Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 819607, 13 pages
doi:10.1155/2012/819607

Research Article

A Globally Convergent Filter-Type Trust Region Method for Semidefinite Programming

Aiqun Huang[1, 2] and Chengxian Xu[1]

[1] *School of Mathematics and Statistics, Xi'an Jiaotong University, Shaanxi, Xi'an 710049, China*
[2] *School of Mathematics and Statistics, Huazhong University of Science and Technology, Hubei, Wuhan 430074, China*

Correspondence should be addressed to Aiqun Huang, huangaiqun776@yahoo.com.cn

Received 17 May 2012; Accepted 29 June 2012

Academic Editor: Soohee Han

When using interior methods for solving semidefinite programming (SDP), one needs to solve a system of linear equations at each iteration. For problems of large size, solving the system of linear equations can be very expensive. In this paper, based on a semismooth equation reformulation using Fischer's function, we propose a filter method with trust region for solving large-scale SDP problems. At each iteration we perform a number of conjugate gradient iterations, but do not need to solve a system of linear equations. Under mild assumptions, the convergence of this algorithm is established. Numerical examples are given to illustrate the convergence results obtained.

1. Introduction

Semidefinite programming (SDP) is convex programming over positive semidefinite matrices. For early application, SDP has been widely used in control theory and combinatorial optimization (see, e.g., [1–3]). Since some algorithms for linear optimization can be extended to many general SDP problems, that aroused much interest in SDP. In the past decade, many algorithms have been proposed for solving SDP, including interior-point methods (IPMs) [4–7], augmented methods [8–10], new Newton-type methods [11], modified barrier methods [12], and regularization approaches [13].

For small and medium sized SDP problems, IPMs are generally efficient. But for large-scale SDP problems, IPMs become very slow. In order to improve this shortcoming, [9, 14] proposed inexact IPMs using an iterative solver to compute a search direction at each iteration. More recently, [13] applied regularization approaches to solve SDP problems. All of these methods are first-order based on a gradient, or inexact second-order based on an approximation of Hessian matrix methods [15].

In this paper, we will extend filter-trust-region methods for solving linear (or nonlinear) programming [16] to large-scale SDP problems and use Lipschitz continuity. Furthermore, the accuracy of this method is controlled by a forcing parameter. It is shown that, under mild assumptions, this algorithm is convergent.

The paper is organized as follows. Some preliminaries are introduced in Section 2. In Section 3, we propose a filter-trust-region method for solving SDP problems, and we study the convergence of this method in Section 4. In Section 5, some numerical examples are presented to demonstrate the convergence results obtained in this paper. Finally, we give some conclusions in Section 6.

In this paper, we use the following common notation for SDP problems: $\mathcal{X}^n$ and $\mathcal{R}^m$ denote the space of $n \times n$ real symmetric matrices and the space of vectors with m dimensions, respectively; $X \succeq 0 (X \succ 0)$ denotes that $X \in \mathcal{X}^n$ is positive semidefinite (positive definite), and $X \preceq 0 (X \prec 0)$ is used to indicate that $X \in \mathcal{X}^n$ is negative semidefinite (negative definite). A superscript T represents transposes of matrices or vectors. For $X, Y \in \mathcal{X}^n$, the standard scalar product on the space of $\mathcal{X}^n$ is defined by

$$\langle X, Y \rangle := X \bullet Y = \operatorname{trace}(XY) = \sum_{i,j=1}^{n} X_{i,j} Y_{i,j}. \tag{1.1}$$

If $X \in \mathcal{X}^n$ and $x \in \mathcal{R}^m$, we denote that $\|X\|_F$ is the Frobenius norm of X, that is, $\|X\|_F = \sqrt{\langle X, X \rangle} = \sqrt{\sum_{i,j=1}^{n} X_{i,j}^2}$ and $\|x\|_2$ is the 2-norm of x, that is, $\|x\|_2 = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{m} x_i^2}$, respectively. Let X be a $p \times q$ matrix. Then we denote by $\operatorname{Vec}(X)$ a pq vector made of columns of X stacked one by one, and the operator $\operatorname{Mat}(\cdot)$ is the inverse of $\operatorname{Vec}(\cdot)$, that is, $\operatorname{Mat}(\operatorname{Vec}(X)) = X$. We also denote that I is identity matrix.

2. Preliminaries

We consider a SDP problem of the form

$$\begin{aligned} &\min && C \bullet X \\ &\text{subject to} && \mathcal{A}(X) = b, \quad X \succeq 0, \end{aligned} \tag{2.1}$$

where $C \in \mathcal{X}^n$, $A^{(i)} \in \mathcal{X}^n$, $i = 1, 2, \ldots, m$, and $b = (b_1, b_2, \ldots, b_m)^T \in \mathcal{R}^m$ are given dates; $\mathcal{A}$ is a linear map from $\mathcal{X}^n$ to $\mathcal{R}^m$ given by

$$\mathcal{A}(X) := \begin{bmatrix} A^{(1)} \bullet X \\ A^{(2)} \bullet X \\ \vdots \\ A^{(m)} \bullet X \end{bmatrix}, \quad X \in \mathcal{X}^n. \tag{2.2}$$

The dual to the problem (2.1) is given by

$$\begin{aligned} &\max && b^T y \\ &\text{subject to} && \mathcal{A}^*(y) + S = C, \quad S \succeq 0, \end{aligned} \tag{2.3}$$

where $\mathcal{A}^*$ is an adjoint operator of $\mathcal{A} : \mathcal{R}^m \to \mathcal{X}^n$ given by

$$\mathcal{A}^*(y) = \sum_{i=1}^{m} y_i A^{(i)}, \quad y \in \mathcal{R}^m. \tag{2.4}$$

Obviously, $X \in \mathcal{X}^n$ and $(y, S) \in \mathcal{R}^m \times \mathcal{X}^n$ are the primal and dual variables, respectively.

It is easily verified that the SDP problem (2.1) is convex. When (2.1) and (2.3) have strictly feasible points, then strong duality holds, see [5, 12]. In this case, a point (X, y, S) is optimal for SDP problems (2.1) and (2.3) if and only if

$$\mathcal{A}(X) = b, \qquad \mathcal{A}^*(y) + S = C, \qquad X \succeq 0, \quad S \succeq 0, \quad \langle X, S \rangle = 0. \tag{2.5}$$

In the sense that (X, y, S) solves SDP problems (2.1) and (2.3) if and only if (X, y, S) solves (2.5) when both SDP problems (2.1) and (2.3) have strictly feasible points.

We now introduce some lemmas which will be used in the sequel.

Lemma 2.1 (see [17]). *Let $A, B \in \mathcal{X}^n$ and let $A \succeq 0$, $B \succeq 0$. Then $\langle A, B \rangle = 0$ if and only if $AB = 0$.*

For $X, S \in \mathcal{X}^n$, we define a mapping $\phi : \mathcal{X}^n \times \mathcal{X}^n \to \mathcal{X}^n$ given by

$$\phi(X, S) := X + S - \sqrt{X^2 + S^2}, \tag{2.6}$$

which is attributed by Fischer to Burmeister (see [18, 19]). This function is nondifferentiable and has a basic property.

Lemma 2.2 (see [20, Lemma 6.1]). *Let ϕ be the Fischer-Burmeister function defined in* (2.6)*. Then*

$$\phi(X, S) = 0 \Longleftrightarrow X \succeq 0, \quad S \succeq 0, \qquad XS = 0. \tag{2.7}$$

In addition, for $\tau > 0$ and $X, S \in \mathcal{X}^n$, we define a mapping $\phi_\tau : \mathcal{X}^n \times \mathcal{X}^n \to \mathcal{X}^n$ by

$$\phi_\tau(X, S) := X + S - \sqrt{X^2 + S^2 + 2\tau^2 I}, \tag{2.8}$$

which is differentiable and has following results.

Lemma 2.3 (see [11, Proposition 2.3]). *Let $\tau > 0$ be any positive number and let ϕ_τ be defined by* (2.8)*. Then*

$$\phi_\tau(X, S) = 0 \Longleftrightarrow X \succ 0, \quad S \succ 0, \ XS = \tau^2 I. \tag{2.9}$$

Lemma 2.4. *Let $\tau > 0$ be any positive number, and let ϕ_τ be defined by* (2.8). *If $\tau \to 0$, we would have*

$$\phi_\tau(X,S) = 0 \Longleftrightarrow X \succeq 0, \quad S \succeq 0, \ XS = 0. \tag{2.10}$$

Proof. The proof can be obtained from Lemmas 2.2 and 2.3. □

Lemma 2.5 (see [20, pages 170–171]). *For any $C \succ 0$, define the linear operator L_C by*

$$L_C[X] := CX + XC, \quad X \in \mathcal{X}^n. \tag{2.11}$$

Then L_C is strictly monotone and so has an inverse L_C^{-1}.

Lemma 2.6 (see [21, Lemma 2]). *Let $X, S, U, V \in \mathcal{X}^n$, and let ϕ_τ be defined by* (2.8). *For any $\tau > 0$, we have that ϕ_τ is Fréchet-differentiable and*

$$\nabla\phi_\tau(X,S)(U,V) = U + V - L_C^{-1}[XU + UX + SV + VS], \tag{2.12}$$

where $C := \sqrt{X^2 + S^2 + 2\tau^2 I}$.

Lemma 2.7 (see [22, Corollary 2.7]). *Let F be a map from $\mathcal{X}^n$ to $\mathcal{X}^n$. If F is locally Lipschitzian on $\mathcal{X}^n$, then F is almost everywhere Fréchet-differentiable on $\mathcal{X}^n$.*

3. The Algorithm

In this section, we will present a filter-trust-region method for solving SDP problems (2.1) and (2.3). Firstly, for a parameter $\tau > 0$, we construct a function:

$$H_\tau(X,y,S) := \begin{pmatrix} \tau \\ \mathcal{A}(X) - b \\ \mathcal{A}^*(y) + S - C \\ X + S - \sqrt{X^2 + S^2 + 2\tau^2 I} \end{pmatrix}, \tag{3.1}$$

where $(X,y,S) \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$.

According to Lemmas 2.1, 2.3 and 2.4, the following theorem is obvious.

Theorem 3.1. *Let $\tau > 0$ and let $H_\tau(X,y,S)$ be defined by* (3.1). *If SDP problems* (2.1) *and* (2.3) *have strictly feasible points, then*

$$H_\tau(X_*, y_*, S_*) = 0 \Longrightarrow (X_*, y_*, S_*) \ \text{solves} \ (2.5). \tag{3.2}$$

In what follows, we will study properties of the function $H_\tau(X,y,S)$. For simplicity, in the remaining sections of this paper, we denote $Z := (X,y,S)$, $Z_k := (X_k, y_k, S_k)$ and $\Delta Z := (\Delta X, \Delta y, \Delta S)$.

Theorem 3.2. *Let $H_\tau(Z)$ be defined by* (3.1)*. For any $Z, \Delta Z \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$ and $\tau > 0$, then $H_\tau(Z)$ is Fréchet-differentiable and*

$$\nabla H_\tau(Z)(\Delta Z) = \begin{pmatrix} \Delta\tau \\ \mathcal{A}(\Delta X) - b \\ \mathcal{A}^*(\Delta y) + \Delta S - C \\ \Delta X + \Delta S - L_C^{-1}[X\Delta X + \Delta X X + S\Delta S + \Delta S S] \end{pmatrix}, \tag{3.3}$$

where $\Delta\tau > 0$ and $C := \sqrt{X^2 + S^2 + 2\tau^2 I}$.

Proof. For any $Z \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$, since $\mathcal{A}(X) - b$ and $\mathcal{A}^*(y) + S - C$ are linear functions and continuous differentiable, it follows that they are also locally Lipschitz continuous. Then, from Lemma 2.7, $\mathcal{A}(X) - b$ and $\mathcal{A}^*(y) + S - C$ are Fréchet-differentiable. Furthermore, $X + S - \sqrt{X^2 + S^2 + 2\tau^2 I}$ is Fréchet-differentiable from Lemma 2.6. Thus, $H_\tau(Z)$ is Fréchet-differentiable and has the form of (3.3). We complete the proof. □

We endow the variable Z with the following norm:

$$\|Z\| = \|(X, y, S)\| := \left(\|X\|_F^2 + \|y\|_2^2 + \|S\|_F^2\right)^{1/2}. \tag{3.4}$$

In addition, we set

$$h(Z) = (h_1(Z), h_2(Z), h_3(Z), h_4(Z))^T, \tag{3.5}$$

where

$$\begin{gathered} h_1(Z) = \|\mathcal{A}(X) - b\|_2, \\ h_2(Z) = \|\mathcal{A}^*(y) + S - C\|_F, \\ h_3(Z) = \left\|X + S - \sqrt{X^2 + S^2 + 2\tau^2 I}\right\|_F \\ h_4(Z) = |\tau|. \end{gathered} \tag{3.6}$$

We also define the function $H_\tau(Z)$ and the vector $h(Z)$ with the following norm:

$$\begin{aligned} \|H_\tau(Z)\| = \|h(Z)\| &= \left(\sum_{i=1}^4 h_i(Z)^2\right)^{1/2} \\ &= \left(\|\mathcal{A}(X) - b\|_2^2 + \|\mathcal{A}^*(y) + S - C\|_F^2 + \left\|X + S - \sqrt{X^2 + S^2 + 2\tau^2 I}\right\|_F^2 + \tau^2\right)^{1/2}. \end{aligned} \tag{3.7}$$

Now, for any $\tau > 0$, we define the merit function $\Psi : \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n \to \mathcal{R}$ by

$$\Psi_\tau(Z) := \frac{1}{2}\|H_\tau(Z)\|^2. \tag{3.8}$$

Lemma 3.3. *For any $\tau > 0$ and $Z \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$, if X and S are nonsingular, then $\Psi_\tau(Z)$ is locally Lipschitz continuous and twice Fréchet-differentiable at every $Z \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$.*

Proof. For any $\tau > 0$, since $\Psi_\tau(Z)$ is convex and continuously differentiable, it follows that $\Psi_\tau(Z)$ is also locally Lipschitz continuous.

In addition, for any $Z \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$, from [20, pages 173–175], $h_3(Z)^2$ is twice Fréchet-differentiable. Furthermore, $h_1(Z)^2$, $h_2(Z)^2$, and $h_4(Z)^2$ are continuous at every $Z \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$ when $\tau > 0$, which, together with Lemma 2.7, $\Psi_\tau(Z)$ is twice Fréchet-differentiable. The proof is completed. □

Lemma 3.4. *Let $H_\tau(Z)$ and $\Psi_\tau(Z)$ be defined by* (3.1) *and* (3.8), *respectively. For any $\tau > 0$, we have*

$$\Psi_\tau(Z) = 0 \Longleftrightarrow H_\tau(Z) = 0. \tag{3.9}$$

Proof. The proof can be immediately obtained from the definition of $H_\tau(Z)$ and $\Psi_\tau(Z)$. □

We follow the classical method for solving $\Psi_\tau(Z) = 0$, which consists some norm of the residual. For any $\tau > 0$, we consider

$$\min \Psi_\tau(Z), \tag{3.10}$$

where $Z \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$. Thus, for any $\tau > 0$, we want to find a minimizer Z^* of $\Psi_\tau(Z)$. Furthermore, if $\Psi_\tau(Z_*) = 0$, then Z_* is also a solution of $H_\tau(Z)$.

In order to state our method for solving (3.10), we consider using a filter mechanism to accept a new point. Just as [16, pages 19–20], the notation of filter is based on that of dominance.

Definition 3.5. For any $\tau > 0$ and any $Z_1, Z_2 \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$, a point Z_1 dominates a point Z_2 if and only if

$$h_i(Z_1) \le h_i(Z_2) \quad \forall i = 1, 2, 3, 4. \tag{3.11}$$

Thus, if iterate Z_1 dominates iterate Z_2, the latter is of no real interest to us since Z_1 is at least as good as Z_2 for each of the components of $h(Z)$. All we need to do is remember iterates that are no dominated by other iterates by using a structure called a filter.

Definition 3.6. Let $F^{(k)}$ be a set of 4-tuples of the following form:

$$(h_1(Z_k), h_2(Z_k), h_3(Z_k), h_4(Z_k)). \tag{3.12}$$

We define $F^{(k)}$ as a filter if $h(Z_k)$ and $h(Z_l)$ belong to $F^{(k)}$, when $k \neq l$, then

$$h_i(Z_k) < h_i(Z_l) \quad \text{for at least one } i \in \{1,2,3,4\}. \tag{3.13}$$

Definition 3.7. A new point Z_k^+ is acceptable for the filter $F^{(k)}$ if and only if

$$\forall\, h(Z_k) \in F^{(k)} \exists i \in \{1,2,3,4\} : h_i\left(Z_k^+\right) \leq h_i(Z_k) - \alpha \|h(Z_k)\|, \tag{3.14}$$

where $\alpha \in (0, 1/\sqrt{4})$ is a small constant.

Now, we formally present our trust region algorithm by using filter techniques.

Algorithm 3.8. The Filter-Trust-Region Algorithm

Step 0. Choose an initial point $Z_0 = (X_0, y_0, S_0) \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$, $\varepsilon > 0$, $0 < \alpha < 1/\sqrt{4}$ and $\tau_0 = \langle X_0, S_0 \rangle / 2n$. The constants η_1, η_2, η_3, μ, γ, γ_1, and γ_2 are also given and satisfy

$$0 < \eta_1 \leq \eta_2 \leq \eta_3 < 1, \quad 0 < \mu < 1, \quad 0 < \gamma < \gamma_1 < 1 \leq \gamma_2. \tag{3.15}$$

Compute $\Psi_{\tau_0}(Z_0)$, set $\Delta_0 = 0.5\|\nabla \Psi_{\tau_0}(Z_0)\|$, $k = 0$ and only $(\mu, -\infty, \mu, \mu)$ in the filter $F^{(0)}$.

Step 1. If $\nabla \Psi_{\tau_k}(Z_k) < \varepsilon$, stop.

Step 2. Compute ΔZ_k by solving the following problem:

$$\begin{aligned} \min \quad & \varphi_k(\Delta Z) \\ \text{s.t.} \quad & \|\Delta Z\| \leq \Delta_k, \end{aligned} \tag{3.16}$$

where

$$\begin{aligned} \varphi_k(\Delta Z) &= \frac{1}{2}\|H_{\tau_k}(Z_k) + \nabla H_{\tau_k}(Z_k)(\Delta Z)\|^2 \\ &= \Psi_{\tau_k}(Z_k) + H_{\tau_k}(Z_k)^T \nabla H_{\tau_k}(Z_k)(\Delta Z) \\ &\quad + \frac{1}{2}(\Delta Z)^T \nabla H_{\tau_k}(Z_k)^T \nabla H_{\tau_k}(Z_k)(\Delta Z). \end{aligned} \tag{3.17}$$

If $\|\Delta Z_k\| < \varepsilon$, stop.

Otherwise, computer the trial point $Z_k^+ = Z_k + \Delta Z_k$.

Step 3. Compute $\Psi_{\tau_k}(Z_k^+)$ and define the following ratio:

$$r_k = \frac{\Psi_{\tau_k}(Z_k) - \Psi_{\tau_k}\left(Z_k^+\right)}{\varphi_k(0) - \varphi_k(\Delta Z_k)}. \tag{3.18}$$

Step 4. If $r_k \geq \eta_1$, set $Z_{k+1} = Z_k^+$.

If $r_k < \eta_1$ but Z_k^+ satisfies (3.14), then add $h(Z_k^+)$ to the filter $F^{(k)}$ and remove all points from $F^{(k)}$ dominated by $h(Z_k^+)$. At the same time, set $Z_{k+1} = Z_k^+$.

Else, set $Z_{k+1} = Z_k$.

Step 5. Update τ_k by choosing

$$\tau_{k+1} \in \begin{cases} \gamma\tau_k & \text{if } Z_{k+1} = Z_k^+, \\ \tau_k & \text{else;} \end{cases} \tag{3.19}$$

and update trust-region radius Δ_k by choosing

$$\Delta_{k+1} := \begin{cases} \gamma\Delta_k, & \text{if } r_k < \eta_1, \\ \gamma_1\Delta_k, & \text{if } r_k \in [\eta_1, \eta_2], \\ \Delta_k, & \text{if } r_k \in (\eta_2, \eta_3), \\ \gamma_2\Delta_k, & \text{if } r_k \geq \eta_3. \end{cases} \tag{3.20}$$

Step 6. Set $k := k + 1$ and go to Step 1.

Remark 3.9. Algorithm 3.8 can be started any $\tau > 0$. In fact, in order to increase the convergent speed greatly, we always choose $\tau_0 = \langle X_0, S_0\rangle / 2n$. In addition, in this algorithm, we fix τ at first, then search Z for $\Psi_\tau(Z) = 0$ to update Z. At last we update τ and repeat.

The following lemma is a generalized case of Proposition 3.1 in [23].

Lemma 3.10. *Algorithm 3.8 is well defined, that is, the inner iteration (Step 2) terminates finitely.*

For the purpose of our analysis, in the sequence of points generated by Algorithm 3.8, we denote $\mathcal{A} = \{k \mid r_k \geq \eta_1\}$, $\mathcal{B} = \{k \mid h(Z_k^+)$ is added to the filter $F^{(k)}\}$, and $\mathcal{C} = \{k \mid Z_{k+1} = Z_k + \Delta Z_k\}$. It is clear that, $\mathcal{C} = \mathcal{A} \bigcup \mathcal{B}$.

Remark 3.11. Lemma 3.3 implies that there exists a constant $0 < M \leq 1$ such that

$$h_i(Z_k) \leq M, \quad \left\|\nabla^2 h_i(Z_k)\right\| \leq M, \quad \left\|\nabla^2 \varphi_k(\Delta Z)\right\| \leq M \tag{3.21}$$

for all $k \in \mathcal{C}$ and $i \in \{1,2,3,4\}$. The second of above inequalities ensures that the constant $0 < M \leq 1$ can also be chosen such that

$$\left\|\nabla^2 \Psi_{\tau_k}(Z_k)\right\| \leq M. \tag{3.22}$$

4. Convergence of Analysis

In this section, we present a proof of global convergence of Algorithm 3.8. First, we make the following assumptions.

Some lemmas will be presented to be used in the subsequent analysis.

(S1) $\varphi_k(0) - \varphi_k(\Delta Z_k) \geq 1/2\|\nabla\Psi_{\tau_k}(Z_k)\| \min\{\Delta_k, \|\nabla\Psi_{\tau_k}(Z_k)\| / \|\nabla H_{\tau_k}(Z_k)^T \nabla H_{\tau_k}(Z_k)\|\}$, where ΔZ_k is a solution of (3.16).

(S2) The iterations generated by Algorithm 3.8 remain in a close, bounded domain.

Lemma 4.1 (see [24]). *Let assumptions (S1) and (S2) hold. If there exists $l_0 > 0$ such that $\|\nabla\Psi_{\tau_k}(Z_k)\| \geq l_0 > 0$ for all k; then there exists $l_1 > 0$ such that $\Delta_k \geq l_1$.*

Lemma 4.2. *Let $\{\tau_k\}$ be the infinite sequence generated by the Algorithm 3.8. Then*

$$\lim_{k\to\infty} \tau_k = 0. \tag{4.1}$$

Proof. Since $|\mathcal{C}| = |\mathcal{A}| = +\infty$, from Steps 4 and 5 of Algorithm 3.8, $\tau_{k+1} = \gamma\tau_k$ and $0 < \gamma < \tau_0 < 1$. Therefore, $\tau_{k+1} = \gamma^k\tau_0$. Moreover,

$$\lim_{k\to\infty} \tau_k = \lim_{k\to\infty} \gamma^k\tau_0 = 0 \tag{4.2}$$

for $0 < \gamma < \tau_0 < 1$, which completes the proof. □

Theorem 4.3. *Let $|\mathcal{C}| < +\infty$, assumptions (S1) and (S2) hold. Then there exists $k \in \mathcal{C}$ such that*

$$\nabla\Psi_{\tau_k}(Z_k) = 0. \tag{4.3}$$

Proof. Suppose that $\nabla\Psi_{\tau_k}(Z_k) \neq 0$ for all $k \in \mathcal{C}$. Then there exists $\omega_0 > 0$ such that

$$\|\nabla\Psi_{\tau_k}(Z_k)\| \geq \omega_0 > 0. \tag{4.4}$$

From Lemma 4.1, there exists $\omega_1 > 0$ such that

$$\Delta_k \geq \omega_1 > 0. \tag{4.5}$$

On the other hand, $|\mathcal{C}| < +\infty$, let N be the last successful iteration, then $Z_{N+1} = Z_{N+2} = \cdots = Z_{N+j}$ $(j \geq 1)$ are unsuccessful iterations. From Steps 4 and 5 of Algorithm 3.8, $r_{N+j} < \eta_1$, for sufficiently large N, we have

$$\lim_{N\to\infty} \Delta_{N+j} = 0, \tag{4.6}$$

which contradicts (4.5). The proof is completed. □

We now consider what happens if the set $\mathcal{A}$ is infinite in the course of Algorithm 3.8.

Theorem 4.4. *Suppose that $|\mathcal{C}| = |\mathcal{A}| = +\infty$, assumptions (S1) and (S2) hold. For any $\tau > 0$ and $Z \in \mathcal{K}^n \times \mathcal{R}^m \times \mathcal{K}^n$, if X and S are nonsingular, then each accumulation point of the infinite sequences generated by Algorithm 3.8 is a stationary point of $\Psi_\tau(Z)$.*

Proof. The proof is by contradiction. Suppose that $\{Z_k\}$ is an infinite sequence generated by Algorithm 3.8, and any accumulation point of $\{Z_k\}$ is not a stationary point of $\Psi_\tau(Z)$. Suppose furthermore that Z_* and τ_* are the accumulation points of $\{Z_k\}$ and $\{\tau_k\}$, respectively. Since Z_* is not a stationary point of $\Psi_\tau(Z)$, then

$$\nabla\Psi_{\tau_*}(Z_*) \neq 0 \tag{4.7}$$

and there exists $\epsilon_0 > 0$ such that

$$\|\nabla \Psi_{\tau_*}(Z_*)\| > \epsilon_0 > 0. \tag{4.8}$$

For some $\epsilon_* > 0$, let $\mathcal{N}(Z_*, \epsilon_*)$ be a neighborhood of Z_*. From (4.8), there exists $\{Z_k\}_{k \in K} \in \mathcal{N}(Z_*, \epsilon_*)$ such that

$$\|\nabla \Psi_{\tau_k}(Z_k)\| \geq \epsilon_0 > 0, \tag{4.9}$$

where $K \subseteq \mathcal{A}$.

For $m, m + \nu \in K$, because

$$\Psi_{\tau_k}(Z_k) - \Psi_{\tau_{k+1}}(Z_{k+1}) \geq \eta_1 \left[\varphi_k(0) - \varphi_k(\Delta Z_k)\right], \tag{4.10}$$

we obtain that

$$\begin{aligned}
\Psi_{\tau_m}(Z_m) - \Psi_{\tau_{m+\nu}}(Z_{m+\nu}) &= \sum_{i=m\in K}^{m+\nu} \left[\Psi_{\tau_i}(Z_i) - \Psi_{\tau_{i+1}}(Z_{i+1})\right] \\
&\geq \eta_1 \sum_{i=m\in K}^{m+\nu} \left[\varphi_k(0) - \varphi_k(\Delta Z_k)\right] \\
&\geq \eta_1 \sum_{i=m\in K}^{m+\nu} \frac{1}{2} \|\nabla \Psi_{\tau_k}(Z_k)\| \min\left\{ \Delta_k, \frac{\|\nabla \Psi_{\tau_k}(Z_k)\|}{\left\|\nabla H_{\tau_k}(Z_k)^T \nabla H_{\tau_k}(Z_k)\right\|} \right\} \\
&\geq \eta_1 \sum_{i=m\in K}^{m+\nu} \frac{1}{2} \epsilon_0 \ \min\left\{ \Delta_k, \frac{\epsilon_0}{\left\|\nabla H_{\tau_k}(Z_k)^T \nabla H_{\tau_k}(Z_k)\right\|} \right\}.
\end{aligned} \tag{4.11}$$

From (4.10), we know that $\Psi_{\tau_k}(Z_k)$ is monotone decreasing and bounded below, which implies that $\Psi_{\tau_m}(Z_m) - \Psi_{\tau_{m+n}}(Z_{m+\nu}) \longrightarrow 0$ for $m \longrightarrow \infty, m \in K$. Thus,

$$\eta_1 \sum_{i=m\in K}^{m+\nu} \frac{1}{2} \epsilon_0 \ \min\left\{ \Delta_k, \frac{\epsilon_0}{\left\|\nabla H_{\tau_k}(Z_k)^T \nabla H_{\tau_k}(Z_k)\right\|} \right\} \longrightarrow 0. \tag{4.12}$$

As a result, we have

$$\lim_{k \to \infty, k \in K} \Delta_k = 0. \tag{4.13}$$

By the update rule of Δ_k, there exists an infinite subsequence $K^\star \subseteq K$, and we have that

$$r_i \leq \eta_1, \quad \lim_{i \to \infty} \Delta_i = 0, \quad i \in K^\star. \tag{4.14}$$

which contradicts $k \in K \subseteq \mathcal{A}$. This completes the proof. □

Table 1

n	m	F-time	F-it.	F-obj.	R-time	R-it.
300	20000	30	20	$3.2566e-12$	63	27
300	25000	72	23	$5.1697e-13$	127	29
400	30000	63	25	$0.2212e-12$	118	32
400	40000	152	31	$6.2008e-14$	202	46
500	30000	176	35	$8.5216e-16$	201	39
500	40000	108	41	$9.1535e-15$	198	52
600	20000	321	47	$8.9660e-17$	485	58
600	60000	298	38	$3.5722e-16$	345	56

In what follows, we investigate the case where the number of iterations added to the filter $F^{(k)}$ in the course of Algorithm 3.8 is infinite.

Theorem 4.5. *Suppose that* $|\mathcal{C}| = |\mathcal{B}| = +\infty$ *but* $|A| < +\infty$, *SDP problems* (2.1) *and* (2.3) *have strictly feasible points. Suppose furthermore that assumptions (S1) and (S2) hold. For any* $\tau > 0$ *and* $Z \in \mathcal{X}^n \times \mathcal{R}^m \times \mathcal{X}^n$, *if X and S are nonsingular, then*

$$\lim_{k\to\infty} \|H_\tau(Z_k)\| = \lim_{k\to\infty} \|\nabla\Psi_{\tau_k}(Z_k)\| = 0. \tag{4.15}$$

Proof. First let $\{\tau_k\}$ be the sequence generated by Algorithm 3.8. From Lemma 4.2, we have

$$\lim_{k\to\infty} \tau_k = 0, \tag{4.16}$$

which, together with assumption (S2), the desired result follows from [16, Lemma 3.1]. □

5. Numerical Experiments

In this section, we describe the results of some numerical experiments with the Algorithm 3.8 for the random sparse SDP considered in [13]. All programs are written in Matlab code and all computations are tested under Matlab 7.1 on Pentium 4.

In addition, in the computations, the following values are assigned to the parameters in the Algorithm: $\eta_1 = 0.1$, $\eta_2 = 0.5$, $\eta_3 = 0.8$, $\mu = 0.1$, $\gamma = 0.2$, $\gamma_1 = 0.5$, and $\gamma_2 = 2$. We also use the stopping criteria is being of $\varepsilon = 10^{-8}$.

In the following Table 1, the first two columns give the size of the matrix C and the dimension of the variable y. In the middle columns, "F-time" denotes the computing time (in seconds), "F-it." denotes the numbers iteration, and "F-obj." defines the value of $\Psi_{\tau_k}(Z_k)$ when our stopping criteria is satisfied. Some numerical results of [13] are shown in the last two columns.

As shown in Table 1, all test problems have been solved just few iterations compared with [13]. Furthermore, this algorithm is less sensitive to the size of SDP problems. Comparatively speaking, our method is attractive and suitable for solving large-scale SDP problems.

6. Conclusions

In this paper, we have proposed a filter-trust-region method for SDP problems. Such a method offers a trade-off between the accuracy of solving the subproblems and the amount of work for solving them. Furthermore, numerical results show that our algorithm is attractive for large-scale SDP problems.

Acknowledgments

The authors would like to thank Professor Florian Jarre for his advice and guidance, Thomas David and Li Luo for their grateful help, and also the referees for their helpful comments. This work is supported by National Natural Science Foundation of China 10971162.

References

[1] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 15 of *SIAM Studies in Applied Mathematics*, SIAM, Philadelphia, Pa, USA, 1994.

[2] M. X. Goemans, "Semidefinite programming in combinatorial optimization," *Mathematical Programming*, vol. 79, no. 1–3, pp. 143–161, 1997.

[3] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of Semidefinite Programming*, Kluwer Acadamic Publishers, Dordrecht, The Netherlands, 2000.

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

[5] F. Alizadeh, J. Pierre, A. Haeberly, and M. L. Overton, "A new primal-dual interiorpoint method for semidefinite programming," in *Proceedings of the 5th SIAM Conference on Applied Linear Algebra*, pp. 113–117, SIAM, Philadelphia, Pa, USA, 1994.

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

[7] 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.

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

[9] K.-C. 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, 2004.

[10] F. Jarre and F. Rendl, "An augmented primal-dual method for linear conic programs," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 808–823, 2008.

[11] C. Kanzow and C. Nagel, "Semidefinite programs: new search directions, smoothing-type methods, and numerical results," *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 1–23, 2002.

[12] M. Kočvara and M. Stingl, "On the solution of large-scale SDP problems by the modified barrier method using iterative solvers," *Mathematical Programming*, vol. 109, no. 2-3, pp. 413–444, 2007.

[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] K.-C. 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.

[15] F. Leibfritz and E. M. E. Mostafa, "An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 1048–1074, 2002.

[16] N. I. M. Gould, S. Leyffer, and P. L. Toint, "A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares," *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 17–38, 2004.

[17] C. Helmberg, *Semidefinite Programming For Combinatorial Optimization*, Konard-Zuse-Zent rum für informationstechnik, Berlin, Germany, 2000.

[18] A. Fischer, "A special Newton-type optimization method," *Optimization*, vol. 24, no. 3-4, pp. 269–284, 1992.

[19] A. Fischer, "A Newton-type method for positive-semidefinite linear complementarity problems," *Journal of Optimization Theory and Applications*, vol. 86, no. 3, pp. 585–608, 1995.
[20] P. Tseng, "Merit functions for semi-definite complementarity problems," *Mathematical Programming*, vol. 83, no. 2, pp. 159–185, 1998.
[21] X. Chen and P. Tseng, "Non-interior continuation methods for solving semidefinite complementarity problems," *Mathematical Programming*, vol. 95, no. 3, pp. 431–474, 2003.
[22] D. Sun and J. Sun, "Semismooth matrix-valued functions," *Mathematics of Operations Research*, vol. 27, no. 1, pp. 150–169, 2002.
[23] H. Jiang, M. Fukushima, L. Qi, and D. Sun, "A trust region method for solving generalized complementarity problems," *SIAM Journal on Optimization*, vol. 8, no. 1, pp. 140–157, 1998.
[24] W. Sun and Y. Yuan, *Optimization Theory and Methods, Nonlinear Programming*, Springer, New York, NY, USA, 2006.