J. Appl. Math. & Informatics Vol. **29***(2011), No. 3 - 4, pp. 573 - 585*
Website: http://www.kcam.biz

A MODIFIED NONMONOTONE FILTER TRUST REGION METHOD FOR SOLVING INEQUALITY CONSTRAINED PROGRAMMING†

HUA WANG* AND DINGGUO PU

Abstract. SQP method is one of the most important methods for solving nonlinear programming. But it may fail if the quadratic subproblem is inconsistent. In this paper, we propose a modified nonmonotone filter trust region method in which the QP subproblem is consistent. By means of nonmonotone filter, this method has no demand on the penalty parameter which is difficult to obtain. Moreover,the restoration phase is not needed any more. Under reasonable conditions, we obtain the global convergence of the algorithm. Some numerical results are presented.

AMS Mathematics Subject Classification : 90C55, 90C30.
Key words and phrases : Nonmonotone, Filter, Inequality constraints, Global convergence.

1. Introduction

Consider the following nonlinear inequality constrained optimization problem(NLP):

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) \le 0, \quad i \in I = \{1, 2, ..., m\} \end{aligned} \tag{1}$$

where $f : R^n \to R$ and $C(x) = (c_1(x), c_2(x), ..., c_m(x))^T : R^n \to R^m$ are continuously differentiable functions. For convenience, let $g(x) = \nabla f(x), A(x) = (\nabla c_1(x), \nabla c_2(x), ..., \nabla c_m(x))$. And f_k refers to $f(x_k)$, C_k to $C(x_k)$, g_k to $g(x_k)$ and A_k to $A(x_k)$, etc.

The nonlinear programming problem (1), arising often in engineering, economy and many fields in the society, is extremely important. There are many

Received June 20, 2010. Revised August 21, 2010. Accepted September 3, 2010.
*Corresponding author.
†This research is supported by the National Natural Science Foundation of China (No.10771162), Young Teacher Research Foundation of Shanghai (No.slx08019) and Shanghai Budget Project (No.1139IA0013).

practical methods for solving problem (1). Among these methods, as we all know, the sequential quadratic programming (SQP) method is one of the most efficient method to solve problem (1). Because its superlinear convergence rate, it has been widely studied [2, 16, 9, 12, 13, 14].

The SQP method generates a sequence $\{x_k\}$ converging to the desired solution by solving the following quadratic programming subproblem

$$\begin{aligned} \min \quad & g(x)^T d + \frac{1}{2} d^T H d \\ \text{s.t.} \quad & C(x) + A(x)^T d \leq 0 \end{aligned} \tag{2}$$

where $H_k \in R^{n \times n}$ is a symmetric positive definite matrix.

For SQP method, however, it may fail if the quadratic subproblem (2) becomes infeasible. Burke and Han [11] gave a modification to this method wherein the QP subproblem is altered in a way which guarantees that the associated constraint region is nonempty for each $x \in R^n$ and for which a reasonable robust convergence theory is established. Recently, Liu and Yuan [10] proposed a method wich is a modified SQP method. Unlike other methods, their method solves two subproblems: one is an unconstrained piecewise quadratic subproblem, the other is quadratic subproblem. Similar methods are given in [5, 6]. Their method has excellent theoretical properties and is implementable. Zhang and Zhang's [4] described another implementable method, which is similar to Liu and Yuan's. At each iteration, it solves two subproblems, but either of them is simple than the subproblem in [10]. It uses a nondifferential exact penalty function as the merit function and gets the global convergence of the algorithm.

However, as we all know, there are several difficulties associated with the use of penalty function, and in particular the choice of the penalty parameter. Hence, in 2002, Fletcher and Leyffer [7] proposed a class of filter methods, which dose not require any penalty parameter and has promising numerical results.

In fact, filter methods exhibits a certain degree of nonmonotonicity. The nonmonotone technique was proposed by Grippo et al. in 1986 [15]. The nonmonotone technique is helpful in overcoming the case where the sequence of iterates is to follow the bottom of curved narrow valleys (a common occurrence in difficult nonlinear problems). There have existed a plenty of literatures about nonmonotone technique, for example, nonmonotone line search approaches [15]-[22], and nonmonotone trust region methods [23]-[28].

In this paper, motivated by the idea in [4, 31], we propose a modified method for solving inequality constrained programming (NLP). This method has the following merits: At each iteration, it solve a linear programming subproblem and a quadratic subproblem. A feasible direction is generated, well with which the iteration points satisfy all the constraints, so it circumvents the difficulties associated with the possible inconsistency of QP subproblem of the original SQP method. By means of nonmonotone filter, the restoration phase, a common feature of the large majority of the filter methods, is not needed, so that the scale of the calculation is decreased in a certain degree. Under reasonable conditions, we obtain the global convergence of the algorithm. In the end, numerical experiments show that this method is effective.

The paper is organized as follows. In section 2, we introduce some preliminary results. The algorithm is presented in Section 3. In Section 4, some global convergence results are proved. And some numerical examples are given in the last section.

2. Preliminaries

In this section, we recall some preliminary results about the filter algorithm which will be used in the sequent analysis.

In filter method, originally proposed by Fletcher and Leyffer [7], the acceptability of steps is determined by comparing the value of constraint violation and the objective function with previous iterates collected in a filter. The new iterate is acceptable for the filter if either feasibility or the objective function value is sufficiently improved in comparison to all iterates bookmarked in the current filter. The promising numerical results lead to a growing interest in filter methods in recent years.

In this paper, define the violation function $h(x)$ by

$$h(x) = \|C(x)^+\|_2^2 \tag{3}$$

where $C(x)^+ = \max\{0, c_j(x) : j \in I\}$.

It is easy to see that $h(x) = 0$ if and only if x is a feasible point. So a trial point should reduce either the value of constraint violation h or the objective function f. To ensure sufficient decrease of at least one of the two criteria, we say that a point x_1 dominates a point x_2 whenever

$$h(x_1) \leq h(x_2) \quad \text{and} \quad f(x_1) \leq f(x_2) \tag{4}$$

All we need to do is to remember iterates that are not dominated by any other iterates using a structure called a filter. A filter is a list $\mathcal{F}$ of pairs of the form (h_i, f_i) such that either

$$h(x_i) \leq h(x_j) \quad \text{or} \quad f(x_i) \leq f(x_j) \tag{5}$$

for $i \neq j$. We thus aim to accept a new iterate x_i only if it is not dominated by any other iterates in the filter.

In practical computation, we do not wish to accept $x_k + d_k$ if its (h, f)-pair is arbitrarily close to that of x_k or that of a point already in the filter. Thus we set a small "margin" around the border of the dominate point of the (h, f) space in which we shall also reject trial points. Formally, we say that a point x is acceptable for the filter if and only if

$$h(x) \leq (1 - \gamma)h_j \quad \text{or} \quad f(x) \leq f_j - \gamma h_j \tag{6}$$

for all $(h_j, f_j) \in \mathcal{F}$, where γ is close to zero. So, there is negligible difference in practice between (6) and (5). As the algorithm progresses, we may want to add a (h, f)-pair to the filter. If $x_k + d_k$ is acceptable for $\mathcal{F}$, then $x_{k+1} = x_k + d_k$, and

$$D_{k+1} = \{(h_j, f_j) | h_j \geq h_k \quad \text{and} \quad f_j - \gamma h_j \geq f_k - \gamma h_k, \quad \forall (h_j, f_j) \in \mathcal{F}\}$$

Filter set is update as the following rule

$$(\mathcal{F}_{k+1}) \quad \mathcal{F}_{k+1} = \mathcal{F}_k \bigcup \{(h_{k+1}, f_{k+1})\} \backslash D_{k+1} \tag{7}$$

We also refer to this operation as "adding $x_k + d_k$ to the filter", although, strictly speaking, it is the (h, f)-pair which is added.

We note that if a point x_k is in the filter or is acceptable for the filter, then any other point x such that

$$h(x) \leq (1 - \gamma) h_k \quad \text{and} \quad f(x) \leq f_k - \gamma h_k \tag{8}$$

is also acceptable for the filter and x_k.

Different from the traditional criteria of filter idea, with nonmonotone technique, we recall that a point x is acceptable to the filter if and only if

$$h(x) \leq (1 - \gamma) \max_{0 \leq r \leq m(k)-1} h_{k-r} \quad \text{or} \quad f(x) \leq \max \left[f_k, \sum_{r=0}^{m(k)-1} \lambda_{kr} f_{k-r} \right] - \gamma h(x) \tag{9}$$

where $(h_{k-r}, f_{k-r}) \in \mathcal{F}$ for $0 \leq r \leq m(k) - 1$, and $0 \leq m(k) \leq \min\{m(k-1) + 1, M\}$, $M \geq 1$ is a given positive constant, $\sum_{r=0}^{m(k)-1} \lambda_{kr} = 1, \lambda_{kr} \in (0, 1)$ and there exists a positive constant λ such that $\lambda_{kr} \geq \lambda$.

Similar to the traditional filter methods, we also need to update the filter set $\mathcal{F}$ at each successful iteration, the technique is comparable to the traditional one except that we do it based on criteria (9) not (6).

3. Description of algorithm

First, we modify the quadratic subproblem of SQP method. Given $x_k \in R^n$, $LP(x_k)$ is defined as following linear programming subproblem

$$\begin{aligned} LP(x_k): \quad & \min \quad z \\ & \text{s.t.} \quad C_k + A_k^T d \leq z e \\ & \qquad\ \ z \geq 0 \\ & \qquad\ \ \|d\| \leq \Delta_k \end{aligned} \tag{10}$$

where $e = (1, 1, ..., 1)^T \in R^m$, Δ_k is a trust region radius. Let its solution be $(\hat{d}_k^T, z_k)^T$. Then quadratic subproblem (2) is replaced by the following convex programming problem

$$\begin{aligned} QP(x_k, H_k): \quad & \min \quad q_k(d) = g_k^T d + \frac{1}{2} d^T H_k d \\ & \text{s.t.} \quad C_k + A_k^T d \leq z_k e \\ & \qquad\ \ \|d\| \leq \Delta_k \end{aligned} \tag{11}$$

Clearly, the convex programming $QP(x_k, H_k)$ is feasible. In fact, $\hat{d}_k$ is a feasible solution of $QP(x_k, H_k)$. If H_k is positive definite then the solution of $QP(x_k, H_k)$ is unique. Let d_k be the solution of $QP(x_k, H_k)$. Then d_k is used as the search direction at the current point x_k.

Borrowed from the usual trust region idea, we also need to define the following predicted reduction for the violation function $h(x)$.

$$\mathrm{pred}_k^c = h(x_k) - \|(C_k + A_k^T d_k)^+\|^2 \tag{12}$$

and the actual reduction

$$\mathrm{ared}_k^c = h(x_k) - h(x_k + d_k) = \|C_k^+\|^2 - \|C(x_k + d_k)^+\|^2 \tag{13}$$

Similarly, to evaluate the descent properties of the step for the objective function, we use the predicted reduction of $f(x)$

$$\mathrm{pred}_k^f = q_k(0) - q_k(d_k) = -q_k(d_k)$$

the actual reduction of $f(x)$

$$\mathrm{ared}_k^f = f(x_k) - f(x_k + d_k) \tag{14}$$

In general trust region method, the step d_k will be accepted if

$$\mathrm{ared}_k^f \geq \rho \mathrm{pred}_k^f \tag{15}$$

where $\rho \in (0, 1)$ is a fixed constant. But in this paper, considering nonmonotone technique, we replace the condition (15) by

$$\mathrm{rared}_k^f \geq \rho \mathrm{pred}_k^f \tag{16}$$

where rared_k^f is the relaxed actual reduction of $f(x)$

$$\mathrm{rared}_k^f = \max\left\{ f(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} f(x_{k-r}) \right\} - f(x_k + d_k). \tag{17}$$

The algorithm is described as follows.

Algorithm A

Step 0. Initialization. Let $0 < \rho < 1, 0 < \gamma < 1, 0 < \lambda \leq 1, 0 < \gamma_0 < \gamma_1 \leq 1 < \gamma_2, M \geq 1$. Choose an initial point $x_0 \in R^n$, a symmetric matrix $H_0 \in R^{n\times n}$ and an initial region radius $\Delta \geq \Delta_{min} > 0, \mathcal{F} = \{(h_0, f_0)\}$. Set $k = 0, m(k) = 0$;

Step 1. Solve $LP(x_k)$ to obtain $\hat{d}_k, z_k$. If $\hat{d}_k = 0$ and $z_k \neq 0$, stop;

Step 2. Solve $QP(x_k, H_k)$ to get d_k. If $d_k = 0$, stop;

Step 3. If $x_k + d_k$ is acceptable to the filter, go to step 4, otherwise go to step 5;

Step 4. If $\mathrm{pred}_k^f \geq \mathrm{pred}_k^c$ and $\mathrm{rared}_k^f < \rho \mathrm{rared}_k^c$, then go to step 5. otherwise go to step 6.

Step 5. $\Delta_k \in [\gamma_0 \Delta_k, \gamma_1 \Delta_k]$, go to step 1;

Step 6. $x_{k+1} = x_k + d_k$, update the filter set. $\Delta_{k+1} \in [\Delta_k, \gamma_2 \Delta_k] \geq \Delta_{min}$. Update H_k to H_{k+1}. $m(k+1) = \min\{m(k)+1, M\}, k = k+1$, and go to step 1.

4. Global convergence

In this section, to present a proof of global convergence of algorithm, we always assume that following conditions hold.

Assumptions:

A1. The objective function f and the constraint functions $c_j(j \in I)$ are twice continuously differentiable.

A2. The iterate $\{x_k\}$ remains in a closed, bounded subset $S \subset R^n$.

A3. There exists two constants $0 < a \leq b$ such that $a\|d\|^2 \leq d^T H_k d \leq b\|d\|^2$, for all k, for all $d \in R^n$.

A4. $h(x_k) - \|(C(x_k) + A(x_k)^T d)^+\|^2 \geq \beta \Delta_k \min\{h(x_k), \Delta_i\}$, where β is a constant.

Remark. Assumptions A1,A2 are the standard assumptions. A3 plays an important role to obtain the convergence results. A4 is the sufficient reduction condition. It guarantees the global convergence in a trust region method. Under the assumptions, f is bounded below and the gradient function $g(x)$ is uniformly continuous in S.

Because no restriction on the constraint functions, the cluster point of the sequence generated by Algorithm A can be either of the two different types of points. Similar to the definition in [3], we give their definition as follows.

Definition. $x \in R^n$ is called

(1) a strong stationary point of problem (1) if x is feasible and there exists a vector $\rho = (\rho_1, \rho_2, ..., \rho_m)^T \in R^m$ such that

$$g(x) + A(x)\rho = 0 \tag{18}$$
$$\rho_i \geq 0, \quad \rho_i c_i(x) = 0 \quad i \in I$$

(2) an infeasible stationary point of problem (1) if x is infeasible and

$$\min_{d \in R^n} \max_{i \in I} \{c_i(x) + \nabla c_i(x)^T d, 0\} = \phi(x)$$

where $\phi(x) = \max_{i \in I}\{c_i(x), 0\}$.

Clearly, a strong stationary point defined above is precisely a KKT point of problem (1). Liu and Yuan [3] proved the following lemma, which is described the properties of infeasible strong stationary point.

Lemma 1. If $x \in R^n$ is an infeasible stationary point, there exists $\rho_0 \geq 0$ and $\rho \in R^m$ such that the following first-order necessary condition

$$\rho_0 g(x) + \sum_{i=1}^{m} \rho_i \nabla c_i(x) = 0$$

$$\rho_i \geq 0 \quad i \in I \tag{19}$$

holds.

Lemma 2. If Algorithm A terminates at x_k, then x_k is either an infeasible stationary point or a strong stationary point.

Proof. See Lemma 3.2 in [4]. □

Lemma 3. Suppose that the assumptions hold, the Algorithm A is well defined.

Proof. We will show that there exists $\delta > 0$ such that step d_k is accepted whenever $\Delta_k \leq \delta$.

Without loss of generality, we can assume that $\|c_k\| \geq \varepsilon$. Then we start with $\delta \in (0, \varepsilon]$ such that the closed δ-ball about x_k lies in S. Since $\delta \leq \varepsilon$, we have that $\Delta_k \leq \delta \leq \varepsilon$. Then by Assumption 4, it holds $\text{pred}_k^c \geq \beta\varepsilon\Delta_k$, and

$$\begin{aligned}
|\text{ared}_k^c - \text{pred}_k^c| &= |h(x_k) - h(x_k + d_k) - (h(x_k) - \|(c_k + A_k^T d_k)^+\|^2)| \quad (20)\\
&= |\|(c_k + A_k^T d_k)^+\|^2 - \|(c_k + (A_k^{'})^T d_k)^+\|^2| \\
&\leq |(\|c_k\|^2 + 2c_k^T A_k^T d_k + d_k^T A_k A_k^T d_k) \\
&\quad -(\|c_k\|^2 + 2c_k^T (A_k^{'})^T d_k) + d_k^T A_k^{'} (A_k^{'})^T d_k)| \\
&\leq 4\sqrt{n}\Delta_k \|c_k\| \|A_k - A_k^{'}\| + 4n\Delta_k^2 \|A_k A_k^T - A_k^{'}(A_k^{'})^T\|
\end{aligned}$$

where $A_k^{'} = A(x_k^{'}), x_k^{'} = x_k + \xi d_k, \xi \in (0, 1)$ denotes some point on the line segment from x_k to $x_k + d_k$. So,

$$\left| \frac{\text{ared}_k^c - \text{pred}_k^c}{\text{pred}_k^c} \right| \leq \frac{4\sqrt{n}\Delta_k \|c_k\| \|A_k - A_k^{'}\| + 4n\Delta_k^2 \|A_k A_k^T - A_k^{'}(A_k^{'})^T\|}{\beta\varepsilon\Delta_k} \to 0 \tag{21}$$

as $\Delta_k \to 0$, which implies $h_k - h(x_k + d_k) > \eta \text{pred}_k^c \geq \eta\beta\varepsilon\Delta_k$. Hence, for all k such that $\Delta_k \leq \delta$, there must exist $\gamma > 0$ such that

$$h(x_k + d_k) \leq (1 - \gamma)h_k \leq (1 - \gamma) \max_{0 \leq r \leq m(k)-1} h_{k-r} \tag{22}$$

That means the trial point $x_k + d_k$ is acceptable to the filter.

To prove the implementation of Algorithm A, we only need to show that if $\text{pred}_k^f \geq \text{pred}_k^c$, it holds $\text{rared}_k^f \geq \rho \text{pred}_k^f$.

In fact,

$$\begin{aligned}
|\text{ared}_k^f - \text{pred}_k^f| &= \left| f(x_k) - f(x_k + d_k) + g_k^T d_k + \frac{1}{2} d_k^T H_k d_k \right| \quad (23)\\
&\leq \left| -g_k^T d_k - \frac{1}{2} d_k^T \nabla^2 f(y_k) d_k + g_k^T d_k + \frac{1}{2} d_k^T H_k d_k \right| \\
&\leq 4\Delta_k^2 \frac{1}{2} \|\nabla^2 f(y_k) - H_k\| \\
&\leq 4n\Delta_k^2 b
\end{aligned}$$

where $y_k = x_k + \xi d_k, \xi \in (0,1)$ denotes some point on the line segment from x_k to $x_k + d_k$. And $b = \frac{1}{2}(\sup \|H_k\| + \max_{x\in S} \|\nabla^2 f(x)\|)$, $\|d_k\| \le 2\sqrt{n}\Delta_k$. Followed by that

$$\left|\frac{\text{ared}_k^c - \text{pred}_k^c}{\text{pred}_k^c}\right| \le \frac{4n\Delta_k^2 b}{\beta\varepsilon\Delta_k} \to 0 \quad \text{as} \quad \Delta_k \to 0 \tag{24}$$

We have $\text{rared}_k^f \ge \text{ared}_k^f \ge \rho\text{pred}_k^f$ for some $\rho \in (0,1)$. Therefore, the trial step is accepted for all $\Delta_k \le \delta$. □

Lemma 4. Suppose that the assumptions hold and Algorithm A dose not terminate finitely, then $\lim_{k\to\infty} h_k = 0$.

Proof. If Algorithm A cannot terminate finitely, then there are infinite points accepted by the filter. We prove the result in two cases by the definition of filter.

(i) $h(x_k + d_k) \le (1-\gamma)\max_{0\le r\le m(k)-1} h_{k-r}$ for all sufficiently large k,

(ii) $f(x_k + d_k) \le \max[f_k, \sum_{r=0}^{m(k)-1} \lambda_{kr} f_{k-r}] - \gamma h_k$ for all sufficiently large k, where f_k refers to $f(x_k)$, h_k to $h(x_k)$ etc.

In view of convenience, let

$$h(x_{l(k)}) = \max_{0\le r\le m(k)-1} h_{k-r}$$

where $k - m(k) + 1 \le l(k) \le k$.

Also, set $h_{k+1} = h(x_k + d_k), f(x_{k+1}) = f(x_k + d_k)$.

(i). Since $m(k+1) \le m(k) + 1$, we have

$$\begin{aligned} h(x_{l(k+1)}) &= \max_{0\le r\le m(k+1)-1} h_{k+1-r} \\ &\le \max_{0\le r\le m(k)} h_{k+1-r} \\ &= \max\{h(x_{l(k)}), h(x_{k+1})\} \\ &= h(x_{l(k)}) \end{aligned} \tag{25}$$

which implies that $\{h(x_{l(k)})\}$ converges. Then

$$h(x_{k+1}) \le (1-\gamma)\max\nolimits_{0\le r\le m(k)-1} h_{k-r},$$

we have

$$h(x_{l(k)}) \le (1-\gamma)h(x_{l(l(k)-1)}). \tag{26}$$

Since $\gamma \in (0,1)$, we deduce that $h(x_{l(k)}) \to 0 (k \to \infty)$.

Therefore

$$h(x_{k+1}) \le (1-\gamma)h(x_{l(k)}) \to 0.$$

holds by the Algorithm A. That is $\lim_{k\to\infty} h(x_k) = 0$.

(ii). Suppose there exists an infinite subsequence S on which

$$f_{k+1} \le \max\left[f_k, \sum_{r=0}^{m(k)-1} \lambda_{kr} f_{k-r}\right] - \gamma h_k$$

Then we first show that for all $k \in S$, it holds

$$f_k \leq f_0 - \lambda\gamma \sum_{r=0}^{k-2} h_r - \gamma h_{k-1} \leq f_0 - \lambda\gamma \sum_{r=0}^{k-1} h_r \tag{27}$$

We prove (27) by induction.

If $k = 1$, we have $f_1 \leq f_0 - \gamma h_0 \leq f_0 - \lambda\gamma h_0$.

Assume that (27) holds for $1, 2, ..., k$, then we consider that (27) holds for $k+1$ in the following two cases.

Case 1. $\max[f_k, \sum_{r=0}^{m(k)-1} \lambda_{kr} f_{k-r}] = f_k$

$$f_{k+1} \leq f_k - \gamma h_k \leq f_0 - \lambda\gamma \sum_{r=0}^{k-2} h_r - \gamma h_k \leq f_0 - \lambda\gamma \sum_{r=0}^{k-1} h_r \tag{28}$$

Case 2. $\max[f_k, \sum_{r=0}^{m(k)-1} \lambda_{kr} f_{k-r}] = \sum_{r=0}^{m(k)-1} \lambda_{kr} f_{k-r}$

Let $p = m(k) - 1$, then

$$\begin{aligned}
f_{k+1} &\leq \sum_{t=0}^{p} \lambda_{kt} f_{k-t} - \gamma h_k \qquad (29) \\
&\leq \sum_{t=0}^{p} \lambda_{kt} \left(f_0 - \lambda\gamma \sum_{r=0}^{k-t-2} h_r - \gamma h_{k-t-1} \right) - \gamma h_k \\
&= \lambda_{k0} \left(f_0 - \lambda\gamma \sum_{r=0}^{k-p-2} h_r - \lambda\gamma \sum_{r=k-p-1}^{k-2} h_r - \gamma h_{k-1} \right) - \gamma h_k \\
&\quad + \lambda_{k1} \left(f_0 - \lambda\gamma \sum_{r=0}^{k-p-2} h_r - \lambda\gamma \sum_{r=k-p-1}^{k-3} h_r - \gamma h_{k-2} \right) \\
&\quad + \cdots + \lambda_{kp} \left(f_0 - \lambda\gamma \sum_{r=0}^{k-p-2} h_r - \gamma h_{k-p-1} \right) \\
&\leq \sum_{t=0}^{p} \lambda_{kr} f_0 - \lambda\gamma \sum_{r=0}^{k-p-2} \left(\sum_{t=0}^{p} \lambda_{kr} \right) h_r - \sum_{t=0}^{p} \lambda_{kr} \gamma h_{k-t-1} - \gamma h_k
\end{aligned}$$

By the fact that $\sum_{t=0}^{p} \lambda_{kt} = 1, \lambda_{kt} \geq \lambda$ and $h_r \geq 0$, we have

$$\begin{aligned}
f_{k+1} &\leq f_0 - \lambda\gamma \sum_{r=0}^{k-p-2} h_r - \lambda\gamma \sum_{r=k-p-1}^{k-1} h_r - \gamma h_k \qquad (30) \\
&= f_0 - \lambda\gamma \sum_{r=0}^{k-1} h_r - \gamma h_k \\
&\leq f_0 - \lambda\gamma \sum_{r=0}^{k} h_r
\end{aligned}$$

Then for all $k \in S$, (27) holds.

Moreover, since $\{f_k\}$ is bounded below, let $k \to \infty$, we can get that

$$\lambda\gamma \sum_{r=0}^{\infty} h_r < \infty$$

It follows that $h_k \to \infty \quad (k \to \infty)$. □

Theorem 1. Suppose $\{x_k\}$ is an infinite sequence generated by Algorithm A. d_k is the solution of $QP(x_k, H_k)$. If the multiplier according to the subproblem (11) is uniform bounded, then $\lim_{k\to\infty} \|d_k\| = 0$.

Proof. Suppose by contradiction that there exists constants $\varepsilon > 0$ and $\bar{k} > 0$ such that $\|d_k\| > \varepsilon$ for all $k > \bar{k}$. Then similar to the proof of Lemma 3, we have

$$\text{pred}_k^f = -q_k(d_k) \geq \beta\varepsilon \min\{\Delta_k, \varepsilon\} \geq \beta\varepsilon \min\{\bar{\Delta}, \varepsilon\} \tag{31}$$

As in the proof of Lemma3, there exists $\rho > 0$ such that $\text{rared}_k^f \geq \rho\text{pred}_k^f$. That is $f_{k+1} \leq \max[f_k, \sum_{r=0}^{m(k)-1} \lambda_{kr} f_{k-r}] - \rho\text{pred}_k^f$. Similar to the proof of Lemma 4, we have

$$\rho \sum_{k=0}^{\infty} \text{pred}_k^f < \infty \tag{32}$$

which implies $\text{pred}_k^f \to 0$. It contradicts (31). Hence the result follows. □

Theorem 2. Suppose $\{x_k\}$ is an infinite sequence generated by Algorithm A, then every cluster point of $\{x_k\}$ is strong stationary point (KKT point) of problem (1).

Proof. Because $\{x_k\}$ lies in a bounded set, there must exists x^*, such that $x_k \to x^*, k \in K$, then by Lemma 4 we have $h(x_k) \to 0, k \in K$. That means x^* is a feasible point. Then $z_k \to 0$. From Theorem 1, we get $d^* = 0$ is the solution of subproblem $QP(x^*, H^*)$. Then by the KKT condition, we obtain

$$\begin{aligned} &g^* + \rho^* A^* = 0 \\ &\rho_i^* \geq 0, \quad \rho^* c_i^* = 0 \quad i \in I. \end{aligned} \tag{33}$$

Therefore, x^* is a KKT point of problem (1). □

5. Numerical tests

In this section, we give some numerical experiences to show the success of proposed method.

(1) Updating of H_k is done by

$$H_{k+1} = H_k + \frac{y_k^T y_k}{y_k^T s_k} - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k}$$

where $y_k = \theta_k \hat{y}_k + (1-\theta_k) H_k s_k$

$$\theta_k = \begin{cases} 1 & if \quad s_k^T \hat{y}_k \le 0.2 s_k^T H_k s_k \\ \frac{0.8 s_k^T H_k s_k}{s_k^T H_k s_k - s_k^T \hat{y}_k} & otherwise \end{cases} \tag{34}$$

and $\hat{y}_k = g_{k+1} - g_k + (A_{k+1} - A_k)\rho_k, s_k = x_{k+1} - x_k$, ρ_k is a multiplier associated with (11).

(2) The stop criteria is $\|d_k\|$ sufficiently small. We assume the error toleration is 10^{-6};

(3) The algorithm parameters were set as follows: $H_0 = I \in R^{n\times n}, \gamma = 0.02, \rho = 0.5, \gamma_0 = 0.1, \gamma_1 = 0.5, \gamma_2 = 2, \Delta_{min} = 10^{-6}, \Delta_0 = 1, M = 3$. The program is written in Matlab.

In Table 1 which presented the results of the numerical experiences, we use the following notations:

Problem represents the number of problems in [29], NI,NF,NG represent the numbers of iterations, function and gradient calculations, respectively.

problem	NI	NF	NG
HS22	7	5	1
HS42	20	16	6
HS43	12	9	2
HS44	5	4	1
HS76	6	3	2
HS86	6	4	4
HS113	12	10	10

Table 1

From the above results, we can see the algorithm in this paper is quite effective.

References

1. R.Fletcher, S.Leyffer, *Nonlinear programming without a penalty function*, Math.Program. **91** (2) (2002) 239-269.
2. P.T.Boggs, J.W.Tolle, P.Wang, *On the local convergence of quasi-newton methods for constrained optimization*, SIAM J.Control Optim. **20** (1982) 161-172.
3. X.W.Liu, Y.X.Yuan, *A robust algorithm for optimization with general equality and inequality constraint*, SIAM J.Sci.Comput. **22** (2) (2002) 517-534.
4. J.L.Zhang, X.S.Zhang, *A robust SQP method for optimization with inequality constraints*, J.Comput.Math. **21** (2) (2003) 247-256.
5. J.L.Zhang, X.S.Zhang, *A modified SQP method with nonmonotone linesearch technique*, J.Global.Optim. **21** (2001) 201-218.
6. G.L.Zhou, *A modified SQP method and its global convergence*, J.Global Optim. **11** (1997) 193-205.
7. R.Fletcher, S.Leyffer, *Nonlinear programming without a penalty function*, Math.Program. **91** (2) (2002) 239-269.

8. P.Y.Nie, C.F.M, *A trust region filter method for general nonlinear programming*, Appl.Math.Comput. **172** (2006) 1000-1017.
9. F.Facchinei, S.Lucidi, *Quadraticly and superlinearly convergent for the solution of inequality constrained optimization problem*, J.Optim.Theory Appl. **85** (1995) 265-289.
10. X.W.Liu, Y.X.Yuan, *A robust algorithm for optimization with general equality and inequality constraints*, SIAM J.Sci.Comput. **22** (2) (2002) 517-534.
11. J.V.Burke, S.P.Han, *A robust SQP method*, Math.Program. **43** (1989) 277-303.
12. S.P.Han, *Superlinearly convergence varialbe metric algorithm fro general nonlinear programming problems*, Math.Program. **11** (1976) 263-282.
13. M.J.D.Powell, *A fast algorithm for nonliear constrained optimization calculations.* In:Waston,G.A.(ed.) Numerical Analysis., pp.144-157. Springer-Verlag, Berlin(1982).
14. M.J.D.Powell, *Variable metric methods for constrained optimization.*In:Bachen,A.,et al.(eds.) Mathematical Programming- The state of Art. Springer-Verlag, Berlin (1982).
15. L.Grippo, F.Lampariello, S.Ludidi, *A nonmonotone line search technique for Newton's method*, SIAM J.Numer.Anal. **23** (1986) 707-716.
16. J.F.Bonnons, E.R.Painer, A.L.Titts, J.L.Zhou, *Avoiding the Maratos effect by means of nonmonotone linesearch, Inequality constrained problems-feasible iterates*, SIAM J.Numer.Anal. **29** (1992) 1187-1202.
17. L.Grippo, F.Lampariello, S.Ladidi, *A truncated Newton method with nonmonotone line search for unconstrained optimization*, J.Optim.Theory Appl. **60** (1989) 401-419.
18. L.Grippo, F.Lampariello, S.Ludidi, *A class of nonmonotone stablization method in unconstrained optimization*, Numer.Math. **59** (1991) 779-805.
19. E.Panier, A.Tits, *A avoiding the Maratos effect by means of nonmonotone line search,I:General constrained problems*, SIAM J.Numer.Anal. **28** (1991) 1183-1195.
20. W.S.Sun, J.Han, J.Sun, *Global convergence of nonmonotone descent methods for unconstrained optimization problems*, J.Comput.Appl.Math. **146** (2002) 89-98.
21. P.L.Toint, *An assessment of nonmonotone line search technique for unconstrained optimization*, SIAM J.Sci.Comput. **17** (1996) 725-739.
22. Z.S.Yu, D.G.Pu, *A new nonmonotone line search technique for unconstrained optimization*, J.Comput.Appl.Math. **219** (2008) 134-144.
23. N.Y.Deng, Y.Xiao, F.J.Zhou, *Nonmonotone trust-region algorithm*, J.Optim.Theory Appl. **26** (1993) 259-285.
24. J.H.Fu, W.Y.Sun, *Nonmonotone adaptive trust-region method for unconstrained optimization problems*, Appl.Math.Comput. **163** (2005) 489-504.
25. X.Ke, J.Han, *A nonmonotone trust-region algorithm for equality constrained optimization*, Sci.in China 38A (1995) 683-695.
26. X.Ke, G.Liu, D.Xu, *A nonmonotone trust-region algorithm for unconstrained optimization*, Chinese Sci.Bull. **41** (1996) 197-202.
27. W.Y.Sun, *Nonmonotone trust region method for solving optimization problems*, Appl.Math.Comput. **156** (2004) 159-174.
28. P.L.Toint, *A nonmonotone trust region algorithm for nonlinear optimization subject to convex constraints*, Math.Program. **77** (1997) 69-94.
29. W.Hock, K.Schittkowski, *Test examples for nonlinear programming codes*, in:Lecture Notes inEconom. and Math.Systems, vol.187, Springer-Verlag, Berlin, 1981.
30. M.Ulbrich, S.Ulbrich, *Nonmonotone trust region methods for nonlinear equality constrained optimization without a penalty function*, Math.Program.Ser.B **95** (2003) 103-135.
31. K.Su, D.G.Pu, *A nonmonotone filter trust region method for nonlinear constrained optimization*, J.Comput.Appl.Math. **223** (1) (2009) 230-239.

Hua Wang received M.Sc. and Ph.D at Tongji University. Since 2008 she has been at Shanghai Lixin University of Commerce. Her research interests include numerical optimization and computational mathematics.

School of Mathematics and Information, Shanghai Lixin University of Commerce. Shanghai, 201620, PR China
e-mail: chengyj819@sina.com

Dingguo Pu received M.Sc. from Fudan University, and Ph.D. from City University of Hong Kong. He is currently a professor at Tongji University. His research interests are numerical optimization and computational mathematics.

Department of Mathematics, Tongji University. Shanghai, 200092, PR China
e-mail: madpu@mail.tongji.edu.cn