Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 293475, 7 pages
http://dx.doi.org/10.1155/2014/293475

Research Article

Improved Filter-SQP Algorithm with Active Set for Constrained Minimax Problems

Zhijun Luo[1] and Lirong Wang[2]

[1] *The Department of Mathematics and Econometrics, Hunan University of Humanities, Science and Technology, Loudi 417000, China*
[2] *The Department of Information Science and Engineering, Hunan University of Humanities, Science and Technology, Loudi 417000, China*

Correspondence should be addressed to Zhijun Luo; ldlzj123@163.com

Received 15 March 2014; Revised 19 July 2014; Accepted 25 August 2014; Published 2 September 2014

Academic Editor: Kazutake Komori

An improved filter-SQP algorithm with active set for constrained finite minimax problems is proposed. Firstly, an active constraint subset is obtained by a pivoting operation procedure. Then, a new quadratic programming (QP) subproblem is constructed based on the active constraint subset. The main search direction d^k is obtained by solving this (QP) subproblem which is feasible at per iteration point and need not to consider the penalty function by using the filter technique. Under some suitable conditions, the global convergence of our algorithm is established. Finally, some numerical results are reported to show the effectiveness of the proposed algorithm.

1. Introduction

Many real life problems, such as engineering, economics, management, finance, and other fields, can be described as the minimax problems, which wants to obtain the objection functions minimum under conditions of the maximum of the functions (such as [1, 2]). In this paper, consider the following constrained minimax optimization problems:

$$\begin{aligned} &\min_{x\in R^n} \quad F(x) \\ &\text{s.t.} \quad g_i(x) \le 0, \quad i \in I = \{1, 2, \ldots, m_1\}, \end{aligned} \tag{1}$$

where

$$F(x) = \max\left\{f_j(x) \mid j \in J = \{1, 2, \ldots, m\}\right\}, \tag{2}$$

$f_j(x)$, $g_i(x)$: $R^n \to R$, are continuously differentiable. For convenience, we denote

$$\begin{gathered} J(x) = \left\{j \in J \mid f_j(x) = F(x)\right\}, \\ I(x) = \left\{i \in I \mid g_i(x) = \varphi(x) \triangleq \max\left\{g_i(x), i \in I;\ 0\right\}\right\}. \end{gathered} \tag{3}$$

Obviously, the objective function $F(x)$ is not necessarily differentiable even if the $f_j(x)$, $j \in J$ are all differentiable. Consequently, the classical algorithms for smooth optimization problems may fail to reach an optimum if they are applied directly to the such constrained minimax optimization problem (1). Taking into account the value of minimax problems, many methods are proposed for solving problem (1). For example, in [3, 4], the minimax optimization problem is viewed as an unconstrained nonsmooth optimization problem, which can be solved by the general methods, such as subgradient methods, bundle methods, and cutting plane methods. The other type of methods which solves the problem (1) is so called smoothing methods, whose approach is to transform the minimax problem (1) into an equivalent smooth constrained nonlinear programming problem as follows:

$$\begin{aligned} &\min \quad z \\ &\text{s.t.} \quad f_j(x) \le z, \quad j \in J, \\ &\qquad g_i(x) \le 0, \quad i \in I, \end{aligned} \tag{4}$$

where $z \in R$ is an artificial variable. Obviously, from the problem (4), the Karush-Kuhn-Tucker (KKT) conditions of (1) can be stated as follows:

$$\begin{gathered}
\sum_{i\in I}\lambda_i \nabla f_i(x) + \sum_{j\in J}\mu_j \nabla g_j(x) = 0 \\
\lambda_i \geq 0, \\
f_i(x) - F(x) \leq 0, \qquad \lambda_i\left(f_i(x) - F(x)\right) = 0, \quad i \in I, \\
\mu_j \geq 0, \qquad g_j(x) \leq 0, \\
\mu_j g_j(x) = 0, \quad j \in J, \\
\sum_{i\in I}\lambda_i = 1,
\end{gathered} \tag{5}$$

where λ_i, μ_j are the corresponding vector. In view of the equivalent relationship between the KKT point of (4) and the stationary point of (1), many methods focus on finding the stationary point of the problem (1), namely, solving (5). And a lot of methods have been proposed to solve minimax problem [5–13]. For finding the minima of convex functions that are not necessarily differentiable, in [5–8], combining the nonmonotone line search with the second-order correction technique, which can effectively avoid Maratos effect, many other effective algorithms for solving the minimax problems are presented, such as [10–13].

To solve the minimax problem efficiently and save time, reduce computations, and reduce the number of iterations, we aim for a fast convergent algorithm. It is well-known that sequential quadratic programming (SQP) can be considered one of the best nonlinear programming methods for smooth constrained optimization problems (see, e.g., [14–20], etc.), which outperforms every other nonlinear programming method in terms of efficiency, accuracy, and percentage of successful solutions over a large number of test problems. Hence, some authors have directly applied SQP techniques to minimax problems and got some satisfactory results (such as [5, 9]). For typical SQP method, the main procedure in the method for minimum problem

$$\min\left\{f(x) \mid g_j(x) \leq 0, j \in J_0\right\} \tag{6}$$

is to solve the following quadratic programming:

$$\begin{aligned}
\min \quad & \nabla f(x)^T d + \frac{1}{2} d^T B d \\
\text{s.t.} \quad & g_j(x) + \nabla g_j(x)^T d \leq 0, \quad j \in J_0,
\end{aligned} \tag{7}$$

where J_0 is an index set and B is an approximation of Hessian matrix of Lagrangian function. Since the objective function $F(x)$ contains the max operator, it is continuous but nondifferentiable even if every constrained function $f_i(x)$ $(i \in I)$ is differentiable. Hence, this method may fail to reach an optimum for the minimax problem. In view of this, and combining with (4), in a similar way to [5], one considers the following quadratic programming through introducing an auxiliary variable z:

$$\begin{aligned}
\min \quad & z + \frac{1}{2} d^T H d \\
\text{s.t.} \quad & f_i(x) + \nabla f_i(x)^T d \leq z, \quad i \in I, \\
& g_j(x) + \nabla g_j(x)^T d \leq 0, \quad j \in J,
\end{aligned} \tag{8}$$

where H is a symmetric positive definite matrix. However, it is well-known that the solution d of (8) may not be a feasible descent direction and can not avoid the Maratos effect. Recently, many researches have extended the popular SQP scheme to the minimax problems (see [21–25], etc.). Jian et al. [22] and Hu et al. [23] process pivoting operation to generate an ε-active constraint subset associated with the current iteration point. At each iteration of their proposed algorithm, a main search direction is obtained by solving a reduced quadratic program which always has a solution.

As an alternative to merit function, in 2002, Fletcher et al. [26] proposed the filter-SQP method for inequality constrained optimization problems instead of classic merit function SQP methods. This method main idea is that a trial point was received if it improves either the objective function or the constraint violation. Furthermore, the global and superlinear local convergence of a trust region filter-SQP method was shown in Ulbrich [27]. In recent years, filter method got high importance and many algorithms have been paid [28–30], and so forth.

In this paper, an improved filter-SQP algorithm with active set for constrained minimax problems (1) is proposed. In the process of the iteration of our algorithm, an active constraint subset first is obtained by a pivoting operation procedure, and then, we construct a new quadratic programming (QP) subproblem based on the active constraint subset. In order to obtain a main search direction d^k for (1), we only need solve this (QP) subproblem which is feasible at per iteration point, and need not to consider the penalty function by using the filter technique. Furthermore, under some mild conditions, the global convergence of our algorithm is established.

The remaining part of this paper is organized as follows: an improved filter-SQP algorithm is proposed in Section 2. In Section 3, we prove that the algorithm is globally convergent. Some preliminary numerical tests are reported in Section 4. And concluding remarks are given in the last section.

2. Improved Filter-SQP Algorithm

As traditional filter technique, define the violation function $h(c(x))$ as follows:

$$h(c(x)) = \left\|c^+(x)\right\|_\infty, \tag{9}$$

where

$$\begin{gathered}
c_j(x) = f_j(x) - F(x) = f_j(x) - z \qquad (\text{let } z \triangleq F(x)), \\
c_j^+(x) = \max\left\{0, c_j(x), g_i(x)\right\}, \quad (i \in I, j \in J).
\end{gathered} \tag{10}$$

It is easy to see that $h(c(x)) = 0$ if and only if x is a feasible point.

Definition 1. A pair $(h(c(x^k)), z(x^k))$ is said to dominate another pair $(h(c(x^l)), z(x^l))$ if and only if both $h(c(x^k)) \leq h(c(x^l))$ and $z(x^k) \leq z(x^l)$.

Definition 2. A filter is a list of pairs $(h(c(x^k)), z(x^k))$ such that no pair dominates any other. A pair $(h(c(x^k)), z(x^k))$ is said to be acceptable for the filter if it is not dominated by any point in the filter.

We use $\mathscr{F}_k \triangleq \{(h(c(x^l)), z(x^l)) \in R^2, l < k\}$ to denote the set of iterations indices l $(l < k)$ such that $(h(c(x^l)), z(x^l))$ is an entry in the current filter. Then, we say that a point x^k is acceptable for the filter if and only if

$$\begin{aligned} & h\left(c\left(x^k\right)\right) \leq (1-\gamma)\, h\left(c\left(x^l\right)\right), \\ \text{or} \quad & z\left(x^k\right) \leq z\left(x^l\right) - \gamma h\left(c\left(x^l\right)\right), \end{aligned} \tag{11}$$

for all $(h(c(x^l)), z(x^l)) \in \mathscr{F}_k$, where γ is close to zero. We may also update the filter which means that the pair $(h(c(x^k)), z(x^k))$ is added to the list of pairs in the filter, and any pairs in the filter that are dominated by $(h(c(x^k)), z(x^k))$ are removed.

As a criterion for accepting or rejecting a trial step, we use the filter technique combined with SQP method.

Algorithm A

Step 0. Given initial point $x^0 \in R^n$, a symmetric positive definite matrix $H_0 \in R^{n\times n}$. Choose parameters $\varepsilon_0 > 0$, $\alpha \in (0, 1/2)$, $\tau \in (2, 3)$, and $0 < \gamma < \beta < 1$. Set $k = 0$, and $\mathscr{F}_0 = \{(u, +\infty)\}$ with some $u \geq \beta h(c(x^k))$.

Step 1. Computation of an active constraint set is as follows.

Step 1.1. Set $i = 0$ and $\varepsilon_{k_i} = \varepsilon_0$.

Step 1.2. Generate an ε-active constraint subset $I(x^k, \varepsilon_{k_i})$, $J(x^k, \varepsilon_{k_i})$ and matrix M_k by

$$\begin{gathered} I\left(x^k, \varepsilon_{k_i}\right) = \left\{i \in I \mid -\varepsilon_{k_i} \leq g_i\left(x^k\right) - \varphi\left(x^k\right) \leq 0\right\}, \\ J\left(x^k, \varepsilon_{k_i}\right) = \left\{j \in J \mid -\varepsilon_{k_i} \leq f_j\left(x^k\right) - F\left(x^k\right) \leq 0\right\}, \\ M_k = \begin{pmatrix} -1 \\ \nabla f_j\left(x^k\right) \\ \nabla g_i\left(x^k\right) \end{pmatrix}, \quad i \in I\left(x^k, \varepsilon_{k_i}\right), \; j \in J\left(x^k, \varepsilon_{k_i}\right). \end{gathered} \tag{12}$$

Step 1.3. If $\det(M_k^T M_k) \geq \varepsilon_{k_i}$, set

$$\begin{gathered} I_k = I\left(x^k, \varepsilon_{k_i}\right), \qquad J_k = I\left(x^k, \varepsilon_{k_i}\right), \\ L_k = I_k \bigcup J_k; \end{gathered} \tag{13}$$

go to Step 2; otherwise let $i = i + 1$, set $\varepsilon_{k_i} = 0.5\varepsilon_{k_i}$, and repeat Step 1.2.

Step 2. Compute (d^k, z_k) by the quadratic problem (14) at x^k. Consider

$$\begin{aligned} \min \quad & z + \frac{1}{2} d^T H_k d \\ \text{s.t.} \quad & f_j\left(x^k\right) + \nabla f_j\left(x^k\right)^T d - F\left(x^k\right) \leq z, \quad j \in J_k, \\ & g_i\left(x^k\right) + \nabla g_i\left(x^k\right)^T d \leq 0, \quad i \in I_k. \end{aligned} \tag{14}$$

Let (λ^k, μ^k) be the corresponding KKT multipliers vector. If $d^k = 0$ then stop.

Step 3. Compute $(\tilde{d}^k, \tilde{z}_k)$ by the quadratic problem (15). Consider

$$\begin{aligned} \min \quad & z + \frac{1}{2}\left(d^k + d\right)^T H_k \left(d^k + d\right) \\ \text{s.t.} \quad & f_j\left(x^k + d^k\right) + \nabla f_j\left(x^k + d^k\right)^T d - F\left(x^k + d^k\right) \leq z, \\ & g_i\left(x^k + d^k\right) + \nabla g_i\left(x^k + d^k\right)^T d \leq -\left\|d^k\right\|^\tau, \end{aligned} \tag{15}$$

where $j \in J_k$, $i \in I_k$. Set $(\tilde{\lambda}^k, \tilde{\mu}^k)$ as the corresponding KKT multipliers vector. If $\|\tilde{d}^k\| > \|d^k\|$, set $\tilde{d}^k = 0$; otherwise, let $d^k = d^k + \tilde{d}^k$.

Step 4. Initial line search: set $\alpha_{k,0} = 1$, $l = 0$.

Step 5. If $x^{k+1} = x^k + \alpha_{k,l} d^k$ is not acceptable for the filter, go to Step 6; otherwise let $\alpha_k = \alpha_{k,l}$, $x^{k+1} = x^k + \alpha_k d^k$, and add x^{k+1} to the filter; go to Step 7.

Step 6. Set $\alpha_{k,l+1} = \alpha_{k,l}/2$, $l = l + 1$, and go to Step 5.

Step 7. Update filter $\mathscr{F}_k$ to $\mathscr{F}_{k+1}$, and obtain H_{k+1} by updating the positive definite matrix H_k using some quasi-Newton formulas. Set $k := k + 1$. Go back to Step 1.

Remark 3. In Step 1, by using the the pivoting operation POP, we obtain an active set $L_k = I_k \bigcup J_k \subseteq I \bigcup J$. Based on the ε-active constraint subset, we construct a new QP (14), which is helpful for discussing the convergence of our algorithm.

Remark 4. Step 1.1–Step 1.3 and Step 4–Step 6 are called inner circle iteration, while Step 1–Step 7 are called outer circle steps.

3. Global Convergence of Algorithm

In this section, we analyze the convergence of the algorithm. The following general assumptions are true throughout this paper.

(*H 1*) The functions $f_j(x)$, $j \in J$, $g_i(x)$, $i \in I$, are continuously differentiable.

$(H\,2)$ $\forall x \in R^n$, the set of vectors

$$\left\{\begin{pmatrix} -1 \\ \nabla f_j(x) \end{pmatrix}, j \in J(x); \begin{pmatrix} 0 \\ \nabla g_i(x) \end{pmatrix}, i \in I(x)\right\} \tag{16}$$

is linearly independent.

$(H\,3)$ There exist a, $b > 0$, such that $a\|d\|^2 \le d^T H_k d \le b\|d\|^2$, for all $k \in R$ and $d \in R^n$.

Similar to Lemmas 2.1 and 2.3 in [22], the following lemma holds which describes some beneficial properties of the pivoting operation POP.

Lemma 5. *Suppose that $H\,1$–$H\,3$ hold and let $x_k \in R^n$. Then*

(1) *the pivoting operation POP can be finished in a finite number of computations; that is, there is no infinite times of loop between Step 1.2 and Step 1.3;*

(2) *if the sequence $\{x^k\}$ of points is bounded, then there exists a constant $\varepsilon > 0$ such that the associated sequence $\{\varepsilon_{k_i}\}$ of parameters generated by POP satisfies $\varepsilon_{k_i} \ge \bar{\varepsilon}$ for all k.*

Lemma 6. *Suppose that $H\,1$–$H\,3$ hold, matrix H_k is symmetric positive definite, and (d^k, z_k) is an optimal solution of* (14). *Then*

(1) $z_k + (1/2)(d^k)^T H_k d^k \le 0$, $z_k \le 0$;

(2) *if $d^k = 0$, then x^k is a K-T point of problem* (1);

(3) *if $d^k \ne 0$, then $z_k < 0$; moreover, d^k is a descent direction of $F(x)$ at point x^k.*

Lemma 7. *If $d^k \ne 0$, Step 4–Step 6 of Algorithm A are well defined; that is, the inner loop Step 5-Step 6 terminates finitely.*

Proof. By contradiction, if the conclusion is false, then Algorithm A will run infinitely between Step 5 and Step 6, so we have $\alpha_{k,l} \to 0$ $(l \to \infty)$ and the point $x^k + \alpha_{k,l}(d^k + \tilde{d}^k)$ is not acceptable for the filter. The following two cases need to be considered.

Case 1. Consider $h(c(x^k)) = 0$.

From the definition of $h(c(x^k))$, we can obtain

$$\begin{aligned} & h\left(c\left(x^k + \alpha_{k,l}\left(d^k + \tilde{d}^k\right)\right)\right) \\ & \quad = \max\Big\{0, f_j\left(x^k\right) + \alpha_{k,l}\nabla f_j\left(x^k\right)^T d^k - z_k + o\left(\alpha_{k,l}\right), \\ & \qquad\qquad g_i\left(x^k\right) + \alpha_{k,l}\nabla g_i\left(x^k\right)^T d^k + o\left(\alpha_{k,l}\right)\Big\}. \end{aligned} \tag{17}$$

Since d^k is a solution of the problem (14), then

$$\nabla f_j\left(x^k\right)^T d^k < 0, \qquad \nabla g_i\left(x^k\right)^T d^k < 0. \tag{18}$$

Together with $\alpha_{k,l} \to 0$, there exists a constant β, such that

$$\begin{aligned} & h\left(c\left(x^k + \alpha_{k,l}\left(d^k + \tilde{d}^k\right)\right)\right) \\ & \quad \le \max\left\{0, \beta\left(f_j\left(x^k\right) - z_k, g_i\left(x^k\right)\right)\right\} \\ & \quad = \beta h\left(c\left(x^k\right)\right) \triangleq (1-\gamma)\, h\left(c\left(x^k\right)\right). \end{aligned} \tag{19}$$

Moreover, for $f_j(x^k) - F(x^k) < 0$ and $1 - \alpha_{k,l} > 0$, we have

$$\begin{aligned} & f_j\left(x^k + \alpha_{k,l}\left(d^k + \tilde{d}^k\right)\right) - F\left(x^k\right) \\ & \quad = f_j\left(x^k\right) + \alpha_{k,l}\nabla f_j\left(x^k\right)^T d^k + o\left(\alpha_{k,l}\right) - F\left(x^k\right) \\ & \quad = \alpha_{k,l}\left(f_j\left(x^k\right) + \nabla f_j\left(x^k\right)^T d^k - F\left(x^k\right)\right) \\ & \qquad + \left(1 - \alpha_{k,l}\right)\left(f_j\left(x^k\right) - F\left(x^k\right)\right) + o\left(\alpha_{k,l}\right) \\ & \quad \le \alpha_{k,l} z_k + o\left(\alpha_{k,l}\right). \end{aligned} \tag{20}$$

With (19) and (20), we conclude that $x^k + \alpha_{k,l}(d^k + \tilde{d}^k)$ must be acceptable for the filter and x^k, which is a contradiction.

Case 2. Consider $h(c(x^k)) > 0$. Considering Taylor's formula, we have

$$\begin{aligned} & f_j\left(x^k + \alpha_{k,l}\left(d^k + \tilde{d}^k\right)\right) - F\left(x^k\right) \\ & \quad = f_j\left(x^k\right) + \alpha_{k,l}\nabla f_j\left(x^k\right)^T\left(d^k + \tilde{d}^k\right) \\ & \qquad - F\left(x^k\right) + \frac{\alpha_{k,l}^2}{2} d^{kT}\nabla f_j^2(y)\, d^k \\ & \quad \le \frac{\alpha_{k,l}^2}{2} d^{kT}\nabla f_j^2(y)\, d^k, \end{aligned} \tag{21}$$

where y denotes some point on the line segment from x^k to $x^k + \alpha_{k,l}(d^k + \tilde{d}^k)$. Since x^k is acceptable for the filter, we have

$$h\left(c\left(x^k\right)\right) \le (1-\gamma)\, h\left(c\left(x^l\right)\right), \tag{22}$$

or

$$z\left(x^k\right) \le z\left(x^l\right) - \gamma h\left(c\left(x^l\right)\right). \tag{23}$$

Similar to Case 1, we can also get the relation

$$h\left(c\left(x^k + \alpha_{k,l}\left(d^k + \tilde{d}^k\right)\right)\right) \le (1-\gamma)\, h\left(c\left(x^k\right)\right). \tag{24}$$

From the assumption, $x^k + \alpha_{k,l}(d^k + \tilde{d}^k)$ is not acceptable for the filter, and we have

$$h\left(c\left(x^k + \alpha_{k,l}\left(d^k + \tilde{d}^k\right)\right)\right) > (1-\gamma)\, h\left(c\left(x^l\right)\right), \tag{25}$$

$$z\left(x^k + \alpha_{k,l}\left(d^k + \tilde{d}^k\right)\right) > z\left(x^l\right) - \gamma h\left(c\left(x^l\right)\right). \tag{26}$$

For the point x^k, if it holds that (22), by $\alpha_{k,l} \to 0$ and (18), we have

$$\begin{aligned} &h\left(c\left(x^k + \alpha_{k,l}\left(d^k + \tilde{d}^k\right)\right)\right) \\ &\quad \leq \max\left\{0, \left(f_j\left(x^k\right) - z_k, g_i\left(x^k\right)\right)\right\} \\ &\quad = h\left(c\left(x^k\right)\right) \leq (1-\gamma)\, h\left(c\left(x^l\right)\right), \end{aligned} \tag{27}$$

which contradicts (25). If inequality (23) holds, by $\alpha_{k,l} \to 0$ and (18), we have

$$z\left(x^k + \alpha_{k,l}\left(d^k + \tilde{d}^k\right)\right) \leq z_k \leq z\left(x^l\right) - \gamma h\left(c\left(x^l\right)\right), \tag{28}$$

which contradicts (26). From the above analysis, the desired conclusion holds. □

Lemma 8. *Suppose that infinite points are added to the filter; then* $\lim_{k\to\infty} h(c(x^k)) = 0$.

In the following of this section, we will show the global convergence of the algorithm.

Theorem 9. *Suppose that H 1–H 3 hold; let $\{x^k\}$ be the sequence of iterates produced by Algorithm A. The algorithm either stops at the KKT point x^k of the problem* (1) *in finite number of steps or generates an infinite sequence $\{x^k\}$ of points such that each accumulation point x^* of $\{x^k\}$ is the KKT point of problem* (1).

Proof. The first statement is easy to show, the only stopping point being in Step 1. Thus, assume that the algorithm generates an infinite sequence $\{x^k\}$, and since $\{d^k, z_k, \lambda^k, \mu^k\}$ is bounded under all above-mentioned assumptions, we can assume without loss of generality that there exists an infinite index set K such that

$$\begin{gathered} x^k \longrightarrow x^*, \quad H_k \longrightarrow H_*, \quad d^k \longrightarrow d^*, \quad z_k \longrightarrow z_*, \\ \lambda^k \longrightarrow \lambda^*, \quad \mu^k \longrightarrow \mu^*, \quad k \in K. \end{gathered} \tag{29}$$

Obviously, according to Lemma 6, it is only necessary to prove that $d^* = 0$.

Let $K_1 = \{k \in K \mid z_k + (1/2) d^{kT} H_k d^k > 0\} \subseteq K$; two cases need to be considered.

Case 1. K_1 is an infinite index set. Suppose by contradiction that $d^* \neq 0$; since

$$\begin{gathered} f_j\left(x^k\right) + \nabla f_j\left(x^k\right)^T d^k - F\left(x^k\right) \leq z_k, \quad j \in J, \\ g_i\left(x^k\right) + \nabla g_i\left(x^k\right)^T d^k \leq 0, \quad i \in I, \end{gathered} \tag{30}$$

in view of $k \in K$, $k \to \infty$, we obtain

$$\begin{gathered} f_j\left(x^*\right) + \nabla f_j\left(x^*\right)^T d^* - F\left(x^*\right) \leq z_*, \quad j \in J, \\ g_i\left(x^*\right) + \nabla g_i\left(x^*\right)^T d^* \leq 0, \quad i \in I. \end{gathered} \tag{31}$$

It is shown that the following corresponding quadratic programming subproblem (32) at x^*

$$\begin{aligned} \min \quad & z + \frac{1}{2} d^T H_* d \\ \text{s.t.} \quad & f_j\left(x^*\right) + \nabla f_j\left(x^*\right)^T d - F\left(x^*\right) \leq z, \quad j \in J, \\ & g_i\left(x^*\right) + \nabla g_i\left(x^*\right)^T d \leq 0, \quad i \in I, \end{aligned} \tag{32}$$

has a nonempty feasible set. Moreover, from $d^* \neq 0$ and Theorem 2.4 in [9], it is not difficult to show that (z_*, d^*) is the unique solution of (32). So, it holds that

$$\begin{gathered} z_* < 0, \qquad \nabla f_j\left(x^*\right)^T d^* \leq z_* < 0, \quad j \in J\left(x^*\right), \\ \nabla g_i\left(x^*\right)^T d^* < 0, \quad i \in I\left(x^*\right). \end{gathered} \tag{33}$$

Considering the KKT conditions of the problem (14), we have

$$\begin{gathered} \lambda_j^* \nabla f_j\left(x^*\right)^T d^{=} d^{*T} H_* d^* = \lambda_j^* \left(f_j\left(x^*\right) - F\left(x^*\right) - z_*\right) \\ z_* + d^{*T} H_* d^* = z_* \left(1 - \lambda_j^*\right) < 0, \end{gathered} \tag{34}$$

which contradicts the definition of K_1.

Case 2. K_1 is a finite index set. That means it holds $z_k < -(1/2) d^{kT} H_k d^k$ for large enough k. There exists a constant $\overline{\alpha} > 0$, for $\alpha \in (0, \overline{\alpha})$; we have

$$\begin{aligned} &F\left(x^k\right) - f_j\left(x^k + \alpha\left(d^k + \tilde{d}^k\right)\right) \\ &\quad = -f_j\left(x^k\right) - \alpha \nabla f_j\left(x^k\right)^T d^k - o(\alpha) + F\left(x^k\right) \\ &\quad \geq -\alpha\left(f_j\left(x^k\right) + \nabla f_j\left(x^k\right)^T d^k - F\left(x^k\right)\right) \\ &\qquad - (1-\alpha)\left(f_j\left(x^k\right) - F\left(x^k\right)\right) + o(\alpha) \\ &\quad \geq -\alpha z_k + o(\alpha) \geq \frac{\alpha}{2} d^{kT} H_k d^k. \end{aligned} \tag{35}$$

Then, for some integer i_0, we have

$$\infty > \sum_{k=i_0}^{\infty}\left(F\left(x^k\right) - f_j\left(x^k + \alpha\left(d^k + \tilde{d}^k\right)\right)\right) \geq \sum_{k=i_0}^{\infty} \frac{a\alpha}{2}\left\|d^k\right\|^2. \tag{36}$$

That means $\|d^k\|^2 \to 0$. Thereby, x^* is a KKT point of problem (1). □

4. Numerical Experiments

In this section, we select some problems in [9, 10] to show the efficiency of our algorithm in Section 2. Some preliminary numerical experiments are tested on an Intel(R) Celeron(R) CPU 2.40 GHz computer. The code of the proposed algorithm is written by using MATLAB 7.0 and utilized the optimization toolbox to solve the quadratic programming (14) and (15). The results show that the proposed algorithm is efficient.

During the numerical experiments, it is chosen at random some parameters as follows.

(1) Consider $\varepsilon_0 = 0.5, \alpha = 0.25, \gamma = 0.1, \beta = 0.2, \tau = 2.25$, and $H_0 = I$, the $n \times n$ unit matrix.

(2) H_k is updated by the BFGS formula similar to [15]. Consider

$$H_{k+1} = H_k - \frac{H_k s_k (H_k s_k)^T}{s_k^T H_k s_k} + \frac{\widehat{\eta}_k \widehat{\eta}_k^T}{s_k^T \widehat{\eta}_k}, \tag{37}$$

where

$$\begin{gathered} s_k = x^{k+1} - x^k, \qquad \widehat{\eta}_k = \theta_k \gamma_k + (1 - \theta_k) H_k s_k, \\ \widehat{\gamma}_k = \nabla_x L\left(x^{k+1}, \lambda^k, \mu^k\right) - \nabla_x L\left(x^k, \lambda^k, \mu^k\right), \\ \nabla_x L\left(x^k, \lambda^k, \mu^k\right) = \sum_{i \in I} \lambda_i \nabla f_i(x) + \sum_{j \in J} \mu_j \nabla g_j(x), \\ \theta_k = \begin{cases} 1, & s_k^T \widehat{\gamma}_k \geq 0.2 s_k^T H_k s_k, \\ \dfrac{0.8 s_k^T H_k s_k}{s_k^T H_k s_k - s_k^T \widehat{\gamma}_k}, & s_k^T \widehat{\gamma}_k < 0.2 s_k^T H_k s_k. \end{cases} \end{gathered} \tag{38}$$

(3) In the implementation, the stopping criterion of Step 2 is changed to $If \|d^k\| \leq 10^{-6}$ STOP.

The algorithm has been tested on some problems from [9, 10]. The results are summarized in Tables 1 and 2. The columns of this table have the following meanings:

No.: the number of the test problem in [9, 10];

n: the dimension of the problem;

m: the number of objective functions;

m_1: the number of inequality constraints;

NT: the number of iterations;

IP: the initial point;

LWM: the proposed Algorithm A;

XUE: the method in [9];

RNM: the method in [10];

ZZM: the method in [21];

FV: the final value of the objective function.

In Table 2, the performance of algorithm LWM is compared with other algorithms. For problem 1 and 2, the results we get are a little better than those in [9] if we choose appropriate initial point. From the iteration results for test problems 3 to 7, it seems that our method is a bit more efficient than that in [10, 21] if the number of iterations is considered.

Table 1: The information of numerical experiments.

No.	n,	m,	m_1,	IP
1 (problem 1 in [9])	2,	3,	0,	$(1,5)^T$
2 (problem 4 in [9])	2,	3,	0,	$(3,1)^T$
3 (problem 1 in [10])	2,	3,	2,	$(0,0)^T$
4 (problem 2 in [10])	2,	6,	2,	$(1,3)^T$
5 (problem 4 in [10])	2,	3,	2,	$(4,2)^T$
6 (problem 5 in [10])	4,	4,	3,	$(0,1,1,0)^T$
7 (problem 6 in [10])	2,	3,	2,	$(0,1)^T$

Table 2: Test results obtained by Algorithm A and other relearnt algorithms.

No.	Method	NT	FV
1 (problem 1 in [9])	LWM	9	1.952224
	XUE	11	1.952224
	ZZM	11	1.952224
2 (problem 4 in [9])	LWM	11	0.616234
	XUE	15	0.616234
	ZZM	11	0.616234
3 (problem 1 in [10])	LWM	5	1.952224
	RNM	10	1.952224
	ZZM	10	1.952224
4 (problem 2 in [10])	LWM	10	0.616432
	RNM	15	0.616234
	ZZM	12	0.616234
5 (problem 4 in [10])	LWM	8	2.250000
	RNM	15	2.250000
	ZZM	11	2.250000
6 (problem 5 in [10])	LWM	21	−44.000000
	RNM	23	−44.000000
	ZZM	23	−43.998000
7 (problem 6 in [10])	LWM	4	2.000000
	RNM	4	2.000000
	ZZM	4	2.000000

5. Concluding Remarks

In this paper, we propose a filter method combining this method with sequential quadratic programming algorithm for inequality constrained minimax problems. With the help of the pivoting operation procedure, an active constraint subset is first obtained. At each iteration, a main search direction is obtained by solving only one quadratic programming subproblem which is feasible at per iteration point and need not to consider the penalty function by using the filter technique. Then, a correction direction is yielded by solving another quadratic programming to avoid Maratos effect and guarantee the global convergence properties under mild conditions. The preliminary numerical results also show that the proposed algorithm is effective.

However, to show that our algorithm is global convergent, we suppose some rigorous conditions such as the hypotheses *H 2-H 3*. We hope that we can get rid of them in our future work. In addition, it is noted that there are still some problems worthy further discussion such as studying the algorithm with inequality and equality constraints. And we can get the main search direction by other techniques, for example, sequential systems of linear equations technique.

Conflict of Interests

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

Acknowledgments

The authors are deeply indebted to the editor Professor Wenyu Sun and the anonymous referees whose insightful comments helped the authors a lot to improve the quality of the paper. The first author would also like to thank Professor Zhibin Zhu for valuable work on numerical experiments. This research was supported by Scientific Research Fund of Hunan Provincial Education Department (nos. 12A077,12C0743,13C453, and 14C0609).

References

[1] A. Baums, “Minimax method in optimizing energy consumption in real-time embedded systems,” *Automatic Control and Computer Sciences*, vol. 43, no. 2, pp. 57–62, 2009.

[2] E. Y. Rapoport, “Minimax optimization of stationary states in systems with distributed parameters,” *Journal of Computer and Systems Sciences International*, vol. 52, no. 2, pp. 165–179, 2013.

[3] E. Polak, D. Q. Mayne, and J. E. Higgins, “Superlinearly convergent algorithm for min-max problems,” *Journal of Optimization Theory and Applications*, vol. 69, no. 3, pp. 407–439, 1991.

[4] R. Reemtsen, “A cutting plane method for solving minimax problems in the complex plane,” *Numerical Algorithms*, vol. 2, no. 3-4, pp. 409–436, 1992.

[5] J. L. Zhou and A. L. Tits, “Nonmonotone line search for minimax problems,” *Journal of Optimization Theory and Applications*, vol. 76, no. 3, pp. 455–476, 1993.

[6] L. Grippo, F. Lampariello, and S. Lucidi, “A nonmonotone line search technique for Newton's method,” *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, 1986.

[7] Y. H. Yu and L. Gao, “Nonmonotone line search algorithm for constrained minimax problems,” *Journal of Optimization Theory and Applications*, vol. 115, no. 2, pp. 419–446, 2002.

[8] F. Wang and Y. Wang, “Nonmonotone algorithm for minimax optimization problems,” *Applied Mathematics and Computation*, vol. 217, no. 13, pp. 6296–6308, 2011.

[9] Y. Xue, “A SQP Method for Minimax Problems,” *Journal of System Science and Math Science*, vol. 22, no. 3, pp. 355–364, 2002 (Chinese).

[10] B. Rustem and Q. Nguyen, “An algorithm for the inequality-constrained discrete min-max problem,” *SIAM Journal on Optimization*, vol. 8, no. 1, pp. 265–283, 1998.

[11] E. Obasanjo, G. Tzallas-Regas, and B. Rustem, “An interior-point algorithm for nonlinear minimax problems,” *Journal of Optimization Theory and Applications*, vol. 144, no. 2, pp. 291–318, 2010.

[12] B. Rustem, S. Žakovic, and P. Parpas, “An interior point algorithm for continuous minimax: implementation and computation,” *Optimization Methods and Software*, vol. 23, no. 6, pp. 911–928, 2008.

[13] Y. Feng, L. Hongwei, Z. Shuisheng, and L. Sanyang, “A smoothing trust-region Newton-CG method for minimax problem,” *Applied Mathematics and Computation*, vol. 199, no. 2, pp. 581–589, 2008.

[14] S. P. Han, “A globally convergent method for nonlinear programming,” *Journal of Optimization Theory and Applications*, vol. 22, no. 3, pp. 297–309, 1977.

[15] M. J. D. Powell, “A fast algorithm for nonlinearly constrained optimization calculations,” in *Numerical Analysis*, pp. 144–157, Springer, Berlin, Germany, 1978.

[16] G. He, Z. Gao, and Y. Lai, “New sequential quadratic programming algorithm with consistent subproblems,” *Science in China A: Mathematics*, vol. 40, no. 2, pp. 137–150, 1997.

[17] E. R. Panier and A. L. Tits, “A superlinearly convergent feasible method for the solution of inequality constrained optimization problems,” *SIAM Journal on Control and Optimization*, vol. 25, no. 4, pp. 934–950, 1987.

[18] Z. Wan, “A modified SQP algorithm for mathematical programs with linear complementarity constraints,” *ACTA Scientiarum Naturalium Universitatis Normalis Hunanensis*, vol. 26, pp. 9–12, 2001.

[19] Z. Zhu, K. Zhang, and J. Jian, “An improved SQP algorithm for inequality constrained optimization,” *Mathematical Methods of Operations Research*, vol. 58, no. 2, pp. 271–282, 2003.

[20] Z. Luo, G. Chen, S. Luo, and Z. Zhu, “Improved feasible SQP algorithm for nonlinear programs with equality constrained sub-problems,” *Journal of Computers*, vol. 8, no. 6, pp. 1496–1503, 2013.

[21] Z. Zhu and C. Zhang, “A superlinearly convergent sequential quadratic programming algorithm for minimax problems,” *Journal of Numerical Methods and Applications*, vol. 27, no. 4, pp. 15–32, 2005.

[22] J. Jian, R. Quan, and Q. Hu, “A new superlinearly convergent SQP algorithm for nonlinear minimax problems,” *Acta Mathematicae Applicatae Sinica*, vol. 23, no. 3, pp. 395–410, 2007.

[23] Q. Hu, Y. Chen, N. Chen, and X. Li, “A modified SQP algorithm for minimax problems,” *Journal of Mathematical Analysis and Applications*, vol. 360, no. 1, pp. 211–222, 2009.

[24] W. Xue, C. Shen, and D. Pu, “A new non-monotone SQP algorithm for the minimax problem,” *International Journal of Computer Mathematics*, vol. 86, no. 7, pp. 1149–1159, 2009.

[25] J. Jian, X. Zhang, R. Quan, and Q. Ma, “Generalized monotone line search SQP algorithm for constrained minimax problems,” *Optimization*, vol. 58, no. 1, pp. 101–131, 2009.

[26] R. Fletcher, S. Leyffer, and P. L. Toint, “On the global convergence of a filter—SQP algorithm,” *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 44–59, 2002.

[27] S. Ulbrich, “On the superlinear local convergence of a filter-SQP method,” *Mathematical Programming*, vol. 100, no. 1, Ser. B, pp. 217–245, 2004.

[28] K. Su and J. Che, “A modified SQP-filter method and its global convergence,” *Applied Mathematics and Computation*, vol. 194, no. 1, pp. 92–101, 2007.

[29] C. Shen, W. Xue, and X. Chen, “Global convergence of a robust filter SQP algorithm,” *European Journal of Operational Research*, vol. 206, no. 1, pp. 34–45, 2010.

[30] C. Gu and D. Zhu, “A non-monotone line search multidimensional filter-SQP method for general nonlinear programming,” *Numerical Algorithms*, vol. 56, no. 4, pp. 537–559, 2011.