

RESEARCH

Open Access



# A nonmonotone flexible filter method for nonlinear constrained optimization

Ke Su<sup>\*</sup>, Xiaochuan Li and Ruyue Hou

<sup>\*</sup>Correspondence:  
pigeonsk@163.com  
College of Mathematics and  
Computer Science, Hebei University,  
180 East Wusi Road, Baoding, China

## Abstract

In this paper, we present a flexible nonmonotone filter method for solving nonlinear constrained optimization problems which are common models in industry. This new method has more flexibility for the acceptance of the trial step compared to the traditional filter methods, and requires less computational costs compared with the monotone-type methods. Moreover, we use a self-adaptive parameter to adjust the acceptance criteria, so that Maratos effect can be avoided a certain degree. Under reasonable assumptions, the proposed algorithm is globally convergent. Numerical tests are presented that confirm the efficiency of the approach.

**MSC:** Primary 90C30; secondary 65K05

**Keywords:** nonmonotone; filter; self-adaptive; global convergence; trust region

## 1 Introduction

We consider the following inequality constrained nonlinear optimization problem

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

where  $x \in \mathbb{R}^n$ , the functions  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  and  $c_i (i \in I): \mathbb{R}^n \rightarrow \mathbb{R}$  are all twice continuously differentiable. For convenience, let  $g(x) = \nabla f(x)$ ,  $c(x) = (c_1(x), c_2(x), \dots, c_m(x))^T$  and  $A(x) = (\nabla c_1(x), \nabla c_2(x), \dots, \nabla c_m(x))$ . 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.

There are various methods for solving the inequality constrained nonlinear optimization problem (P). For example, sequential quadratic programming methods, trust region approaches [1], penalty methods and interior point methods [2]. But in these works, a penalty or Lagrange function is always used to test the acceptability of the iterates. 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. In 2002, Fletcher and Leyffer [3] proposed a class of filter methods, which does not require any penalty parameter and has promising numerical results. Consequently, filter technique has employed to many approaches, for instance, SLP methods [4], SQP methods [5, 6], interior point approaches [7] and derivative-free optimization [8, 9]. Furthermore, Fletcher *et al.* [5] proved the global convergence of the filter-SQP method, then Ulbrich and Ulbrich [10] showed its superlin-

ear local convergence. But the filter methods also encounter the Maratos effect. Maratos effect, observed by Maratos in his PhD thesis in 1978, means some steps that make good progress toward a solution are rejected by the merit function. To overcome the drawback in filter methods, Ulbrich [11] introduced a new filter method using the Lagrangian function instead of the objective function as the acceptance criterion. After that, Nie and Ma [12] used a fixed scalar to combine the objective function and violation constraint function as one measure in the entry of the filter. But both of them used the fixed criterion to decide whether accept a trial point or not, that means the criterion is invariable no matter what improvements made by the trial point. Actually, if we can change the criterion according to the different improvements made by the current trial point, we can avoid Maratos effect to a certain degree, and decrease the computational costs as well.

On the other hand, the promising numerical results of filter methods owe to their non-monotonicity in a certain degree. Based on this property, some other non-monotone-type filter methods are proposed [13–15]. Gould and Toint [16] also introduced a new non-monotone filter method using the area of the region in  $h - f$  plane as the criteria to decide whether a trial point is acceptable or not, where  $h = h(x)$  is the constraint violation function and  $f = f(x)$  is the objective function at the current point  $x$ .

Motivated by the idea and methods above, we proposed a class of nonmonotone filter trust region methods with self-adaptive parameter for solving problem (P). Our method improves previous non-monotone filter method. Unlike Ulbrich [11], we do not use a Lagrangian function in the filter but use the similar type of function as that in Nie and Ma [12]. Moreover, different from Nie and Ma [12], the parameter in our method is not fixed but variable, that means the criterion is adjusted according to the different improvements. To avoid the trial point from falling into a ‘valley’, we also add the non-monotonic technique into the criterion. Different from existing SQP-filter methods, we use a quadratic subproblem that always feasible to avoid the feasible restoration, hence decrease the scale of the calculation to a certain degree.

This paper is organized as follows: in Section 2, we introduce the feasible SQP subproblem and the non-monotonic flexible filter. We propose the non-monotone filter method with self-adaptive parameter in Section 3. Section 4 presents the global convergence properties and some numerical results are reported in Section 5. We end our presentation in short conclusion in Section 6.

## 2 The modified SQP subproblem and the non-monotone flexible filter method

### 2.1 The modified SQP subproblem

Our algorithm is an SQP method, to avoid the infeasibility of the quadratic subproblem, we choose a quadratic program that presented by Zhou [17]. At the  $k$ th iterate, we compute a trial step by solving the following quadratic problem,

$$\begin{aligned}
 Q(x_k, H_k, \rho_k): \quad & \min g_k^T d + \frac{1}{2} d^T H_k d \\
 \text{s.t.} \quad & c_j(x_k) + \nabla c_j(x_k)^T d \leq \Psi^+(x^k, \rho_k), \quad j \in I, \\
 & \|d\| \leq \rho_k,
 \end{aligned} \tag{1}$$

where

$$\Psi^+(x_k, \rho_k) = \max\{\Psi(x_k, \rho_k), 0\}, \tag{2}$$

$$\Psi(x_k, \rho_k) = \min\{\bar{\Psi}(x_k; d_k) : \|d_k\| \leq \rho_k\} \tag{3}$$

and  $\bar{\Psi}(x_k; d_k)$  is the first order approximation to  $\Psi(x_k + d_k) = \max\{c_j(x_k + d_k) : j \in I\}$ , namely

$$\bar{\Psi}(x_k; d_k) = \max\{c_j(x_k) + \nabla c_j(x_k)^T d_k : j \in I\} \tag{4}$$

and  $\rho_k > 0$ . We notice that these convex programs have the following properties.

We can condense the above definitions by the following form

$$\Psi^+(x_k, \rho_k) = \max\left\{ \min_{\|d_k\| \leq \rho_k} \left\{ \max_{j \in I} \{c_j(x_k) + \nabla c_j(x_k)^T d_k\} \right\}, 0 \right\}. \tag{5}$$

**Lemma 1** [17] *If  $d_k = 0$  is the solution to  $Q(x_k, H_k, \rho_k)$ , then  $x_k$  is a KKT point of the problem (P).*

*Proof* The proof is similar to that of Lemma 4.1 in [16]. □

### 2.2 The non-monotone flexible filter with a self-adaptive parameter

In traditional filter method, originally proposed by Fletcher and Leyffer [3], the acceptability of iterates is determined by comparing the value of constraint violation and the objective function with previous iterates collected in a filter. Define the violation function  $h(x)$  by  $h(x) = \|c(x)^+\|_\infty$ , where  $c_i(x)^+ = \max\{c_i(x), 0, i \in I\}$ . Obviously,  $h(x) = 0$  if and only if  $x$  is a feasible point. So a trial point should either reduce the value of constraint violation or the objective function  $f$ .

Definition of filter set is based on the definition of dominance as following,

**Definition 1** A pair  $(h_k, f_k)$  is dominated by  $(h_j, f_j)$  if and only if  $h_k \leq h_j$  and  $f_k \leq f_j$  for each  $j \neq k$ .

**Definition 2** A filter set  $\mathcal{F}$  is a set of pairs  $(h, f)$  such that no pair dominates any other.

To ensure the convergence, some additional conditions are required to decide whether to accept a trial point to the filter or not. The traditional acceptable criterion is as following.

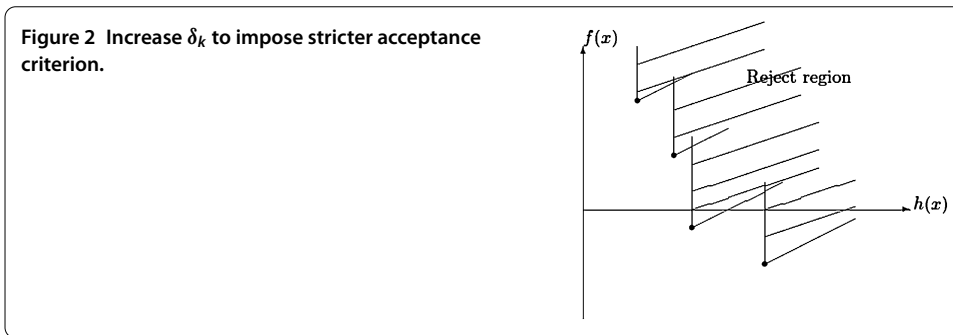
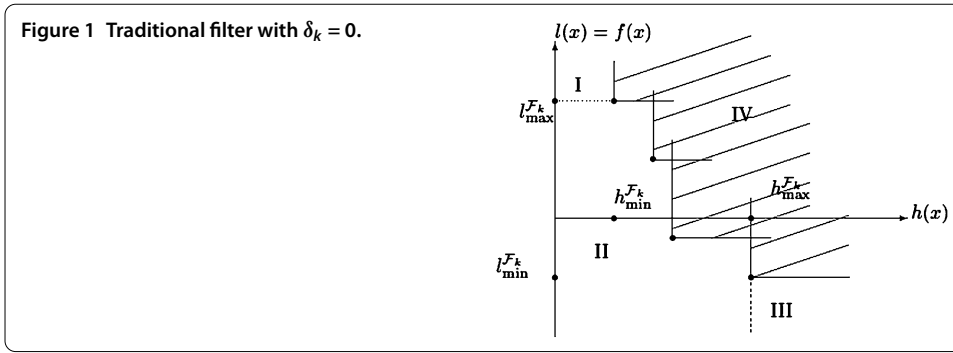
**Definition 3** A trial point  $x$  is called acceptable to the filter if and only if

$$\text{either } h(x) \leq \beta h_j \text{ or } f(x) \leq f_j - \gamma h_j \text{ for } \forall (h_j, f_j) \in \mathcal{F}, \tag{6}$$

where  $0 < \gamma < \beta < 1$  are constants. In practice,  $\beta$  is close to 1 and  $\gamma$  close to 0.

Actually, in traditional filter method, some good point such as superlinear convergent step may be rejected due to the increase of both objective function value and constraint violation value compared to other entries in filter. That is the reason why the Maratos effect occurs. So motivated by [12], we substitute the original objective function  $f(x_k)$  at the  $k$ th iterate by the following function

$$l(x_k) = f(x_k) + \delta_k h(x_k) = f(x_k) + \delta_k \|c(x_k)^+\|_\infty, \tag{7}$$



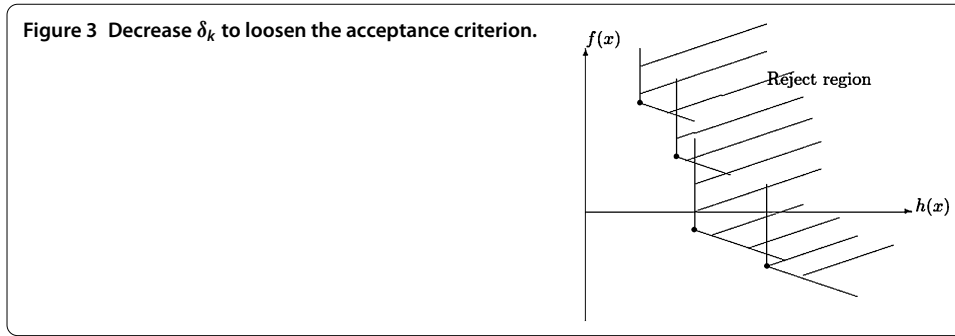
where  $c_i(x_k)^+ = \max\{c_i(x_k), 0\}$  for  $i = 1, 2, \dots, m$ . Here  $\delta_k$  is a self-adaptive parameter at  $k$ th iterate, it can be changed according to the different improvements that made by the current trial point. Note that the traditional filter methods are the special cases with  $\delta_k = 0$ , and we hope overcome the Maratos effect with suitable  $\delta_k$ .

We aim to reduce the value of both  $h(x)$  and  $l(x)$ . By original criterion, the trial point is acceptable if and only if (6) holds. Nie and Ma [12] proposed a trust region filter method with a given penalty parameter which is negative, but in this paper, different from [12], the parameter  $\delta_k$  is a variable scalar which is changed according to the different improvements caused by the trial point. Specifically, at the beginning, let  $\delta_0 = 0$ , that is what the traditional filter method does, and  $f(x_k) = l(x_k)$  (see Figure 1).

There are four regions in the right-hand half space I, II, III, IV. At the current iterate  $k$ , if the trial point  $x_k$  moves into the region IV, that means the pair  $(h_k, l_k)$  is located in region IV, we say that the trial point is rejected according to our criterion. If  $x_k$  moves into the region I, II, or III, we accept it, but need to adjust the parameter  $\delta_k$  in the criterion. For region III, we say that the algorithm does not make a good improvement, since we do not want to accept points with larger constraint violation. Thus we intend to impose stricter acceptance criterion, that means to increase the value of  $\delta_k$ , which will result in the bigger reject area and smaller acceptable area (see Figure 2). So update  $\delta_k$  as following:

$$\delta_{k+1} = \min \left\{ \rho_k, \delta_k + \left| \frac{l_k - l_k^+}{h_k - h_k^+} \right| \right\}. \tag{8}$$

If  $x_k$  moves into the region II, we say that the algorithm makes good improvement since it reduces not only the objective function  $l(x_k)$  but also the constraint violation  $h(x_k)$ , so we intend to loosen the acceptance criterion to hope for more improvements. That means to decrease the value of  $\delta_k$  so that make the reject area become smaller and the acceptable



area bigger (see Figure 3). So update  $\delta_k$  as following:

$$\delta_{k+1} = \max \left\{ -\rho_k, \delta_k - \left| \frac{l_k - l_k^+}{h_k - h_k^+} \right| \right\}. \tag{9}$$

If  $x_k$  moves into region I, we will also accept it because the value of constraint violation does decrease, and we can also accept the increase of  $l(x_k)$  in finite steps. Meanwhile, the value of  $\delta_k$  will not be changed. If  $x_k$  moves into region IV, that means this trial point is rejected, and  $\delta_k$  also should be remained in the next iterate.

As we all know, because of the non-monotone properties of filter method in a certain degree, it has the good numerical results. Su and Pu [13] also proposed a modified non-monotone filter method to exhibit a further non-monotone technique. Motivated by this, we loosen the acceptance criterion by non-monotonic technique and give the following criteria.

**Definition 4** A point  $x$  is acceptable to the filter if and only if

$$h(x) \leq \beta \max_{0 \leq r \leq m(k)-1} h_{k-r} \quad \text{or} \quad l(x) \leq \max \left[ l_k, \sum_{r=0}^{m(k)-1} \lambda_{kr} l_{k-r} \right] - \gamma h(x), \tag{10}$$

where  $(h_{k-r}, l_{k-r}) \in \mathcal{F}$  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  $\lambda$  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 equivalent to the traditional method with the modified acceptance rule (10).

To control the infeasibility, an upper bound condition of violation function is needed, namely  $h(x) \leq u$ , where  $u$  is a positive scalar, which can be implemented in the algorithm by initiating the filter with the pair  $(u, -\infty)$ .

### 3 A nonmonotone flexible filter algorithm

At the current  $k$ th iterate, the trial point  $x_k$  is accepted by our algorithm if it satisfies two conditions, first is accepted by the filter set, second is sufficiently reduction. We define the sufficient reduction condition is as following:

$$\text{rared}_k^l \geq \eta \text{pred}_k^f \quad \text{and} \quad h_{l(k)} \leq \alpha_1 \|d_k\|_\infty^{\alpha_2}, \tag{11}$$

where  $\alpha_1, \alpha_2$  are constants, the relaxed actual reduction  $\text{rared}_k^l$  and the predicted reduction  $\text{pred}_k^f$  are defined as

$$\text{rared}_k^l = l(x_k + d_k) - \max \left\{ l(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} l_{k-r} \right\}, \tag{12}$$

$$\text{pred}_k^f = -g_k^T d_k - \frac{1}{2} d_k^T H_k d_k, \tag{13}$$

$$h_{l(k)} = \max_{0 \leq r \leq m(k)-1} h_{k-r} \tag{14}$$

and the matrix  $H_k$  is the Hessian matrix  $\nabla^2 f(x_k)$  or an approximate to it,  $\sum_{r=0}^{m(k)-1} \lambda_{kr} = 1$ ,  $\lambda_{kr} \in (0, 1)$ ,  $0 \leq m(k) \leq \min\{m(k-1) + 1, M\}$ ,  $M \geq 1$  is a given positive constant.

A formal description of the algorithm is given as follows.

**Algorithm A**

- Step 0.* Let  $0 < \rho_0 < 1$ ,  $0 < \gamma < \beta < 1$ ,  $0 < \lambda \leq 1$ ,  $0 < \gamma_0 < \gamma_1 \leq 1 < \gamma_2$ ,  $M \geq 1$ ,  $u > 0$ ,  $\alpha_1 = \alpha_2 = 0.5$ . Choose an initial point  $x_0 \in R^n$ , a symmetric matrix  $H_0 \in R^{n \times n}$  and an initial region radius  $\Delta_0 \geq \Delta_{\min} > 0$ ,  $\mathcal{F}_0 = \{(u, -\infty)\}$ . Set  $k = 0$ ,  $m(k) = 0$ .
- Step 1.* Solve the subproblem  $Q(x_k, H_k, \rho_k)$ , if  $\|d_k\| = 0$ , stop.
- Step 2.* Let  $x_k^+ = x_k + d_k$ , compute  $h_k^+, l_k^+$ .
- Step 3.* If  $x_k^+$  is acceptable to the filter  $\mathcal{F}_k$ , go to step 4, otherwise go to step 5.
- Step 4.* If  $x_k^+$  is located in the region I or region IV, let  $\delta_{k+1} = \delta_k$ , if  $x_k^+$  is located in the region II, let  $\delta_{k+1}$  is updated by (9), if  $x_k^+$  is in the region III, let  $\delta_{k+1}$  is updated by (8).
- Step 5.* If  $\text{rared}_k^l \leq \eta \text{pred}_k^f$  and  $h_l(k) \leq \alpha_1 \|d_k\|_\infty^{\alpha_2}$ , then go to step 6, otherwise go to step 7.
- Step 6.* Let  $\rho_k \in [\gamma_0 \rho_k, \gamma_1 \rho_k]$ , go to step 1.
- Step 7.* Let  $x_{k+1} = x_k^+$ , update the filter set.  $\rho_{k+1} \in [\rho_k, \gamma_2 \rho_k] \geq \rho_{\min}$ , update  $H_k$  to  $H_{k+1}$ ,  $m(k+1) = \min\{m(k) + 1, M\}$ ,  $k = k + 1$  and go to step 1.

**Remark 1** At the beginning of each iteration, we always set  $\rho_k \geq \rho_{\min}$ , which will avoid too small trust region radius.

**Remark 2** In above algorithm, let  $M$  be a nonnegative integer. For each  $k$ , let  $m(k)$  satisfy

$$m(0) = 0, \quad 0 \leq m(k) \leq \min\{m(k-1) + 1, M\} \quad \text{for } k \geq 1.$$

In fact, if  $M = 1$ , the algorithm actual is a monotone method, the nonmonotonicity is showed as  $M > 1$ .

**4 The convergent properties**

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  $c_i$  ( $i \in I = \{1, 2, \dots, m\}$ ) are twice continuously differentiable.
- A2. For all  $k$ ,  $x_k$  and  $x_k + d_k$  all remain in a closed, bounded convex subset  $S \subset R^n$ .

- A3. The matrix sequence  $\{H_k\}$  is uniformly bounded.
- A4. The functions  $A = \nabla c$  are uniformly bounded on  $S$ .

By the above assumptions, we can suppose that there exist constants  $\nu_1, \nu_2, \nu_3$  such that  $\|f(x)\| \leq \nu_1, \|\nabla f(x)\| \leq \nu_1, \|\nabla^2 f(x)\| \leq \nu_1, \|c(x)\| \leq \nu_2, \|\nabla c(x)\| \leq \nu_2, \|\nabla^2 c(x)\| \leq \nu_2$ .

**Definition 5** [2] The Mangasarian-Fromowitz constraint qualification (MFCQ) is said to be satisfied at a point  $x \in \mathbb{R}^n$  with respect to the underlying constraint system  $g(x) \leq 0$ , if there is a  $z \in \mathbb{R}^n$  such that

$$\nabla c_i(x)^T z < 0, \quad i \in \{i : c_i(x) \geq 0, i \in I\}. \tag{15}$$

**Lemma 2** [12] *Let Assumptions hold, and let  $\bar{x}$  be a feasible point of problem (P) at which MFCQ holds but which is not a KKT point. Then there exists a neighborhood  $N$  of  $\bar{x}$  and positive constants  $\xi_1, \xi_2, \xi_3$  such that for all  $x_k \in N \cap S$  and all  $\rho_k$  for which*

$$\xi_2 h_k \leq \rho_k \leq \xi_3 \tag{16}$$

*it follows that SQP subproblem has a feasible solution  $d_k$ , and the predicted reduction satisfies*

$$\text{pred}_k^f \geq \frac{1}{3} \rho_k \xi_1. \tag{17}$$

*If  $\rho_k \leq (1 - \eta_3)\xi_1/3\nu_2$ , then*

$$f(x_k) - f(x_k^+) \geq \eta \text{pred}_k^f, \tag{18}$$

*where  $\eta < \eta_3$ .*

*If  $h_k > 0$  and  $\rho_k \leq \sqrt{\frac{2\beta h_k}{n^2 \nu_2}}$  then  $h(x_k^+) \leq \beta h_k$ .*

**Lemma 3** *Suppose that Assumptions hold, then Algorithm A is well defined.*

*Proof* We will show that the trial point  $x_k^+$  is acceptable to the filter when  $\rho_k$ , is small enough. We consider the following two cases.

Case 1.  $h_k = 0$ .

To prove the implementation of Algorithm A, we have to show for all  $k$  such that  $\rho_k \leq \delta$  it holds  $\text{ared}_k^l \geq \eta \text{pred}_k^f$ . We know  $\text{ared}_k^l = l(x_k) - l(x_k^+)$ .

In fact,

$$\begin{aligned} |\text{ared}_k^l - \text{pred}_k^f| &= \left| l(x_k) - l(x_k^+) + g_k^T d_k + \frac{1}{2} d_k^T H_k d_k \right| \\ &= \left| f(x_k) + \delta_k h(x_k) - f(x_k^+) - \delta_{k+1} h(x_k + d_k) + g_k^T d_k + \frac{1}{2} d_k^T H_k d_k \right| \\ &\leq \left| \frac{1}{2} d_k^T (\nabla^2 f(y_k) - H_k) d_k + \delta_k h(x_k) - \delta_{k+1} h(x_k^+) \right| \\ &\leq \rho_k^2 \frac{1}{2} \|\nabla^2 f(y_k) - H_k\| + |\delta_{k+1}| h(x_k^+), \end{aligned} \tag{19}$$

where  $y_k = x_k + \xi d_k$ ,  $\xi \in (0, 1)$  denotes some point on the line segment from  $x_k$  to  $x_k^+$ . By the update of  $\delta_k$  and the definition of  $h(x_k^+)$ , we know  $|\delta_{k+1}| \leq \rho_k$ ,

$$h(x_k^+) = \|c^+(x_k + d_k)\| = \left\| c^+(x_k) + A(x_k)d_k + \frac{1}{2}d_k^T \nabla^2 c(s_k)d_k \right\| \leq v_2 \rho_k + \frac{1}{2} v_2 \rho_k^2,$$

where  $s_k$  denotes some point in the line from  $x_k$  to  $x_k^+$ .

Hence we obtain that

$$\begin{aligned} |\text{ared}_k^l - \text{pred}_k^f| &\leq \rho_k^2 b + \rho_k \left( v_2 \rho_k + \frac{1}{2} v_2 \rho_k^2 \right) \\ &\leq \left( b + v_2 \left( 1 + \frac{1}{2} \delta \right) \right) \rho_k^2, \end{aligned} \tag{20}$$

where  $b = \frac{1}{2}(\sup \|H_k\| + \max_{x \in S} \|\nabla^2 f(x)\|)$ , together with Lemma 2

$$|\text{pred}_k^f| = \left| -g_k^T d_k - \frac{1}{2} d_k^T H_k d_k \right| \geq \frac{1}{3} \xi_1 \rho_k.$$

We have

$$\left| \frac{\text{ared}_k^l - \text{pred}_k^f}{\text{pred}_k^f} \right| \leq \frac{(b + v_2(1 + \frac{1}{2}\delta))\rho_k^2}{\frac{1}{3}\xi_1\rho_k} \rightarrow 0 \quad \text{as } \rho_k \rightarrow 0. \tag{21}$$

We deduce that  $\text{rared}_k^l \geq \text{ared}_k^f \geq \eta \text{pred}_k^f$  for some  $\eta \in (0, 1)$ , since

$$\text{rared}_k^l = \max \left\{ l(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} l_{k-r} \right\} - l(x_k^+) \geq l(x_k) - l(x_k^+) = \text{ared}_k^l. \tag{22}$$

By  $\max\{l(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} l_{k-r}\} - l(x_k^+) \geq \eta \text{pred}_k^f > \gamma h(x_k)$ , we can see

$$l(x_k^+) \leq \max \left\{ l(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} l_{k-r} \right\} - \gamma h(x_k),$$

so  $x_k^+$  is acceptable to the filter.

Case 2.  $h_k > 0$ .

There exists a constant  $\delta > 0$  and  $k_0$  such that  $\rho_k \leq \delta$  when  $k < k_0$ . Let  $\delta = \sqrt{\frac{2\beta h_k}{n^2 M_2}}$  by Lemma 2, we have  $h_k^+ \leq \beta h_k$ , that is  $h_k^+ \leq \beta \max_{0 \leq j \leq m(k)-1} \{h_{k-j}\}$ . So  $x_k^+$  must be acceptable to the filter by the definition.

With the similar analysis to case 1, we have

$$\begin{aligned} |\text{ared}_k^l - \text{pred}_k^f| &\leq \left| \frac{1}{2} d_k^T (\nabla^2 f(y_k) - H_k) d_k + \sigma_k h(x_k) - \sigma_{k+1} h(x_k^+) \right| \\ &\leq \rho_k^2 \frac{1}{2} \|\nabla^2 f(y_k) - H_k\| + |\sigma_k| h(x_k) + |\sigma_{k+1}| h(x_k^+) \\ &\leq \rho_k^2 b + \rho_k (1 + \beta h(x_k)) \\ &\leq \left( b + \frac{1 + \beta}{\xi_2} \right) \rho_k^2. \end{aligned} \tag{23}$$



Then it holds

$$\left| \frac{\text{ared}_k^f - \text{pred}_k^f}{\text{pred}_k^f} \right| \leq \frac{(b + \frac{1+\beta}{\xi_2})\rho_k^2}{\frac{1}{3}\xi_1\rho_k} \rightarrow 0 \quad \text{as } \rho_k \rightarrow 0. \tag{24}$$

The conclusion follows. This is the end of proof. □

**Lemma 4** *Suppose that Assumptions hold and Algorithm A does not terminated finitely, then  $\lim_{k \rightarrow \infty} h_k = 0$ .*

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

- (i)  $K_1 = \{k | h_k^+ \leq \beta \max_{0 \leq r \leq m(k)-1} h_{k-r}\}$  is an infinite set.
- (ii)  $K_2 = \{k | l_k^+ \leq \max[l_k, \sum_{r=0}^{m(k)-1} \lambda_{kr} l_{k-r}] - \gamma h_k\}$  is an infinite set.

In view of convenience, let

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

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

- (i) Since  $m(k+1) \leq m(k) + 1$ , we have

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

which implies that  $\{h(x_{l(k)})\}$  converges. Then by  $h(x_{k+1}) \leq \beta \max_{0 \leq r \leq m(k)-1} [h(x_{k-r})]$ , we have

$$h(x_{l(k)}) \leq \beta h(x_{l(l(k)-1)}). \tag{26}$$

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

Therefore

$$h(x_{k+1}) \leq \beta h(x_{l(k)}) \rightarrow 0$$

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

- (ii) We first show that for all  $k \in S$ , it holds

$$l_k \leq l_0 - \lambda\gamma \sum_{r=0}^{k-2} h_r - \gamma h_{k-1} \leq l_0 - \lambda\gamma \sum_{r=0}^{k-1} h_r. \tag{27}$$

We prove this by induction.

If  $k = 1$ , we have  $l_1 \leq l_0 - \gamma h_0 \leq l_0 - \lambda\gamma h_0$ .

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

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

$$l_k^+ \leq l_k - \gamma h_k \leq l_0 - \lambda\gamma \sum_{r=0}^{k-1} h_r - \gamma h_k \leq l_0 - \lambda\gamma \sum_{r=0}^k h_r. \tag{28}$$

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

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

$$\begin{aligned} l_{k+1} &\leq \sum_{t=0}^p \lambda_{kt} l_{k-t} - \gamma h_k \\ &\leq \sum_{t=0}^p \lambda_{kt} \left( l_0 - \lambda\gamma \sum_{r=0}^{k-t-2} h_r - \gamma h_{k-t-1} \right) - \gamma h_k \\ &= \lambda_{k0} \left( l_0 - \lambda\gamma \sum_{r=0}^{k-p-2} h_r - \lambda\gamma \sum_{r=k-p-1}^{k-2} h_r - \gamma h_{k-1} \right) - \gamma h_k \\ &\quad + \lambda_{k1} \left( l_0 - \lambda\gamma \sum_{r=0}^{k-p-2} h_r - \lambda\gamma \sum_{r=k-p-1}^{k-3} h_r - \gamma h_{k-2} \right) \\ &\quad + \dots + \lambda_{kp} \left( l_0 - \lambda\gamma \sum_{r=0}^{k-p-2} h_r - \gamma h_{k-p-1} \right) \\ &\leq \sum_{t=0}^p \lambda_{kt} l_0 - \lambda\gamma \sum_{r=0}^{k-p-2} \left( \sum_{t=0}^p \lambda_{kt} \right) h_r - \sum_{t=0}^p \lambda_{kt} \gamma h_{k-t-1} - \gamma h_k. \end{aligned} \tag{29}$$

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

$$\begin{aligned} l_{k+1} &\leq l_0 - \lambda\gamma \sum_{r=0}^{k-p-2} h_r - \lambda\gamma \sum_{r=k-p-1}^{k-1} h_r - \gamma h_k \\ &= l_0 - \lambda\gamma \sum_{r=0}^{k-1} h_r - \gamma h_k \\ &\leq l_0 - \lambda\gamma \sum_{r=0}^k h_r. \end{aligned} \tag{30}$$

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

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

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

It follows that  $h_k \rightarrow 0 (k \rightarrow \infty).$  □

**Lemma 5** *Suppose that Assumptions hold. If Algorithm A does not termination finitely, then  $\lim_{k \rightarrow \infty} \|d_k\| = 0.$*

*Proof* Suppose by contradiction that there exist constants  $\epsilon > 0$  and  $\bar{k} > 0$  such that  $\|d_k\| > \epsilon$  for all  $k > \bar{k}$ .

Then by Lemma 2,  $\text{pred}_k^f > \frac{1}{3}\xi_1 \|\rho_k\| > \frac{1}{3}\xi_1 \|d_k\| > \frac{1}{3}\xi\epsilon > 0$ , because of  $\text{pred}_k^l \geq \eta \text{pred}_k^f$ , we have  $\max[l_k, \sum_{r=0}^{m(k)-1} \lambda_{kr} l_{k-r}] - l_{k+1} \geq \eta \text{pred}_k^f$ . We take the sum at the both sides, together with the sequence  $l_k$  is bounded below, we have  $\eta \sum \text{pred}_k^f < \infty$ , that follows  $\text{pred}_k^f \rightarrow 0$  as  $k \rightarrow \infty$ , which contradicts to  $\text{pred}_k^f > 0$ . Hence the conclusion follows.  $\square$

**Theorem 1** *Suppose  $\{x_k\}$  is an infinite sequence generated by Algorithm A. Then every cluster point of  $\{x_k\}$  is a KKT point of problem (P).*

### 5 Numerical results

In this section, we give some numerical experiments to show the success of our proposed method. All examples are chosen from [18] and [19].

(1) [2] 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  $y_k = \theta_k \hat{y}_k + (1 - \theta_k) H_k s_k$ ,

$$\theta_k = \begin{cases} 1, & s_k^T \hat{y}_k \geq 0.2 s_k^T H_k s_k, \\ \frac{0.8 s_k^T H_k s_k}{s_k^T H_k s_k - s_k^T \hat{y}_k}, & \text{otherwise} \end{cases} \tag{31}$$

and  $\hat{y}_k = g_{k+1} - g_k$ ,  $s_k = x_{k+1} - x_k$ .

(2) We assume the error tolerance is  $10^{-6}$ .

(3) The algorithm parameters were set as follows:  $H_0 = I \in R^{n \times n}$ ,  $\beta = 0.9$ ,  $\gamma = 0.1$ ,  $\rho = 0.5$ ,  $\alpha_1 = \alpha_2 = 0.5$ ,  $\sigma_0 = -0.1$ ,  $\Delta_{\min} = 10^{-6}$ ,  $\Delta_0 = 1$ . The program is written in Matlab.

In Table 1, the problems are numbered in the same way as in Hock and Schittkowski [18] and Schittkowski [19]. For example, ‘HS2’ is the problem 2 in Hock and Schittkowski [18] and ‘S216’ is the problem 216 in Schittkowski [19]. Some equality constrained problems are also included in our test problems, such as S216, S235, S252 and so on. NF, NG represent the number of function and gradient calculations respectively. In Table 1, the results in first column are calculated by Algorithm A, those in second column are calculated by traditional filter method, which are shown in [3], those in third column are calculated by Matlab function ‘fmincon’, compared the three methods, our algorithm has a smaller number of function calculations and gradient calculations.

To show the effect of the non-monotone method, we also list the numerical results in Table 2, these tests are done for  $M = 1$ ,  $M = 3$  and  $M = 10$  respectively, that means the degree of non-monotonicity is increasing.

First numerical results show that the nonmonotone algorithm is more effective than monotone one for most test examples and our algorithm is effective and satisfactory.

### 6 Conclusions

In our method, the criterion used to test the trial points is flexible, the refuse region is variable according to the different improvement made by the previous trial point, while in

**Table 1** The numerical results of different algorithm

	Algorithm A (NG-NF)	Filter (NG-NF)	Matlab (NF)
HS2	27-36	19-32	28
HS6	9-9	37-41	23
HS11	59-82	-	32
HS13	102-102	-	203
HS14	6-6	6-6	23
HS15	102-192	24-46	50
HS16	17-17	22-34	23
HS17	44-44	44-44	15
HS18	38-47	36-43	40
HS19	8-8	8-8	27
HS20	17-17	21-34	63
HS21	8-8	8-8	15
HS22	2-2	2-2	19
HS23	7-7	7-7	31
HS41	15-15	15-15	41
HS45	2-2	2-2	20
HS59	10-40	13-46	53
HS64	54-62	57-86	301
HS65	28-28	40-40	44
HS72	52-72	38-50	101
HS73	1-22	1-22	35
HS106	17-55	-	509
HS108	7-7	14-29	182
S216	4-13	3-13	21
S235	36-38	36-38	110
S252	18-34	58-58	139
S265	2-2	2-2	17
S269	9-9	14-31	48

**Table 2** The results of different  $M$  in our algorithm (*i.e.* using different degree of nonmonotone)

	$M = 1$ (NG-NF)	$M = 3$ (NG-NF)	$M = 10$ (NF)
HS2	28-40	27-36	-
HS6	26-36	9-9	9-9
HS11	29-58	59-82	39-98
HS13	102-102	102-102	102-102
HS14	6-6	6-6	6-6
HS15	18-34	102-192	27-60
HS16	63-70	17-17	17-17
HS17	44-44	44-44	44-44
HS18	30-37	38-47	-
HS19	23-23	8-8	8-8
HS20	85-92	17-17	17-17

the traditional filter methods, the elements in the filter structure are fixed. By the numerical results, we also find the new method has more effective results and less computational costs than not only the traditional methods but also the Matlab algorithms. Moreover, the use and adjustment of the self-adaptive parameter in our method is a good way to balance the value of objective function and the violation constraint function. On the other hand, the application of non-monotone in criterion avoids the Maratos effect to a certain degree, because more trial points are accepted by the filter according to the algorithm. We also compared the results of different nonmonotonic degree, although we can not decide which value of  $M$  is the best one, at least the results with nonmonotone is better than that with monotone technique.

**Competing interests**

The authors declare that they have no competing interests.

**Authors' contributions**

All authors contributed equally to the writing of this paper. All authors read and approved the final manuscript.

**Acknowledgements**

This research is supported by the National Natural Science Foundation of China (No. 11101115), the Natural Science Foundation of Hebei Province (No. 2014201033) and the Key Research Foundation of Educational Bureau of Hebei Province (No. ZD2015069). In addition, we would like to show our deepest gratitude to the editor and the anonymous reviewers who have helped to improve the paper.

Received: 1 June 2016 Accepted: 28 September 2016 Published online: 12 October 2016

**References**

1. Zhang J. A robust trust region method for nonlinear optimization with inequality constraint. *Appl Math Comput.* 2006;176:688-99.
2. Nocedal J, Wright S. *Numerical optimization.* New York: Springer; 1999.
3. Fletcher R, Leyffer S. Nonlinear programming without a penalty function. *Math Program.* 2002;91:239-69.
4. Chin C, Fletcher R. On the global convergence of an SLP-filter algorithm takes EQP steps. *Math Program.* 2003;96:161-77.
5. Fletcher R, Gould N, Leyffer S, Toint P, Wachter A. A global convergence of a trust region SQP-filter algorithm for general nonlinear programming. *SIAM J Optim.* 2002;13:635-60.
6. Fletcher R, Leyffer S, Toint P. On the global convergence of a filter-SQP algorithm. *SIAM J Optim.* 2002;13:44-59.
7. Ulbrich M, Ulbrich S, Vicente L. A global convergent primal-dual interior-point method for nonconvex nonlinear programming. *Math Program.* 2004;100:379-410.
8. Audet C, Dennis J. A pattern search filter method for nonlinear programming without derivatives. *SIAM J Optim.* 2004;14:980-1010.
9. Karas E, Riberio A, Sagastizabalc C, Solodov M. A bundle filter method for nonsmooth convex constrained optimization. *Math Program.* 2009;116:297-320.
10. Ulbrich M, Ulbrich S. Nonmonotone trust region methods for nonlinear equality constrained optimization without a penalty function. *Math Program.* 2003;95:103-35.
11. Ulbrich S. On the superlinear local convergence of a filter-SQP method. *Math Program.* 2004;100:217-45.
12. Nie P, Ma C. A trust region filter method for general nonlinear programming. *Appl Math Comput.* 2006;172:1000-17.
13. Su K, Pu D. A nonmonotone filter trust region method for nonlinear constrained optimization. *J Comput Appl Math.* 2009;223:230-9.
14. Chen Z. A penalty-free-type nonmonotone trust region method for nonlinear constrained optimization. *Appl Math Comput.* 2006;173:1014-46.
15. Chen Z, Zhang X. A nonmonotone trust region algorithm with nonmonotone penalty parameters for constrained optimization. *J Comput Appl Math.* 2004;172:7-39.
16. Gould N, Toint P. Global convergence of a non-monotone trust-region SQP-filter algorithm for nonlinear programming. *Nonconvex Optim Appl.* 2006;82:125-50.
17. Zhou G. A modified SQP method and its global convergence. *J Glob Optim.* 1997;11:193-205.
18. Hock W, Schittkowski K. *Test examples for nonlinear programming codes. Lecture notes in economics and mathematics system.* New York: Springer; 1981.
19. Schittkowski K. *More test examples for nonlinear mathematical programming codes.* New York: Springer; 1987.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---