J Glob Optim (2008) 41:203–217
DOI 10.1007/s10898-007-9219-0

A globally and superlinearly convergent modified SQP-filter method

Ke Su

Received: 27 June 2006 / Accepted: 5 May 2007 / Published online: 3 August 2007

Abstract In this paper, we presented a modified SQP-filter method based on the modified quadratic subproblem proposed by Zhou (J. Global Optim. 11, 193–2005, 1997). In contrast with the SQP methods, each iteration this algorithm only needs to solve one quadratic programming subproblems and it is always feasible. Moreover, it has no demand on the initial point. With the filter technique, the algorithm shows good numerical results. Under some conditions, the globally and superlinearly convergent properties are given.

Keywords Constrained optimization · KKT point · Sequential quadratic programming · Global convergence · Superlinear convergence

1 Introduction

In this paper, we consider the following nonlinear inequality constrained optimization problem:

$$(P)\ \min\ f(x)$$
$$\text{s.t.}\ \ g_j(x) \leq 0, \quad j \in I = \{1, 2, \cdots, m\} \tag{1}$$

where $x \in R^n$, $f : R^n \rightarrow R$ and $g_j(j \in I) : R^n \rightarrow R$ are assumed to be twice continuously differentiable.

There are many practical methods for solving problem (P). For example, trust region methods, gradient projection approaches, QP-free methods [16] and so on [4]. Among these methods, as we all know, the sequential quadratic programming(SQP) method is one of the

K. Su (✉)
Department of Mathematics, Tongji University, ShangHai 200092, P.R.China
e-mail: pigeonsk@163.com

K. Su
College of Mathematics and Computer, Hebei University, Baoding, 071002, P.R.China

most efficient methods to solve problem (P). Because its superlinear convergence rate, it has been widely studied [1,2,5,9,11,12,15].

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

$$\begin{aligned} &\min \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\ &\text{s.t. } g_j(x_k) + \nabla g_j(x_k)^T d \le 0, \quad j \in I = \{1, 2, \cdots, m\} \end{aligned} \tag{2}$$

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

The SQP algorithms have two serious shortcomings. First, in order to obtain a search direction, one must solve one or more quadratic programming subproblems per iteration, and the computation amount of this type is very large. Second, the SQP algorithms require that the related quadratic programming subproblems to be solvable per iteration, but it is difficult to be satisfied. Moreover, the solutions of the sequential quadratic subproblem may be unbounded, which leads to the sequence generated by the method is divergence.

Based on the above reasons, Burke and Han [3], Zhou [14], and Zhang and Zhang [13] modified the quadratic subproblem respectively to ensure that their methods are globally convergent. However, Burke and Han's method is only a conceptual method and can not be implementable practically. Zhou's method is based on the exact linear search. Zhang's method focus on the inexact line search, but there is much difficult to choose the penalty parameter in penalty function, which is used as a merit function. For this case, we adopt the filter technique, which is proposed by Fletcher and Leyffer [6] in 2002. After that, Fletcher et al. [7,8] combined this method with SQP, then get the global convergence. Without penalty function, filter methods have several advantages over penalty function methods. A penalty parameter estimate, which could be problematic to obtain, is not required.

Our idea is to combine the subproblem proposed in [14] and filter technique. The algorithm proposed in this paper has the following merits: it requires to solve only one QP subproblem with only a subset of the constraints which are estimate as active, the initial point is arbitrary, the subproblem is feasible at each iterate point, and need not to consider the penalty parameter. In the end, under mild conditions, its global convergence and local superlinear convergence are obtained.

This paper is organized as follows. In Sect. 2, we review some definitions and preliminary results that will be used in the latter sections. Section 3 introduces the algorithm. The global convergence theory for the method is presented in Sect. 4. In Sect. 5, we study the local superlinear convergence of the proposed algorithm. Some numerical examples are given in the last section.

The symbols we use in this paper are standard. For convenience, we list some of them as follows:

(1) $f'(x, d) = \lim_{\lambda \downarrow 0} (f(x + \lambda d) - f(x))/\lambda$;

(2) $g'(x)$ is Frechet derivative of g at x;

(3) $\|x\|_\infty = \max\{|x_j|, \quad j = 1, 2, \cdots, n\}$;

(4) $I = \{1, 2, \cdots, m\}$.

2 Preliminaries

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

2.1 Signs and lemmas

Define function $\Phi(x)$, $\Psi(x)$ by

$$\Phi(x) = \max\{0, g_j(x) : j \in I\} \tag{3}$$

$$\Psi(x) = \max\{g_j(x) : j \in I\} \tag{4}$$

For $\forall x, d \in R^n$, let $\Psi^*(x; d)$ be the first order approximation to $\Psi(x + d)$, namely

$$\Psi^*(x; d) = \max\{g_j(x) + \nabla g_j(x)^T d : j \in I\} \tag{5}$$

For $\forall \sigma > 0$, function $\Psi(x, \sigma)$, $\Psi^0(x, \sigma) : R^n \times R^+ \to R$ are defined as follows:

$$\Psi(x, \sigma) = \min\{\Psi^*(x; d) : \|d\| \le \sigma\} \tag{6}$$

$$\Psi^0(x, \sigma) = \max\{\Psi(x, \sigma), 0\} \tag{7}$$

Remark (6) equals to the following linear programming:

$$LP(x, \sigma) : \quad \min\{z : \quad g_j(x) + \nabla g_j(x)^T d \le z, \quad j \in I, \quad \|d\| \le \sigma\} \tag{8}$$

Denote

$$\theta(x, \sigma) = \Psi(x, \sigma) - \Psi(x) \tag{9}$$

$$\theta^0(x, \sigma) = \Psi^0(x, \sigma) - \Psi(x) \tag{10}$$

$$F = \{x : g_j(x) \le 0 \;\; j \in I\} = \{x : \Psi(x) \le 0\} \tag{11}$$

$$F^c = \{x : \Psi(x) > 0\} \tag{12}$$

Definition 1 [3] Mangasarian-Fromotz constraint qualification (MFCQ) is said to be satisfied by $g(x) \le 0$ at x if $\exists z \in R^n$ such that

$$\nabla g_j(x)^T z < 0 \quad \forall j \in \{j \in I | g_j(x) \ge 0\}$$

Lemma 1 [14] $\forall x \in F^c$, *if MFCQ is satisfied at* x, *then* $\theta(x, \sigma) < 0$ $(\forall \sigma > 0)$.

Lemma 2 [14] $\Psi(x, \sigma), \Psi^0(x, \sigma), \theta(x, \sigma), \theta^0(x, \sigma)$ *are all continuous on* $R^n \times R^+$.

Lemma 3 [14] $\forall x \in F^c$, *if* $\theta(x, \sigma) < 0$, *then* $\theta^0(x, \sigma) < 0$.

2.2 The notion of a filter

To avoid using the classical merit function with penalty term, in which the penalty parameter is difficult to obtain, we adopt the filter technique, which is proposed by Fletcher and Leyffer [6]. The acceptability of steps is determined by comparing the constraint violation and objective function value 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.

Define the violation function $h(x)$ by

$$h(x) = \|g(x)^+\|_\infty \tag{13}$$

where $g(x)^+ = \max\{0, g_j(x) : \ j \in I\}$

It is easy to see that $h(x) = 0$ if and only if x is a feasible point. So, a trial step should reduce either the constraint value h or the function value 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

$$h(x_1) \leq h(x_2) \quad \text{and} \quad f(x_1) \leq f(x_2) \tag{14}$$

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

$$h(x_i) \leq h(x_j) \quad \text{or} \quad f(x_i) \leq f(x_j) \tag{15}$$

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

$$h(x) \leq (1-\gamma)h_j \quad \text{or} \quad f(x) \leq f_j - \gamma h_j \tag{16}$$

for all $(h_j, f_j) \in \mathcal{F}$, where γ is close to zero. So, there is negligible difference in practice between (16) and (15). 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

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

Filter set is update as the following rule

$$(\mathcal{F}_{k+1}) \quad \mathcal{F}_{k+1} = \mathcal{F}_k \bigcup \{(h_{k+1}, f_{k+1})\} \setminus D_{k+1} \tag{17}$$

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

$$h(x) \leq (1-\gamma)h_k \quad \text{and} \quad f(x) \leq f_k - \gamma h_k \tag{18}$$

is also acceptable for the filter and x_k.

3 Description of the algorithm

Given $x \in R^n$, $\sigma > 0$. $D(x, \sigma)$ is defined as the following set

$$D(x, \sigma) = \{d | g_j(x) + \nabla g_j(x)^T d \leq \Psi^0(x, \sigma), \quad j \in I\}$$

If d^* is the solution of $LP(x, \sigma)$, the $d^* \in D(x, \sigma)$, hence $D(x, \sigma)$ is nonempty. The quadratic subproblem (2) is replaced by the following convex programming problem

$$Q(x_k, H_k, \sigma_k) : \quad \min \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d$$
$$\text{s.t.} \quad g_j(x_k) + \nabla g_j(x_k)^T d \leq \Psi^0(x_k, \sigma_k) \quad j \in L_k \tag{19}$$

where L_k is the set of approximate active indices of the point x_k. Clearly, by the above statement, the convex programming $Q(x_k, H_k, \sigma_k)$ is feasible. If H_k is positive definite, then the solution of $Q(x_k, H_k, \sigma_k)$ is unique. The convex programming problem has the following properties:

Theorem 1 [14] *Suppose that $x_k \in R^n$, $H_k \in R^{n\times n}$ is a symmetric positive definite matrix. If MFCQ is satisfied at x_k, then*

(1) *The convex programming problem $Q(x_k, H_k, \sigma_k)$ has a unique solution d_k which satisfies KKT conditions, i.e. there exist vectors $U^k = (u_j^k \ \ j \in L_k)$ such that*

(a) $g_j(x_k) + \nabla g_j(x_k)^T d_k \leq \Psi^0(x_k, \sigma_k) \quad j \in L_k$;
(b) $u_j^k \geq 0 \quad j \in L_k$;
(c) $\nabla f(x_k) + H_k d_k + A_k U^k = 0, \quad A_k = (\nabla g_j(x_k) \ \ j \in L_k)$;
(d) $u_j^k(g_j(x_k) + \nabla g_j(x_k)^T d_k) = 0 \ \ j \in L_k$;
(2) *If $d_k = 0$ is the solution of $Q(x_k, H_k, \sigma_k)$, then x_k is a KKT point of problem (P).*

Lemma 4 $\forall x \in F, \ \ d \in D(x, \sigma)$, *then* $\Psi^*(x; d) = 0$

Now, the algorithm for the solution of problem (P) can be stated as follows.

Algorithm A

Step 0: Initialization:

Given $x_0 \in R^n$, Σ is a compact set which consists of symmetric positive definite matrices. $H_0 \in \Sigma, k = 0, \epsilon_0 > 0, \sigma_r > \sigma_l > 0, \sigma_0 \in [\sigma_l, \sigma_r], C > 0, \eta, \alpha_1, \alpha_2 \in (0, 1)$, initial filter set $\mathcal{F}_0$;

Step 1: Computation of an 'active' constraint set L_k:

S1.1 Let $i = 0, \epsilon_{k,i} = \epsilon_0$;

S1.2 Set

$$L_{k,i} = \{j \in I \mid -\epsilon_{k,i} \leq g_j(x_k) - \Phi(x_k) \leq 0\}$$

$$A_{k,i} = (\nabla g_j(x_k) \ \ j \in L_{k,i})$$

If $det(A_{k,i}^T A_{k,i}) \geq \epsilon_{k,i}$, let $L_k = L_{k,i}$, $A_k = A_{k,i}$, $i_k = i$, go to step 2;

S1.3 Set $i = i + 1$, $\epsilon_{k,i} = \epsilon_{k,i-1}/2$, and go to S1.2 (inner loop A);

Step 2: Computation of the direction d_k: Compute $\Psi(x, \sigma)$, $\Psi^0(x, \sigma)$,let d_k be the solution of convex programming problem $Q(x_k, H_k, \sigma_k)$. If $d_k = 0$, then x_k is a KKT point of problem (P). If $\|d_k\| \geq C$, go to Step 4;

Step 3: Test to accept the trial step:

If $x_k + d_k$ is not acceptable for the filter.

If $h_k > \|d_k\|\{\eta, \alpha_1\|d_k\|^{\alpha_2}\}$, call Restoration Algorithm (Algorithm B) to obtain $x_k^r = x_k + s_k^r$, and go to Step 2. Otherwise go to Step 4.

If $x_k + d_k$ is acceptable for the filter, let $x_{k+1} = x_k + d_k$ and add x_{k+1} to the filter, go to Step 8;

Step 4: Computation of the direction q_k:

Let A_k^1 be the matrix whose rows are $|L_k|$ linearly independent rows of A_k, and A_k^2 be the matrix whose rows are the remaining $n - |L_k|$ rows of A_k. We might denote $A_k = \begin{pmatrix} A_k^1 \\ A_k^2 \end{pmatrix}$.

Like A_k, we might as well let $\nabla f(x_k) = \begin{pmatrix} \nabla f_1(x_k) \\ \nabla f_2(x_k) \end{pmatrix}$. Compute

$$\rho_k = -\nabla f(x_k)^T d_k, \qquad \pi_k = -(A_k^1)^{-1} \nabla f_1(x_k),$$
$$\tilde{d}_k = \frac{-\rho_k((A_k^1)^{-1})^T e}{1 + 2|e^T \pi_k|}, \qquad q_k = \rho_k(d_k + \bar{d}_k) \tag{20}$$

where $\bar{d}_k = \begin{pmatrix} \tilde{d}_k \\ 0 \end{pmatrix}$, $e = (1, 1, \cdots, 1)^T \in R^{|L_k|}$;

Step 5: $\alpha_{k,0} = 1, l = 0$;

Step 6: If $x_k + \alpha_{k,l} q_k$ is not acceptable for the filter, the go to Step 7. Otherwise let $\alpha_k = \alpha_{k,l}, x_{k+1} = x_k + \alpha_k q_k$ and add x_{k+1} to the filter, go to Step 8;

Step 7: $\alpha_{k,l+1} = \alpha_{k,l}/2, l = l + 1$, go to Step 6 (inner loop B);

Step 8: Update:

Choose $H_{k+1} \in \sum, \sigma_{k+1} \in [\sigma_l, \sigma_r], k = k + 1$. If $h_k > \|d_k\|\{\eta, \alpha_1\|d_k\|^{\alpha_2}\}$, call Restoration Algorithm (Algorithm B) to obtain $x_k^r = x_k + s_k^r$, and go to Step 2. Or else, go to Step 1.

Remark (1) H_{k+1} can be obtained by iterative formula.

(2) Whether in Zhou's or in Zhang's algorithm, the penalty function is needed. In this paper, the penalty function is substituted by filter, which avoid the penalty parameter estimate. Practical experience shows that they exhibit a certain degree of nonmonotonicity which can be beneficial.

(3) When the solution of (19) is unacceptable, it generates a revised direction by solving a system of linear equation, which takes full advantage of good property of d.

If $h_k > \|d_k\|\{\eta, \alpha_1\|d_k\|^{\alpha_2}\}$, we give the restoration algorithm (Algorithm B) to compute the x_k^r such that $h(x_k^r) \le \eta \min\{h_k^I, \alpha_1\|d_k\|^\theta\}$, where $2 < \theta \le 3$, $h_k^I = \min\{h_i | h_i > 0, (h_i, f_i) \in \mathcal{F}\}$.

In a restoration algorithm, it is therefore desired to decrease the value of $h(x)$. The direct way is utilized Newton method or the similar ways to attack $g(x + s)^+ = 0$. We now give the restoration algorithm.

Algorithm B

Step 1: Let $x_k^0 = x_k, \Delta_k^0 = \sigma_k, j = 0, \eta, \bar{\eta} \in (0, 1), 2 < \theta \le 3$;

Step 2: If $h(x_k^j) \le \eta\{h_k^I, \alpha_1\|d_k\|^\theta\}$, then let $x_k^r = x_k^j$ and stop;

Step 3: Compute

$$\begin{aligned} &\min\ h(x_k^j) - \|(g_k^j + A_k^j d)^+\|_\infty \\ &\text{s.t.}\ \ \|d_k\| \le \Delta_k^j \end{aligned} \tag{21}$$

to get s_k^j. Let $r_k^j = \dfrac{h(x_k^j) - h(x_k^j + d)}{h(x_k^j) - \|(g_k^j + A_k^j d)^+\|_\infty}$;

Step 4: If $r_k^j \le \bar{\eta}$, then let $x_k^{j+1} = x_k^j, \Delta_k^{j+1} = \frac{1}{2}\Delta_k^j, j = j + 1$ and go to step 3. Otherwise, let $x_k^{j+1} = x_k^j + s_k^j, \Delta_k^{j+1} = 2\Delta_k^j$, get A_k^{j+1}, $j = j + 1$ and go to step 2.

The above restoration algorithm is a Newton method for $g(x)^+ = 0$. This method is utilized frequently [10]. Of course, there are other restoration algorithms, such as interior point restoration algorithm, SLP restoration algorithm and so on.

4 Global convergence of algorithm

In the sequential analysis, we always assume that following conditions hold.

Assumptions

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

A2 For any $x \in R^n$, the vectors $\{\nabla g_j(x), \quad j \in I(x)\}$ are linearly independent, where $I(x) = \{j \in I | g_j(x) = \Phi(x)\}$.

A3 The iterate $\{x_k\}$ remain in a closed, bounded convex subsets $S \subset R^n$.

A4 When solving (21), we have $h(x_k^j) - \|(g(x_k^j) + A_k^j d)^+\| \geq \beta \min\{h(x_k^j), \Delta_k^j\}$ where $\beta > 0$ is a constant.

A5 There exist 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 $d \in R^n$.

(A1) and (A3) are the standard assumptions. (A2) is necessary for the following Lemma 5. (A4) is the sufficient reduction condition and it is every moderate.

Without loss of generality, we may assume that there exists $M > 0$ such that $\|x_k\| \leq M$. From Theorem 1, we also can assume that $\|U^k\| \leq M$. A important consequence of the assumption A1 and A3 is that they together directly ensure that for all k, it holds

$$0 < h_k \leq h_{\max} \quad \text{and} \quad f_{\min} \leq f_k \tag{22}$$

for some constants $f_{\min}, h_{\max} > 0$. Thus the part of the (h, f) space in which the (h, f)-pairs associated with the filter iterates lie is restricted to the rectangle

$$A_0 = [0, h_{\max}] \times [f_{\min}, \infty] \tag{23}$$

Before we show the global convergence of Algorithm A, we must to ensure the algorithm is implementable.

Lemma 5 *For any iterate k, the index i_k defined in step 1 is finite, which means that the inner loop A terminates in finite number of times.*

Proof Suppose by a contradiction that Algorithm A will run infinitely between Step 1.2 and Step 1.3, so we have

$$\det(A_{k,i}^T A_{k,i}) < \frac{1}{2^i}\epsilon_0 \tag{24}$$

By the definition of $L_{k,i}$, we can see that $L_{k,i+1} \subseteq L_{k,i}$. And there are only finite possible subsets of I, so we have $L_{k,i+1} \equiv L_{k,i}$ for large enough i. We denote it by L_k^*, now letting $i \to \infty$, then we obtain

$$\det(A_{L_k^*}^T A_{L_k^*}) = 0 \;\; \text{and} \;\; L_k^* = I(x_k) \tag{25}$$

which contradicts the Assumption A2. □

Lemma 6 *If $d_k \neq 0$, then it holds*

$$\begin{aligned} \nabla f(x_k)^T d_k < 0, \qquad & \nabla f(x_k)^T q_k \leq -\frac{1}{2}\rho_k^2 < 0 \\ \nabla g_j(x_k)^T d_k = 0, \quad & \nabla g_j(x_k)^T q_k \leq -\frac{\rho_k^2}{1 + 2|e^T \pi_k|} < 0 \end{aligned} \tag{26}$$

Proof We can see $I(x_k) \subseteq L_k$ by the definition of L_k. If $d_k \neq 0$, then by (19), we have

$$\nabla f(x_k)^T d_k \leq -\frac{1}{2} d_k^T H_k d_k < 0$$

If x_k is not a feasible point, we have $\nabla g_j(x_k)^T d_k \leq 0$. If x_k is a feasible point, then by the definition of $I(x_k)$ and Lemma 1, we have $\nabla g_j(x_k)^T d_k \leq 0 \quad j \in I(x_k)$.

In addition, from (20), we obtain

$$\begin{aligned}\nabla f(x_k)^T q_k &= \rho_k \nabla f(x_k)^T (d_k + \bar{d}_k)\\ &= \rho_k(\nabla f(x_k)^T d_k + \nabla f(x_k)^T \bar{d}_k)\\ &= \rho_k(-\rho_k + \nabla f_1(x_k)^T \tilde{d}_k)\\ &= \rho_k\left(-\rho_k + \frac{\rho_k \pi_k^T e}{1+2|e^T\pi_k|}\right)\\ &\le -\frac{1}{2}\rho_k^2 < 0 \end{aligned} \tag{27}$$

Moreover,

$$A_k^T \bar{d}_k = (A_k^1)^T \tilde{d}_k = \frac{-\rho_k (A_k^1)^T((A_k^1)^{-1})^T e}{1+2|e^T\pi_k|} = -\frac{\rho_k e}{1+2|e^T\pi_k|}$$

$$\nabla g_j(x_k)^T q_k = \rho_k(\nabla g_j(x_k)^T d_k + \nabla g_j(x_k)^T \bar{d}_k) \le -\frac{\rho_k^2}{1+2|e^T\pi_k|} < 0 \quad j \in I(x_k)$$

The claim holds. □

Lemma 7 *The inner loop B terminates in finite number of times.*

Proof By contradiction, if the conclusion is false, then the Algorithm A will run infinitely between Step 6 and Step 7, so we have

$$\alpha_{k,l} \to 0 \quad (l \to \infty)$$

and $x_k + \alpha_{k,l} q_k$ is not acceptable for the filter, we consider it in the following two cases:

Case 1 $h(x_k) = 0$:
By the definition of $h(x_k)$, we have

$$\begin{aligned} h(x_k + \alpha_{k,l} q_k) &= \max\{0, g_j(x_k + \alpha_{k,l} q_k)\}\\ &= \max\{0, g_j(x_k) + \alpha_{k,l}\nabla g_j(x_k)^T q_k + o(\|\alpha_{k,l} q_k\|^2)\}\end{aligned} \tag{28}$$

From Lemma 6, we have $\nabla g_j(x_k)^T q_k < 0$. Together with $\alpha_{k,l} \to 0$, we obtain that there must exist a constant β, such that

$$h(x_k + \alpha_{k,l} q_k) \le \max\{0, \beta g_j(x_k)\} = \beta h(x_k) \stackrel{\triangle}{=} (1-\gamma) h(x_k) \tag{29}$$

Moreover, by Lemma 6, $\nabla f(x_k)^T q_k \le -\frac{1}{2}\rho_k^2 < 0$. Then

$$f(x_k + \alpha_{k,l} q_k) = f(x_k) + \alpha_{k,l}\nabla f(x_k)^T q_k + O(\|\alpha_{k,l} q_k\|^2) \le f(x_k) \tag{30}$$

With (29) and (30), we conclude that $x_k + \alpha_{k,l} q_k$ must be acceptable for the filter and x_k, which is a contradiction.

Case 2 $h(x_k) \neq 0$
Similar to Case 1, we can also get the relation

$$h(x_k + \alpha_{k,l} q_k) \le (1-\gamma) h(x_k) \tag{31}$$

Since x_k is acceptable for the filter, we have

$$h_k \le (1-\gamma)h_j \quad \text{or} \quad f_k \le f_j - \gamma h_j \quad \forall (h_j, f_j) \in F \tag{32}$$

By the assumption, $x_k + \alpha_{k,l} q_k$ is not acceptable for the filter, so we have

$$h(x_k + \alpha_{k,l} q_k) > (1-\gamma)h_j \tag{33}$$

and

$$f(x_k + \alpha_{k,l} q_k) > f_j - \gamma h_j \tag{34}$$

for the point x_k, if it holds that $h_k \le (1-\gamma)h_j$, then by Lemma 6 and $\alpha_{k,l} \to 0$,

$$\begin{aligned} h(x_k + \alpha_{k,l} q_k) &= \max\{0, g_j(x_k + \alpha_{k,l} q_k)\} \\ &\le \max\{0, g_j(x_k)\} \le h_k \le (1-\gamma)h_j \end{aligned} \tag{35}$$

which contradicts (33).

If it holds $f_k \le f_j - \gamma h_j$, then, also by Lemma 6 and $\alpha_{k,l} \to 0$, we get

$$f(x_k + \alpha_{k,l} q_k) = f(x_k) + \alpha_{k,l} \nabla f(x_k)^T q_k + O(\|\alpha_{k,l} q_k\|^2) \le f_k \le f_j - \gamma h_j \tag{36}$$

which contradicts (34).

Based on the above analysis, together with Case 1, we can see that the claim holds.

Lemma 7 means that there exists a constant $\bar{\alpha} > 0$, such that $\alpha_k \ge \bar{\alpha}$ for large enough k. □

Lemma 8 *The Restoration Algorithm B terminates in a finite number of iteration.*

Proof It is similar to Lemma 1 in [10].

By the above statement, we see that Algorithm A is implementable. Now, we turn to prove the global convergence of Algorithm A. □

Theorem 2 [14]: *Assume that the MFCQ is satisfied at $x_0 \in R^n$. Let $\sigma_l > 0$ and $F = \{x | g(x) \le 0\}$, then there exists a neighbor $N(x_0)$ of x_0 such that*

(1) *the MFCQ is satisfied at any point in $N(x_0)$;*
(2) *if $x_0 \in F$, then $\Psi^0(x, \sigma) = 0$ for all $x \in N(x_0)$ and $\sigma \ge \sigma_l$;*
(3) *if $x_0 \in F$, then*

$$\sup \left\{ \sum_{j=1}^{m} \mu_j : H \in \Sigma, x \in N(x_0), \sigma \in [\sigma_l, \sigma_r] \right\} < \infty$$

where $\Sigma \subset R^{n\times n}$ is a compact set which consists of symmetric positive definite matrices and $0 < \sigma_l < \sigma_r$.

Lemma 9 *Suppose that infinite points are added to the filter, then $\lim_{k\to\infty, k\in K} h_k = 0$, where K is an infinite set.*

Proof If the lemma were not true, there would have an infinite subsequence K_1, such that for $\forall k \in K_1$,

$$h_k \geq \epsilon > 0$$

At each iteration k, (h_k, f_k) is added to the filter. By (17), we can deduce that (h, f)-pair be added to the filter at a large stage within the square

$$[h_k - \gamma\epsilon, h_k] \times [f_k - \gamma\epsilon, f_k]$$

even if (h_k, f_k) is later removed from the filter. Now observe these squares whose area are all $\gamma^2\epsilon^2$. As a consequence, the set $[0, h_{\max}] \times [f_{\min}, \infty] \cap \{(h, f) | f \leq \kappa_f\}$ is completely covered by at most finite number of such squares, for any choice of $\kappa_f \geq f_{\min}$. Since $(h_k, f_k)(k \in K_1)$ keep on being added to the filter, this implies that f_k tends to infinite when k tends to infinite. Without loss of generality, we can that $f_{k+1} \geq f_k$, for k large enough. Then,

$$h_{k+1} \leq (1 - \gamma)h_k \leq h_k - \gamma\epsilon$$

Therefore, $h_k \to 0(k \to \infty)$, which is a contradiction. The conclusion follows. □

Theorem 3 *Suppose the Algorithm A is applied to problem (P), and the Assumption A1–A4 hold. Let $\{x_k\}$ be the sequence of iterates produced by the algorithm. Then there are two following possible cases:*

(A) *The iteration terminates at a KKT point;*
(B) *Any accumulation point of $\{x_k\}$ is a KKT point of problem (P).*

Proof (A) It is evident according to the algorithm and Lemma 9 and Theorem 2.

(B) By the construction of Algorithm A, there are two cycles between Step 1 and Step 8, one generates the sequence $\{x_k\}$ with the form $x_{k+1} = x_k + d_k$, the other generates it with the form $x_{k+1} = x_k + \alpha_k q_k$. We prove that the claim according to the two cycles. □

Case 1 Suppose there are infinite points gotten by the relation $x_{k+1} = x_k + d_k$, by Assumption A3, there must exists a point x^* such that $x_k \to x^*(k \in K)$, where K is a infinite index set. Also, by Lemma 9, $h(x_k) \to 0, \ (k \in K)$, that means x^* is feasible point and $\Psi^0(x_k, \sigma_k) \to 0, \ (k \in K)$. Suppose x^* is not a KKT point, let $K_1 = \{k \in K | \nabla f(x_k)^T d_k > -\frac{1}{2} d_k^T H_k d_k\} \subset K$.

(i) K_1 is an infinite index set.

If $\lim_{k \in K_1, k \to \infty} \|d_k\| = 0$, then it is easy to see that x^* is a KKT point. It is a contradiction. So, without loss of generality, we suppose that $\|d_k\| > \epsilon$ for $k \in K_1$.

By $h(x_k) \to 0 \quad (k \in K)$ and Assumption A5, we can assume $\exists k_0$, for $k > k_0, k \in K_1$, it holds

$$h(x_k) \leq \frac{a\epsilon^2}{2M} \leq \frac{a\|d_k\|^2}{2M} \leq \frac{d_k^T H_k d_k}{2M} \tag{37}$$

While by KKT condition of the problem (2) and Theorem 1, we have $\nabla f(x_k) + H_k d_k + A_k U_k = 0, \ A_k = (\nabla g_j(x_k), \ j \in L_k)$. Together with (37), we obtain that for all $k \in K_1, k > k_0$, it holds

$$\begin{aligned}\nabla f(x_k)^T d_k &= -d_k^T H_k d_k - d_k^T A_k U^k \\ &= -d_k^T H_k d_k - (U^k)^T g(x_k) \\ &\le -d_k^T H_k d_k + \|U^k\|_\infty h(x_k) \\ &\le Mh(x_k) - d_k^T H_k d_k \\ &\le -\frac{1}{2} d_k^T H_k d_k \end{aligned} \tag{38}$$

which contradicts the definition of K_1.

(ii) K_1 is a finite index set.

That means it holds $\nabla f(x_k)^T d_k \le -\frac{1}{2} d_k^T H_k d_k$ for large enough k.

We have

$$f(x_k) - f(x_{k+1}) = -\nabla f(x_k)^T d_k + O(\|d_k\|^2) \ge -\nabla f(x_k)^T d_k \ge \frac{1}{2} d_k^T H_k d_k \ge \frac{a}{2}\|d_k\|^2$$

Because f is bounded below, for some integer i_0, we have

$$\infty > \sum_{k=i_0}^{\infty} (f(x_k) - f(x_{k+1})) \ge \sum_{k=i_0}^{\infty} \frac{a}{2}\|d_k\|^2$$

Then

$$\sum_{k=i_0}^{\infty} \|d_k\|^2 < +\infty$$

That means $\|d_k\| \to 0$. Hence x^* is a KKT point.

Case 2 Suppose there are infinite points gotten from the relation $x_{k+1} = x_k + \alpha_k q_k$:

Suppose also by contradiction that $\|d_k\| > \epsilon,\ k \in K$. By Lemma 3, we have $\alpha_k > \bar{\alpha} > 0$. Without loss of generality, we can assume that $q_k \to q^*$. Since $\|d_k\| > \epsilon$, it easy to see that $\nabla f(x^*)^T q^* < 0$. By Lemma 2, then

$$\begin{aligned} 0 &= \lim_{k\in K} (f(x_{k+1}) - f(x_k)) \\ &= \lim_{k\in K} (\alpha_k \nabla f(x_k)^T q_k + O(\|\alpha_k q_k\|^2)) \\ &\le \lim_{k\in K} (\alpha_k \nabla f(x_k)^T q_k) \\ &\le \bar{\alpha} \nabla f(x^*)^T q^* < 0 \end{aligned} \tag{39}$$

which is a contradiction.

Combined Case 1 and Case 2, we can see that the claim holds. □

5 Superlinear convergence of algorithm

In order to study the superlinear convergent property, we need some stronger regularity assumptions.

Assumptions

B1: $H_k \to H^*$ as $k \to \infty$.

B2: The second-order sufficiently conditions are satisfied at the KKT point x^* and the corresponding multiplier vector λ^*, i.e.

$$d^T \nabla^2_{xx} L(x^*, \lambda^*) d > 0, \ \forall d \in \{d | \nabla g_j(x^*)^T d = 0, \ \ j \in I(x^*)\}$$

where $L(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x), \ I(x^*) = \{j | g_j(x^*) = 0\}$.

B3: At x^*, strict complementarity slackness and linear independence of the gradients of the active constraints hold.

B4: Matrices $H_k, \ k = 1, 2, \ldots$ are symmetric positive definite and satisfy the following condition

$$\lim_{k \to \infty} \frac{\|(H_k - \nabla^2_{xx} L(x^*, \lambda^*))d\|}{\|d_k\|} = 0$$

Lemma 10 [13] *Sequence $\{x_k\}$ converges to the solution x^* of problem (P).*

From Theorem 3 and Lemma 9, we know that $\|d_k\| \to 0$. So, it is natural that $\|d_k\|$ satisfies $\|d_k\| \leq \sigma_k$ for k sufficiently large. Lemma 8 implies that $\Psi^0(x_k, \sigma_k) = 0$ when k is large enough. So the sequence $Q(x_k, H_k, \sigma_k)$ is equivalent to the following quadratic programming subproblem when k is sufficiently large.

$$\begin{aligned} &\min \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\ &s.t. \ \ g_j(x_k) + \nabla g_j(x_k)^T d \leq 0 \quad j \in L_k \end{aligned} \tag{40}$$

Lemma 11 *It holds, for $k \to \infty$, that*

$$L_k \equiv I(x^*) = I_*, \ \ d_k \to 0, \ \ \lambda_k \to (\lambda^*_j, j \in I_*), \ \ U^k \to \lambda^*$$

where (d_k, λ_k) is the KKT pair of the above quadratic programming subproblem.

Proof By the statements above and Lemma 9, $H_k \to H^*$, it holds that $d_k \to 0$ as $k \to \infty$. According to Lemma 5 and $x_k \to x^*$, it follows that $I_* \subset L \equiv L_k$.

First, we prove that

$$\lambda_k \to (\lambda^*_j, j \in L)$$

Since x^* is a KKT point of problem (P), we have

$$\nabla f(x^*) + A_* \lambda^*_L = 0, \ \ \lambda^*_L \geq 0, \ \ \lambda^*_j = 0 \ j \in I \setminus L$$

where $\lambda^*_L = (\lambda^*_j, \ j \in L), \ \ A_* = (\nabla g_j(x^*), \ j \in L)$.

From Lemma 5 and $x_k \to x^*$, it following that

$$A^T_* A_* \text{ is nonsingular, and } (A^T_k A_k)^{-1} \to (A^T_* A_*)^{-1}$$

So $\lambda^*_L = -(A^T_* A_*)^{-1} A^T_* \nabla f(x^*)$

Moreover, by KKT condition of problem (2), we have

$$\nabla f(x_k) + H_k d_k + A_k \lambda_k = 0$$

Hence, $\lambda_k = -(A^T_k A_k)^{-1} A^T_k (\nabla f(x_k) + H_k d_k) \to -(A^T_* A_*)^{-1} A^T_* \nabla f(x^*) = \lambda^*_L$.

While, it is easy to see that $U^k \to \lambda^*$.

Second, we prove that $L \subset I_*$.

For $j_0 \notin L$, if $j_0 \notin I_*$ by contradiction, there must be a constant $\xi_0 > 0$ such that $g_{j_0}(x^*) \leq -\xi_0 < 0$. Again, since $g_{j_0}(x)$ is continuously differentiable, and $d_k \to 0$ ($k \to \infty$), we have for k large enough

$$g_{j_0}(x^*) + \nabla g_{j_0}(x^*)^T d_k \leq -\frac{\xi_0}{2} < 0$$

which means $j_0 \notin L$, contradicts the above assumption. Hence $L \equiv L_k \equiv I_*$. □

Lemma 12 *Suppose Assumption A1–A5, B1–B4 hold, then $x_{k+1} = x_k + d_k$ for k sufficiently large.*

Proof Suppose x_k is acceptable for the filter, we will show that for k sufficiently large, $x_k + d_k$ is acceptable for the filter.

From Lemma 9 and Lemma 11, we know that $d_k \to 0$, $h_k \to 0$ as $k \to \infty$. Also, by the construction of Algorithm A, we have $h(x_k + d_k) = o(\|d_k\|^2)$. So, we just need to show that $f(x_k + d_k) \le f(x_k) + \gamma h(x_k)$. Let $s_k = f(x_k + d_k) - f(x_k) - \gamma h(x_k)$, we have

$$s_k = \nabla f(x_k)^T d_k + \frac{1}{2} d_k^T \nabla^2 f(x_k) d_k + o(\|d_k\|^2)$$

While by the KKT condition of the problem (2) and $h_k \to 0$, we have

$$\nabla f(x_k)^T d_k = -d_k^T H_k d_k - \sum_{j=1}^{m} u_j^k \nabla g_j(x_k)^T d_k$$

$$g_j(x_k) + \nabla g_j(x_k)^T d_k + \frac{1}{2} d_k^T \nabla^2 g_j(x_k) d_k = o(\|d_k\|^2)$$

Then it holds

$$\begin{aligned} s_k &= -d_k^T H_k d_k + \sum_{j=1}^{m} u_j^k g_j(x_k) + \frac{1}{2} d_k^T \nabla^2 L(x_k, U^k) d_k + o(\|d_k\|^2) \\ &= -\frac{1}{2} d_k^T H_k d_k + \sum_{j=1}^{m} u_j^k g_j(x_k) + \frac{1}{2} d_k^T (\nabla^2 L(x_k, U^k) - H_k) d_k + o(\|d_k\|^2) \end{aligned} \tag{41}$$

According to $u_j^k \to \lambda_j^* > 0,\ g_j(x_k) \to g_j(x^*) < 0,\ j \in I_*$ and Assumption A5, we have

$$\begin{aligned} s_k \le &-\frac{a}{2}\|d_k\|^2 + \frac{1}{2} d_k^T (\nabla_{xx}^2 L(x_k, U^k) - \nabla_{xx}^2 L(x^*, \lambda^*)) d_k \\ &+ \frac{1}{2} d_k^T (\nabla_{xx}^2 L(x^*, \lambda^*) - H_k) d_k + o(\|d_k\|^2) \end{aligned} \tag{42}$$

Since $x_k \to x^*$, $U^k \to \lambda^*$ and Assumption A3, then

$$d_k^T (\nabla_{xx}^2 L(x_k, U^k) - \nabla_{xx}^2 L(x^*, \lambda^*)) d_k = o(\|d_k\|^2)$$

Assumption B4 implies that

$$d_k^T (\nabla_{xx}^2 L(x^*, \lambda^*) - H_k) d_k = o(\|d_k\|^2)$$

Therefore, when k is sufficiently large, it holds

$$s_k \le -\frac{a}{2}\|d_k\|^2 + o(\|d_k\|^2) \le 0$$

Hence, for all k large enough, $x_k + d_k$ is acceptable for the filter.

In view of Lemma 12 and the way of Theorem 5.2 in [5], it is easy to get the convergence theorem as follows. □

Theorem 4 *Under all stated assumptions, the algorithm is superlinear convergent, i.e. the sequence $\{x_k\}$ generated by the algorithm satisfies $\|x_{k+1} - x^*\| = o(\|x_k - x^*\|)$.*

6 Numerical experiments

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

(1) Updating of H_k is done by

$$H_{k+1} = \begin{cases} H_k & \text{if } s_k^T y_k \le 0 \\ H_k + \dfrac{y_k^T y_k}{y_k^T s_k} - \dfrac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} & \text{if } s_k^T y_k > 0 \end{cases} \tag{43}$$

(2) The stop criteria is $\|d_k\|$ sufficiently small.

(3) If an equality constraint $h(x) = 0$ exists in the original problem, it is most easily handle as two corresponding inequalities $h(x) \le 0$ and $h(x) \ge 0$, and we can apply the above algorithm.

(4) The algorithm parameters were set as follows: $\sigma_l = 1, \sigma_r = 2, \gamma = 0.1, H_0 = I \in R^{n\times n}$. The program is written in Matlab.

Example 1

$$\begin{aligned} &min\ f(x) = x - \frac{1}{2} + \frac{1}{2}cos^2(x) \\ &s.t.\ \ x \ge 0 \end{aligned}$$

$$x_0 = 2, x^* = 0, f(x^*) = 0, \text{iterate} = 2.$$

Example 2

$$\begin{aligned} &min\ f(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 \\ &s.t.\ \ 6 - x_1^2 - x_2^2 - x_3^2 - x_4^2 \le 0 \end{aligned}$$

$$x_0 = (2, 2, 2, 2)^T, x^* = (1.2247, 1.2247, 1.2247, 1.2247)^T, f(x^*) = 6, \text{iterate} = 5.$$

Example 3

$$\begin{aligned} &min\ f(x) = \sum_{i=1}^{3} x_i^2 x_{i+1}^2 + x_1 x_4 \\ &s.t.\ \ 4 - \sum_{i=1}^{4} x_i \le 0 \\ &\qquad 1 - \sum_{i=1}^{4} (-1)^{i+1} x_i \le 0 \end{aligned}$$

$$x_0 = (2.5, 1.5, 0, 0)^T, x^* = (1.2400, 0.7533, 1.2600, 0.7467)^T, f(x^*) = 3.5844, \text{iterate} = 6.$$

Example 4

$$\begin{aligned} &min\ f(x) = \frac{4}{3}(x_1^2 - x_1 x_2 + x_2^2)^{\frac{3}{4}} - x_3 \\ &\text{s.t.}\ \ x \ge 0 \\ &\qquad x_3 \le 2 \end{aligned}$$

$$x_0 = (0, 0.25, 0)^T, x^* = (0, 0, 2)^T, f(x^*) = -2, \text{iterate} = 7.$$

Comparing with the results in [13] and [14], the computation in this paper is less than that method in [13] and [14]. So, the algorithm is effective.

Acknowledgements The author would like to thank the tutor, Prof. Dingguo Pu for his valuable suggestions and comments for improving the paper. This work is supported by the National Natural Science Foundation of China (No. 10571137).

References

1. Boggs, P.T., Tolle, J.W., Wang, P.: On the local convergence of quasi-newton methods for constrained optimization. SIAM J. Control Optim. **20**, 161–171 (1982)
2. Bonnons, J.F., Painer, E.R., Titts, A.L., Zhou, J.L.: Avoiding the Maratos effect by means of nonmontone linesearch, Inequality constrained problems-feasible iterates. SIAM J. Numer. Anal. **29**, 1187–1202 (1992)
3. Burke, J.V., Han, S.P.: A robust SQP method. Math. Program. **43**, 277–303 (1989)
4. Du, D.Z., Pardalos, P.M., Wu, W.L.: Mathematical Theory of Optimization. Kluwer Academic Publishers, Boston (2001)
5. Facchinei, F., Lucidi, S.: Quadraticly and superlinearly convergent for the solution of inequality constrained optimization problem. J. Optim. Theory Appl. **85**, 265–289 (1995)
6. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Math. Program. **91**, 239–269 (2002)
7. Fletcher, R., Leyffer, S., Toint, P.L.: On the global convergence of a filter-SQP algorithm. SIAM J. Optim. **13**, 44–59 (2002)
8. Fletcher, R., Gould, N.I.M., Leyffer, S., Toint, P.L., Wachter, A.: Global convergence of a trust region SQP-filter algorithm for general nonlinear programming. SIAM J. Optim. **13**, 635–660 (2002)
9. Han, S.P.: Superlinearly convergence variable metric algorithm for general nonlinear programming problems. Math. Program. **11**, 263–282 (1976)
10. Nie, P.Y., Ma, C.F.: A trust region filter mehtod for general nonlinear programming. Appl. Math. Comput. **172**, 1000–1017 (2006)
11. Powell, M.J.D.: A fast algorithm for nonlinear constrained optimization calculations. In: Waston, G.A. (ed.) Numerical Analysis., pp. 144–157. Springer-Verlag, Berlin (1978)
12. Powell, M.J.D.: Variable metric methods for constrained optimization. In: Bachen, A., et al. (eds.) Mathematical Programming-The state of Art. Springer-Verlag, Berlin (1982)
13. Zhang, J.L., Zhang, X.S.: A modified SQP method with nonmonotone linesearch technique. J. Global Optim. **21**, 201–218 (2001)
14. Zhou, G.L.: A modified SQP method and its global convergence. J. Global Optim. **11**, 193–205 (1997)
15. Zhu, Z.B., Zhang, K.C., Jian, J.B.: An improved SQP algorithm for inequality constrained optimization. Math. Methods Oper. Res. **58**, 271–282 (2003)
16. Zhu, Z.B.: A globally and superlinearly convergent feasible QP-free method for nonlinear programming. Appl. Math. Comput. **168**, 519–539 (2005)