - b}{1 + \|b\|}, \qquad \text{dinf} = \frac{C + S - \mathscr{A}^*(y)}{1 + \|C\|_F}, \\ \text{gap} &= \frac{\left|b^T y - \langle C, X\rangle\right|}{1 + |\langle C, X\rangle| + \left|b^T y\right|}. \end{aligned} \tag{41}$$

We set the maximum number of iterations allowed in Algorithm 1 and SDPAD to 20,000.

We use the same strategy for updating the parameter $\mu > 0$ SDPAD (beta 2). In particular, given some integer $h > 0$, let

$$\begin{aligned} \text{pvd}(k) &= e^{\sum_{j=k-h+1}^{k} \ln(\text{pinf}(j)/\text{dinf}(j))/h}, \\ \text{dvp}(k) &= e^{\sum_{j=k-h+1}^{k} \ln(\text{dinf}(j)/\text{pinf}(j))/h}. \end{aligned} \tag{42}$$

For $k = h, 2h, \ldots$, if $\text{pvd}(k) > \eta$, then set $\mu = \max\{\min\{\mu/\gamma, \mu_{\max}\}, \mu_{\min}\}$. Otherwise, if $\text{dvp}(k) > \eta$, then set $\mu = \max\{\min\{\gamma\mu, \mu_{\max}\}, \mu_{\min}\}$. Here $0 < \mu_{\min} < \mu_{\max} < \infty$.

We set $\mu_{\min} = 10^{-2}$, $\mu_{\max} = 10^2$, $\gamma = 0.75$, and $\eta = 1.5$ for our test problems. The parameter ρ for updating X is set to 1.618. We choose the initial iterate $y^0 = 0$, $X^0 = I$, and $S^0 = 0$.

Let $f(y, S) = (1/2)\|\mathscr{A}^*(y) + S - C\|_F^2$. Since the other parts of L_μ are linear, the choice of λ_k is mainly depending on $f(y, S)$. We set $\lambda_0 = \|AA^T\|_F$ and choose λ_{k+1} as the Barzilai-Borwein step size [38] of $f(y, S)$ with the following safeguard:

$$\lambda_{k+1} = \begin{cases} \left\|AA^T\right\|_F, & \text{if } p_k^T z_k \leq 0, \\ \max\left\{\dfrac{\|z_k\|^2}{p_k^T z_k}, \left\|AA^T\right\|_F\right\}, & \text{otherwise,} \end{cases} \tag{43}$$

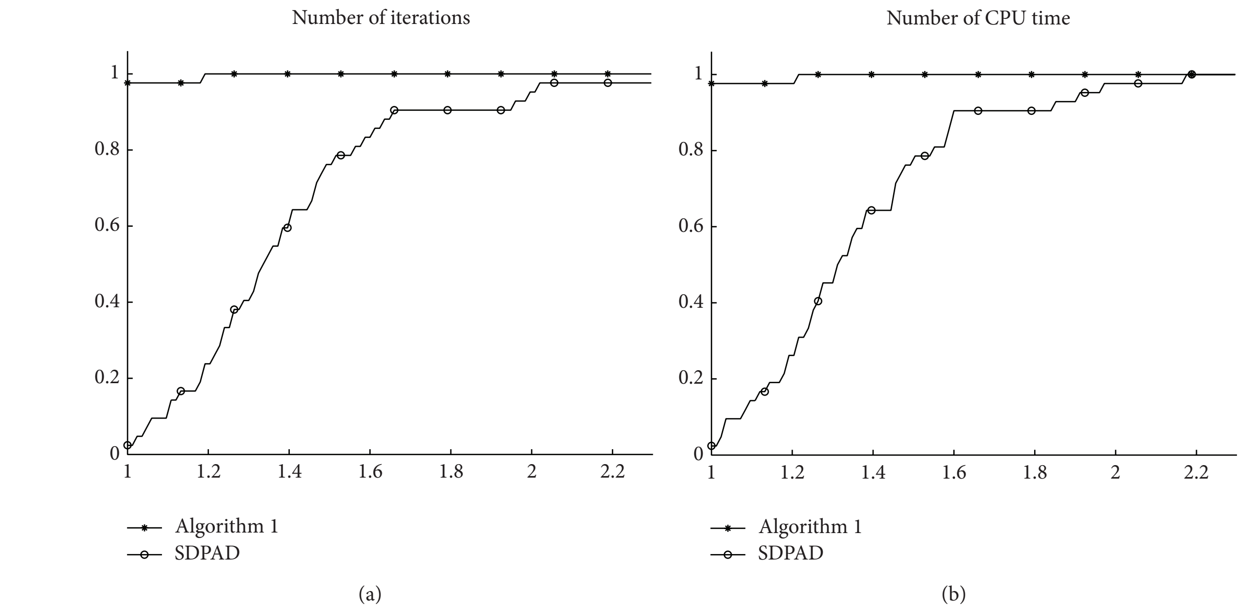

Figure 1: Performance profiles for SDPAD and the present method for number of iterations (a) and CPU time (b).

Table 1: Numerical results compared with [16] for computing frequency assignment problems.

Name	n	m	Algorithm 1 in this paper					SDPAD				
			pinf	dinf	gap	itr	cpu	pinf	dinf	gap	itr	cpu
fap01	52	1378	$9.34e-7$	$2.47e-7$	$4.71e-5$	622	1.13	$9.77e-7$	$1.22e-7$	$5.61e-5$	653	0.67
fap02	61	1866	$9.39e-7$	$9.01e-7$	$5.84e-5$	1399	1.67	$9.98e-7$	$8.92e-7$	$7.27e-5$	1720	1.83
fap03	65	2145	$7.54e-7$	$4.75e-7$	$5.82e-6$	914	1.06	$9.84e-7$	$4.88e-7$	$3.97e-5$	870	0.98
fap04	81	3321	$9.55e-7$	$9.97e-7$	$4.44e-5$	852	1.70	$9.54e-7$	$9.96e-7$	$4.45e-5$	874	1.71
fap05	84	3570	$9.83e-7$	$8.53e-7$	$3.68e-6$	1155	2.38	$9.98e-7$	$8.29e-7$	$4.27e-6$	1198	2.37
fap06	93	4371	$9.93e-7$	$6.44e-7$	$3.19e-6$	653	1.48	$9.89e-7$	$6.45e-7$	$3.40e-6$	653	1.46
fap07	98	4851	$9.97e-7$	$8.44e-7$	$1.00e-5$	648	1.50	$9.99e-7$	$8.20e-7$	$9.76e-6$	667	1.52
fap08	120	7260	$6.65e-7$	$9.98e-7$	$2.58e-6$	725	2.51	$6.63e-7$	$9.96e-7$	$2.57e-6$	725	2.46
fap09	174	15225	$9.98e-7$	$4.34e-7$	$8.42e-7$	438	3.05	$9.90e-7$	$3.04e-7$	$1.02e-6$	464	3.17
fap10	183	14479	$8.33e-7$	$1.00e-6$	$1.14e-4$	2278	21.55	$9.17e-7$	$1.00e-6$	$1.29e-4$	2313	21.87
fap11	252	24292	$9.60e-7$	$9.99e-7$	$2.53e-4$	2462	52.68	$9.65e-7$	$1.00e-6$	$2.16e-4$	2585	54.54
fap12	369	26462	$9.06e-7$	$9.99e-7$	$2.40e-4$	3197	2 : 28	$7.61e-7$	$1.00e-6$	$2.20e-4$	3394	2 : 35
fap25	2118	322924	$8.85e-7$	$1.00e-6$	$1.13e-4$	5152	7 : 54 : 54	$1.00e-6$	$9.67e-7$	$1.13e-4$	5495	8 : 20 : 12
fap36	4110	1154467	$9.93e-7$	$9.78e-7$	$3.08e-5$	4256	47 : 54 : 26	$1.38e-6$	$1.26e-7$	$1.68e-5$	5000	116 : 28 : 06

where $p_k = y^{k+1} - y^k$, and $z_k = \nabla_y f(y^{k+1}, S^{k+1}) - \nabla_y f(y^k, S^k)$. Clearly, this choice of λ_k ensures that the matrix $\lambda_k I - AA^T$ is positive definite for any $k \geq 1$.

5. Numerical Results

In this section, we report our numerical results. We compare solutions obtained from Algorithm 1 and SDPAD on the SDP relaxations of frequency assignment problems and binary integer quadratic programs problems. All the procedures were carried out by MATLAB 2011b on a 3.10 GHz Core i5 PC with 4 GB of RAM under Windows 7.

In Tables 1 and 2, the first column gives the problem name; some notations have been also used in column headers, n: the size of the matrix X; m: the total number of equality and inequality constraints; "itr": the number of iterations; "cpu": the CPU time in the format of "hours, minutes, and seconds."

5.1. Frequency Assignment Relaxation. In this subsection, we consider SDPs arising from semidefinite relaxation of frequency assignment problems (fap) [39]. The explicit description of the SDP form is given in [40]. For a given undirected graph $G = (V, E)$ with vertex set $V = \{1, \ldots, r\}$ and edge set $E \subset V \times V$, assume $W = (w_{ij}) \in \mathcal{S}^r$ is a weight matrix for G. If the edge $(i, j) \notin E$, we suppose $w_{ij} = w_{ji} = 0$.

Table 2: Numerical results compared with [16] for computing binary integer quadratic programs problem.

Name	n	Algorithm 1 in this paper					SDPAD				
		pinf	dinf	gap	itr	cpu	pinf	dinf	gap	itr	cpu
be100.1	101	$9.37e-7$	$9.96e-7$	$5.52e-7$	1464	3.76	$9.70e-7$	$8.82e-7$	$4.96e-7$	2012	4.79
be100.2	101	$5.19e-7$	$9.95e-7$	$2.89e-7$	1322	3.24	$1.00e-6$	$9.05e-7$	$2.81e-7$	1744	4.24
be120.3.1	121	$9.96e-7$	$8.66e-7$	$5.77e-7$	2214	6.82	$9.82e-7$	$9.99e-7$	$7.08e-7$	2447	7.45
be120.3.2	121	$7.33e-7$	$9.72e-7$	$8.85e-7$	1968	6.20	$8.60e-7$	$9.99e-7$	$1.08e-6$	2405	7.53
be120.8.1	121	$9.95e-7$	$9.99e-7$	$7.24e-7$	1618	4.78	$9.99e-7$	$7.73e-7$	$8.24e-7$	2006	5.87
be120.8.2	121	$9.31e-7$	$9.10e-7$	$4.26e-7$	3033	9.56	$5.63e-7$	$1.00e-6$	$9.99e-7$	3415	10.64
be150.3.1	151	$4.33e-7$	$9.96e-7$	$8.89e-7$	2030	9.08	$8.36e-7$	$9.96e-7$	$3.23e-7$	2557	11.36
be150.3.2	151	$9.94e-7$	$8.58e-7$	$3.97e-7$	2244	10.16	$9.45e-7$	$9.98e-7$	$5.42e-7$	3143	14.01
be150.8.1	151	$9.85e-7$	$8.08e-7$	$3.84e-7$	1694	7.33	$9.97e-7$	$9.38e-7$	$1.40e-7$	2255	9.63
be150.8.2	151	$9.40e-7$	$9.72e-7$	$8.63e-7$	1829	8.10	$9.98e-7$	$7.46e-7$	$6.61e-7$	2386	10.28
be200.3.1	201	$5.48e-7$	$9.97e-7$	$9.56e-7$	2031	13.90	$9.05e-7$	$9.99e-7$	$1.22e-6$	2840	19.19
be200.3.2	201	$9.79e-7$	$9.77e-7$	$5.43e-7$	2254	16.03	$8.03e-7$	$9.99e-7$	$5.73e-7$	3276	23.20
be200.8.1	201	$5.64e-7$	$9.98e-7$	$6.72e-7$	3068	21.90	$9.98e-7$	$6.61e-7$	$4.54e-7$	4154	29.47
be200.8.2	201	$9.07e-7$	$9.59e-7$	$5.35e-7$	1817	11.98	$7.72e-7$	$9.98e-7$	$4.65e-7$	2918	19.11
be250.1	251	$8.78e-7$	$9.90e-7$	$4.90e-7$	3638	36.27	$1.00e-6$	$9.76e-7$	$5.54e-7$	5336	52.52
be250.2	251	$8.06e-7$	$9.94e-7$	$8.81e-7$	3280	32.22	$9.99e-7$	$7.95e-7$	$5.36e-7$	5111	49.91
bqp50-1	51	$9.65e-7$	$9.57e-7$	$1.21e-7$	3334	3.73	$1.00e-6$	$5.76e-7$	$5.36e-7$	2800	3.08
bqp50-2	51	$6.80e-7$	$9.92e-7$	$7.91e-7$	4278	4.15	$1.00e-6$	$6.85e-7$	$4.32e-7$	6975	6.60
bqp100-1	101	$5.06e-7$	$9.96e-7$	$3.33e-7$	1558	3.58	$9.18e-7$	$9.98e-7$	$5.16e-7$	1917	4.32
bqp100-2	101	$6.78e-7$	$1.00e-6$	$7.25e-7$	2887	6.43	$9.99e-7$	$9.57e-7$	$6.68e-7$	3438	7.56
bqp250-1	251	$8.36e-7$	$9.48e-7$	$8.33e-7$	3135	30.35	$9.24e-7$	$9.99e-7$	$1.00e-6$	4943	48.14
bqp250-2	251	$6.57e-7$	$9.99e-7$	$7.89e-7$	3467	32.96	$9.86e-7$	$1.00e-6$	$7.41e-7$	5091	48.01
bqp500-1	501	$6.09e-7$	$9.98e-7$	$1.33e-6$	4676	3 : 20	$9.99e-7$	$9.31e-7$	$3.38e-7$	6931	4 : 54
bqp500-2	501	$9.99e-7$	$7.53e-7$	$4.51e-7$	5307	3 : 49	$6.01e-7$	$9.99e-7$	$1.01e-7$	10580	7 : 31
gka1a	51	$9.98e-7$	$2.77e-7$	$3.16e-8$	2240	2.05	$8.43e-7$	$9.81e-7$	$2.15e-6$	2635	2.33
gka2a	61	$8.95e-7$	$9.84e-7$	$2.50e-6$	1326	1.36	$5.88e-7$	$9.98e-7$	$2.39e-6$	2594	2.51
gka3a	71	$9.04e-7$	$8.92e-7$	$5.39e-7$	1098	1.58	$9.84e-7$	$9.98e-7$	$5.21e-7$	1328	1.88
gka4a	81	$9.99e-7$	$9.93e-7$	$8.21e-7$	1371	2.14	$9.98e-7$	$8.02e-7$	$5.02e-7$	2273	3.39
gka5a	51	$8.14e-7$	$9.83e-7$	$1.29e-7$	1268	1.24	$9.97e-7$	$6.58e-7$	$9.02e-9$	1392	1.33
gka6a	31	$8.84e-7$	$7.95e-7$	$3.49e-7$	927	0.60	$1.98e-7$	$9.96e-7$	$1.15e-6$	979	0.62
gka7a	31	$9.12e-7$	$9.96e-7$	$9.07e-7$	948	0.65	$1.00e-6$	$7.50e-7$	$7.33e-7$	1906	1.24
gka8a	101	$8.56e-7$	$9.94e-7$	$2.51e-6$	2521	5.02	$9.91e-7$	$8.54e-7$	$8.65e-7$	5804	10.91
gka9b	101	$5.46e-7$	$1.41e-7$	$1.83e-5$	1263	3.02	$6.45e-7$	$1.73e-7$	$1.88e-5$	1313	3.07
gka10b	126	$9.93e-7$	$8.92e-7$	$2.83e-5$	1775	8.11	$9.97e-7$	$7.65e-7$	$2.45e-5$	1810	8.32
gka6c	91	$9.88e-7$	$7.13e-7$	$1.76e-7$	4002	7.92	$5.56e-7$	$9.99e-7$	$4.07e-7$	5122	9.95
gka7c	101	$8.70e-7$	$9.96e-7$	$1.06e-7$	4478	9.69	$6.83e-7$	$9.99e-7$	$6.47e-7$	5314	11.45
gka9d	101	$7.71e-7$	$9.99e-7$	$3.52e-9$	1091	2.69	$9.92e-7$	$8.58e-7$	$1.04e-7$	1500	3.64
gka10d	101	$5.17e-7$	$9.95e-7$	$3.44e-8$	1423	3.38	$8.61e-7$	$9.96e-7$	$1.01e-6$	1798	4.23
gka4e	201	$9.08e-7$	$7.60e-7$	$4.28e-7$	3534	24.79	$1.00e-6$	$7.22e-7$	$5.43e-7$	4754	33.12
gka5e	201	$8.25e-7$	$9.91e-7$	$3.67e-7$	3153	22.09	$9.97e-7$	$8.91e-7$	$2.95e-7$	4157	28.95
gka4f	501	$9.64e-7$	$9.93e-7$	$1.08e-6$	5153	4 : 14	$9.98e-7$	$7.23e-7$	$4.24e-7$	7529	6 : 08
gka5f	501	$7.66e-7$	$9.97e-7$	$8.13e-7$	4660	3 : 48	$9.99e-7$	$9.66e-7$	$9.37e-7$	7023	5 : 41

For a given edge subset $T \subseteq E$, we can formulate the problem as

$$\begin{aligned} \min \quad & \left\langle \frac{1}{2k}\operatorname{Diag}(We) + \frac{k-1}{2k}W, X \right\rangle \\ \text{s.t.} \quad & X_{ij} \geq \frac{-1}{k-1}, \quad \forall (i,j) \in E \setminus T, \\ & X_{ij} = \frac{-1}{k-1}, \quad \forall (i,j) \in T, \\ & \operatorname{diag}(X) = e, \quad X \succeq 0, \end{aligned} \tag{44}$$

where $e \in \mathscr{R}^r$ is the vector of all ones, $\operatorname{Diag}(x)$ is a diagonal matrix with x as the diagonal entries, and $\operatorname{diag}(X)$ is a vector of the diagonal entries of matrix X. The constraints $X_{ij} = -1/(k-1)$ were replaced by $X_{ij}/\sqrt{2} = -1/(\sqrt{2}(k-1))$ and $X_{ij} \geq -1/(k-1)$ by $X_{ij}/\sqrt{2} \geq -1/(\sqrt{2}(k-1))$. So we have $\mathscr{A}\mathscr{A}^* = I$. We set h to 50 for updating the penalty parameter μ.

We did not run SDPAD on our own computer on the problem "fap36" and the results presented here were taken from Table 1 in [16]. From Table 1, it can be observed that Algorithm 1 is often faster than SDPAD for achieving a duality gap of the same order. The infeasibility achieved by Algorithm 1 is satisfactory as well.

5.2. Binary Integer Quadratic Programs Problem. In this subsection, we present numerical results of Algorithm 1 and SDPAD on binary integer quadratic (BIQ) problems [41] through SDP relaxations which have the following form:

$$\begin{aligned} \min \quad & \left\langle \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix}, X \right\rangle \\ \text{s.t.} \quad & X_{ii} - X_{n,i} = 0, \quad i = 1, \ldots, n-1, \\ & X_{nn} = 1, \quad X \succeq 0, \end{aligned} \tag{45}$$

where $Q \in \mathscr{R}^{(n-1)\times(n-1)}$. The constraints $X_{ii} - X_{n,i} = 0$ were replaced by $\sqrt{2/3}(X_{ii} - X_{n,i}) = 0$ and the matrix Q was scaled by its Frobenious norm. We set h to 50 for updating the penalty parameter μ.

Table 2 lists the results of Algorithm 1 and SDPAD on the BIQ problems. By comparing the results in Table 2, we can conclude that Algorithm 1 applied to BIQ problems is superior to SDPAD in terms of CPU time and number of iterations. In addition, the accuracy of the approximate optimal solutions computed by Algorithm 1 is as good as that obtained by SDPAD.

Figure 1 shows the performance profiles [42] of Algorithm 1 and SDPAD for the number of iterations, Figure 1(a), and CPU time, Figure 1(b). We observe that Algorithm 1 is better than SDPAD in terms of number of iterations and CPU time.

6. Conclusion

In this paper, a novel linearizing alternating direction augmented Lagrangian approach is proposed for solving semidefinite programs (SDP). The algorithm updates the dual variables without solving any system of linear equations. Moreover, all the variables are updated in closed forms. Preliminary numerical results show the efficiency of the proposed algorithm. However, there are still some unsettled issues for implementation. For example, efficient strategies to update penalty parameter μ and choose step size λ_k deserve more work for applications of the algorithm.

Conflict of Interests

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

Acknowledgments

The authors would like to express their sincere thanks to the anonymous referees and editors for their careful review of the paper and the valuable comments, which have greatly improved the earlier version of this paper. This paper is supported by the Science and Technology Key Project of Education Department of Henan Province (14A110024).

References

[1] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM Journal on Optimization*, vol. 5, no. 1, pp. 13–51, 1995.

[2] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, Pa, USA, 1994.

[3] T. Fujie and M. Kojima, "Semidefinite programming relaxation for nonconvex quadratic programs," *Journal of Global Optimization*, vol. 10, no. 4, pp. 367–380, 1997.

[4] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, "Solving Euclidean distance matrix completion problems via semidefinite programming," *Computational Optimization and Applications*, vol. 12, no. 1-3, pp. 13–30, 1999.

[5] M. Anjos and J. B. Lasserre, *Handbook on Semidefinite, Conic and Polynomial Optimization*, Springer, New York, NY, USA, 2012.

[6] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Studies in Applied Mathematics, SIAM, Philadelphia, Pa, USA, 1994.

[7] R. D. C. Monteiro, "Primal-dual path-following algorithms for semidefinite programming," *SIAM Journal on Optimization*, vol. 7, no. 3, pp. 663–678, 1997.

[8] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, "An interior-point method for semidefinite programming," *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 342–361, 1996.

[9] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[10] K. Toh and M. Kojima, "Solving some large scale semidefinite programs via the conjugate residual method," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 669–691, 2002.

[11] K. Toh, "Solving large scale semidefinite programs via an iterative solver on the augmented systems," *SIAM Journal on Optimization*, vol. 14, no. 3, pp. 670–698, 2003.

[12] J. Povh, F. Rendl, and A. Wiegele, "A boundary point method to solve semidefinite programs," *Computing*, vol. 78, no. 3, pp. 277–286, 2006.

[13] J. Malick, J. Povh, F. Rendl, and A. Wiegele, "Regularization methods for semidefinite programming," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 336–356, 2009.

[14] A. Huang and C. Xu, "A trust region method for solving semidefinite programs," *Computational Optimization and Applications*, vol. 55, no. 1, pp. 49–71, 2013.

[15] X. Zhao, D. Sun, and K. Toh, "A Newton-CG augmented Lagrangian method for semidefinite programming," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1737–1765, 2010.

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

[17] Z. Wen, D. Goldfarb, S. Ma, and K. Scheinberg, "Row by row methods for semidefinite programming," Tech. Rep., Department of IEOR, Columbia University, 2009.

[18] Y. Xu, W. Sun, and L. Qi, "A feasible direction method for the semidefinite program with box constraints," *Applied Mathematics Letters*, vol. 24, no. 11, pp. 1874–1881, 2011.

[19] V. G. Zhadan and A. A. Orlov, "Dual interior point methods for linear semidefinite programming problems," *Computational Mathematics and Mathematical Physics*, vol. 51, no. 12, pp. 2031–2051, 2011.

[20] H. Lin, "An inexact spectral bundle method for convex quadratic semidefinite programming," *Computational Optimization and Applications*, vol. 53, no. 1, pp. 45–89, 2011.

[21] J. Sun and S. Zhang, "A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs," *European Journal of Operational Research*, vol. 207, no. 3, pp. 1210–1220, 2010.

[22] A. Auslender and C. H. Ramírez, "Penalty and barrier methods for convex semidefinite programming," *Mathematical Methods of Operations Research*, vol. 63, no. 2, pp. 195–219, 2006.

[23] S. Zhang, J. Ang, and J. Sun, "An alternating direction method for solving convex nonlinear semidefinite programming problems," *Optimization*, vol. 62, no. 4, pp. 527–543, 2013.

[24] X. X. Huang, X. Q. Yang, and K. L. Teo, "Lower-order penalization approach to nonlinear semidefinite programming," *Journal of Optimization Theory and Applications*, vol. 132, no. 1, pp. 1–20, 2007.

[25] M. Aroztegui, J. Herskovits, J. R. Roche, and E. Bazn, "A feasible direction interior point algorithm for nonlinear semidefinite programming," *Structural and Multidisciplinary Optimization*, vol. 132, no. 1, pp. 1–20, 2007.

[26] L. Yang and B. Yu, "A homotopy method for nonlinear semidefinite programming," *Computational Optimization and Applications*, vol. 56, no. 1, pp. 81–96, 2013.

[27] C. Kanzow, C. Nagel, H. Kato, and M. Fukushima, "Successive linearization methods for nonlinear semidefinite programs," *Computational Optimization and Applications*, vol. 31, no. 3, pp. 251–273, 2005.

[28] H. Yamashita, H. Yabe, and K. Harada, "A primal-dual interior point method for nonlinear semidefinite programming," *Mathematical Programming*, vol. 135, no. 1-2, pp. 89–121, 2012.

[29] Z. Lu, A. Nemirovski, and R. D. C. Monteiro, "Large-scale semidefinite programming via a saddle point mirror-prox algorithm," *Mathematical Programming*, vol. 109, no. 2-3, pp. 211–237, 2006.

[30] R. D. C. Monteiro, C. Ortiz, and B. F. Svaiter, "A first-order block-decomposition method for solving two-easy-block structured semidefinite programs," *Mathematical Programming Computation*, vol. 6, no. 2, pp. 103–150, 2014.

[31] J. H. Chen, T. B. Yang, and S. H. Zhu, "Efficient low-rank stochastic gradient descent methods for solving semidefinite programs," in *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pp. 122–130, 2014.

[32] G. Q. Wang and Y. Q. Bai, "A new primal-dual path-following interior-point algorithm for semidefinite optimization," *Journal of Mathematical Analysis and Applications*, vol. 353, no. 1, pp. 339–349, 2009.

[33] Z. Yu, "Solving semidefinite programming problems via alternating direction methods," *Journal of Computational and Applied Mathematics*, vol. 193, no. 2, pp. 437–445, 2006.

[34] M. H. Xu and T. Wu, "A class of linearized proximal alternating direction methods," *Journal of Optimization Theory and Applications*, vol. 151, no. 2, pp. 321–337, 2011.

[35] P. Tseng, "Alternating projection-proximal methods for convex programming and variational inequalities," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 951–965, 1997.

[36] K. C. Kiwiel, C. H. Rosa, and A. Ruszczynski, "Proximal decomposition via alternating linearization," *SIAM Journal on Optimization*, vol. 9, no. 3, pp. 668–689, 1999.

[37] B. S. He, H. Yang, and S. L. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *Journal of Optimization Theory and Applications*, vol. 106, no. 2, pp. 337–356, 2000.

[38] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[39] A. Eisenblätter and M. A. Grötschel, "Frequency planning and ramifications of coloring," *Discussiones Mathematicae. Graph Theory*, vol. 22, no. 1, pp. 51–88, 2002.

[40] S. Burer, R. D. C. Monteiro, and Y. Zhang, "A computational study of a gradient-based log-barrier algorithm for a class of large-scale SDPs," *Mathematical Programming*, vol. 95, no. 2, pp. 359–379, 2003.

[41] A. Wiegele, "Biq Mac Library—a collection of Max-Cut and quadratic 0-1 programming instances of medium size," Tech. Rep., 2007.

[42] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.