Journal of Computational Mathematics
Vol.30, No.3, 2012, 262–278.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1109-m3567

A NEW TRUST-REGION ALGORITHM FOR FINITE MINIMAX PROBLEM*

Fusheng Wang   Chuanlong Wang
Department of Mathematics, Taiyuan Normal University, Taiyuan 030012, China
Email: fswang2005@163.com   clwang218@126.com
Li Wang
Department of Mathematics, University of California, San Diego, USA
Email: agathapasley@gmail.com

Abstract

In this paper, a new trust region algorithm for minimax optimization problems is proposed, which solves only one quadratic subproblem based on a new approximation model at each iteration. The approach is different with the traditional algorithms that usually require to solve two quadratic subproblems. Moreover, to avoid Maratos effect, the nonmonotone strategy is employed. The analysis shows that, under standard conditions, the algorithm has global and superlinear convergence. Preliminary numerical experiments are conducted to show the effiency of the new method.

Mathematics subject classification: 90C47, 49K35.
Key words: Trust-region methods, Minimax optimization, Nonmonotone strategy, Global convergence, Superlinear convergence.

1. Introduction

In this paper, we study the finite minimax problem of the form

$$(P): \min_{x\in\mathbf{R}^n} \max_{1\le i\le m} f_i(x), \tag{1.1}$$

where $f_i:\mathbf{R}^n\to\mathbf{R}$ are twice continuously differentiable. Many problems of interest in real world applications can be modeled as finite minimax problems (P). This class of problems occur, for instance, in curve fitting, $\mathbf{L}_1$ and $\mathbf{L}_\infty$ approximation problems, systems of nonlinear equations [1], problems finding feasible points of systems of inequalities, nonlinear programming problems, multiobjective problems, engineering design, optimal control and etc., which show that the finite minimax problem is a very important class of nonsmooth optimization problems. At present, many algorithms have been developed, which can be classified into two classes. One is that, the problem (P) can be viewed as an unconstrained nondifferentiable optimization problems, so we can use the methods for solving general nondifferentiable optimization problems, such as subgradient methods, bundle methods and cutting plane methods to solve it(see [2-7]). The other is that, in view of the particular structure of its nondifferentiability, it's also suitable to make use of smooth optimization methods which based on the well-known fact that the problem (P) is equivalent to a smooth optimization problem on the $n+1$ variables (x,z):

$$\begin{cases} \min\limits_{(x,z)\in\mathbf{R}^{n+1}} z, \\ s.t. \quad f_i(x)-z\le 0, \quad i=1,\cdots,m, \end{cases} \tag{1.2}$$

* Received October 30, 2010 / Revised version received July 22, 2011 / Accepted July 29, 2011 / Published online May 7, 2012 /

where $x \in \mathbf{R}^n$ and $z \in \mathbf{R}$. Many line search algorithms were proposed by using this features (see [8-16]), under mild assumptions, these methods have good properties of both global convergence and locally superlinear convergence.

In history, most algorithms on minimax problems are line search rather than trust region based [17]. It's well known that trust-region methods are very efficient for smooth optimization problems, and they usually induce strong global convergence. Fletcher [18, 19] first applied trust-region methods to a class of composite nonsmooth optimization problems, and proposed a good trust-region algorithm, where the minimax problem can be regarded as its special case. Furthermore, Yuan [20] proved that it had a rate of superlinear convergence. However, it requires to compute the exact Hessian matrices $\nabla^2 f_i(x),\ i = 1, \cdots, m$ in the two subproblems at each iteration, which causes many gradient evaluations. Other trust-region methods that can be used to solve minimax problems were presented in [17], for example, Algorithm 11.3.1 thereof, but it is suitable to the general nonsmooth optimization problems.

Recently, the nonmonotone strategy has attracted attention from more and more researchers (see[21-24]), since Panier and Tits [21] indicated that, as an improvement of strategy of monotonic relaxation, it can prevent the Maratos effect when applied to the SQP algorithms for smooth optimization problems. Xue [8] proposed a new approximation model to the objective maximum function for minimax problems, which demonstrates some good properties. In addition, Wang and Zhang [25] proposed an algorithm based on trust region methods. Motivated by [8, 21], in this paper we develop a new trust region algorithm for finite minimax problems. On the one hand, unlike the line search methods such as [8–10, 15] in which the approximation Hessian matrices B_k have to be positive definite, the new algorithm does not require B_k to keep positive definite. On the other hand, the new algorithm is also different from [25]. First, it solves only one quadratic subproblem at each iteration. Second, it employs the nonmonotone strategy to overcome the Maratos effect. Third, it employs a new approximation model proposed in [8], and the corresponding subproblem is more stable. Under mild conditions, the global and superlinear convergence are obtained. Preliminary numerical experiments show that the proposed algorithm is robust and efficient.

The paper is organized as follows: In Section 2, we briefly recall trust-region methods; The algorithm is presented in Section 3. In Section 4, we analyze the global convergence and the rate of local convergence. In Section 5, we report some numerical results. Finally, we end the paper with conclusions.

We shall use the following notations and terminology. Unless otherwise stated, the vector norm used in this paper is Euclidean vector norm on $\mathbf{R}^n$, and the matrix norm is the induced operator norm on $\mathbf{R}^{n\times n}$. In addition, we denote

$$\phi(x) = \max_{1\le i\le m} f_i(x), \qquad \phi_k = \phi(x_k), \tag{1.3a}$$

$$I_A(x) = \Big\{i : f_i(x) = \phi(x)\Big\}, \qquad I_N(x) = \{i : f_i(x) < \phi(x)\}, \tag{1.3b}$$

$$f(x) = (f_1(x), \cdots, f_m(x))^T, \qquad \nabla f(x) = (\nabla f_1(x), \cdots, \nabla f_m(x)), \tag{1.3c}$$

$$F(x) = diag(f_1(x), \cdots, f_m(x)), \qquad e = (1, \cdots, 1)^T, \quad e \in \mathbf{R}^m. \tag{1.3d}$$

2. Trust Region Methods

Without loss of generality, in this section, we consider unconstrained optimization problem $\min_{x\in\mathbf{R}^n} f(x)$.

Trust-region methods have been paid much attention since their emergence ([17,26,27]) because of their strong global convergence. For easy reference, let us recall the trust-region methods first. Trust region methods are based on a local quadratic model of $f(x_k+s)-f(x_k)$ around the k-th iterate x_k defined by (2.1a), and they produce a trial step by solving the following trust region subproblem:

$$\begin{cases} \min g_k^T s + \frac{1}{2}s^T B_k s = m_k(s), & (2.1a) \\ s.t. \|s\|_2 \le \Delta_k, & (2.1b) \end{cases}$$

where step $s \in R^n$, $g_k = \nabla f(x_k)$, B_k is an $n \times n$ symmetric matrix which approximates the Hessian of objective function or chosen to be the exact Hessian matrix $B_k = \nabla^2 f(x_k)$, and $\Delta_k > 0$ is a trust-region radius. Let s_k be the solution of subproblem (2.1), $Pred_k = m_k(0) - m_k(s_k)$ be the predicted reduction in the approximate model $m_k(s)$, $Ared_k = f(x_k) - f(x_k+s_k)$ be the actual reduction in the objective function. The ratio between the actual reduction and the predicted reduction $r_k = Ared_k/Pred_k$ plays a very important role in the trust-region methods, since this ratio is used to decide whether the trial step is accepted and how to adjust the trust-region radius Δ_k.

3. The Algorithm

For problem (P), the Lagrangian function is defined by

$$L(x,\lambda) = \sum_{i=1}^{m} \lambda_i f_i(x). \tag{3.1}$$

In this paper, we intend to use (1.2) to develop a new SQP algorithm for problem (P). According to the idea of SQP algorithm, the QP subproblem is as follows

$$\begin{cases} \min\limits_{(d,z)\in \mathbf{R}^{n+1}} \frac{1}{2}\langle d, B_k d\rangle + z, \\ s.t. \quad \langle \nabla f_i(x_k), d\rangle - z \le \phi(x_k) - f_i(x_k), \quad i = 1,\cdots,m. \end{cases} \tag{3.2}$$

In [8,9], Xue proposed SQP line search algorithms for solving finite minimax problems based on a special subproblem

$$\begin{cases} \min\limits_{(d,z)\in \mathbf{R}^{n+1}} \frac{1}{2}\langle d, B_k d\rangle + \frac{\gamma}{2}z^2 + z = m_k(d,z), \\ s.t. \quad \langle \nabla f_i(x_k), d\rangle - z \le \phi(x_k) - f_i(x_k), \quad i = 1,\cdots,m, \end{cases} \tag{3.3}$$

where $\gamma \in (0,1)$ is a small scalar, in practice a value of γ much closer to zero would be used (typical value that we have used is $\gamma = 10^{-5}$), B_k is an $n \times n$ symmetric matrix which approximates to the Hessian matrix of Lagrangian function (3.1), and the notation $\langle , \rangle$ is the inner product of two vectors.

The approximation model of (3.3) has some advantages over that of (3.2) [8, 9]. In fact, in the trust region subproblem (3.2), the Hessian matrix of its quadratic objective function is

$$\begin{pmatrix} B_k & 0 \\ 0 & 0 \end{pmatrix}, \tag{3.4}$$

while in the trust region subproblem (3.3), the Hessian matrix of its quadratic objective function is

$$\begin{pmatrix} B_k & 0 \\ 0 & \gamma \end{pmatrix}. \tag{3.5}$$

Obviously, (3.4) is a singular matrix even if B_k is positive definite, (3.5) is a positive definite matrix when B_k is positive definite. Thus, subproblem (3.3) is more stable than subproblem (3.2) in general.

In this paper, we employ model (3.3) to produce an algorithm. However, according to the KKT conditions of problem (P), we can not directly use the solution of (3.3) to be the trial step of SQP algorithm, we need to make some transformations for them.

Let $\{x_k\} \subset \mathbf{R}^n$ be a sequence generated by the algorithm, at kth iteration, we want to compute an SQP trial step d_k first, and then to set the next iterate x_{k+1} from x_k. Suppose that $(\tilde{d}_k, \tilde{z}_k)$ is the solution of following trust-region subproblem

$$\begin{cases} \min\limits_{(d,z)\in\mathbf{R}^{n+1}} \frac{1}{2}\langle d, B_k d\rangle + \frac{\gamma}{2}z^2 + z = m_k(d,z), \\ s.t. \quad \langle \nabla f_i(x_k), d\rangle - z \le \phi(x_k) - f_i(x_k), \quad i = 1, \cdots, m, \\ \|d\|_\infty \le \Delta_k, \end{cases} \tag{3.6}$$

where $\gamma \in (0,1]$ is a scalar, and $\tilde{\lambda}_k$ is the corresponding multiplier. Set

$$d_k = \frac{\tilde{d}_k}{(1+\gamma\tilde{z}_k)}, \quad \lambda_k = \frac{\tilde{\lambda}_k}{(1+\gamma\tilde{z}_k)}. \tag{3.7}$$

Then d_k is just the SQP trial step which we need, (3.7) is called as standardization.

It is well-known that the trust-region methods induce global convergence if the objective function ϕ is smooth. We will show below (Theorem 4.1) that it still makes sense even if ϕ is not smooth. In addition, we employ $\phi(x)$ to be the merit function. Now we give the algorithm as follows:

Algorithm 3.1.

step 0 Given initial values $x_0 \in \mathbf{R}^n$, $\varepsilon > 0$, some integer $\hat{M} > 0$, Δ_{max}, $\Delta_0 \in (0, \Delta_{max})$, $B_0 = I$, $\tau > 0$, $0 < \tau_1 < 1 < \tau_2$, $m(k) = 0$, $k := 0$.

step 1 (Computation of trial step)
Solve the quadratic program (3.6).
Assume that $(\tilde{d}_k, \tilde{z}_k)$ is the solution of (3.6), set

$$d_k = \frac{\tilde{d}_k}{(1+\gamma\tilde{z}_k)}, \ \lambda_k = \frac{\tilde{\lambda}_k}{(1+\gamma\tilde{z}_k)}, \ z_k = \tilde{z}_k.$$

If $\|d_k\| \le \varepsilon$, stop; Otherwise,

step 2 (Compute the ratio between the actual reduction and the predicted reduction using nonmonotone strategy)

$$r_k = \frac{\phi(x_{l(k)}) - \phi(x_k + d_k)}{m_k(0,0) - m_k(d_k, z_k)}, \tag{3.8}$$

where $\phi(x_{l(k)} = \max\limits_{0\le j\le m(k)} \phi(x_{k-j})$.

step 3 If $r_k > \tau$, $x_{k+1} = x_k + d_k$; Otherwise, $x_{k+1} = x_k$;

step 4 (Update Δ_k)
If $r_k < 0.25, \Delta_{k+1} = \tau_1\Delta_k$; goto *step 7*;
if $r_k \ge 0.75$ and $\|d_k\| = \Delta_k$, $\Delta_{k+1} = \min(\tau_2\Delta_k, \Delta_{max})$;
Otherwise, $\Delta_{k+1} = \Delta_k$.

step 5 $m(k) = \min(m(k-1)+1, \hat{M})$.
step 6 Update B_k
step 7 $k := k+1$, return to *step 1*.

Remark 3.1. In [8–10], updating B_k to B_{k+1} is restricted to use the Powell's modification of BFGS formula, which ensures that the matrices $\{B_k\}$ keep positive definite. In Algorithm 3.1, however, there is no such strict restriction, and we can use any Quasi-Newton updating formula.

We recall the following well-known result on minimax problems (see [7]).

Lemma 3.1. *A point $x^* \in \boldsymbol{R}^n$ is a critical point for problem (P) if and only if there exists a vector $\lambda^* \in \boldsymbol{R}^m$ such that*

$$\begin{cases} \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) = 0, \quad \sum_{i=1}^{m} \lambda_i^* = 1, \\ \lambda_i^* \geq 0, \quad \lambda_i^*(f_i(x^*) - \phi(x^*)) = 0, \quad i = 1, \cdots, m. \end{cases} \tag{3.9}$$

Remark 3.2. Note that (3.9) is just the KKT conditions of problem (P), which is the first-order necessary conditions of problem (1.2) at the point $(x^*, \phi(x^*))$. Provided that Assumption 2 (below in section 4) is satisfied.

Lemma 3.2. *Suppose that $(\tilde{d}_k, \tilde{z}_k)$ is the solution of SQP trust-region subproblem* (3.6). *If $\tilde{d}_k = 0$, then $\tilde{z}_k = 0$ holds.*

Proof. Since $(\tilde{d}_k, \tilde{z}_k)$ is the solution of SQP trust-region subproblem (3.6), when $\tilde{d}_k = 0$, the quantity $\tilde{z}_k$ is feasible if and only if it is nonnegative. Hence, to be optimal, $\tilde{z}_k$ must be equal to zero. □

In order to show that the stopping criterion is reasonable in Algorithm 3.1, we give the following theorem:

Theorem 3.1. *Suppose that $\{x_k\}$ and $\{d_k\}$ are generated by Algorithm* 3.1. *If $d_k = 0$, then x_k is the KKT point of the problem (P).*

Proof. Denote $(\tilde{d}_k, \tilde{z}_k)$ is the solution of subproblem (3.6). Then there exists multipliers $\tilde{\lambda}_k \in \mathbf{R}^m$ and $u_k \in \mathbf{R}$ satisfying

$$(B_k + u_k I)\tilde{d}_k + \nabla f(x_k)\tilde{\lambda}_k = 0, \tag{3.10a}$$

$$(\gamma \tilde{z}_k + 1) - e^T \tilde{\lambda}_k = 0, \tag{3.10b}$$

$$(F(x_k) + diag(\nabla f_1(x_k)^T \tilde{d}_k, \cdots, \nabla f_m(x_k)^T \tilde{d}_k) - (\phi(x_k) + \tilde{z}_k)I)\tilde{\lambda}_k = 0, \tag{3.10c}$$

$$u_k(\|\tilde{d}_k\| - \Delta_k) = 0, \tag{3.10d}$$

$$\nabla f(x_k)^T \tilde{d}_k - \tilde{z}_k e \leq \phi(x_k)e - f(x_k), \tag{3.10e}$$

$$\|d_k\| \leq \Delta_k, \quad \tilde{\lambda}_k \geq 0, \quad u_k \geq 0. \tag{3.10f}$$

By the second equation in (3.7) and (3.10b), we have

$$\sum_{i=1}^{m} \lambda_{k,i} = 1, \tag{3.11}$$

If $d_k = 0$, it follows from (3.7) that $\tilde{d}_k = 0$ holds. By Lemma 3.2, $\tilde{z}_k = 0$ holds too, which gives $\lambda_k = \tilde{\lambda}_k$. Substituting $\tilde{d}_k = 0$ into (3.10a) yields

$$\sum_{i=1}^{m} \lambda_{k,i} \nabla f_i(x_k) = 0. \tag{3.12}$$

It follows from (3.10c) and (3.10f) that

$$\lambda_{k,i} \geq 0, \quad \lambda_{k,i}(f_i(x_k) - \phi(x_k)) = 0, \quad i = 1, \cdots, m.$$

By Lemma 3.1, x_k is the KKT point of the problem (P). □

Remark 3.3. It is worth pointing out that, the denominators in (3.7) are not equal to zero when the algorithm does not stop. In fact, if $1 + \gamma \tilde{z}_k = 0$, it follows from (3.10a)-(3.10b) that $\tilde{\lambda}_k = 0$, and $\tilde{d}_k = 0$, which means that $d_k = 0$. By the terminal criterion, the algorithm is stopping.

4. Convergence Analysis

4.1. Global convergence

We suppose that the following standard assumptions hold throughout the analysis.

Assumption 1. For any point $x_0 \in \mathbf{R}^n$, the level set $L(x_0) = \{x \in \mathbf{R}^n : \phi(x) \leq \phi(x_0)\}$ is compact.

Assumption 2. For each $x \in L(x_0)$, the vectors

$$\begin{pmatrix} \nabla f_i(x) \\ -1 \end{pmatrix}, \quad i \in I_A(x),$$

are linearly independent.

Assumption 3. $\{B_k\}$ is uniformly bounded, in the sense that, there exists $M > 0$ such that $\|B_k\| \leq M$ holds, for $k = 1, 2, \cdots$.

Remark 4.1. Assumptions 1 and 2 are common assumptions in the literature on the minimax problems, where Assumption 1 is introduced in order to ensure the existence of a solution to problem (P) and Assumption 2 is a condition that ensures the constrained problem (1.2) and the original problem (P) have the same first-order necessary conditions (3.9). Assumption 3 is generally required for global convergence in the context of trust-region methods.

Theorem 4.1. *Suppose that $\{x_k\}$ and $\{d_k\}$ are generated by Algorithm* 3.1*, and Assumptions* 1-3 *are satisfied. If the algorithm does not stop in finite steps, then every accumulation point of $\{x_k\}$ is a KKT point of problem (P).*

Proof. Since Assumption 2 holds, problem (P) and problem (1.2) have the same KKT conditions (3.9). Thus, we can draw support from the KKT point of problem (1.2) to investigate the KKT point of original problem (P). By Assumption 1, the sequence $\{x_k\}$ is contained in the compact set $L(x_0)$, and hence it converges to the set of its accumulation points if the algorithm does not stop in a finite step.

Suppose on contrary that the conclusion does not hold. First, we claim that $\Delta_k \to 0$ holds.

In fact, if Δ_k does not converge to zero, then there exist $\delta > 0$ and infinite k, for sake of simplicity, we denote its set by K_0, such that

$$\Delta_k \geq \delta \quad and \quad r_k \geq 0.25, \quad \forall k \in K_0. \tag{4.1}$$

Without loss of generality, suppose that

$$\lim_{k \in K_0, k \to \infty} x_k = \bar{x}.$$

By hypothesis, $\bar{x}$ is not the KKT point of problem (1.2). Therefore, $(0,0)$ is not the solution of following problem:

$$\begin{cases} \min\limits_{(d,z) \in \mathbf{R}^{n+1}} z + \frac{M}{2}\|d\|^2, \\ s.t. \quad \langle \nabla f_i(\bar{x}), d \rangle - z \leq \phi(\bar{x}) - f_i(\bar{x}), \quad i = 1, \cdots, m, \\ \|d\|_\infty \leq \frac{\delta}{2}. \end{cases} \tag{4.2}$$

We denote that $(\bar{d}, \bar{z})$ is the solution of above problem (4.2). Since $(0,0)$ is a feasible solution of problem (4.2), we have

$$\bar{z} + \frac{M}{2}\|\bar{d}\|^2 < 0. \tag{4.3}$$

Denote

$$\bar{\mu} = -\bar{z} - \frac{M}{2}\|\bar{d}\|^2.$$

It follows from (4.3) that $\bar{\mu} > 0$. It's not difficult to prove the following inequality

$$m_k(0,0) - m_k(d_k, z_k) = -z_k - \frac{\gamma}{2}z_k^2 - \frac{1}{2}\langle d_k, B_k d_k \rangle > \frac{1}{2}\bar{\mu} > 0, \tag{4.4}$$

which holds for sufficiently large $k \in K_0$. In fact, since $(\bar{d}, \bar{z})$ is the solution of problem (4.2) and (d_k, z_k) is the solution of problem (3.6), it follows from (4.1) that $(\bar{d}, \bar{z})$ is a feasible solution of subproblem (3.6). By Assumption 3, we obtain

$$z_k + \frac{\gamma}{2}z_k^2 + \frac{1}{2}\langle d_k, B_k d_k \rangle \leq \bar{z} + \frac{\gamma}{2}\bar{z}^2 + \frac{1}{2}\langle \bar{d}, B_k \bar{d} \rangle \leq \bar{z} + \frac{\gamma}{2}\bar{z}^2 + \frac{M}{2}\|\bar{d}\|^2.$$

Thus, for a sufficiently large $k \in K_0$, we have

$$-z_k - \frac{\gamma}{2}z_k^2 - \frac{1}{2}\langle d_k, B_k d_k \rangle \geq -\bar{z} - \frac{\gamma}{2}\bar{z}^2 - \frac{M}{2}\|\bar{d}\|^2 = \bar{\mu} - \frac{\gamma}{2}\bar{z}^2.$$

Since γ is a given scalar, we can choose γ small enough, such that $\bar{\mu} - \frac{\gamma}{2}\bar{z}^2 > \frac{1}{2}\bar{\mu}$. Hence, the statement (4.4) is correct.

Consequently, it follows from (4.1) and (4.4) that

$$\phi(x_{l(k)}) - \phi(x_{k+1}) \geq \frac{1}{4}\Big(m_k(0,0) - m_k(d_k, z_k)\Big) \geq \frac{1}{8}\bar{\mu}$$

holds for a sufficiently large $k \in K_0$. This contradicts with $\lim_{k \to \infty} \phi(x_k) = \phi(\bar{x})$, and implies that $\Delta_k \to 0$. Furthermore, we have

$$\lim_{k \to \infty} \|d_k\| = 0. \tag{4.5}$$

Since $\Delta_k \to 0$, there exists a subsequence $K_1 \subseteq K_0$, such that

$$r_k < 0.25, \quad \forall k \in K_1. \tag{4.6}$$

We may as well suppose that

$$\lim_{k \in K_1, k \to \infty} x_k = \hat{x}.$$

By the hypothesis, $\hat{x}$ is not a KKT point of problem (1.2). Let $(\hat{d}, \hat{z})$ is the solution of the following problem:

$$\begin{cases} \min\limits_{(d,z)\in \mathbf{R}^{n+1}} z + \frac{M}{2}\|d\|^2, \\ s.t. \quad \langle \nabla f_i(\hat{x}), d\rangle - z \le \phi(\hat{x}) - f_i(\hat{x}), \quad 1 \le i \le m, \quad \|d\|_\infty \le 1. \end{cases} \tag{4.7}$$

We obtain

$$\hat{z} + \frac{M}{2}\|\hat{d}\|^2 < 0.$$

Denote

$$\hat{\mu} = -\hat{z} - \frac{M}{2}\|\hat{d}\|^2 > 0.$$

It can be verified that $(\Delta_k \hat{d}, \Delta_k \hat{z})$ is a feasible solution of problem

$$\begin{cases} \min\limits_{(d,z)\in \mathbf{R}^{n+1}} z + \frac{M}{2}\|d\|^2, \\ s.t. \quad \langle \nabla f_i(\hat{x}), d\rangle - z \le \phi(\hat{x}) - f_i(\hat{x}), \quad i = 1, \cdots, m, \\ \|d\|_\infty \le \Delta_k, \quad (\Delta_k < 1). \end{cases} \tag{4.8}$$

Hence, there holds

$$\hat{z} + \frac{M}{2}\|\hat{d}\|^2 \le \Delta_k \hat{z} + \frac{1}{2}\Delta_k^2 M \|\hat{d}\|^2,$$

which immediately yields that

$$(1 - \Delta_k)\hat{z} \le -\frac{1}{2}M(1 - \Delta_k)(1 + \Delta_k)\|\hat{d}\|^2.$$

By simplification, we have

$$\hat{z} \le -\frac{1}{2}M(1 + \Delta_k)\|\hat{d}\|^2.$$

Therefore, we obtain

$$\hat{\mu} = -\hat{z} - \frac{M}{2}\|\hat{d}\|^2 \ge \frac{1}{2}M\Delta_k\|\hat{d}\|^2. \tag{4.9}$$

It follows that

$$\begin{aligned} & m_k(0,0) - m_k(d_k, z_k) \\ = & -z_k - \frac{\gamma}{2}z_k^2 - \frac{1}{2}\langle d_k, B_k d_k\rangle \ge -\hat{z} - \frac{\gamma}{2}\hat{z}^2 - \frac{M}{2}\|\hat{d}\|^2 \\ \ge & \frac{1}{2}(-\hat{z} - \frac{M}{2}\|\hat{d}\|^2) \ge \frac{1}{4}M\Delta_k\|\hat{d}\|^2. \end{aligned} \tag{4.10}$$

In addition, we have

$$\begin{aligned}
\phi(x_{j+1}) - \phi(x_j) &= \max_{1\le i\le m} f_i(x_j + d_j) - \phi(x_j) \\
&= \max_{1\le i\le m} \{f_i(x_j) + \langle \nabla f_i(x_j), d_j\rangle + \frac{1}{2}\langle d_j, B_j d_j\rangle + o(\|d_j\|)\} - \phi(x_j) \\
&\le \max_{1\le i\le m} \{\phi(x_j) + z_j + \frac{1}{2}\langle d_j, B_j d_j\rangle + o(\|d_j\|)\} - \phi(x_j) \\
&= z_j + \frac{1}{2}\langle d_j, B_j d_j\rangle + o(\|d_j\|), \qquad (4.11)
\end{aligned}$$

where the inequality holds, as $\|\nabla^2 f_i(x_j) - B_j\|$ is bounded and $\langle \nabla f_i(x_j), d_j\rangle - z_j \le \phi(x_j) - f_i(x_j),\ i = 1, \cdots, m$. It follows from (4.11) that

$$\begin{aligned}
Ared_k &= \phi(x_{l(k)}) - \phi(x_{k+1}) \\
&= \Big(\phi(x_{l(k)}) - \phi(x_{l(k)+1})\Big) + \Big(\phi(x_{l(k)+1}) - \phi(x_{l(k)+2})\Big) + \cdots + \Big(\phi(x_k) - \phi(x_{k+1})\Big) \\
&\ge \Big(- z_{l(k)} - \frac{1}{2}\langle d_{l(k)}, B_{l(k)} d_{l(k)}\rangle + o(\|d_{l(k)}\|)\Big) + \cdots + \Big(- z_k - \frac{1}{2}\langle d_k, B_k d_k\rangle + o(\|d_k\|)\Big) \\
&= \Big(- z_{l(k)} - \frac{\gamma}{2} z_{l(k)}^2 - \frac{1}{2}\langle d_{l(k)}, B_{l(k)} d_{l(k)}\rangle + \frac{\gamma}{2} z_{l(k)}^2 + o(\|d_{l(k)}\|)\Big) + \cdots \\
&\quad + \Big(- z_k - \frac{\gamma}{2} z_k^2 - \frac{1}{2}\langle d_k, B_k d_k\rangle + \frac{\gamma}{2} z_k^2 + o(\|d_k\|)\Big) \\
&\ge Pred_k + o(\|d_k\|), \qquad (4.12)
\end{aligned}$$

where $k - M \le k - m(k) \le l(k) \le k$. In the last inequality we have used the facts

$$Pred_k = -z_k - \frac{\gamma}{2} z_k^2 - \frac{1}{2}\langle d_k, B_k d_k\rangle,$$

and for every $j = l(k), l(k)+1, \cdots, k-1$,

$$-z_j - \frac{\gamma}{2} z_j^2 - \frac{1}{2}\langle d_j, B_j d_j\rangle \ge 0.$$

It follows from (4.5), (4.10) and (4.12) that

$$\lim_{k\to\infty} r_k = \lim_{k\to\infty} \frac{Ared_k}{Pred_k} \ge 1,$$

which contradicts (4.6). Hence the theorem is proved. □

Remark 4.2. Similarly, just as the case of smooth optimization, Assumption 3 in Theorem 4.1 can be replaced by the following condition:

$$\sum_{k=1}^{\infty} \frac{1}{1 + \max_{1\le i\le k} \|B_k\|} = +\infty.$$

Proposition 4.1. *Suppose that Assumption 2 is satisfied, the sequence $\{x_k\}$ is generated by Algorithm 3.1, and $\{\lambda_k\}$ is the corresponding sequence of their multipliers. Then, the convergent subsequence $\{x_{k_l}\}$ implies that the corresponding multiplier's subsequence $\{\lambda_{k_l}\}$ is convergent. Furthermore, if $\{x_k\}$ is convergent, then $\{\lambda_k\}$ is also convergent.*

Proof. By Algorithm 3.1, $\{x_k\}$ has a convergent subsequence $\{x_{k_l}\}$. For simplicity, denote $\{x_{k_l}\}$ by $\{x_k\}$, and x^* is its limit point; Denote $\{\lambda_{k_l}\}$ by $\{\lambda_k\}$. By Theorem 4.1, x^* is a KKT point of problem (P).

Note that $\{\lambda_k\}$ is in the compact set

$$\bigwedge = \Big\{\lambda \in \mathbf{R}^m : \sum_{i=1}^{m} \lambda_i = 1 \quad and \quad \lambda_i \geq 0, \forall i = 1, 2, \cdots, m\Big\},$$

there exist a subsequence $\{\lambda_{k_i}\}$ and $\lambda^* \in \bigwedge$ such that $\lambda_{k_i} \to \lambda^*$, as $i \to \infty$. We claim that the sequence $\{\lambda_k\}$ converges to λ^*. In fact, on the contrary, there exists another subsequence $\{\lambda_{k_j}\}$ and $\bar{\lambda}^* \in \bigwedge$, $(\bar{\lambda}^* \neq \lambda^*)$, such that $\lambda_{k_j} \to \bar{\lambda}^*$, as $j \to \infty$. It follows from $\lim\limits_{k\to\infty} x_k = x^*$ and Lemma 3.1 that

$$\sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) = 0, \quad \sum_{i=1}^{m} \lambda_i^* = 1,$$

$$\sum_{i=1}^{m} \bar{\lambda}_i^* \nabla f_i(x^*) = 0, \quad \sum_{i=1}^{m} \bar{\lambda}_i^* = 1.$$

This yields that

$$\sum_{i=1}^{m} (\bar{\lambda}_i^* - \lambda_i^*) \begin{pmatrix} \nabla f_i(x) \\ -1 \end{pmatrix} = 0.$$

By Assumption 2, we immediately obtain

$$\bar{\lambda}_i^* = \lambda_i^*, \quad i = 1, \cdots, m,$$

which contradicts $\bar{\lambda}^* \neq \lambda^*$. In addition, if $\{x_k\}$ is convergent, it is clear, from the above process, that $\{\lambda_k\}$ is also convergent. Hence the proposition is proved. □

4.2. Locally Superlinear Convergence

To discuss superlinear convergence, we need to make the following assumption.

Assumption 4. For every initial point $x_0 \in \mathbf{R}^n$, $\nabla^2 f_i(x),\ i = 1, \cdots, m$ are Lipschitz uniformly continuous on the level set $L(x_0)$, that is, there exist positive scalars $L_i,\ i = 1, \cdots, m$, for every $x_1, x_2 \in L(x_0)$, it holds

$$\|\nabla^2 f_i(x_1) - \nabla^2 f_i(x_2)\| \leq L_i \|x_1 - x_2\|, \quad i = 1, 2, \cdots, m. \tag{4.13}$$

Now we establish a necessary and sufficient conditions for the rate of superlinear convergence.

Theorem 4.2. *Suppose that*

(1). *$\{x_k\}$ and $\{d_k\}$ are generated by Algorithm* 3.1, *$\{x_k\}$ converges to x^*, and x^* is the KKT point of problem (P).*

(2). $\sum\limits_{i=1}^{m} \lambda_i^* \nabla^2 f_i(x^*)$ *is nonsingular.*

Then the convergence rate is superlinear if and only if

$$\lim_{k\to\infty} \frac{\|\sum\limits_{i=1}^{m} \lambda_i^* \nabla f_i(x_k + d_k)\|}{\|d_k\|} = 0. \tag{4.14}$$

Proof. Since x^* is the KKT point of problem (P), we have

$$\sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) = 0. \tag{4.15}$$

Note that $\sum_{i=1}^{m} \lambda_i^* \nabla^2 f_i(x^*)$ is nonsingular, expanding $\sum_{i=1}^{m} \lambda_i^* \nabla f_i(x_k + d_k)$ around x^* gives

$$\begin{aligned}
&\Big\| \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x_k + d_k) \Big\| \\
&= \Big\| \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 f_i(\xi_k)(x_k + d_k - x^*) \Big\| \\
&= \mathcal{O}(\|x_k + d_k - x^*\|).
\end{aligned} \tag{4.16}$$

Sufficiency: If (4.14) holds, it suffices to prove that

$$\lim_{k\to\infty} \frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} = 0. \tag{4.17}$$

In fact, it follows from (4.16) that

$$\lim_{k\to\infty} \frac{\|x_k + d_k - x^*\|}{\|d_k\|} = 0. \tag{4.18}$$

Denote $\delta_k = \|x_k + d_k - x^*\| / \|x_k - x^*\|$, we have

$$\frac{\|x_k + d_k - x^*\|}{\|d_k\|} \geq \frac{\|x_k + d_k - x^*\|}{\|x_k + d_k - x^*\| + \|x_k - x^*\|} = \frac{\delta_k}{1 + \delta_k}.$$

According to (4.18), we have $\lim_{k\to\infty} \delta_k = 0$. Hence, (4.17) holds.

Necessary: If (4.17) holds, since

$$\frac{\|x_k - x^*\| - \|x_k + d_k - x^*\|}{\|x_k - x^*\|} \leq \frac{\|d_k\|}{\|x_k - x^*\|} \leq \frac{\|x_k - x^*\| + \|x_k + d_k - x^*\|}{\|x_k - x^*\|},$$

we have

$$\lim_{k\to\infty} \frac{\|d_k\|}{\|x_k - x^*\|} = 1.$$

It follows from (4.16) and (4.17)that

$$\begin{aligned}
&\lim_{k\to\infty} \frac{\Big\| \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x_k + d_k) \Big\|}{\|d_k\|} \\
&= \lim_{k\to\infty} \frac{\mathcal{O}(\|x_k + d_k - x^*\|)}{\|d_k\|} \\
&= \lim_{k\to\infty} \frac{\mathcal{O}(\|x_k + d_k - x^*\|)}{\|x_k - x_*\|} \\
&= 0.
\end{aligned}$$

Thus, (4.14) holds, and the theorem is proved. □

Theorem 4.3. *Suppose that*

(1). *Assumption* 4 *is satisfied.*

(2). $\{x_k\}$ *and* $\{d_k\}$ *are generated by Algorithm* 3.1, $\{x_k\}$ *converges to* x^*, *and* x^* *is the KKT point of problem (P).*

(3). $\sum_{i=1}^m \lambda_i^* \nabla^2 f_i(x^*)$ *is nonsingular, and*

$$\Big\| \sum_{i=1}^m (\lambda_{k,i} - \lambda_i^*) \nabla f_i(x_k) \Big\| = o(\|d_k\|).$$

(4). *The multipliers* $u_k \to 0$, *as* $k \to \infty$.

Then the convergence rate is superlinear if and only if

$$\lim_{k\to\infty} \frac{\|(B_k - \sum\limits_{i=1}^m \lambda_i^* \nabla^2 f_i(x^*)) d_k\|}{\|d_k\|} = 0. \tag{4.19}$$

Proof. Firstly, since $\| \sum\limits_{i=1}^m (\lambda_{k,i} - \lambda_i^*) \nabla f_i(x_k) \| = o(\|d_k\|)$, we obtain

$$\begin{aligned} &\Big\| (B_k - \sum_{i=1}^m \lambda_i^* \nabla^2 f_i(x^*)) d_k \Big\| = o(\|d_k\|) \\ \Leftrightarrow &\Big\| (B_k - \sum_{i=1}^m \lambda_i^* \nabla^2 f_i(x^*)) d_k + \sum_{i=1}^m (\lambda_{k,i} - \lambda_i^*) \nabla f_i(x_k) \Big\| = o(\|d_k\|). \end{aligned} \tag{4.20}$$

Secondly, from (3.7), (3.10a) and (3.10b) we have

$$(B_k + u_k I) d_k + \sum_{i=1}^m \lambda_{k,i} \nabla f_i(x_k) = 0. \tag{4.21}$$

It follows that

$$\begin{aligned} &(B_k - \sum_{i=1}^m \lambda_i^* \nabla^2 f_i(x^*)) d_k + \sum_{i=1}^m (\lambda_{k,i} - \lambda_i^*) \nabla f_i(x_k) \\ &= (B_k + u_k I) d_k + \sum_{i=1}^m \lambda_{k,i} \nabla f_i(x_k) - \sum_{i=1}^m \lambda_i^* \nabla^2 f_i(x^*) d_k - u_k d_k - \sum_{i=1}^m \lambda_i^* \nabla f_i(x_k) \\ &= -\sum_{i=1}^m \lambda_i^* \nabla^2 f_i(x^*) d_k - u_k d_k - \sum_{i=1}^m \lambda_i^* \nabla f_i(x_k) \\ &= \sum_{i=1}^m \lambda_i^* (\nabla f_i(x_k + d_k) - \nabla f_i(x_k) - \nabla^2 f_i(x^*) d_k) - \sum_{i=1}^m \lambda_i^* \nabla f_i(x_k + d_k) - u_k d_k. \end{aligned} \tag{4.22}$$

By Assumption 4, denoting $L = \frac{1}{2} \sum\limits_{i=1}^m \lambda_i^* L_i$ gives

$$\begin{aligned} &\Big\| \sum_{i=1}^m \lambda_i^* (\nabla f_i(x_k + d_k) - \nabla f_i(x_k) - \nabla^2 f_i(x^*) d_k) \Big\| \\ &\le \frac{1}{2} \sum_{i=1}^m \lambda_i^* L_i \Big(\|x_k - x^*\| + \|x_{k+1} - x^*\| \Big) \|d_k\| \\ &= L \Big(\|x_k - x^*\| + \|x_{k+1} - x^*\| \Big) \|d_k\| \\ &= o(\|d_k\|). \end{aligned} \tag{4.23}$$

Since $\|u_k d_k\|/\|d_k\| \leq \|u_k\|$ and $u_k \to 0$, we have

$$\|u_k d_k\| = o(\|d_k\|). \tag{4.24}$$

It follows from (4.22)– (4.24) that

$$\Big(B_k - \sum_{i=1}^{m} \lambda_i^* \nabla^2 f_i(x^*)\Big) d_k + \sum_{i=1}^{m} \Big(\lambda_{k,i} - \lambda_i^*\Big) \nabla f_i(x_k)\| = o(\|d_k\|)$$

$$\Leftrightarrow \Big\| \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x_k + d_k) \Big\| = o(\|d_k\|). \tag{4.25}$$

By Theorem 4.2, the convergence rate is superlinear if and only if

$$\Big\| \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x_k + d_k) \Big\| = o(\|d_k\|). \tag{4.26}$$

Therefore, it follows from (4.20), (4.25) and (4.26) that the conclusion is correct, and the proof is complete. □

5. Numerical Results

In order to validate the proposed approach from a computational point of view, some preliminary numerical experiments were carried out. The algorithm was coded in MATLAB 6.5, and the tests were performed on a PC computer with CPU Pentium 4, 2.00GHz. In the implementation, $\Delta_0 = 1, \Delta_{max} = 50, \tau_1 = 0.5, \tau_2 = 2, \gamma = 10^{-5}$, $\tau = 10^{-3}$, $\hat{M} = 5$, $B_0 = I$(the identity matrix), and the tolerance was set to $\varepsilon = 1.0e-05$. The program terminates if $\|d\| \leq \varepsilon$ is satisfied.

The test results are shown in Tables 1 and 2. In Table 1, the performance of our proposed Algorithm 3.1 (NTR) is compared with that of Algorithm 11.3.1 proposed in [17] (CGT), the algorithm in [8] (Xue), the algorithm in [10] (JQH) and the algorithm in [25] (WZH), where the test results for "CGT" algorithm were obtained by the program coded in MATLAB 6.5, the test results for "Xue", "JQH" and "WZH" were directly quoted from [8], [10] and [25] respectively. In addition, to make such comparison meaningful, the approximation matrices B_k were all updated by the Powell's modification of BFGS formula in Table 1. In Table 2, to show the advantage of Algorithm 3.1 over line search algorithms in the choice of approximation matrices B_k, instead of using Powell's modification of BFGS formula, SR1 update formula is employed, which make a comparison between Algorithm 3.1 and "Xue", where the test results for "Xue" were obtained by the program coded in MATLAB 6.5.

In the tables, "NI", "NF" and "NG" represent the total number of iteration, the total number of function evaluations and the total number of gradient evaluations respectively; $\phi(x)$ and $\|d\|$ stand for the optimal value of objective function and the norm of the solution of SQP subproblem respectively.

In addition, we also set a maximum iteration number of $50(n+m)$ to terminate the computation when this limit is reached, and mark the character "x" on all output items; The character "F" will be marked in case that the algorithm fails, and the character "-" represents that such items were lacked in the results of relevant algorithms. The tested examples were all from [25].

As is demonstrated in Tables 1 and 2, we can see that:

Table 1: Test Results obtained by Algorithm 3.1 and other relearnt algorithms.

Fun.	n/m	Methods	NI	NF	NG	$\phi(x)$	$\|d\|$
1	2/3	NTR	6	6	6	1.9522	2.6382e-06
		CGT	6	11	6	1.9522	1.0273e-07
		Xue	6	11	6	1.9522	2.9694e-06
		JQH	7	-	-	1.9522	7.2261e-07
		WZH	6	6	6	1.9522	8.2341e-07
2	2/3	NTR	5	5	5	2.0000	9.7055e-10
		CGT	3	5	3	2.0000	7.5480e-07
		Xue	5	9	5	2.0000	4.7033e-09
		JQH	5	-	-	2.0000	6.2246e-08
		WZH	5	5	5	2.0000	9.7022e-10
3	4/4	NTR	11	13	11	-43.0000	1.5189e-07
		CGT	15	29	7	-43.0000	3.9833e-06
		Xue	10	23	10	-44.0000	3.7726e-05
		JQH	11	-	-	-44.0000	2.0230e-07
		WZH	11	15	11	-44.0000	1.4745e-06
4	2/3	NTR	10	12	10	0.6164	5.8861e-06
		CGT	12	23	8	0.6164	4.8023e-07
		Xue	8	15	8	0.6164	1.1560e-06
		JQH	8	-	-	0.6164	7.2757e-08
		WZH	10	20	10	0.6164	2.4353e-06
5	3/6	NTR	9	9	9	3.5997	3.7477e-06
		CGT	22	43	10	3.5997	7.4495e-06
		Xue	10	19	10	3.5997	4.3608e-07
		JQH	11	-	-	3.5997	1.0469e-06
		WZH	11	11	11	3.5997	3.9095e-07
6	7/5	NTR	19	23	19	6.7868e+02	6.0676e-07
		CGT	31	61	18	6.7868e+02	4.5610e-06
		Xue	14	40	14	6.8063e+02	1.0945e-05
		JQH	-	-	-	-	-
		WZH	19	44	19	6.7868e+02	4.1400e-06
7	10/9	NTR	15	16	15	24.3062	8.0741e-07
		CGT	15	29	13	24.3062	6.5809e-07
		Xue	17	34	17	24.3062	2.7316e-05
		JQH	38	-	-	24.3062	7.9208e-06
		WZH	16	24	16	24.3062	8.9702e-07
8	20/18	NTR	23	29	23	1.3125e+02	1.3058e-06
		CGT	36	71	23	1.3125e+02	1.3254e-06
		Xue	24	53	24	1.3261e+02	7.9283e-06
		JQH	74	-	-	1.3261e+02	1.6223e-05
		WZH	21	48	21	1.3125e+02	1.3537e-06
9	3/30	NTR	15	28	15	0.0508	3.9540e-06
		CGT	8	15	8	0.0508	6.3160e-11
		Xue	8	15	8	0.0508	2.5070e-07
		JQH	-	-	-	-	-
		WZH	7	7	7	0.0508	4.2244e-06
10	2/20	NTR	8	11	8	4.6934	5.4319e-07
		CGT	20	39	8	4.6934	1.5302e-05
		Xue	-	-	-	-	-
		JQH	-	-	-	-	-
		WZH	10	31	10	4.6934	5.2753e-07
11	4/40	NTR	12	12	12	1.1571e+02	1.2666e-07
		CGT	10	19	10	1.1571e+02	5.8611e-07
		Xue	-	-	-	-	-
		JQH	-	-	-	-	-
		WZH	13	13	13	1.1571e+02	5.4233e-07

Table 2: comparisons of NTR and Xne

Fun.	n/m	Methods	NI	NF	NG	$\phi(x)$	$\|d\|$
1	2/3	NTR	6	6	6	1.9522245548	1.8106851102e-06
		Xue	6	12	6	1.9522245163	5.0113823266e-07
2	2/3	NTR	5	5	5	2.0000000042	9.7054950921e-10
		Xue	5	10	5	2.0000000041	9.4141059916e-10
3	4/4	NTR	10	13	10	-43.9999999970	7.6867194355e-09
		Xue	9	23	9	-43.9999999778	2.5805395664e-07
4	2/3	NTR	11	11	11	0.6164324356	1.3771568725e-010
		Xue	F	F	F	F	F
5	3/6	NTR	9	9	9	3.5997193004	7.8459691769e-08
		Xue	10	20	10	3.5997193324	2.8809727197e-07
6	7/5	NTR	16	18	16	6.7867963789e+02	7.1887200299e-06
		Xue	16	49	16	6.7867963788e+02	1.7692068617e-06
7	10/9	NTR	17	18	17	24.3062090718	1.4384595588e-09
		Xue	11	27	11	24.3062103112	2.7280903150e-07
8	20/18	NTR	17	23	17	1.3124767075e+02	4.6726726354e-06
		Xue	F	F	F	F	F
9	3/30	NTR	13	25	13	0.0508163280	1.8303080727e-07
		Xue	8	16	8	0.0508163265	5.0470039777e-07
10	2/20	NTR	9	16	9	4.6934565606	3.6812892634e-06
		Xue	9	22	9	4.6935277641	3.0250953851e-05
11	4/40	NTR	14	15	14	1.15706439521e+02	1.6105869108e-07
		Xue	F	F	F	F	F

1. From Table 1, under the condition of using Powell's modification of BFGS formula in all algorithms which can guarantee that the update matrix B_k keeps positive definite, the new algorithm (NTR) performs much better than the "CGT", "Xue" and "JQH" algorithms in terms of the number of iterations, function evaluations and gradient evaluations in general.

When comparing the new algorithm (NTR) and the algorithm in [25](WZH), we observe that:

(i) Though the performance is the same in all terms for problems 1 and 2, "NTR" algorithm performed much better than "WZH" algorithm in all terms for problems 7, 8, 10 and 11;

(ii) For problems 3, 4 and 6, though the performance is the same in terms of the number of iterations and gradient evaluations, the number of function evaluations required by "NTR" is smaller than that of "WZH" ;

(iii) For problems 3 and 5, "WZH" requires fewer iterations, function and gradient evaluations, but the difference is not significant. This means that in most cases, the new algorithm performs better than the algorithm in [25].

2. From Table 2, we can find that, under the condition of using SR1 update formula in all algorithms which can not guarantee that the update matrix B_k keeps positive definite, for all test problems, although B_k may be indefinite, the new algorithm (NTR) worked very well, but "Xue" algorithm failed for problems 4, 10 and 11. The reason is that the latter is a line-search based algorithm, and it requires that the approximation Hessian matrix B_k keeps positive definite.

Through above preliminary tests, it is clear that the new algorithm proposed in this paper is robust and competitive.

6. Conclusions

In this paper, we described a new trust-region algorithm for finite minimax problems. The algorithm uses a new approximation model in trust region scheme, and solves only one quadratic subproblem at each iteration. In particular, it does not require B_k to be positive definite. In addition, it employs nonmonotone strategy to avoid Maratos effect. The analysis shows that, under mild conditions, the new algorithm has strong global convergence and a rate of superlinear convergence. Preliminary numerical tests indicate that the new algorithm is robust and efficient.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (11171250,11071184); The Natural Science Foundation of Shanxi Province of China(2011011002-2); Foundation of Taiyuan Normal University(A0647). The authors are grateful to the editors and the anonymous referees for their careful reading of the original draft, and their detailed suggestions which improved the paper.

References

[1] G.A. Watson, The minimax solution of an overdetermined systemof nonlinear equations, *Journal of the Institute for Mathematics and its Applications*, **23** (1979), 167-180.

[2] M.L. Overton, Algorithms for nonlinear l_1 and l_∞ fitting, Nonlinear optimization, 1981, Edited by M. J. D. Powell, Academic Press, London, England, 1982, 213-221.

[3] M. Gaudioso, M.F. Monaco, A boundle type approach to the unconstrained minimization of convex nonsmooth functions, *Math. Program.*, **23** (1982), 216-226.

[4] X. Chen, Superlinear convergence of smoothing quasi-Newton methods for nonsmooth equations, *J. Comput. Appl. Math.*, **80** (1997), 105-126.

[5] X. Chen, Smoothing methods for complementarity problems and their applications: a survey, *Journal of the Operations Research Society of Japan*, **43** (2000), 32-47.

[6] P.P. Shen, Y.J. Wang, A new pruning test for finding all global minimizers of nonsmooth functions, *Appl. Math. Comput.*, **168** (2005), 739-755.

[7] G. Di Pillo, L. Grippo, S. Lucidi, A smooth method for the finite minimax problem, *Math. Program.*, **60** (1993), 187-214.

[8] Y. Xue, The sequential quadratic programming method for solving minimax problem, *Journal of Systems Science and Mathematical Sciences*, **22** (2002), 355-364.

[9] Y. Xue, A Newton like algorithm for solving minimax problem, *Journal on Numerical Methods and Computer Applications*, **2** (2004), 108-115.

[10] J.B. Jian, R. Quan, Q.J. Hu, A new superlinearly convergent SQP algorithm for nonlinear minimax problem, *Acta Mathematicae Applicatae Sinica, English series*, **23** (2007), 395-410.

[11] Z.B. Zhu, X. Cai, J.B. Jian, An improved SQP algorithm for solving minimax problems, *Appl. Math. Lett.*, **22** (2009), 464-469.

[12] E. Polak, D.H. Mayne, J.E. Higgins, Superlinearly convergent algorithm for min-max problems, *J. Optimiz. Theorey Appl.*, **69** (1991), 407-439.

[13] A. Vardi, New minimax algorithm, *J. Optimiz. Theorey Appl.*, **75** (1992), 613-634.

[14] S.P. Han, Variable metric methods for minimizing a class of nondifferentiable functions, *Math. Program.*, **20** (1981), 1-13.

[15] J.L. Zhou, A.L. Tits, Nonmonotone line search for minimax problems, *J. Optimiz. Theorey Appl.*, **76** (1993), 455-476.

[16] W.Y. Sun, On the convergence of an iterative method for the minimax problem, *J. Aust. Math. Soc., Ser. B*, **39** (1997), 280-292.

[17] A.R. Conn, N.I. M. Gould, P.L. Toint, Trust-Region Methods, MPS/SIAM, Philadeiphia, 2000.

[18] R. Fletcher, Second order correction for nondifferentiable optimization problems, in: G.A. Watson, ed., *Numerical Analysis*, (1982), 85-114.

[19] R. Fletcher, A model algorithm for composite nondifferentiable optimization problems, *Mathematical Programming Study*, **17** (1982), 67-76.

[20] Y. Yuan, On the superlinear convergence of a trust-region algorithm for nonsmooth optimization, *Math. Program.*, **31** (1985), 269-285.

[21] E.R. Panier, A.L. Tits, Avoiding the Maratos effect by means of a nonmonotone line search, Part 1: General constrained problems, *SIAM J. Numer. Anal.*, **28** (1991), 1183-1195.

[22] J.Y. Fan, W.B. Ai, Q.Y. Zhang, A line search and trust region algorithm with trust region radius converging to zero, *J. Comput. Math.*, **22**:6 (2004), 865-873.

[23] Q.Y. Zhou, W.Y. Sun, An adaptive nonmonotonic trust region method with curvilinear searches, *J. Comput. Math.*, **6** (2006), 761-771.

[24] J.T. Mo, K.C. Zhang, Z.X. Wei, A nonmonotone trust region method for unconstrained optimization, *Appl. Math. Comput.*, **171** (2005), 371-384.

[25] F.S. Wang, K.C. Zhang, A hybrid algorithm for nonlinear minimax problems, *Ann. Oper. Res.*, **164** (2008), 167-191.

[26] M.J.D. Powell, On the global convergence of trust region algorithm for unconstrained minimization, *Math. Program.*, **29** (1984), 297-303.

[27] Y.X. Yuan, W.Y. Sun, Optimization Theory and Methods, Science Press, Beijing, China, 1997.

[28] L. Luksan, A compact variable metric algorithm for nonlinear minimax approximation, *Computing*, **36** (1986), 355-373.