Acta Mathematica Sinica, English Series
Oct., 2014, Vol. 30, No. 10, pp. 1810–1826
Published online: September 15, 2014
DOI: 10.1007/s10114-014-3241-1
Http://www.ActaMath.com

Acta Mathematica Sinica, English Series

A Filter Method for Nonlinear Semidefinite Programming with Global Convergence

Zhi Bin ZHU **Hua Li ZHU**
School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin 541004, P. R. China
E-mail: *zhuzbma@hotmail.com* *zhuhuali06@gmail.com*

Abstract In this study, a new filter algorithm is presented for solving the nonlinear semidefinite programming. This algorithm is inspired by the classical sequential quadratic programming method. Unlike the traditional filter methods, the sufficient descent is ensured by changing the step size instead of the trust region radius. Under some suitable conditions, the global convergence is obtained. In the end, some numerical experiments are given to show that the algorithm is effective.

Keywords Cone programming, nonlinear semidefinite programming, filter method, step size, global convergence

MR(2010) Subject Classification 90C22, 65K05

1 Introduction

1.1 Motivation

It is considered the following nonlinear semidefinite programming (SDP) problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) = 0, \quad i \in I = \{1, 2, \dots, l\}, \\ & \mathbf{A}(x) \preceq 0, \end{aligned} \tag{1.1}$$

where $x \in \mathbb{R}^n$, the functions $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^l$, $\mathbf{A} : \mathbb{R}^n \to S^m$ are sufficiently smooth, and S^m denotes the set of the m-th order real symmetric matrices. $\mathbf{A}(x) \preceq 0$ (or $\mathbf{A}(x) \prec 0$) means that $\mathbf{A}(x)$ is negative semidefinite (or negative definite).

Nonlinear SDP (1.1) is an extension of the standard linear SDP which has been attracting a lot of research in recent decade [7, 21, 23, 24, 26]. Recently, several papers have studied on theoretical properties and numerical methods for solving (1.1). For example, sequence SDP methods [20], original dual interior point methods [27], augmented Lagrangian method [22], successive linearization methods [1], interior point methods [12], and so on. The above methods

Received April 25, 2013, revised October 23, 2013, accepted December 2, 2013
Supported by National Natural Science Foundation of China (Grant Nos. 11061011 and 11361018), Guangxi Fund for Distinguished Young Scholars (Grant No. 2012GXNSFFA060003) and the Guangxi Fund (Grant No. 2013GXNSFDA019002); the first author would like to thank the Project of Guangxi Innovation Team "Optimization method and its engineering application" (Grant No. 2014GXNSFFA118001); this work is also supported by Guangxi Experiment Center of Information Science and Guangxi Key Laboratory of Automatic Detecting Technology and Instruments

all use penalty function to get the global convergence. However, the choice of penalty parameter for the penalty function often leads to some numerical difficulties and bad conditions.

Given an initial point x_0, a sequence $\{x_k\}$ is generated and close to x_* which satisfies some optimal conditions for (1.1). For the k-th iteration, the search direction d_k is obtained by solving the following quadratic semidefinite subproblem (QSD):

$$\begin{aligned} \min \quad & q_k(d) = Df(x_k)^T d + \tfrac{1}{2} d^T M_k d \\ \text{s.t.} \quad & g_i(x_k) + Dg_i(x_k)^T d = 0, \quad i \in I, \\ & \mathbf{A}_k(d) \preceq 0, \end{aligned} \tag{1.2}$$

where $M_k \succ 0$, $A_k(d) = \mathbf{A}(x_k) + \sum_{i=1}^n d_i \frac{\partial \mathbf{A}(x_k)}{\partial_{x_i}}$. The general iteration formula at x_k is $x_{k+1} = x_k + t_k d_k$, where t_k is the step size. Let the minimum value $t_k^{\min}$ be defined as follows:

$$t_k^{\min} \triangleq \frac{1}{2} \begin{cases} \min\left\{1-\beta, \dfrac{\gamma h(x_k)}{-Df(x_k)^T d_k}, \dfrac{\delta h(x_k)^2}{-Df(x_k)^T d_k}, \right. \\ \qquad \left. \dfrac{h(x_k)}{(l+1) d_k^T M_k d_k} \right\}, & \text{if } Df(x_k)^T d_k < 0, \\ \min\left\{1-\beta, \dfrac{h(x_k)}{(l+1) d_k^T M_k d_k} \right\}, & \text{otherwise}, \end{cases} \tag{1.3}$$

where $0 < \delta, \gamma, \beta < 1$, while $h(x_k)$ is the value of the constraint violation $h(x)$ (please see the definition (2.1)) at x_k.

An efficient algorithm [20] was proposed to solve the problem (1.1) by using a series of QSD (1.2). However, when QSD (1.2) was solved, there was a problem that (1.2) was not consistent or its solution was unbounded. The filter method [3] was firstly proposed to solve the nonlinear programming. It is well known that this method can avoid the choice of penalty function, so it follows that the filter method has been widely studied (see [4]). The sequence semidefinite programming method [5] was applied for the general nonlinear SDP. The sequence semidefinite programming (SSP) method [15, 16] based on the filter technique was proposed to avoid penalty function for solving (1.1), where the search direction was controlled by changing the trust region radius in each trial step.

In this study, we present a new filter algorithm to solve the problem (1.1). Here, the filter technique is applied to QSD (1.2). Unlike existing filter methods, we ensure the sufficient descent by changing the step size instead of the trust region radius (see [25]), and it doesn't need any penalty function. Finally, the global convergence is got under some suitable assumptions.

1.2 Notations

Some notations are introduced as follows. We use $Df(x)$ and $Dg(x)$ to express the derivative of functions $f(x)$ and $g(x)$ at points x, respectively. For the matrices $A, B \in \mathbb{R}^{m\times m}$, their trace product is defined as $\langle A, B\rangle = A \bullet B = \text{tr}(AB^T)$, where $\text{tr}(C) := \sum_{i=1}^m c_{ii}$ denotes the trace of the matrix $C \in \mathbb{R}^{m\times m}$. It is easy to see that the operator $\bullet$ defines a scalar product on the set of matrices $\mathbb{R}^{m\times m}$. In our analysis, $\|\cdot\|$ denotes the Frobenius norm for a matrix and the 2-norm for a vector. Furthermore, for any given matrix $A \in S^m$, $\lambda_j(A)$ denotes the j-th eigenvalue with $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_m(A)$, and A_+ denotes the matrix defined by

$$A_+ := P\text{diag}((\lambda_1)_+, \ldots, (\lambda_m)_+)P^T,$$

where $(\lambda_j)_+ = \max\{0, \lambda_j\}$, while P satisfies the spectrum decomposition $A = P\mathrm{diag}(\lambda_1, \ldots, \lambda_m)P^T$.

For a given matrix-value function $\mathbf{A}(\cdot)$, its differential operator is evaluated at x with the following notation:

$$D\mathbf{A}(x) := \left(\frac{\partial \mathbf{A}(x)}{\partial x_i}\right)_{i=1}^n = \left(\frac{\partial \mathbf{A}(x)}{\partial x_1}, \ldots, \frac{\partial \mathbf{A}(x)}{\partial x_n}\right)^T.$$

The above notation generates the following operator:

$$D\mathbf{A}(x)y = \sum_{i=1}^n y_i \frac{\partial \mathbf{A}(x)}{\partial x_i}, \quad \forall y \in \mathbb{R}^n.$$

We define the adjoint operator as follows:

$$D\mathbf{A}(x)^* Z = \left(\left\langle \frac{\partial \mathbf{A}(x)}{\partial x_1}, Z\right\rangle, \ldots, \left\langle \frac{\partial \mathbf{A}(x)}{\partial x_n}, Z\right\rangle\right)^T, \quad \forall Z \in S^m.$$

2 Filter

In this section, we define the constraint violation as

$$h(x) = \lambda_1(\mathbf{A}(x))_+ + \|g(x)\|. \tag{2.1}$$

Definition 2.1 *A point x_1 is called to dominate another point x_2 in the filter F iff*

$$f(x_1) \le f(x_2) \quad \text{and} \quad h(x_1) \le h(x_2), \quad \forall x_2 \in F. \tag{2.2}$$

When a new point x_k is added into the filter F, other points in the filter F, which are dominated by x_k, must be removed from F, see [3].

In practical application, a point x_k is determined whether to be accepted by filter via the following criterion.

Definition 2.2 *A point x_k is called to be accepted by the filter if*

$$h(x_k) < \beta h(x_j) \quad \text{or} \quad f(x_k) < f(x_j) - \gamma h(x_k), \quad \forall x_j \in F, \tag{2.3}$$

where γ, β are constant such that $0 < \gamma < \beta < 1$, and $\gamma \to 0,\ \beta \to 1$.

Definition 2.3 *If the iteration point x_k is acceptable to the filter, and it holds that*

$$f(x_k) - f(x_k + t_k d_k) \ge \eta(q_k(0) - q_k(t_k d_k)) \quad \text{and} \quad \Delta q^k = q_k(0) - q_k(t_k d_k) > \delta h^2(x_k), \tag{2.4}$$

where $0 < \delta, \eta < 1$, then the corresponding iteration is said to be an f-type iteration. If the right inequality of (2.4) does not hold, then the iteration is said to be an h-type iteration. When the h-iteration iteration exists, the current iteration point x_k will be included into the filter.

3 QSD Filter Algorithm

We now state the QSD filter algorithm for solving (1.1) as follows. For convenience, we give the following notations:

$$\tau^k = \min_{x_t \in F_k} \{h(x_t)\}, \quad P_k^j = h(x_k^{j+1}) - h(x_k^j),$$

where F_k is the filter at the k-th iteration.

Algorithm 3.1

(S.0) Choose $x_0 \in \mathbb{R}^n$, $M_0 \succ 0$, $0 < \delta, \eta < 1$, $0 < \gamma < \beta < 1$, $0 < \eta_1 < \beta$, $F_0 := \{x_0\}, k := 0$.

(S.1) Set $t_k = 1$.

(S.2) Calculate the search direction d_k by solving QSD (1.2). If $d_k = 0$, the KKT point x_k is obtained and stop.

(S.3) If QSD (1.2) is incompatible, we get x_k^r by using Algorithm 3.2, and set $x_k := x_k^r$. Go back to (S.1).

(S.4) If $t_k < t_k^{\min}$, go to Algorithm 3.2. Set $x_k := x_k^r$ and go back to (S.1).

(S.5) Filtering criterion: if $x_k + t_k d_k$ is accepted by the filter, go to (S.6), otherwise set $t_k := \frac{1}{2} t_k$, go to (S.4).

(S.6) Compute $r_k = \frac{f(x_k) - f(x_k + t_k d_k)}{q_k(0) - q_k(t_k d_k)}$. If $r_k < \eta$ and $\Delta q^k = q_k(0) - q_k(t_k d_k) > \delta h^2(x_k)$, set $t_k := \frac{1}{2} t_k$, and go back to (S.4).

(S.7) If $\Delta q^k = q_k(0) - q_k(t_k d_k) \leq \delta h^2(x_k)$, add the point x_k into the filter, update F_{k+1} and τ^{k+1}, otherwise, set $F_{k+1} = F_k,\ \tau^{k+1} = \tau^k$.

(S.8) Update M_{k+1}, and let $x_{k+1} = x_k + t_k d_k,\ k := k+1$. Go back to (S.1).

By using the following Recovery algorithm, we can reduce the constraint violation $h(x_k)$.

Algorithm 3.2 (Recovery algorithm)

(R.0) Let $x_k^0 = x_k,\ t_k^0 = 1,\ 0 < \eta_1 < \beta$, and set $j := 0$.

(R.1) Solve the following subproblem

$$\begin{array}{ll} \min & q_k^j(t_k^j d_k^j) = \|g(x_k^j) + Dg(x_k^j)^T t_k^j d_k^j\| + \lambda_1(\mathbf{A}(x) + D\mathbf{A}(x) t_k^j d_k^j)_+ - h(x_k^j) \\ \text{s.t.} & 1 \leq \|d_k^j\| \leq 2 \end{array} \tag{3.1}$$

to get d_k^j.

(R.2) Calculate

$$r_k^j = \frac{P_k^j}{q_k^j(t_k d_k^j)}. \tag{3.2}$$

(R.3) If $r_k^j \leq \eta$, set $x_k^{j+1} = x_k^j,\ t_k^{j+1} = \frac{1}{2} t_k^j,\ j := j+1$, go back to (R.2). Otherwise, set $x_k^{j+1} = x_k^j + t_k^j d_k^j$, $t_k^{j+1} = 2t_k^j$, $j := j+1$, and go to (R.4).

(R.4) If $h(x_k^j) \leq \min\{\eta_1 \tau^k, \|t_k^j d_k^j\|\}$, set $x_k^r = x_k^j$, and go back to (S.1), otherwise, go back to (R.1).

4 Convergence Analysis

In this section, we establish the global convergence of the QSD filter algorithm for nonlinear SDP (1.1). First, we make the following general assumptions.

Assumption 4.1

(A.1) Objective function $f(x)$ and constraint functions $g(x), \mathbf{A}(x)$ are twice continuous differentiable on an open set containing X.

(A.2) The sequence $\{x_k\} \in X$ is bounded.

(A.3) $\forall x_k \in F_k$, $\{Dg_i(x_k), i \in I\}$ is linearly independent.

(A.4) For solving (3.1), we have

$$\begin{aligned} q_k^j(t_k^j d_k^j) &= \|g(x_k^j) + Dg(x_k^j)^T t_k^j d_k^j\| + \lambda_1(\mathbf{A}(x_k^j) + D\mathbf{A}(x_k^j) t_k^j d_k^j)_+ - h(x_k^j) \\ &\leq -\eta_2 \min\{h(x_k^j), t_k^j d_k^j\}. \end{aligned}$$

(A.5) $\forall x_*$, if λ_*, Z_* satisfy that

$$D\mathbf{A}(x_*)^* Z_* + \sum_{j=1}^{p} \lambda_{*j} Dh_j(x_*) = 0, \quad \mathrm{Tr}(Z_* \mathbf{A}(x_*)) = 0, Z_* \succeq 0, \mathbf{A}(x_*) \preceq 0, h(x_*) = 0,$$

it holds that $Z_* = 0$.

(A.6) There exists a constant $M > 0$ such that the sequence of Hessian matrices $\{M_k\}$ satisfies $\|M_k\| \le M$ for all k.

According to Assumption 4.1, we know that the Hessian matrices of f and the constraint functions g_i and $\mathbf{A}(x)$ are bounded on X. Without loss of generality, we assume that there is a constant M such that $\|D^2 f(x_k)\| \le M$, $\|D^2 g_i(x_k)\| \le M$, $\|D^2 \mathbf{A}(x_k)\| \le M$ for all $x_k \in X$.

$x_* \in \mathbb{R}^n$ is called to be a stationary point of the original problem (1.1), if x_* is a feasible point of (1.1), and the corresponding Lagrange multiplier $(\lambda_*, Z_*) \in \mathbb{R}^l \times S^m$ satisfies the KKT condition as follows,

$$\begin{aligned} &Df(x_*) + D\mathbf{A}(x_*)^* Z_* + Dg(x_*)^T \lambda_* = 0, \\ &\langle Z_*, A(x_*) \rangle = 0, \\ &Z_* \succeq 0. \end{aligned}$$

In addition to the KKT condition, it is also needed the FJ necessary condition defined by below. If x_* is a feasible point of (1.1) and the direction set

$$\{d \in \mathbb{R}^n | Df(x_*)^T d < 0, Dg(x_*)^T d = 0, E_*^T (D\mathbf{A}(x_*)d) E_* \prec 0\} = \emptyset, \tag{4.1}$$

where the columns of E_* are the standard orthogonal eigenvectors corresponding to those zero eigenvalues, we call the point $x_* \in S^n$ as an FJ point.

Suppose Assumption 4.1 (A.5) holds, it is easy to see that every FJ point satisfies the KKT condition [15].

The following results are based on Assumption 4.1. First, we investigate the optimal properties of Algorithm 3.1.

Lemma 4.1 $\tau^k = \min_{x_t \in F_k} \{h(x_t)\} > 0$.

Proof If the conclusion is not true, then $h(x_{k^*}) = 0$ for some $x_{k^*} \in F_{k^*}$. Since $d = 0$ is a feasible point for the QSD (1.2) and $h(x_{k^*}) = 0$, we have

$$q_{k^*}(0) - q_{k^*}(d_{k^*}) = -Df(x_{k^*})^T d_{k^*} - \frac{1}{2} d_{k^*}^T M_{k^*} d_{k^*} > 0.$$

Combining with $t_{k^*} \in (0, 1]$ and $M_{k^*} \succ 0$, we have

$$\begin{aligned} q_{k^*}(0) - q_{k^*}(t_{k^*} d_{k^*}) &= -t_{k^*} \left(Df(x_{k^*})^T d_{k^*} + \frac{1}{2} t_{k^*} d_{k^*}^T M_{k^*} d_{k^*} \right) \\ &= -t_{k^*} \left(Df(x_{k^*})^T d_{k^*} + \frac{1}{2} d_{k^*}^T M_{k^*} d_{k^*} \right) + \frac{1}{2} (t_{k^*} - t_{k^*}^2) d_{k^*}^T M_{k^*} d_{k^*} \\ &> 0 = \delta h(x_{k^*})^2, \end{aligned}$$

which contradicts the definition of h-type iteration, so the point x_{k^*} will not be added into the filter. The proof is finished. □

Similarly, we can obtain the following result [17].

Lemma 4.2 *Suppose there are an infinite number of points to be added to the filter. Then* $\lim_{k \to \infty} h(x_k) = 0$.

Proof If the result is not true, there would have infinite components in K_1 which is defined as follows,

$$K_1 = \{k \mid h(x_k) > \varepsilon\}.$$

Since Assumption 4.1 holds, without loss of generality, we assume that $|f(x_k)| \leq M$ for all k, where M is a positive constant. Then we analyze with two cases.

(1) If $\min_{i\in K_1}\{f(x_1)\}$ exists. Let $f(x_{kc}) = \min_{i\in K_1}\{f(x_1)\}$ and $h(x_{kc})$ be the corresponding value related to (2.1). Then, according to the definition of the filter, the other components, which lie behind x_k in the filter, satisfy $h(x_k) \leq h(x_{kc})$ and $f(x_k) \geq f(x_{kc})$. Then, all the filter points, which enter the filter behind x_{kc}, can be covered with a square, whose area is no more than $2Mh(x_{kc})$. We consider the area lies to the south-west of the filter in this square. When a new point x_{kc} enters the filter, the nest point x_{kc+1} should lie to south-west of the point in the filter F_{kc}, and the area which lies to south-west of the F_{kc+1} in the square is smaller than that of F_{kc}. Therefore, we think that the area is reduced if a new point enter the filter. If a new point enters K_1 of the filter, the area of the square more than $(1-\beta)\gamma\varepsilon^2$, will be reduced. In fact, when a point is added to the filter, its h value is less than every point, which lies to the left of this point, to more than $(1-\beta)\varepsilon$, its f value is less than every point, which lies to the below of the point, to more than $\gamma\varepsilon$. Therefore, the area of the square, more than $(1-\beta)\gamma\varepsilon^2$ will be reduced. Thus, the area will be reduced to zero after finite time. When the area is zero, it means that a point can not enter K_1, which contradicts the infiniteness of K_1.

(2) If $\min_{i\in K_1}\{f(x_1)\}$ does not exists. From the conditions in this lemma, let $f(x_{kc}) = \inf_{i\in K_1}\{f(x_1)\}$. From the definition of inf, there exist $f(x_{kc}) \geq f(x_c)$ and $f(x_{kc}) \leq f(x_c) + \gamma\varepsilon$. Then, according to the definition of the filter, the other components, which lie behind in the filter, satisfy $h(x_k) \leq h(x_{kc})$ and $f(x_k) \geq f(x_{kc}) - \gamma\varepsilon$. Using the same techniques as that in the case (1), the result is got.

Thus, the conclusion is obtained. □

Lemma 4.3 *If there are just finite points to be added into the filter and infinite points to be added into the sequence, then* $\lim_{k\to\infty} h(x_k) = 0$.

Proof If the result is not true. There would have an infinite components in K_1, which is defined as follows:

$$K_1 = \{k \mid h(x_k) > \varrho\}. \tag{4.2}$$

Since $f(x_k)$ is bounded by Assumption 4.1, there exists some K_2 such that

$$+\infty > \sum_{k\geq K_2} f(x_k) - f(x_{k+1}), \tag{4.3}$$

$$f(x_k) - f(x_{k+1}) = f(x_k) - f(x_k + t_k d_k) \geq \eta(q_k(0) - q_k(t_k d_k)) \geq \eta\delta h^2(x_k), \quad \forall k > K_2, \tag{4.4}$$

so $f(x_k)$ is monotonically decreasing. However,

$$\sum_{k\geq K_2} f(x_k) - f(x_{k+1}) > \sum_{k\in K_1, k\geq K_2} \delta\varrho^2 = +\infty,$$

which contradicts (4.3). So, the conclusion is true. □

For the restoration algorithm, similar to [18, 19], we can obtain the following result, which shows that Algorithm 3.2 is well defined.

Lemma 4.4 *The recovery algorithm terminates finitely.*

Proof If the result is not true, there exists a positive parameter ϱ satisfies $h(x_k^j) > \varrho$ for all upper index j. Since $q_k^j(t_k^j d_k^j) \le -\eta_2 \min\{h(x_k^j), t_k^j d_k^j\} \le 0$, Recovery algorithm (R.3) yields $P_k^j = h(x_k^{j+1}) - h(x_k^j) \le 0$ (when $r_k^j \le \eta$, we have $P_k^j = h(x_k^{j+1}) - h(x_k^j) = h(x_k^j) - h(x_k^j) = 0$, when $r_k^j > \eta$, $P_k^j = h(x_k^{j+1}) - h(x_k^j) = \frac{1}{r_k^j} q_k^j(t_k^j d_k^j) \le 0$). It is easy to see that the sequence $\{h(x_k^j)\}$ is monotonously decreasing and

$$+\infty > \sum_{j=1}^{\infty} h(x_k^{j-1}) - h(x_k^j) \ge -\eta \sum_{j=1}^{\infty} q_k^j(t_k^j d_k^j) \ge \eta\eta_2 \sum_{r_k^j \ge \eta} \min\{\varrho, \|t_k^j d_k^j\|\},$$

while $\|d_k^j\| \ge 1$, and $\lim_{r_k^j \ge \eta} t_k^j d_k^j \to 0$. So $\lim_{j\to\infty} t_k^j = 0$.

Based on the Taylor expanding, it follows that

$$\begin{aligned} h(x_k^j + t_k^j d_k^j) &= \|g(x_k^j + t_k^j d_k^j)\| + \lambda_1(\mathbf{A}(x_k^j + t_k^j d_k^j))_+ \\ &= \|g(x_k^j) + Dg(x_k^j)^T t_k^j d_k^j + o(t_k^j d_k^j)\| + \lambda_1(\mathbf{A}(x_k^j) + D\mathbf{A}(x_k^j) t_k^j d_k^j + o(t_k^j d_k^j))_+ \\ &= \|g(x_k^j) + Dg(x_k^j)^T t_k^j d_k^j\| + \lambda_1(\mathbf{A}(x_k^j) + D\mathbf{A}(x_k^j) t_k^j d_k^j)_+ + o(t_k^j d_k^j), \end{aligned}$$

i.e.,

$$h(x_k^j) - h(x_k^j + t_k^j d_k^j) = q_k^j(t_k^j d_k^j) - o(t_k^j d_k^j) = q_k^j(t_k^j d_k^j) - o(t_k^j).$$

So $q_k^j(t_k^j d_k^j) = P_k^j + o(t_k^j)$ as $t_k^j \to 0$. Since $t_k^{j+1} = 2t_k^j$ from Recovery Algorithm 3.2 (R.3), we known that $\{t_k^j\}$ is a strictly monotone increasing sequence when t_k^j is small enough. It contradicts $\lim_{j\to\infty} t_k^j = 0$. Thus, the conclusion holds. □

Next, we would introduce the following result [3].

Lemma 4.5 *Consider minimizing a quadratic function $\phi(\alpha) : \mathbb{R} \to \mathbb{R}$ with $\phi(0) < 0$ on the interval $\alpha \in [0, 1]$. A necessary and sufficient condition for the minimizer to be at $\alpha = 1$ is $\phi'' + \phi' \le 0$. In this case, it follows that $\phi(0) - \phi(1) \ge -\frac{1}{2}\phi'(0)$.*

Lemma 4.6 *Suppose Assumption* 4.1 *holds. If d_k is a feasible solution of the* QSD (1.2) *at x_k, then it follows that*

$$f(x_k + t_k d_k) - f(x_k) \le q_k(t_k d_k) + M\|t_k d_k\|^2, \tag{4.5}$$

$$h(x_k + t_k d_k) \le (1 - t_k)h(x_k) + \frac{1}{2}(l + 1)M\|t_k d_k\|^2. \tag{4.6}$$

Proof Under Assumption 4.1(A.6), based on the Taylor expanding, it is easy to see that

$$f(x_k + t_k d_k) = f(x_k) + Df(x_k)^T t_k d_k + \frac{1}{2}(t_k d_k)^T D^2 f(y)(t_k d_k),$$

where y locates on the line segment between x_k and $x_k + t_k d_k$. So

$$f(x_k + t_k d_k) - f(x_k) = q_k(t_k d_k) + \frac{1}{2}(t_k d_k)^T (D^2 f(y) - M_k)(t_k d_k) \le q_k(t_k d_k) + M\|t_k d_k\|^2.$$

Moreover, from the Taylor expandings about $g(x_k + t_k d_k)$ and $\mathbf{A}(x_k + t_k d_k)$ at the point x_k, we have

$$\begin{aligned} &\|g(x_k + t_k d_k)\| \le \|g(x_k) + t_k Dg(x_k)^T d_k\| + \frac{1}{2} lM\|t_k d_k\|^2, \\ &\lambda_1(\mathbf{A}(x_k + t_k d_k))_+ \le \lambda_1(\mathbf{A}_k(t_k d_k))_+ + \frac{1}{2}M\|t_k d_k\|^2. \end{aligned} \tag{4.7}$$

Since d_k is a feasible solution of the QSD (1.2) at the point x_k, it holds that

$$\mathbf{A}(x_k) + \sum_{i=1}^{n} d_{ki} \frac{\partial \mathbf{A}(x_k)}{\partial_{x_i}} \preceq 0,$$

$$g_i(x_k) + Dg_i(x_k)^T d_k = 0, \quad i = 1, 2, \ldots, l.$$

Thereby, we have

$$\mathbf{A}_k(t_k d_k) = \mathbf{A}(x_k) + t_k \sum_{i=1}^{n} d_{ki} \frac{\partial \mathbf{A}(x_k)}{\partial_{x_i}} \preceq (1 - t_k)\mathbf{A}(x_k),$$

$$g_i(x_k) + t_k Dg_i(x_k)^T d_k = (1 - t_k) g_i(x_k), \quad i = 1, 2, \ldots, l.$$

Combining with (4.7), we can get

$$\|g(x_k + t_k d_k)\| \le (1 - t_k)\|g(x_k)\| + \frac{1}{2} lM \|t_k d_k\|^2,$$

$$\lambda_1(\mathbf{A}(x_k + t_k d_k))_+ \le (1 - t_k)\lambda_1(\mathbf{A}(x_k))_+ + \frac{1}{2} M \|t_k d_k\|^2,$$

that is,

$$h(x_k + t_k d_k) \le (1 - t_k) h(x_k) + \frac{1}{2}(l + 1) M \|t_k d_k\|^2.$$

The conclusion holds. □

To obtain two main results of this paper, we establish the following important result.

Lemma 4.7 *Suppose Assumption* 4.1 *holds, and* $x_* \in X$ *is a feasible point of the original problem* (1.1) *but not the* KKT *point. Then, there exist a neighborhood* N *at* x_* *and some positive constants* ε, μ, κ, *such that* $\forall x_k \in N \cap X$, *the feasible set of the* QSD (1.2) *is not empty and the feasible direction* d_k *of* QSD (1.2) *satisfies*

$$\mu h(x_k) \le \|d_k\| \le \kappa, \tag{4.8}$$

and

$$q_k(0) - q_k(d_k) \ge \frac{1}{3}\|d_k\|\varepsilon. \tag{4.9}$$

If $\|t_k d_k\| \le \frac{(1-\eta)\varepsilon}{3M}$, *we have*

$$\frac{f(x_k) - f(x_k + t_k d_k)}{q_k(0) - q_k(t_k d_k)} \ge \eta. \tag{4.10}$$

Proof Suppose Assumption 4.1 holds, x_* is not an FJ point, there exists a vector $d_* \in \mathbb{R}^n$ with $\|d_*\| = 1$ and satisfies (4.1). Set

$$A_k := (Dg(x_k)(Dg(x_k))^T)^{-1} Dg(x_k), \quad p_k := \begin{cases} -A_k^T g(x_k), & \text{if } \ l \ge 1, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$s_k := \begin{cases} (I - Dg(x_k)^T A_k) d_* / \|(I - Dg(x_k)^T A_k) d_*\|, & \text{if } \ l \ge 1, \\ d_*, & \text{otherwise.} \end{cases}$$

Based on (4.1) and the continuity of Df and $D\mathbf{A}$, there exist a small neighborhood N and a positive constant ε such that

$$s_k^T Df(x_k) < -\varepsilon \quad \text{and} \quad E_k^T (D\mathbf{A}(x_k) s_k) E_k \prec -\varepsilon I \tag{4.11}$$

for all $x_k \in N$, where the columns of E_k are the standard orthogonal eigenvectors corresponding to these zero eigenvalues of $\mathbf{A}(x_k)$.

Let $P_k = [E_k, F_k]$ be the orthogonal matrix, such that

$$P_k^T \mathbf{A}(x_k) P_k = \begin{pmatrix} E_k^T \mathbf{A}(x_k) E_k & E_k^T \mathbf{A}(x_k) F_k \\ F_k^T \mathbf{A}(x_k) E_k & F_k^T \mathbf{A}(x_k) F_k \end{pmatrix} = \begin{pmatrix} \Lambda_{E_k} & 0 \\ 0 & \Lambda_{F_k} \end{pmatrix},$$

where Λ_{E_k} and Λ_{F_k} are diagonal matrices. Without loss of generality, we assume $\Lambda_{F_k} \prec 0$ for any $x_k \in N \cap X$.

For any v_k with $v_k \geq \|p_k\|$, we define $d_k^\alpha = p_k + \alpha v_k s_k,\ \alpha \in [0,1]$. When $\alpha = 1$, since p_k, s_k are orthogonal and $\|s_k\| = 1$, we have

$$v_k \leq \|d_k^1\| = \sqrt{\|p_k\|^2 + v_k^2} \leq \sqrt{2} v_k. \tag{4.12}$$

Next, we prove that d_k^1 is a feasible solution of (1.2). First, from the definition of p_k, it is easy to obtain $g(x_k) + Dg(x_k)^T d_k^\alpha = 0, \alpha \in [0,1]$. So d_k^1 satisfies the equality constraint conditions of (1.2). Moreover,

$$\begin{aligned} \mathbf{A}(x_k) + D\mathbf{A}(x_k) d_k^1 &= P_k \left(\begin{pmatrix} \Lambda_{E_k} & 0 \\ 0 & \Lambda_{F_k} \end{pmatrix} + \begin{pmatrix} E_k^T D\mathbf{A}(x_k) d_k^1 E_k & E_k^T D\mathbf{A}(x_k) d_k^1 F_k \\ F_k^T D\mathbf{A}(x_k) d_k^1 E_k & F_k^T D\mathbf{A}(x_k) d_k^1 F_k \end{pmatrix} \right) P_k^T \\ &= P_k \begin{pmatrix} \Lambda_{E_k} + E_k^T D\mathbf{A}(x_k) d_k^1 E_k & E_k^T D\mathbf{A}(x_k) d_k^1 F_k \\ F_k^T D\mathbf{A}(x_k) d_k^1 E_k & \Lambda_{F_k} + F_k^T D\mathbf{A}(x_k) d_k^1 F_k \end{pmatrix} P_k^T. \end{aligned}$$

From the boundedness of $D\mathbf{A}(x_k)$ on $N \cap X$, there exist two positive parameters $\bar{a}$ and $\bar{c}$, of which one is independent from d_k^1, such that

$$\|F_k^T D\mathbf{A}(x_k) d_k^1 F_k\| \leq \|d_k^1\| \bar{a}, \quad \lambda_1(\Lambda_{F_k}) < -\bar{c}, \quad \forall x_k \in N \cap X. \tag{4.13}$$

It follows that

$$\Lambda_{F_k} + F_k^T D\mathbf{A}(x_k) d_k^1 F_k \preceq \Lambda_{F_k} + \|d_k^1\| \bar{a} I \prec (\|d_k^1\| \bar{a} - \bar{c}) I.$$

Then, if $\|d_k^1\| \leq \frac{\bar{c}}{\bar{a}}$, we have

$$\Theta := \Lambda_{F_k} + F_k^T D\mathbf{A}(x_k) d_k^1 F_k \prec 0, \quad \forall x_k \in N \cap X.$$

On the other hand, it holds that, for all $x_k \in N \cap X$,

$$\begin{aligned} &\Lambda_{E_k} + E_k^T D\mathbf{A}(x_k) d_k^1 E_k - E_k^T D\mathbf{A}(x_k) d_k^1 F_k \Theta^{-1} F_k^T D\mathbf{A}(x_k) d_k^1 E_k \\ &\quad = \Lambda_{E_k} + E_k^T D\mathbf{A}(x_k) p_k E_k + v_k E_k^T D\mathbf{A}(x_k) s_k E_k \\ &\qquad - E_k^T D\mathbf{A}(x_k) p_k F_k \Theta^{-1} F_k^T D\mathbf{A}(x_k) p_k E_k - 2 v_k E_k^T D\mathbf{A}(x_k) p_k F_k \Theta^{-1} F_k v_k^T D\mathbf{A}(x_k) s_k E_k \\ &\qquad - v_k^2 E_k^T D\mathbf{A}(x_k) s_k F_k \Theta^{-1} F_k^T D\mathbf{A}(x_k) s_k E_k \\ &\quad \preceq v_k(-\varepsilon I + v_k(-E_k^T D\mathbf{A}(x_k) s_k F_k \Theta^{-1} F_k^T D\mathbf{A}(x_k) s_k E_k)) + \Lambda_{E_k} \\ &\qquad + E_k^T D\mathbf{A}(x_k) p_k E_k - E_k^T D\mathbf{A}(x_k) p_k F_k \Theta^{-1} F_k^T D\mathbf{A}(x_k) p_k E_k \\ &\qquad - 2 v_k E_k^T D\mathbf{A}(x_k) p_k F_k \Theta^{-1} F_k^T D\mathbf{A}(x_k) s_k E_k. \end{aligned} \tag{4.14}$$

Let $B_r := \{x_k \mid \|x_k - x_*\| \leq r\}$ with the radius $r > 0$, and we define the non-negative value $b_r := \max_{x \in B_r} \{\lambda_1(-E_k^T D\mathbf{A}(x) s_k F_k \Theta^{-1} F_k^T D\mathbf{A}(x) s_k E_k)\}$. Two cases are discussed as follows.

The first case: For $\bar{r} > 0$, with $b_{\bar{r}} = 0$, we have

$$E_k^T D\mathbf{A}(x) s_k F_k \Theta^{-1} F_k^T D\mathbf{A}(x) s_k F_k \equiv 0,$$

as $-E_k^T D\mathbf{A}(x)S_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x)S_kF_k \succeq 0$ for all $x_k \in B_{\bar r}$.

From (4.14), we obtain the sufficient condition for $\mathbf{A}(x_k)+D\mathbf{A}(x_k)d_k^1 \preceq 0$ as follows:

$$\begin{aligned}&-v_k\varepsilon I+\Lambda_{E_k}+E_k^T D\mathbf{A}(x_k)p_kE_k-E_k^T D\mathbf{A}(x_k)p_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x_k)p_kE_k\\&\quad-2v_kE_k^T D\mathbf{A}(x_k)p_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x_k)s_kE_k\preceq 0.\end{aligned}\tag{4.15}$$

Moreover, let

$$\begin{aligned}\Psi_1 &:= E_k^T D\mathbf{A}(x_k)p_kE_k-E_k^T D\mathbf{A}(x_k)p_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x_k)p_kE_k,\\ \Psi_2 &:= 2\bar cE_k^T D\mathbf{A}(x_k)p_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x_k)s_kE_k/\bar a.\end{aligned}$$

Then (4.15) holds under the following condition

$$v_k\ge\max\left\{0,\frac{\lambda_1(\Lambda_{E_k})+\|\Psi_1\|+\|\Psi_2\|}{\varepsilon}\right\}=O(h(x_k)).$$

The second case: $b_r\neq 0$ for all $r>0$. In this case, we know that $\|p_k\|\to 0$ and $\frac{1}{b_r}$ increases as $r\to 0$ for all $x_k\in N\cap B_r$. So, there exists a sufficiently small $\bar r$ which satisfies $v_k\le\frac{\varepsilon}{2b_{\bar r}}$.

When $v_k\le\frac{\varepsilon}{2b_{\bar r}}$, we have

$$-\varepsilon I+v_kE_k^T D\mathbf{A}(x_k)s_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x_k)s_kE_k\preceq-\tfrac{\varepsilon}{2}.\tag{4.16}$$

From (4.14) and (4.16) we know the sufficient condition for $\mathbf{A}(x_k)+D\mathbf{A}(x_k)d_k^1\preceq 0$ as follows:

$$\begin{aligned}&-v_k\frac{\varepsilon}{2}I+\Lambda_{E_k}+E_k^T D\mathbf{A}(x_k)p_kE_k-E_k^T D\mathbf{A}(x_k)p_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x_k)p_kE_k\\&\quad-2v_kE_k^T D\mathbf{A}(x_k)p_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x_k)s_kE_k\preceq 0.\end{aligned}\tag{4.17}$$

Moreover, let

$$\begin{aligned}\Psi_1 &:= E_k^T D\mathbf{A}(x_k)p_kE_k-E_k^T D\mathbf{A}(x_k)p_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x_k)p_kE_k,\\ \Psi_2 &:= 2\bar cE_k^T D\mathbf{A}(x_k)p_kF_k\Theta^{-1}F_k^T D\mathbf{A}(x_k)s_kE_k/\bar a.\end{aligned}$$

Then (4.17) is true under the following condition:

$$v_k\ge\max\left\{0,\frac{2(\lambda_1(\Lambda_{E_k})+\|\Psi_1\|+\|\Psi_2\|)}{\varepsilon}\right\}=O(h(x_k)).$$

Hence, if $\kappa\le\min\{\frac{\bar c}{\bar a},\frac{\sqrt{2\varepsilon}}{2b_{\bar r}}\}$, (4.8) holds for any $x_k\in N\cap B_{\bar r}$ in this case. Combining the above two cases, d_k^1 satisfies the negative semidefinite constraint conditions of (1.2). So d_k^1 is a feasible solution of (1.2). The first conclusion is proved.

From the definition of p_k, we have

$$\|d_k^1\|\ge v_k\ge\|p_k\|=O(g(x_k)).\tag{4.18}$$

From the negative semidefinite constraint condition, we have

$$\mathbf{A}(x_k)\preceq-D\mathbf{A}(x_k)d_k^1\quad\text{and}\quad\lambda_1(\mathbf{A}(x_k))_+\le M\|d_k^1\|.$$

So,

$$\|d_k^1\|\ge\frac{1}{M}\lambda_1(\mathbf{A}(x_k))_+=O(\lambda_1(\mathbf{A}(x_k))_+).\tag{4.19}$$

Combining with (4.18) and (4.19), it holds that $\|d_k^1\|\ge O(h(x_k))$. Furthermore, there exists some sufficiently large parameters μ, such that (4.8) satisfies for any $x_k\in N\cap X$.

Let $\phi(\alpha) = q_k(p_k + \alpha v_k S_k)$. It is easy to have that

$$\phi'(\alpha) = v_k s_k^T Dq_k(p_k + \alpha v_k s_k) = v_k s_k^T (Df(x_k) + M_k(p_k + \alpha v_k s_k)).$$

Using (4.11), if $\|d_k\| \leq \frac{\sqrt{2}}{2}\frac{\varepsilon}{M}$, we have

$$\begin{aligned}\phi'(0) &= v_k s_k^T(Df(x_k) + M_k p_k) \leq v_k(s_k^T M_k p_k - \varepsilon)\\ &\leq v_k(Mv_k - \varepsilon) \leq \sqrt{2}\|d_k\|\left(\frac{\sqrt{2}}{2}M\|d_k\| - \varepsilon\right) \leq 0,\end{aligned}$$

and $\phi'' = v_k^2 s_k^T M_k s_k \leq v_k^2 M \leq \|d_k\|^2 M$. Then

$$\phi'' + \phi'(0) \leq M\|d_k\|^2 + \sqrt{2}\|d_k\|\left(\frac{\sqrt{2}}{2}M\|d_k\| - \varepsilon\right) \leq \sqrt{2}\|d_k\|(\sqrt{2}M\|d_k\| - \varepsilon) \leq 0.$$

Furthermore, if $\|d_k\| \leq (1 - \frac{2\sqrt{2}}{3})\frac{\varepsilon}{M}$, we have

$$q_k(0) - q_k(d_k) \geq \frac{\sqrt{2}}{2}\|d_k\|\left(\varepsilon - \frac{\sqrt{2}}{2}M\|d_k\|\right) = \frac{\sqrt{2}}{4}\|d_k\|(\varepsilon - M\|d_k\|) \geq \frac{1}{3}\|d_k\|\varepsilon, \tag{4.20}$$

and

$$q_k(0) - q_k(t_k d_k) \geq \frac{1}{3}\|t_k d_k\|\varepsilon. \tag{4.21}$$

The second conclusion is proved.

According to (4.6) and (4.20), if $\|t_k d_k\| \leq \frac{(1-\eta)\varepsilon}{3M}$, we have

$$\begin{aligned}\frac{f(x_k) - f(x_k + t_k d_k)}{q_k(0) - q_k(t_k d_k)} &\geq 1 - \frac{\|t_k d_k\|^2 M}{q_k(0) - q_k(t_k d_k)} \geq 1 - \frac{3\|t_k d_k\|^2 M}{\|t_k d_k\|\varepsilon}\\ &= 1 - \frac{3\|t_k d_k\| M}{\varepsilon} \geq \eta.\end{aligned}$$

The last conclusion is proved. So, the conclusion holds. □

Based on Lemma 4.4, we can get one point x_k^j with $h(x_k^j) < \min\{\eta_1\tau^k, \|t_k^i d_k^j\|\}$ from the recovery algorithm. Then, according to Lemma 4.7, the loop between $(S.1)$ and $(S.3)$ terminates finitely.

Theorem 4.8 *Suppose Assumption* 4.1 *holds. A new point will be added into the sequence* $\{x_k\}$.

Proof Based on the Taylor expanding about $f(x_k + t_k d_k)$ at the point x_k and $M_k \succ 0$, it is easy to get

$$\begin{aligned}f(x_k + t_k d_k) &= f(x_k) + t_k Df(x_k)^T d_k + \frac{1}{2}(t_k d_k)^T D^2 f(y)(t_k d_k)\\ &< f(x_k) + t_k Df(x_k)^T d_k + \frac{1}{2}M(t_k d_k)^T(t_k d_k),\end{aligned}$$

where y locates on the segment between x_k and $x_k + t_k d_k$. When $q_k(0) - q_k(t_k d_k) > \delta h(x_k)^2$, we have

$$t_k Df(x_k)^T d_k < -\delta h(x_k)^2 - \frac{1}{2}t_k^2 d_k^T M_k d_k < 0,$$

and

$$\frac{f(x_k) - f(x_k + t_k d_k)}{q_k(0) - q_k(t_k d_k)} \geq 1 - \frac{\|t_k d_k\|^2 M}{q_k(0) - q_k(t_k d_k)} \geq 1 - \frac{\frac{M}{\|M_k\|}(\delta h(x_k)^2 + t_k Df(x_k)^T d_k)}{\delta h(x_k)^2}. \tag{4.22}$$

There exists $\bar{t}_{k1} \geq \frac{\delta h(x_k)^2}{-Df(x_k)^T d_k} \geq 2t_k^{\min}$ such that

$$\frac{f(x_k) - f(x_k + t_k d_k)}{q_k(0) - q_k(t_k d_k)} \geq \eta, \quad \forall t_k \leq \bar{t}_{k1}.$$

There are two cases to be discussed as follows.

The first case: $h(x_k) = 0$, there exists $\bar{t}_{k2} \geq t_k^{\min} = 0$, such that $h(x_k + t_k d_k) \leq \frac{1}{2}(l + 1)M\|t_k d_k\|^2 < \beta\tau^k$, $\forall t_k \leq \bar{t}_{k2}$. Therefore, $x_k + t_k d_k$ satisfies filter criterion.

The second case: $h(x_k) > 0$. Just as (S.3), we only need to consider the situation involving Algorithm 3.2.

According Lemma 4.4, a point x_k^r, which is generated by Algorithm 3.2, satisfies $h(x_k^r) \leq \eta_1\tau^k < \beta\tau^k$. So, there exists $\bar{t}_{k3} \geq \frac{2h(x_k)}{(l+1)d_k^T M_k d_k} \geq 2t_k^{\min}$, such that

$$h(x_k + t_k d_k) = (1 - t_k)h(x_k) + O(\|t_k d_k\|^2) \leq \beta\tau^k, \quad \forall t_k \leq \bar{t}_{k3}.$$

Therefore, $x_k + t_k d_k$ satisfies filter criterion.

Combining the above two cases with (4.22), we can get a new point $x_{k+1} = x_k + t_k d_k$ which is added into the sequence $\{x_k\}$. □

Theorem 4.9 *Suppose Assumption* 4.1 *holds. The sequence* $\{x_k\}$ *generated by Algorithm* 3.1*, either terminates at the* KKT *point, or produces an accumulation point which satisfies the* KKT *conditions.*

Proof First, we consider the case that $\{x_k\}$ contains an infinite number of h-type iterations. For an h-type iteration, x_k is always entered into the filter for a complete iteration, so it follows from Lemma 4.2 that $h(x_k) \to 0$ on this subsequence. It must also follow that $\tau_k \to 0$. Moreover, only h-type iteration can reset τ_k, so there exists a thinner infinite subsequence on which $\tau_{k+1} < h(x_k) = \tau_k$ is satisfied. Because X is bounded, there exist an accumulation point x_* and a subsequence index K, such that $x_k \to x_*,\ h(x_k) \to 0,\ k \in K$, and $\tau_{k+1} < h(x_k) = \tau_k$. So x_* is a feasible point. If x_* is not a KKT point, we show that this leads to a contradiction. Lemma 4.7 shows that the subproblem (1.2) is compatible at x_k, and

$$q_k(0) - q_k(t_k d_k) \geq \frac{1}{3}\|t_k d_k\|\varepsilon > \delta h^2(x_k), \quad \forall 0 < t_k \leq 1.$$

Thus, for $k \in K$ large enough, an f-type iteration will hold. This contradicts with the fact that the subsequence is generated by h-type iterations. So x_* is a KKT point.

Next, we consider the alternative case that the sequence $\{x_k\}$ contains only a finite number of h-type iterations. Hence, there exists an index K_1 such that all iterations are f-type iterations for all $k \geq K_1$. It follows that x_{k+1} is always acceptable to x_k, and

$$f(x_k) - f(x_k + t_k d_k) \geq \eta(q_k(0) - q_k(t_k d_k)) \geq \delta h(x_k)^2 > 0.$$

So, the sequence $\{f(x_k)\}$ is strictly monotonically decreasing for $k \geq K_1$. Therefore, it follows from Lemma 4.3 that $\lim h(x_k) \to 0$, hence any accumulation point x_* is a feasible point. Since $f(x)$ is bounded on X, it also follows that $\sum_{k \geq K_1}(f(x_k) - f(x_{k+1}))$ is convergent. If one accumulation point is not a KKT point, there exist a subsequence K_2 and a constant $\varrho > 0$, such that $\|d_k\| > \varrho$ for all $k \in K_2$.

From (4.6), if $h(x_k) \leq \beta\tau^k$ and $\|t_k d_k\| \leq \sqrt{\frac{2\beta\tau^k}{(l+1)M_k}}$, we have

$$h(x_k + t_k d_k) \leq \beta\tau^k. \tag{4.23}$$

We analyze by contradiction. If $\|t_k d_k\| \leq \frac{(1-\eta)\varepsilon}{3M}$, we have

$$f(x_k) - f(x_k + t_k d_k) \geq \eta(q_k(0) - q_k(t_k d_k)).$$

It follows as above that sufficient condition for accepting an f-type point is that

$$t_k \mu h(x_k) \leq \|t_k d_k\| \leq \min\{t_k \kappa, \tfrac{2\tau^k}{(l+1)M_k \kappa}, \tfrac{(1-\eta)\varepsilon}{3M}\}. \tag{4.24}$$

Since $\min\{\frac{2\tau^k}{(l+1)M_k\kappa}, \frac{(1-\eta)\varepsilon}{3M}\}$ on the right-hand side of (4.24) is a constant recorded as $\overline{td}$, while the left side converges to zero and t_k is decreasing in the inner loop, if $\kappa \geq \overline{td}$, (4.24) holds with $\|t_k d_k\| \geq \frac{1}{2}\overline{td}$; if $\kappa < \overline{td}$, (4.24) holds with $t_k = 1$. We then know from (4.24) that $\|t_k d_k\| \geq \min\{\frac{1}{2}\overline{td}, \varrho\}$.

According to the fact that $\min\{\frac{1}{2}\overline{td}, \varrho\} \gg h^2(x_k)$, we have

$$q_k(0) - q_k(t_k d_k) \geq \frac{1}{6}\varepsilon \min\{\overline{td}, 2\varrho\} \geq \delta h^2(x_k).$$

Then

$$\begin{aligned}\sum_{k\geq K_1} f(x_k) - f(x_{k+1}) &= \sum_{k\geq K_1} f(x_k) - f(x_k + t_k d_k) \geq \sum_{k\in K_1, k\geq K_2} \eta(q_k(0) - q_k(t_k d_k)) \\ &\geq \sum_{k\in K_1, k\geq K_2} \frac{1}{6}\eta\varepsilon \min\{\overline{td}, 2\varrho\} = +\infty,\end{aligned}$$

which contradicts the fact that $\sum_{k\geq K_1}(f(x_k) - f(x_{k+1}))$ is convergent. Thus x_* is a KKT point. The proof is finished. □

5 Numerical Experiments

In this section, a MATLAB code is written for the filter algorithm presented in Section 3. We use Jos Sturm' SeDuMi code [8] to test the feasibility of problem (1.2) and Restoration algorithm. The link between the MATLAB code and the SeDuMi is provided by the parser YALMIP [2].

In order to make a preliminary test of the algorithm, we select some examples of the publicly available benchmark collection COMPleib [10, 11], and some references therein.

With the data contained in COMPleib, it is possible to construct particular nonlinear semidefinite optimization problems arising in feedback control design [10]. We consider in our numerical tests only the basic Static (or reduced order) Output Feedback, H_2-SDP and H_∞-BMI problem. The reader can find more details on the motivation of this problem [2, 6, 9, 13, 14].

The following NLSDP formulation of the H_2-SDP and H_∞-BMI problems are considered,

$$\min\{\mathrm{Tr}(LB_1B_1^T) | A_F^T L + LA_F + C_F^T C_F = 0, A_F^T V + VA_F \prec 0, V \succ 0\} \tag{5.1}$$

$$\begin{aligned}&\min\ \mathrm{Tr}(X) \\ &\text{s.t.}\quad A_F Q + QA_F^T + B_1 B_1^T \preceq 0, Q \succ 0, \\ &\qquad \begin{bmatrix} X & C_F Q \\ QC_F^T & Q \end{bmatrix} \succeq 0,\end{aligned} \tag{5.2}$$

where $A_F = A + BFC$ and $C_F = C_1 + D_{12}FC$. The data A, B_1, B, C_1, C and D_{12} are extracted from COMPleib. In the problem (H_2-NSDP) the variables are the matrices L, V and F. The

L, V are symmetric and real, but the F, associated with SDP control law, is in general not square.

For all these numerical tests, the parameter values are selected as follows: $\delta = 0.2$, $\eta = 0.1$, $\gamma = 0.1$, $\beta = 0.9$, $\eta_1 = 0.5$, $\varepsilon = 10^{-6}$, and

$$F_0 = \begin{bmatrix} 1 \cdots 1 \\ \vdots \ddots \vdots \\ 1 \cdots 1 \end{bmatrix}_{n3\times n5}, \qquad L_0 = V_0 = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}_{n1\times n1}. \tag{5.3}$$

The results obtained for ($H2$-NSDP) are given in the next table.

Table 1 The detail information of numerical experiments

Date set	$n1$	$n3$	$n5$	P	f-itr	h-itr	FLV-norm	$f(x^*)$	$h(x^*)$	CPU-time
AC1	5	2	3	36	0	18	1.613720E-10	4.400000E-03	1.146600E-10	7.910595
AC2	5	5	3	45	0	17	5.345901E-11	5.600000E-03	1.645729E-10	7.32821
AC3	5	2	4	38	0	11	3.260891E-07	2.098550E+01	1.594100E-12	4.79347
AC4	4	2	1	22	0	2	1.993300E-08	1.215917E+02	5.329769E-13	1.349766
AC5	4	4	2	28	0	1	1.825143E-11	-5.198035E+00	5.329769E-13	0.461226
AC6	7	7	2	70	0	1	9.673774E-09	1.871739E+02	4.786215E-08	1.608399
AC7	9	1	1	91	1	1	4.213213E-12	1.542351E-03	9.317730E-12	2.722906
AC8	9	2	1	95	0	1	3.692761E-08	-3.348557E+00	3.776700E-07	2.132902
AC12	4	1	3	23	17	13	4.797330E-07	-9.643317E+00	1.007377E-09	14.120128
REA1	4	4	2	28	0	3	2.176100E-07	1.899313E+00	3.510300E-11	1.859684
REA2	4	4	2	28	0	2	2.176100E-07	3.332248E+00	4.941137E-11	1.705555
ROC2	10	1	1	154	2	6	9.361000E-07	-4.335303E-04	1.844348E-11	13.7017
HE1	4	2	2	24	0	7	3.527703E-10	-1.926451E-01	1.713281E-11	2.777024
HE2	4	4	2	28	0	1	4.616000E-10	-3.714650E+01	2.910100E-11	0.663644
HE3	8	10	4	112	0	23	1.116385E-09	1.101073E+00	8.991523E-10	29.580323
HE5	8	4	4	88	0	2	1.467400E-10	1.526623E+01	3.214600E-11	3.433427
DIS1	8	8	4	104	5	28	3.145000E-08	-6.805860E+01	6.180000E-10	32.989876
DIS2	3	3	2	18	4	2	3.379800E-08	-2.728800E+00	5.231000E-11	2.446329
DIS3	6	6	4	66	6	8	1.373000E-07	-4.834950E+01	2.180800E-10	9.572058
DIS4	6	6	4	66	8	11	4.224800E-09	-1.262960E+01	6.298700E-11	16.753892
DIS5	4	3	2	26	0	1	2.548700E-11	-4.010000E-02	8.696200E-11	0.513626
TG1	10	10	2	130	0	2	5.999700E-09	3.846617E+02	5.047515E-11	7.832082
WEC1	10	10	3	140	0	6	6.197960E-08	6.919465E+03	6.919465E+03	13.320438
EB1	10	2	1	112	1	2	2.913952E-08	-6.786449E-01	2.451396E-12	7.945267
EB2	10	2	1	112	1	3	2.000544E-12	-7.946488E-01	2.482662E-12	8.515224
EB3	10	2	1	112	0	5	3.580103E-11	-5.353461E+00	5.034508E-10	9.107625
NN1	3	3	1	15	0	2	1.614203E-10	-6.611111E+00	4.636953E-11	1.184888
NN2	2	2	1	8	0	1	1.023375E-10	-2.500000E+00	1.153396E-12	0.435164
NN3	4	1	1	21	2	3	5.872802E-07	6.908271E-01	1.495332E-10	2.354346
NN5	7	7	1	63	0	2	8.429720E-07	7.972685E+03	4.687160E-11	2.482027
NN6	9	9	1	99	0	2	5.671721E-09	2.410376E+05	3.011982E-10	4.130557
NN7	9	3	1	93	0	5	1.534142E-08	1.232876E+00	7.378014E-12	7.953678
NN10	8	2	3	78	0	3	2.864549E-07	0.000000E+00	1.063402E-10	6.269913
NN12	16	3	3	281	0	5	7.167800E-11	1.950000E-02	1.742000E-11	14.483057
NN13	6	6	2	54	0	1	3.512800E-12	-1.031476E+03	2.938816E-12	0.686661
NN14	6	3	2	48	0	1	5.290971E-11	6.972569E+02	8.789821E-11	0.709556
NN15	3	4	2	20	0	7	1.063860E-11	6.089412E-03	2.335593E-11	2.271124

The results obtained for H_∞-BMI are given in the next table.

Table 2 The detail information of numerical experiments

Date set	$n1$	$n3$	$n4$	$n5$	P	f-itr	h-itr	FLV-norm	$f(x^*)$	$h(x^*)$	CPU-time
AC1	5	3	2	3	27	6	2	1.270074E-08	4.803525E-02	5.621072E-12	3.361243
AC2	5	3	5	3	39	5	4	4.330750E-08	1.999563E-01	1.788460E-09	4.649376
AC3	5	2	5	4	38	4	2	2.871300E-10	2.873052E+01	2.299779E-12	3.221253
AC6	7	2	6	4	51	6	1	1.024628E-08	2.792167E+01	2.253981E-07	5.442844
AC7	9	1	1	2	48	1	2	1.073717E-08	1.553869E-03	5.437485E-07	2.407462
AC8	9	1	2	5	53	16	4	6.241822E-08	4.530955E+00	9.897606E-09	13.819975
AC15	4	2	6	3	37	4	5	5.460600E-07	2.788135E+02	6.021619E-07	4.546079
AC16	4	2	6	5	41	3	3	6.902790E-07	1.844990E+02	3.229229E-08	2.956544
AC17	4	1	4	2	22	2	2	5.591396E-10	1.894837E+01	1.067774E-09	1.836845
REA1	4	2	4	3	26	4	6	1.459189E-09	4.211260E+00	4.491661E-09	3.594247
HE2	4	2	4	2	24	1	7	2.167184E-08	1.210467E+01	5.173020E-08	2.807093
DIS1	8	4	8	4	88	37	3	9.857044E-07	4.617351E+01	5.679505E-09	18.537
DIS2	3	2	3	2	16	3	2	1.441517E-09	6.136059E+00	1.757512E-09	1.806747
DIS3	6	4	6	4	58	3	2	1.315240E-08	7.950581E+00	1.530289E-08	2.922814
DIS4	6	4	6	6	66	3	4	2.368565E-11	5.789123E+00	9.437750E-12	4.1261
MFP	4	3	4	2	26	2	8	9.010079E+01	9.010079E+01	4.583344E-08	3.767903
EB1	10	1	2	1	60	2	12	6.127437E-07	3.307067E+00	2.096467E-08	15.333061
EB3	10	1	2	1	59	0	2	4.723363E-09	8.845746E-01	7.848463E-09	2.508197
NN2	2	1	2	1	7	2	2	1.073834E-10	2.501830E+00	3.415311E-10	1.262522
NN4	4	2	4	3	26	4	2	7.889474E-11	7.293037E+00	6.516140E-12	2.187475
NN8	3	2	3	2	16	1	5	1.529508E-12	5.268370E+00	1.679649E-12	1.983806
NN10	8	3	2	3	48	0	1	5.042174E-12	2.050249E-12	2.759750E-11	0.615728
NN15	3	2	3	2	16	3	1	2.185543E-07	7.834253E-03	7.148670E-11	1.325343
NN17	3	2	2	1	11	8	1	9.127584E-07	7.769225E-01	2.02E-09	3.046686
PSM	7	2	5	3	49	1	3	5.918773E-08	4.176420E+00	8.08E-07	2.37508

Date set = the name of the example of COMPleib.

$n1$ = the dimension of the variable L, V.

$n3$ = the row number of the variable F.

$n5$ = the column number of the variable F.

$P = n1 * (n1 + 1) + n3 * n5$, number of variable.

FLV-norm = $\|\text{vec}(F); \text{vec}(L); \text{vec}(V)\|$, infinity norm used in Algorithm.

f-iter = the number of f-iterations.

h-iter = the number of h-iterations.

cpu time = the total cpu time (sec.) including restoration and the inner loops.

$f(x^*)$ = the value of f at the optimum.

$$h(x^*) = \|A_{F^*}^T L^* + L^* A_{F^*} + C_{F^*} C_{F^*}\|_2 + \lambda_1(A_{F^*}^T V^* + V^* A_{F^*})_+ + \lambda_1(-V^*)_+,$$

the value of h at the optimum.

6 Conclusion

In this paper, we propose a filter algorithm by changing the step size for nonlinear semidefinite programming. The global convergence of the filter method was obtained under quite mild assumptions, like MFCQ, boundedness, etc. The QSD subproblems at each step of the algorithm are actually linear semidefinite programming problems. We have performed some numerical experiments which are applied to optimal SDP problems. The restoration algorithm is described in a very simple way, which is just tried to obtain a feasible point by minimizing the constraint violation function h.

Acknowledgements The authors would like to thank the anonymous referee, whose constructive comments led to a considerable revision of the original paper.

References

[1] Christian, K., Christian, N., Hirokazu, K., et al.: Successive linearization methods for nonlinear semidefinite programs. *Comput. Optimiz. and Applic.*, **31**(3), 251–273 (2005)

[2] Fberg, J. L.: YALMIP: A toolbox for modeling and optimization in matlab. *Computer Aided Control Systems Design*, **9**, 284–289 (2004)

[3] Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. *Math. Programming*, **91**(2), 239–269 (2002)

[4] Fletcher, R., Leyffer, S., Toint, P. L.: On the global convergence of a filter-SQP algorthm. *SIAM J. Optimiz.*, **13**(1), 44–59 (2002)

[5] Freund, R. W., Jarre, F., Vogelbusch, C. H.: Nonlinear semidefinite programming: sensitivity, convergence, and an application in passive reduced order modeling. *Math. Programming*, **109**(2–3), 581–611 (2007)

[6] Gomez, W., Ramirez, H.: A filter algorithm for nonlinear semidefinite programming. *Comput. Applied Math.*, **29**(2), 297–328 (2010)

[7] Helmberg, C.: Semidefinite Programming for Combinatorial Optimization, Konrad-Zuse-Zentrum fur Informationstechnik, Berlin, 2000

[8] Huang, Y. S., MacTralance, A. G. J.: Multivariable Feedback: A Quasi-classical Approach. Lectures Notes in Control and Information, Springer-Verlag, New York, 1982

[9] Leibfritz, F., Lipinski, W.: Description of the brnchmark examples in compleib. Technical report, University of Trier, Department of Mathematics, D-54286 Trier, Germany, 2003

[10] Leibfritz, F.: Compleib: Constrained matrix-optimization problem library — a collection of test examples for nonlinear semidefinite programming, control system design and related problems. Technical report, University of Trier, Department of Mathematics, D-54286 Trier, Germany, 2004

[11] Leibfritz, F., Lipinski, W.: Compleib 1.0-user manual and quick reference. Technical report, University of Trier, Department of Mathematics, D-54286 Trier, Germany, 2004

[12] Leibfritz, F., Mostafa, E. M. E.: An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems. *SIAM J. Optimiz.*, **12**(4), 1048–1074 (2002)

[13] Leibfritz, F., Mostafa, M. E.: An interior point constrained trust region method for a special class of nonlinear semidefinite programming problems. *Math. Programming*, **12**(4), 1048–1074 (2004)

[14] Leibfritz, F., Mostafa, M. E.: Trust region methods for solving the optimal output feed-back design problem. *Intern. J. Control*, **76**(5), 501–519 (2003)

[15] Li, C. J., Sun, W. Y.: A filter-successive linearization methods for nonlinear semidefinite programs. *Sci. China, Ser. A*, **39**(8), 977–995 (2009)

[16] Li, C. J., Sun, W. Y.: Some properties for nonconvex semidefinite programming. *Numer. Math. J. Chin. Univ.*, **30**, 184–192 (2008)

[17] Nie, P. Y.: A trust region filter method for general non-linear programming. *Appl. Math. Comput.*, **172**(2), 1000–1017 (2006)

[18] Nie, P. Y.: Sequential penalty quadratic programming filter methods for nonlinear programming. *Nonlinear Anal.*: *Real World Applications*, **8**(1), 118–129 (2007)

[19] Nie, P. Y.: Composite-step like filter method for equality constraint problems. *J. Comput. Math.*, **21**(5), 613-624 (2003)

[20] Rafael, C., Hector, R. C.: A global algorithm for nonlinear semidefinite programming. *SIAM J. Optimiz.*, **15**(1), 303–318 (2004)

[21] Sun, D. F.: The strong second-order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their imlications. *Math. Operations Research*, **31**(4), 761–776 (2006)

[22] Sun, D. F., Sun, J., Zhang. L. W.: The rate of convergence of the augmented Lagrangian method for nonlinear semidefinite programming. *Math. Programming*, **114**(2), 349–391 (2008)

[23] Todd, M.: Semidefinite optimization. *Acta Numerica*, **10**, 515–560 (2001)

[24] Vandenberghe, L., Boyd, S.: Semidefinite programming. *SIAM Review*, **38**(1), 49–95 (1996)

[25] Wang, X. L., Zhu, Z. B., Huang, Q. Q.: An SQP-filter method for inequality constrained optimization and its global convergence. *Appl. Math. Comput.*, **217**(24), 10224–10230 (2011)

[26] Wolkowicz, H., Saigal, R., Vandenberghe, L.: Handbook of Semidefinite Programming Theory, Algorithms, and Applications, Kluwer Academic Publishers, Boston-Dordrecht-London, 2000

[27] Yamashita, Y., Yabe, H.: Local and superlinear convergence of a primal-dual interior point method for nonlinear semidefinite programming. *Math. Programming*, **132**(1–2), 1–30 (2012)