Chin. Ann. Math.
35B(2), 2014, 209–224
DOI: 10.1007/s11401-014-0826-z

Chinese Annals of Mathematics, Series B
© The Editorial Office of CAM and
Springer-Verlag Berlin Heidelberg 2014

A Dwindling Filter Algorithm with a Modified Subproblem for Nonlinear Inequality Constrained Optimization*

Chao GU[1] Detong ZHU[2]

Abstract The authors propose a dwindling filter algorithm with Zhou's modified subproblem for nonlinear inequality constrained optimization. The feasibility restoration phase, which is always used in the traditional filter method, is not needed. Under mild conditions, global convergence and local superlinear convergence rates are obtained. Numerical results demonstrate that the new algorithm is effective.

Keywords Modified subproblem, Dwindling filter, Feasibility restoration phase, Convergence, Constrained optimization
2000 MR Subject Classification 49M37, 65K05, 90C30

1 Introduction

Consider the following nonlinear inequality constrained optimization:

$$\min \quad f(x), \tag{1.1a}$$

$$\text{s.t.} \quad g_j(x) \le 0, \quad j \in I, \tag{1.1b}$$

where $x \in \mathbb{R}^n$, $f\colon \mathbb{R}^n \to \mathbb{R}$ and $g_j\colon \mathbb{R}^n \to \mathbb{R}$ $(j = 1, \cdots, m)$ are continuously differentiable. The problem has become highly important in recent years (see [6, 9–10, 12, 14–15, 20–21]).

It is well-known that the sequential quadratic programming (or SQP for short) method is one of the most effective methods to solve (1.1). Because of its superlinear convergence rate, it is a topic of much active research. However, 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 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 that the sequence generated by the method is divergent. Based on the above reasons, Zhou [19] modified the quadratic subproblem to make it feasible and bounded.

Manuscript received October 4, 2012. Revised January 16, 2013.
[1]School of Mathematics and Information, Shanghai Lixin University of Commerce, Shanghai 201620, China. E-mail: chaogumath@gmail.com
[2]Department of Mathematics, Shanghai Normal University, Shanghai 200234, China.
E-mail: dtzhu@shnu.edu.cn
*This work was supported by the National Natural Science Foundation of China (Nos. 11201304, 11371253), the Innovation Program of Shanghai Municipal Education Commission (No. 12YZ174) and the Group of Accounting and Governance Disciplines (No. 10kq03).

The filter idea was first presented by Fletcher and Leyffer [3] for nonlinear programming (or NLP for short), offering an alternative to penalty functions, as a tool to guarantee global convergence of algorithms for nonlinear programming. Filter methods were successfully applied to solving various optimization problems, including complementarity and variational inequality problems (see [3–4, 7–8, 11, 13–14, 16–17]). Recently, Chen and Sun [1] proposed a dwindling multidimensional filter line search method for unconstrained optimization. The envelope of the dwindling filter becomes thinner and thinner as the step size approaches zero, which leads to more flexibility for the acceptance of the trial step.

In this paper, we propose a dwindling filter algorithm with Zhou's modified quadratic subproblem for nonlinear inequality constrained optimization. The algorithm has the following merits: It requires to solve only one QP subproblem with only a subset of the constraints which are estimated as active; the initial point is arbitrary; the subproblem is feasible at each iterate point; the feasibility restoration phase, which is always used in the traditional filter methods, is not needed. This paper can be outlined as follows. In Section 2, we state the new algorithm. The global convergence of the new algorithm is proved in Section 3. The local Q-superlinear convergence rate is established in Section 4. Some numerical results are given in Section 5.

2 Description of the Algorithm

Define functions $\Phi(x)$ and $\Psi(x)$ by

$$\Phi(x) = \max\{0, g_j(x) : j \in I\}, \tag{2.1}$$

$$\Psi(x) = \max\{g_j(x) : j \in I\}. \tag{2.2}$$

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

$$\Psi^*(x, d) = \max\{g_j(x) + \nabla g_j(x)^{\mathrm{T}} d : j \in I\}. \tag{2.3}$$

$\forall \sigma > 0$, functions $\Psi(x, \sigma)$ and $\Psi^0(x, \sigma)$: $\mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}$ are defined as follows:

$$\Psi(x, \sigma) = \min\{\Psi^*(x, d) : \|d\| \le \sigma\}, \tag{2.4}$$

$$\Psi^0(x, \sigma) = \max\{\Psi(x, \sigma), 0\}. \tag{2.5}$$

(2.4) equals the following linear programming:

$$\mathrm{LP}(x, \sigma) : \min\{z : g_j(x) + \nabla g_j(x)^{\mathrm{T}} d \le z, j \in I, \|d\| \le \sigma\}. \tag{2.6}$$

Denote

$$\theta(x, \sigma) = \Psi(x, \sigma) - \Psi(x), \tag{2.7}$$

$$\theta^0(x, \sigma) = \Psi(x, \sigma)^0 - \Psi(x), \tag{2.8}$$

$$F = \{x : g_j(x) \le 0 : j \in I\} = \{x : \Psi(x) \le 0\}, \tag{2.9}$$

$$F^{\mathrm{c}} = \{x : \Psi(x) > 0\}. \tag{2.10}$$

Definition 2.1 *Mangasarian-Fromotz constraint qualification* (*or MFCQ for short*) *is said to be satisfied by* $g(x) \le 0$ *at* x *if* $\exists\, z \in \mathbb{R}^n$, *such that*

$$\nabla g_j(x)^{\mathrm{T}} z < 0, \quad \forall j \in \{j \in I \mid g_j(x) > 0\}.$$

Lemma 2.1 *$\forall x \in F^{c}$, if MFCQ is satisfied at x, then $\theta(x,\sigma) < 0$, $\forall \sigma < 0$.*

Lemma 2.2 *$\Psi(x,\sigma)$, $\Psi^0(x,\sigma)$, $\theta(x,\sigma)$ and $\theta^0(x,\sigma)$ are all continuous on $\mathbb{R}^n \times R^+$.*

Lemma 2.3 *$\forall x \in F^{c}$, if $\theta(x,\sigma) < 0$, then $\theta^0(x,\sigma) < 0$.*

For the details of Lemmas 2.1–2.3, see [19]. Given $x \in \mathbb{R}^n$ and $\sigma > 0$, $D(x,\sigma)$ is defined by the following set:

$$D(x,\sigma) = \{d \mid g_j(x) + \nabla g_j(x)^{\mathrm{T}} d \le \Psi^0(x,\sigma), j \in I\}.$$

If d^* is the solution of $\mathrm{LP}(x,\sigma)$, then $d^* \in D(x,\sigma)$ and hence $D(x,\sigma)$ is nonempty. We obtain the direction d_k from the following convex programming problem $Q(x_k, H_k, \sigma_k)$:

$$\min \Big\{ \nabla f(x_k)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} H_k d \Big\}, \tag{2.11a}$$

$$\text{s.t.} \quad g_j(x_k) + \nabla g_j(x_k)^{\mathrm{T}} d \le \Psi^0(x_k,\sigma_k), \quad j \in L_k, \tag{2.11b}$$

where L_k is the set of approximate active indices of the point x_k. Clearly, by the above statement, the convex programming is feasible.

Let us measure the inequality constraint violation at x by

$$h(x) = \|g(x)^+\|, \tag{2.12}$$

where $g_j(x)^+ = \max\{0, g_j(x)\}$, $j \in I$. The basic idea of the filter method is to interpret the optimization problem as a bi-objective optimization problem with the two goals of minimizing the objective function $f(x)$ and the constraint violation $h(x)$. In the traditional filter method, a trial point $x_k(\alpha_{k,l}) = x_k + \alpha_{k,l} d_k$ is called acceptance to the filter if and only if

$$h(x_k(\alpha_{k,l})) \le (1-\gamma_h) h(x_j) \quad \text{or} \quad f(x_k(\alpha_{k,l})) \le f(x_j) - \gamma_f h(x_j)$$

for all $(h(x_j), f(x_j)) \in \mathcal{F}$. Different from the above idea, we call that a trial point $x_k(\alpha_{k,l})$ is acceptable to the dwindling filter if and only if

$$h(x_k(\alpha_{k,l})) \le h(x_j) - \phi(\alpha_{k,l})\gamma_h h(x_j) \quad \text{or} \quad f(x_k(\alpha_{k,l})) \le f(x_j) - \phi(\alpha_{k,l})\gamma_f h(x_j) \tag{2.13}$$

for all $(h(x_j), f(x_j)) \in \mathcal{F}$, where $\phi(\alpha)$ is a dwindling function defined by Chen and Sun [1].

Definition 2.2 *$\phi(\alpha) : [0,1] \to \mathbb{R}$ is a dwindling function if it is a monotonically increasing and continuous function such that*

$$\phi(\alpha) = 0 \Leftrightarrow \alpha = 0, \tag{2.14}$$

$$\alpha = 1 \Leftrightarrow \phi(\alpha) = 1, \tag{2.15}$$

$$\lim_{\alpha \to 0} \frac{\phi(\alpha)}{\alpha} = 0. \tag{2.16}$$

For example, $\phi(\alpha) = \alpha^{\frac{3}{2}}$ satisfies (2.14)–(2.16). A decreasing sequence of step sizes $\alpha_{k,l} \in (0,1]$ $(l = 0,1,2,\cdots)$ is tried until (2.13) is satisfied, in which the envelope of the dwindling filter becomes thinner and thinner as the step size approaches zero. If $\phi(\alpha) = 1$, the dwindling filter is reduced to the traditional filter. Later, if $h(x_k) > 0$, the filter is augmented for a new iteration using the update formula

$$\mathcal{F}_{k+1} = \mathcal{F}_k \cup \{(h, f) \in \mathbb{R}^2 : h \geq h(x_k) - \phi(\alpha)\gamma_h h(x_k),\ f \geq f(x_k) - \phi(\alpha)\gamma_f h(x_k)\}. \tag{2.17}$$

It is easy to see that

$$\min\{h : (h, f) \in \mathcal{F}_k\} > 0 \tag{2.18}$$

for all k.

Algorithm 2.1 Given starting point x_0, Σ is a compact set which consists of symmetric positive definite matrices, $H_0 \in \Sigma$, $\mathcal{F}_0 = \{(h, f) \in \mathbb{R}^2\colon h \geq \widetilde{h} - \phi(\alpha)\gamma_h \widetilde{h}\}$, $\widetilde{h} > h(x_0)$, γ_f, $\gamma_h \in (0, 1)$, $\eta_f \in (0, \frac{1}{2})$, $0 < \tau_1 \leq \tau_2 < 1$, $\epsilon_0 > 0$, $\epsilon > 0$.

Step 1 Compute $f(x_k)$, $g(x_k)$, $h(x_k)$, $\nabla f(x_k)$, $\nabla g(x_k)$.

Step 2 Compute an active constraint set L_k.

(1) Let $i = 0$ and $\epsilon_{k,i} = \varepsilon_0$.

(2) Set

$$\begin{aligned} L_{k,i} &= \{j \in I \mid -\epsilon_{k,i} \leq g_j(x_k) - \Phi(x_k) \leq 0\}, \\ A_{k,i} &= (\nabla g_j(x_k), j \in L_k). \end{aligned}$$

If $\det(A_{k,i}^{\mathrm{T}} A_{k,i}) \geq \epsilon_{k,i}$, let $L_k = L_{k,i}$, $A_k = A_{k,i}$, and go to Step 3.

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

Step 3 Compute d_k from the convex programming problem and set $\widetilde{x}_{k+1} = x_k + d_k$. If $\|d_k\| + h(x_k) \leq \epsilon$, stop.

Case 1 $-\nabla f(x_k)^{\mathrm{T}} d_k > h(x_k)$ holds: If

$$f(\widetilde{x}_{k+1}) \leq f(x_k) + \eta_f \nabla f(x_k)^{\mathrm{T}} d_k \tag{2.19}$$

holds, set $x_{k+1} = \widetilde{x}_{k+1}$ and go to Step 5.

Case 2 $-\nabla f(x_k)^{\mathrm{T}} d_k > h(x_k)$ is not satisfied: If

$$h(\widetilde{x}_{k+1}) \leq (1 - \gamma_h) h(x_k) \tag{2.20a}$$

or

$$f(\widetilde{x}_{k+1}) \leq f(x_k) - \gamma_f h(x_k) \tag{2.20b}$$

holds, set $x_{k+1} = \widetilde{x}_{k+1}$ and go to Step 5.

Step 4 Computation of 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

$$\begin{aligned} \rho_k &= -\nabla f(x_k)^{\mathrm{T}} d_k, \qquad \pi_k = -(A_k^1)^{-1} \nabla f_1(x_k), \\ \widetilde{d}_k &= \frac{-\rho_k((A_k^1)^{-1})^{\mathrm{T}} e}{1 + 2|e^{\mathrm{T}} \pi_k|}, \quad q_k = \rho_k(d_k + \overline{d}_k), \end{aligned}$$

where $\overline{d}_k = \binom{\tilde{d}_k}{0}$, $e = (1, 1, \cdots, 1)^{\mathrm{T}} \in R^{|L_k|}$.

(1) Set $\alpha_{k,0} = 1$ and $l \leftarrow 0$.

(2) Compute $x_k(\alpha_{k,l}) = x_k + \alpha_{k,l} q_k$. If $(h(x_k(\alpha_{k,l})), f(x_k(\alpha_{k,l}))) \in \mathcal{F}_k$, go to Step 4(3).

Case 1 $-\alpha_{k,l} \nabla f(x_k)^{\mathrm{T}} q_k > h(x_k)$ holds: If

$$f(x_k(\alpha_{k,l})) \le f(x_k) + \alpha_{k,l} \eta_f \nabla f(x_k)^{\mathrm{T}} q_k \tag{2.21}$$

holds, set $x_{k+1} = x_k(\alpha_{k,l})$ and go to Step 5.

Case 2 $-\alpha_{k,l} \nabla f(x_k)^{\mathrm{T}} q_k > h(x_k)$ is not satisfied: If

$$h(x_k(\alpha_{k,l})) \le h(x_k) - \phi(\alpha_{k,l}) \gamma_h h(x_k) \tag{2.22a}$$

or

$$f(x_k(\alpha_{k,l})) \le f(x_k) - \phi(\alpha_{k,l}) \gamma_f h(x_k) \tag{2.22c}$$
(2.22b)

holds, set $x_{k+1} = x_k(\alpha_{k,l})$ and go to Step 5.

(3) Choose $\alpha_{k,l+1} \in [\tau_1 \alpha_{k,l}, \tau_2 \alpha_{k,l}]$, set $l \leftarrow l + 1$, and go back to Step 4(2) (inner loop B).

Step 5 If either $-\nabla f(x_k)^{\mathrm{T}} d_k > h(x_k)$ or $-\alpha_{k,l} \nabla f(x_k)^{\mathrm{T}} q_k > h(x_k)$ is not satisfied, augment the filter. Choose $H_{k+1} \in \Sigma$, $\sigma_{k+1} \in [\sigma_l, \sigma_r]$, $k \leftarrow k + 1$, and go back to Step 1.

3 Global Convergence

Assumption 3.1 (1) The functions $f(x)$ and $g_j(x)$ $(j \in \{0, 1, 2, \cdots, m\})$ are twice continuously differentiable and bounded on $\mathbb{R}^n$.

(2) The iterate $\{x_k\}$ remains in a compacted subset $\mathbb{S} \subset \mathbb{R}^n$.

(3) There exist two constants a and b such that $a\|p\|^2 \le p^{\mathrm{T}} B_k p \le b\|p\|^2$ for all k, where $p \in \mathbb{R}^n$, $0 < a \le b$.

(4) The Mangasarian-Fromovitz constraint qualification (MFCQ) holds.

Lemma 3.1 (see [19]) *Suppose that $x_k \in \mathbb{R}^n$, $H_k \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix. If MFCQ is satisfied at x_k, then 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) $g_j(x_k) + \nabla g_j(x_k)^{\mathrm{T}} d_k \le \Psi^0(x_k, \sigma_k),\ j \in L_k$;

(b) $u_j^k \ge 0,\ j \in L_k$;

(c) $\nabla f(x_k) + H_k d_k + A_k U_k = 0,\ A_k = (\nabla g_j(x_k),\ j \in L_k)$;

(d) $u_j^k (g_j(x_k) + \nabla g_j(x_k)^{\mathrm{T}} d_k) = 0,\ j \in L_k$.

Lemma 3.2 (see [15]) *For any iterate k, the index i defined in Step 2 in Algorithm 2.1 is finite, which means that the inner loop A terminates in a finite number of times.*

Lemma 3.3 (see [15]) *If $d_k \ne 0$, then it holds that*

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

Lemma 3.4 *Suppose that Assumption* 3.1 *holds. Then trial point* $x_k(\alpha_{k,l})$ *could not be rejected by* x_k *if* $\alpha_{k,l}$ *is sufficiently small.*

Proof Suppose that $\widetilde{x}_{k+1} = x_k + d_k$ is rejected by x_k. From Lemma 3.3, we get $\nabla f(x_k)^{\mathrm{T}} q_k < 0$. The second-order Taylor expansion of $f(x)$ implies that

$$f(x_k(\alpha_{k,l})) - f(x_k) + \phi(\alpha_{k,l})\gamma_f h(x_k) = \alpha_{k,l}\nabla f(x_k)^{\mathrm{T}} q_k + \phi(\alpha_{k,l})\gamma_f h(x_k) + O(\alpha_{k,l}^2\|q_k\|^2).$$

Since $\phi(\alpha_{k,l}) = o(\alpha_{k,l})$ as $l \to \infty$, $f(x_k(\alpha_{k,l})) \leq f(x_k) - \phi(\alpha_{k,l})\gamma_f h(x_k)$ holds for sufficiently small $\alpha_{k,l}$.

Remark 3.1 Without any assumption, we prove that the trial point could not be rejected by the current iterate.

Lemma 3.5 *The mechanisms of the filter ensure that for all* k*,*

$$(h(x_k), f(x_k)) \notin \mathcal{F}_k. \tag{3.1}$$

Proof The proof is done by induction. Note that

$$\mathcal{F}_0 = \{(h, f) \in \mathbb{R}^2 : h \geq \widetilde{h} - \phi(\alpha)\gamma_h \widetilde{h}\},$$

where $\widetilde{h} > h(x_0)$. Since $\phi(\alpha) = o(\alpha)$ as $l \to \infty$, $h(x_0) \leq \widetilde{h} - \phi(\alpha)\gamma_h\widetilde{h}$ holds for sufficiently small α. The claim is valid for $k = 0$.

Suppose that the claim is true for k. If $(h(\widetilde{x}_{k+1}), f(\widetilde{x}_{k+1})) \notin \mathcal{F}_{k+1}$ with $\widetilde{x}_{k+1} = x_k + d_k$ holds, the claim is true. Otherwise, we consider the trial point $x(\alpha_{k,l}) = x_k + \alpha_{k,l} q_k$. It is evident that $(h(x_{k+1}), f(x_{k+1})) \notin \mathcal{F}_k$. In the following we need to prove that $(h(x_{k+1}), f(x_{k+1})) \notin \mathcal{F}_{k+1}$. There are two cases.

Case 1 $h(x_k) = 0$.

We have $\nabla f(x_k)^{\mathrm{T}} q_k < 0$ from Lemma 3.3 and $-\alpha_{k,l}\nabla f(x_k)^{\mathrm{T}} q_k > h(x_k)$. So (2.21) must be satisfied, i.e., $\mathcal{F}_{k+1} = \mathcal{F}_k$ and $(h(x_{k+1}), f(x_{k+1})) \notin \mathcal{F}_{k+1}$.

Case 2 $h(x_k) > 0$.

If $-\alpha_{k,l}\nabla f(x_k)^{\mathrm{T}} q_k > h(x_k)$ holds, the proof is similar to Case 1. Otherwise, consider that the filter is augmented in iteration k, i.e.,

$$\mathcal{F}_{k+1} = \mathcal{F}_k \cup \{(h, f) \in \mathbb{R}^2 : h \geq h(x_k) - \phi(\alpha)\gamma_h h(x_k),\ f \geq f(x_k) - \phi(\alpha)\gamma_f h(x_k)\}.$$

By $(h(x_{k+1}), f(x_{k+1})) \notin \mathcal{F}_k$ and

$$f(x_{k+1}) \leq f(x_k) - \phi(\alpha_k)\gamma_f h(x_k) \quad \text{(by the proof of Lemma 3.4)},$$

we obtain $(h(x_{k+1}), f(x_{k+1})) \notin \mathcal{F}_{k+1}$.

Lemma 3.6 *Suppose that Assumption* 3.1 *holds. Then the inner loop B terminates in a finite number of iterations.*

Proof The proof is done by contradiction. Suppose that the inner loop does not terminate in a finite number of iterations. In this case, the algorithm will always reject the trial points $\widetilde{x}$ and $x(\alpha_{k,l})$, which leads to $\alpha_{k,l} \to 0$. Therefore, we may consider two cases.

If $-\alpha_{k,l}\nabla f(x_k)^{\mathrm{T}}q_k > h(x_k)$ holds, the trial point is required to satisfy (2.21). Since q_k is a descent direction, there exists a constant $c_1 > 0$ such that, for $\alpha_{k,l} \le c_1$, (2.21) must be satisfied. If, on the other hand, $-\alpha_{k,l}\nabla f(x_k)^{\mathrm{T}}q_k > h(x_k)$ does not hold, the trial point is required to satisfy (2.22). From Lemma 3.4, there exists a constant $c_2 > 0$ such that, for $\alpha_{k,l} \le c_2$, (2.22) must be satisfied.

From Lemma 3.5, we have $(h(x_k), f(x_k)) \notin \mathcal{F}_k$, i.e.,

$$h(x_k) \le h_j - \phi(\alpha)\gamma_h h_j \quad \text{or} \quad f(x_k) \le f_j - \phi(\alpha)\gamma_f h_j \quad \text{for all } (\theta_j, f_j) \in \mathcal{F}.$$

Suppose that $x(\alpha_{k,l}) = x_k + \alpha_{k,l}q_k$ is rejected by the filter. Hence

$$h(x_k(\alpha_{k,l})) > h_j - \phi(\alpha)\gamma_h h_j \tag{3.2a}$$

or

$$f(x_k(\alpha_{k,l})) > f_j - \phi(\alpha)\gamma_f h_j \tag{3.2b}$$

for all $(h_j, f_j) \in \mathcal{F}$. If $h(x_k) \le h_j - \phi(\alpha)\gamma_h h_j$ holds, from Lemma 3.3 and $\alpha_{k,l} \to 0$, we obtain that there exists a constant $c_3 > 0$, for $\alpha_{k,l} \le c_3$,

$$h(x_k(\alpha_{k,l})) = \max\{0, g_j(x_k + \alpha_{k,l}q_k)\} \le \max\{0, g_j(x_k)\} = h(x_k) \le h_j - \phi(\alpha)\gamma_h h_j,$$

which contradicts (3.2a). If $f(x_k) \le f_j - \phi(\alpha)\gamma_f h_j$ holds, from Lemma 3.3 and $\alpha_{k,l} \to 0$, we get that there exists a constant $c_4 > 0$, such that for $\alpha_{k,l} \le c_4$,

$$f(x_k(\alpha_{k,l})) = f(x_k) + \alpha_{k,l}\nabla f(x_k)^{\mathrm{T}}q_k + O(\alpha_{k,l}^2\|q_k\|^2) \le f(x_k) \le f_j - \phi(\alpha)\gamma_f h_j,$$

which contradicts (3.2b). Choose $\alpha_{k,l} \le \min\{c_1, c_2, c_3, c_4\}$, and the inner loop B terminates in a finite number of iterations.

Lemma 3.7 *Suppose that infinite points are added to the filter. Then there exists a subsequence $\mathcal{G}$ in which the filter has been augmented such that*

$$\lim_{k\to\infty, k\in\mathcal{G}} h(x_k) = 0.$$

Proof The proof is done by contradiction. Suppose that there exists an infinite subsequence $\{k_i\}$ of $\mathcal{G}$ for which

$$h(x_{k_i}) \ge \epsilon \tag{3.3}$$

for some $\epsilon > 0$. At each iteration k_i, $(h(x_{k_i}), f(x_{k_i}))$ is added to the filter, which means that no other (h, f) can be added to the filter at a later stage within the square

$$[h(x_{k_i}) - \gamma_h\phi(\alpha_{k_i})\epsilon, h(x_{k_i})] \times [f(x_{k_i}) - \gamma_f\phi(\alpha_{k_i})\epsilon, f(x_{k_i})].$$

Now observe that the area of the each of these squares is at least $\gamma_h\gamma_f\phi(\alpha_{k_i})^2\epsilon^2$. As a consequence, if there exists an infinite subsequence $\{k_{i_j}\} \subseteq \{k_i\}$ such that $\phi(\alpha_{k_{i_j}}) \ge \epsilon$ as $j \to \infty$, the set $[0, h_{\max}] \times [f_{\min}, \infty] \cap \{(h, f) \mid f \le \kappa_f\}$ is completely covered by at most a finite number of such squares. This is in contradiction to the infinite subsequence $\{k_{i_j}\}$. If $\phi(\alpha_{k_i}) \to 0$ as $i \to \infty$, we get

$$\alpha_{k_i} \to 0. \tag{3.4}$$

Since $\lim\limits_{i\to\infty} \alpha_{k_i} = 0$, we have $\lim\limits_{i\to\infty} \alpha_{k_i,l_i} = 0$. Lemma 3.6 implies that there exists a constant c_5 such that for $\alpha_{k_i,l_i} \le c_5$, α_{k_i,l_i} is accepted by Algorithm 2.1, which contradicts (3.4).

Lemma 3.8 *Suppose that finite points are added to the filter. Then*

$$\lim_{k\to\infty} h(x_k) = 0.$$

Proof From Assumption 3.1, there exists a $K \in \mathbb{N}$ so that for all iterations $k \geq K$, there is no point adding to the filter. If $\widetilde{x}_{k+1} = x_k + d_k$ is accepted by Algorithm 2.1, we then have that for all $k \geq K$,

$$f(\widetilde{x}_{k+1}) \leq f(x_k) + \eta_f \nabla f(x_k)^{\mathrm{T}} d_k \leq f(x_k) - \eta_f h(x_k). \tag{3.5}$$

Suppose that $\widetilde{x}_{k+1} = x_k + d_k$ is rejected by Algorithm 2.1. In this case, the trial point $x_{k+1} = x_k + \alpha_k q_k$ is accepted. Since $h(x_k) < \alpha_k(-\nabla f(x_k)^{\mathrm{T}} q_k)$, we have

$$f(x_{k+1}) \leq f(x_k) + \alpha_k \eta_f \nabla f(x_k)^{\mathrm{T}} q_k \leq f(x_k) - \eta_f h(x_k),$$

which is the same as (3.5). We conclude that

$$f(x_{K+i}) - f(x_K) \leq -\eta_f \sum_{k=K}^{K+i-1} h(x_k).$$

Since $\{f_{K+i}\}$ is bounded below as $i \to \infty$, we have $\lim\limits_{k\to\infty} h(x_k) = 0$.

Lemma 3.9 *Suppose that Assumptions* 3.1 *holds. Then*

$$\lim_{k\to\infty} h(x_k) = 0.$$

Proof The proof is similar to that of Theorem 1 in [16].

Theorem 3.1 *Suppose that Assumptions* 3.1 *holds. Then*

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

Proof Suppose that the claim is not true, i.e., there exists a subsequence infinite index set $\mathcal{K}$ and a constant $\epsilon > 0$ so that $\|d_k\| \geq \epsilon$ for all $k \in \mathcal{K}$. It follows from Lemmas 3.1 and 3.3 that

$$\begin{aligned} h(x_k) + \nabla f(x_k)^{\mathrm{T}} d_k &= h(x_k) - d_k^{\mathrm{T}} H_k d_k - d_k^{\mathrm{T}} A_k U_k \\ &= h(x_k) - d_k^{\mathrm{T}} H_k d_k + g(x_k)^{\mathrm{T}} U_k \\ &\leq -d_k^{\mathrm{T}} H_k d_k + (1 - c_6) h(x_k) \\ &\leq -a\|d_k\|^2 + (1 - c_6) h(x_k) \\ &\leq -a\epsilon^2 + (1 - c_6) h(x_k). \end{aligned} \tag{3.6}$$

If $1 - c_6 \leq 0$, $h(x_k) < -\nabla f(x_k)^{\mathrm{T}} d_k$ holds. If, on the other hand, $1 - c_6 > 0$, $h(x_k) < -\nabla f(x_k)^{\mathrm{T}} d_k$

also holds with $h(x_k) < \frac{a\epsilon^2}{1-c_6}$ since $h(x_k) \to 0$ as $k \to \infty$. Similarly, we have

$$\begin{aligned} h(x_k) + \nabla f(x_k)^{\mathrm{T}} q_k &\le h(x_k) - \frac{1}{2}\rho_k^2 \\ &= h(x_k) - \frac{1}{2}(-d_k^{\mathrm{T}} H_k d_k - d_k^{\mathrm{T}} A_k U_k)^2 \\ &= h(x_k) - \frac{1}{2}[(d_k^{\mathrm{T}} H_k d_k)^2 + 2 d_k^{\mathrm{T}} H_k d_k d_k^{\mathrm{T}} A_k U_k + (d_k^{\mathrm{T}} A_k U_k)^2] \\ &= -\frac{1}{2}(d_k^{\mathrm{T}} H_k d_k)^2 - d_k^{\mathrm{T}} H_k d_k d_k^{\mathrm{T}} A_k U_k - \frac{1}{2}(d_k^{\mathrm{T}} A_k U_k)^2 + h(x_k) \\ &\le -\frac{1}{2} a^2 \|d_k\|^4 + d_k^{\mathrm{T}} H_k d_k g(x_k)^{\mathrm{T}} U_k - \frac{1}{2}(g(x_k)^{\mathrm{T}} U_k)^2 + h(x_k) \\ &\le -\frac{1}{2} a^2 \epsilon^4 + (b\epsilon^2 c_6 + 1) h(x_k). \end{aligned} \tag{3.7}$$

When $h(x_k) < \frac{a^2\epsilon^4}{2(b\epsilon^2 c_6+1)}$, $h(x_k) < -\nabla f(x_k)^{\mathrm{T}} q_k \le -\alpha \nabla f(x_k)^{\mathrm{T}} q_k$ is satisfied. Define

$$\zeta = \begin{cases} \dfrac{a^2\epsilon^4}{2(b\epsilon^2 c_6 + 1)}, & \text{if } 1 - c_6 \le 0, \\ \min\left\{\dfrac{a\epsilon^2}{1-c_6}, \dfrac{a^2\epsilon^4}{2(b\epsilon^2 c_6 + 1)}\right\}, & \text{otherwise.} \end{cases} \tag{3.8}$$

Choose K_1 such that for all $k \ge K_1$, $h(x_k) \le \zeta$, and then the trail point must be accepted by (2.19) or (2.21). We denote the two sets of indices of those iterations in which (2.19) holds by $\mathcal{K}_1 \subset \mathcal{K}$ and in which (2.21) holds by $\mathcal{K}_2 \subset \mathcal{K}$. There are three cases.

Case 1 If $\mathcal{K}_1$ is infinite and $\mathcal{K}_2$ is finite, there exists a K_2, such that for all iterations $k \ge K_2$ and $k \in \mathcal{K}_1$, (2.19) holds, i.e.,

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\ge -\eta_f \nabla f(x_k)^{\mathrm{T}} d_k \\ &= \eta_f (d_k^{\mathrm{T}} H_k d_k - d_k^{\mathrm{T}} A_k U_k) \\ &\ge \eta_f a \|d_k\|^2 - \eta_f c_6 h(x_k). \end{aligned} \tag{3.9}$$

From the proof of Lemma 3.8, one can conclude that

$$\lim_{k\to\infty} \eta_f a \|d_k\|^2 - \eta_f c_6 h(x_k) = 0.$$

$\lim\limits_{k\to\infty} h(x_k) = 0$ implies that $\lim\limits_{\substack{k\to\infty \\ k\in\mathcal{K}_1}} \|d_k\| = 0$, which is a contradiction.

Case 2 If $\mathcal{K}_2$ is infinite and $\mathcal{K}_1$ is finite, there exists a K_3, such that for all iterations $k \ge K_3$ and $k \in \mathcal{K}_2$,

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\ge -\alpha_k \eta_f \nabla f(x_k)^{\mathrm{T}} q_k \\ &\ge \frac{1}{2}\eta_f \alpha_k \rho_k^2 \\ &\ge \frac{1}{2} a^2 \epsilon^4 \alpha_k - b\epsilon^2 c_6 \alpha_k h(x_k). \end{aligned} \tag{3.10}$$

From the proof of Lemma 3.8 and $\lim\limits_{k\to\infty} h(x_k) = 0$, we have $\lim\limits_{\substack{k\to\infty \\ k\in\mathcal{K}_2}} \alpha_k = 0$, which contradicts the proof of Lemma 3.6.

Case 3 If $\mathcal{K}_1$ is infinite and $\mathcal{K}_2$ is infinite, we have

$$\begin{aligned}\infty &> \sum_k f(x_k) - f(x_{k+1}) \geq \sum_{k\in\mathcal{K}} f(x_k) - f(x_{k+1}) \\ &\geq \sum_{k\in\mathcal{K}_1} -\eta_f \nabla f(x_k)^{\mathrm{T}} d_k + \sum_{k\in\mathcal{K}_2} -\alpha_k\eta_f \nabla f(x_k)^{\mathrm{T}} q_k \\ &\geq \sum_{k\in\mathcal{K}_1} \eta_f a\epsilon^2 - \eta_f c_6 h(x_k) + \sum_{k\in\mathcal{K}_2} \frac{1}{2}a^2\epsilon^4\alpha_k - b\epsilon^2 c_6\alpha_k h(x_k) \\ &= \infty,\end{aligned}$$

which is a contradiction.

Remark 3.2 The result of Theorem 3.1 is stronger than that in [16]. The reason is that the feasibility restoration phase, which is always used in the traditional filter method, is not needed.

4 Local Convergence

In order to analyze the local convergence rate of the proposed algorithm more assumptions are needed.

Assumption 4.1 (1) x_* is a KKT point of (1.1). Strict complementarity slackness and linear independence of the gradients of the active constraints hold.

(2) The second-order sufficient condition holds at x_*, i.e., there exists a constant $\widetilde{\tau} > 0$ such that

$$p^{\mathrm{T}}\nabla^2_{xx}L(x_*,\lambda_*)p \geq \widetilde{\tau}\|p\|^2 \quad \text{with} \quad \nabla g_j(x_*)^{\mathrm{T}}p = 0, j \in I^*,$$

where $I^* = \{j \mid g_j(x^*) = 0, j \in I\}$.

(3) $x_k \to x_*$.

(4)

$$\lim_{k\to\infty} \frac{\|[H_k - \nabla^2\mathcal{L}(x_*,\lambda_*)]d_k\|}{\|d_k\|} = 0. \tag{4.1}$$

Algorithm 4.1 Given starting point x_0, Σ is a compact set which consists of symmetric positive definite matrices, $H_0 \in \Sigma$, $\mathcal{F}_0 = \{(h,f) \in \mathbb{R}^2 : h \geq \widetilde{h} - \phi(\alpha)\gamma_h\widetilde{h}\}$, $\widetilde{h} > h(x_0)$, γ_f, $\gamma_h \in (0,1)$, $\tau \in (2,3)$, $\eta_f \in (0,\frac{1}{2})$, $0 < \tau_1 \leq \tau_2 < 1$, $\epsilon_0 > 0$, $\epsilon > 0$.

Step 1 Compute $f(x_k)$, $g(x_k)$, $h(x_k)$, $\nabla f(x_k)$, $\nabla g(x_k)$.

Step 2 Compute an active constraint set L_k.

(1) Let $i = 0$ and $\epsilon_{k,i} = \varepsilon_0$.

(2) Set

$$\begin{aligned}L_{k,i} &= \{j \in I \mid -\epsilon_{k,i} \leq g_j(x_k) - \Phi(x_k) \leq 0\}, \\ A_{k,i} &= (\nabla g_j(x_k), j \in L_k).\end{aligned}$$

If $\det(A_{k,i}^{\mathrm{T}}A_{k,i}) \geq \epsilon_{k,i}$, let $L_k = L_{k,i}$, $A_k = A_{k,i}$, and go to Step 3.

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

Step 3 Compute d_k from the convex programming problem and set $\widetilde{x}_{k+1} = x_k + d_k$. If $\|d_k\| + h(x_k) \leq \epsilon$, stop.

Step 4 Let $P_kA_k = \begin{pmatrix} A_k^1 \\ A_k^2 \end{pmatrix}$, where P_k is a permutation matrix and A_k^1 is invertible. Solve linear equations: $(A_k^1)^{\mathrm{T}}\widehat{d}_k = -\|d_k\|^{\tau}e + F(x_k + d_k)$, where $F(x_k + d_k) = (g_j(x_k + d_k), j \in L_k)$. Set $d_k^{\mathrm{soc}} = P_k^{\mathrm{T}}\begin{pmatrix} \widehat{d}_k \\ 0 \end{pmatrix}$ and $\overline{x}_k = x_k + d_k + d_k^{\mathrm{soc}}$.

Case 1 $-\nabla f(x_k)^{\mathrm{T}}d_k > h(x_k)$ holds. If

$$f(\overline{x}_{k+1}) \leq f(x_k) + \eta_f \nabla f(x_k)^{\mathrm{T}}d_k \tag{4.2}$$

holds, set $x_{k+1} = \overline{x}_{k+1}$ and go to Step 5.

Case 2 $-\nabla f(x_k)^{\mathrm{T}}d_k > h(x_k)$ is not satisfied. If

$$h(\overline{x}_{k+1}) \leq (1 - \gamma_h)h(x_k) \tag{4.3a}$$

or

$$f(\overline{x}_{k+1}) \leq f(x_k) - \gamma_f h(x_k) \tag{4.3b}$$

holds, set $x_{k+1} = \overline{x}_{k+1}$ and go to Step 5.

Step 5 Computation of 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

$$\begin{aligned} \rho_k &= -\nabla f(x_k)^{\mathrm{T}}d_k, \qquad & \pi_k &= -(A_k^1)^{-1}\nabla f_1(x_k), \\ \widetilde{d}_k &= \frac{-\rho_k((A_k^1)^{-1})^{\mathrm{T}}e}{1 + 2|e^{\mathrm{T}}\pi_k|}, & q_k &= \rho_k(d_k + \overline{d}_k), \end{aligned}$$

where $\overline{d}_k = \begin{pmatrix} \widetilde{d}_k \\ 0 \end{pmatrix}$, $e = (1, 1, \cdots, 1)^{\mathrm{T}} \in \mathbb{R}^{|L_k|}$.

(1) Set $\alpha_{k,0} = 1$ and $l \leftarrow 0$.

(2) Compute $x_k(\alpha_{k,l}) = x_k + \alpha_{k,l}q_k$. If $(h(x_k(\alpha_{k,l})), f(x_k(\alpha_{k,l}))) \in \mathcal{F}_k$, go to Step 5(3).

Case 1 $-\alpha_{k,l}\nabla f(x_k)^{\mathrm{T}}q_k > h(x_k)$ holds. If

$$f(x_k(\alpha_{k,l})) \leq f(x_k) + \alpha_{k,l}\eta_f \nabla f(x_k)^{\mathrm{T}}q_k \tag{4.4}$$

holds, set $x_{k+1} = x_k(\alpha_{k,l})$ and go to Step 6.

Case 2 $-\alpha_{k,l}\nabla f(x_k)^{\mathrm{T}}q_k > h(x_k)$ is not satisfied. If

$$h(x_k(\alpha_{k,l})) \leq h(x_k) - \phi(\alpha_{k,l})\gamma_h h(x_k) \tag{4.5a}$$

or

$$f(x_k(\alpha_{k,l})) \leq f(x_k) - \phi(\alpha_{k,l})\gamma_f h(x_k) \tag{4.5b}$$

holds, set $x_{k+1} = x_k(\alpha_{k,l})$ and go to Step 6.

(3) Choose $\alpha_{k,l+1} \in [\tau_1\alpha_{k,l}, \tau_2\alpha_{k,l}]$, set $l \leftarrow l + 1$, and go back to Step 5(2) (inner loop B).

Step 6 If either $-\nabla f(x_k)^{\mathrm{T}}d_k > h(x_k)$ or $-\alpha_{k,l}\nabla f(x_k)^{\mathrm{T}}q_k > h(x_k)$ is not satisfied, augment the filter. Choose $H_{k+1} \in \Sigma$, $\sigma_{k+1} \in [\sigma_l, \sigma_r]$, $k \leftarrow k + 1$, and go back to Step 1.

Theorem 3.1 shows that $\|d_k\| \to 0$ as $k \to \infty$. So, it is natural that $\|d_k\|$ satisfies $\|d_k\| \le \sigma_k$ for sufficiently large k. (2.3)–(2.5) imply 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

$$\min \left\{ \nabla f(x_k)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} H_k d \right\}, \tag{4.6a}$$

$$\text{s.t.} \quad g_j(x_k) + \nabla g_j(x_k)^{\mathrm{T}} d \le 0, \quad j \in L_k. \tag{4.6b}$$

Lemma 4.1 (see [15]) *It holds, for $k \to \infty$, that*

$$L_k \equiv I(x^*) = I^*, \quad \|d_k\| \to 0, \quad \lambda_k \to (\lambda_j^*, j \in I^*), \quad U_k \to \lambda^*,$$

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

Lemma 4.2 *Suppose that Assumption* 4.1 *holds. Then the full step $x_{k+1} = x_k + d_k$ or $x_{k+1} = x_k + d_k + d_k^{\mathrm{soc}}$ is acceptable to the current filter $\mathcal{F}_k$ for sufficiently large k.*

Proof Suppose that the full step $x_{k+1} = x_k + d_k$ is rejected by the dwindling filter. According to (4.6) and the invertibility of A_k^1,

$$\|d_k^{\mathrm{soc}}\| = \left\| P_k^{\mathrm{T}} \begin{pmatrix} \widehat{d}_k \\ 0 \end{pmatrix} \right\| \le \delta_1 \|\widehat{d}_k\| = \delta_1 (A_k^1)^{-T} (-\|d_k\|^\tau e + F(x_k + d_k)) = O(\|d_k\|^2).$$

For $j \notin I^*$ and sufficiently large k, there exists a constant $c_7 > 0$ such that $g_j(x_k) \le -c_7$. Using Taylor's theorem, we can write

$$\begin{aligned} g_j(x_k + d_k + d_k^{\mathrm{soc}}) &= g_j(x_k) + \nabla g_j(x_k)^{\mathrm{T}} (d_k + d_k^{\mathrm{soc}}) + \frac{1}{2} (d_k + d_k^{\mathrm{soc}})^{\mathrm{T}} \nabla^2 g_j(\vartheta_k)(d_k + d_k^{\mathrm{soc}}) \\ &\le -c_7 + O(\|d_k\|), \end{aligned}$$

where ϑ_k is between x_k and $x_k + d_k + d_k^{\mathrm{soc}}$. Since $\|d_k\| \to 0$ as $k \to \infty$, the right-hand side term is negative. As for $j \in I^*$, using Taylor's theorem and the definition of d_k^{soc},

$$\begin{aligned} g_j(x_k + d_k + d_k^{\mathrm{soc}}) &= g_j(x_k + d_k) + \nabla g_j(x_k)^{\mathrm{T}} d_k^{\mathrm{soc}} + \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 g_j(\varrho_k) d_k^{\mathrm{soc}} \\ &\quad + \frac{1}{2} (d_k^{\mathrm{soc}})^{\mathrm{T}} \nabla^2 g_j(\xi_k) d_k^{\mathrm{soc}} \\ &= g_j(x_k + d_k) + \nabla g_j(x_k)^{\mathrm{T}} P_k^{\mathrm{T}} \begin{pmatrix} \widehat{d}_k \\ 0 \end{pmatrix} + + O(\|d_k\|^3) \\ &= g_j(x_k + d_k) + [(A_k^1)^{\mathrm{T}} \widehat{d}_k]_j + O(\|d_k\|^3) \\ &= -\|d_k\|^\tau + O(\|d_k\|^3), \end{aligned}$$

where $\tau \in (2, 3)$, ϱ_k is between x_k and $x_k + d_k$ and ξ_k is between $x_k + d_k$ and $x_k + d_k + d_k^{\mathrm{soc}}$. From the two cases, we obtain that $h(x_k + d_k + d_k^{\mathrm{soc}}) = \|g_j(x_k + d_k + d_k^{\mathrm{soc}})^+\| = \max\{0, g_j(x_k + d_k + d_k^{\mathrm{soc}})\} = 0$. From the update formula of the dwindling filter, we have

$$\min\{h : (h, f) \in \mathcal{F}_k\} > 0 \tag{4.7}$$

for all k. Therefore, $x_{k+1} = x_k + d_k + d_k^{\mathrm{soc}}$ is acceptable to the current filter $\mathcal{F}_k$ for sufficiently large k.

Theorem 4.1 *Suppose that Assumption* 4.1 *holds. Then the full step* $x_{k+1} = x_k + d_k$ *or* $x_{k+1} = x_k + d_k + d_k^{\mathrm{soc}}$ *is taken by the algorithm for sufficiently large* k. *Further, the sequence* $\{x_k\}$ *converges to* x_* *Q-superlinearly.*

Proof Suppose that the point $x_{k+1} = x_k + d_k$ is not accepted by the filter $\mathcal{F}_k$. In that case, consider $x_{k+1} = x_k + d_k + d_k^{\mathrm{soc}}$. From Lemma 4.2, we have that $x_{k+1} = x_k + d_k + d_k^{\mathrm{soc}}$ is acceptable to the current filter $\mathcal{F}_k$ for sufficiently large k. It is similar to the proof of Theorem 3.1 that $-\nabla f(x_k)^{\mathrm{T}} d_k > [h(x_k)]^{\nu}, \nu \in (0, \frac{1}{2})$ holds for sufficiently large k. Then we have

$$h(x_k) \leq [-\nabla f(x_k)^{\mathrm{T}} d_k]^{\frac{1}{\nu}} = o(\|d_k\|^2).$$

Combining with Lemma 3.1 and $d_k^{\mathrm{soc}} = O(\|d_k\|^2)$, we obtain that

$$\begin{aligned}
\nabla f(x_k)^{\mathrm{T}} d_k^{\mathrm{soc}} &= -U_k^{\mathrm{T}} A_k^{\mathrm{T}} d_k^{\mathrm{soc}} - d_k^{\mathrm{T}} H_k d_k^{\mathrm{soc}} \\
&= U_k^{\mathrm{T}} \|d_k\|^{\tau} e + F(x_k + d_k) + o(\|d_k\|^2) \\
&= \frac{1}{2} d_k^{\mathrm{T}} \Big(\sum_{j \in I^*} u_{k,i} \nabla^2 g_j(x_k) \Big) d_k + o(\|d_k\|^2).
\end{aligned} \tag{4.8}$$

By (4.8) and Taylor's expansion, we have

$$\begin{aligned}
&\quad f(x_k + d_k + d_k^{\mathrm{soc}}) - f(x_k) - \eta_f \nabla f(x_k)^{\mathrm{T}} d_k \\
&= \nabla f(x_k)^{\mathrm{T}} (d_k + d_k^{\mathrm{soc}}) + \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 f(x_k) d_k + o(\|d_k\|^2) - \eta_f \nabla f(x_k)^{\mathrm{T}} d_k \\
&= (1 - \eta_f) \nabla f(x_k)^{\mathrm{T}} d_k + \nabla f(x_k)^{\mathrm{T}} d_k^{\mathrm{soc}} + \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 f(x_k) d_k + o(\|d_k\|^2) \\
&= (1 - \eta_f)(-d_k^{\mathrm{T}} H_k d_k - d_k^{\mathrm{T}} A_k U_k) + \frac{1}{2} d_k^{\mathrm{T}} \Big(\sum_{j \in I^*} u_{k,i} \nabla^2 g_j(x_k) \Big) d_k \\
&\quad + \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 f(x_k) d_k + o(\|d_k\|^2) \\
&\leq (\eta_f - 1) d_k^{\mathrm{T}} H_k d_k + \frac{1}{2} d_k^{\mathrm{T}} \nabla^2 \mathcal{L}(x_k, U_k) d_k + (1 - c_6) h(x_k) + o(\|d_k\|^2) \\
&= \Big(\eta_f - \frac{1}{2} \Big) d_k^{\mathrm{T}} H_k d_k + \frac{1}{2} d_k^{\mathrm{T}} (\nabla^2 \mathcal{L}(x_k, U_k) - H_k) d_k + o(\|d_k\|^2) \\
&\leq a \Big(\eta_f - \frac{1}{2} \Big) \|d_k\|^2 + o(\|d_k\|^2).
\end{aligned}$$

Since $\|d_k\| \to 0$ as $k \to \infty$, $\eta_f < \frac{1}{2}$ and $a > 0$, we get

$$f(x_k + d_k + d_k^{\mathrm{soc}}) \leq f(x_k) + \eta_f \nabla f(x_k)^{\mathrm{T}} d_k$$

for sufficient large k. By Assumption 4.1(4), the sequence $\{x_k\}$ converges to x_* Q-superlinearly.

5 Numerical Experiments

In this section, we present the numerical results of Algorithm 2.1 on an HP i5 personal computer with 2G memory. The selected parameter values are: $\gamma_f = 0.5$, $\gamma_h = 0.5$, $\tau = 2.5$, $\eta_f = 0.5$, $\tau_1 = \tau_2 = 0.5$, $\epsilon_0 = 10^{-6}$, $\epsilon = 10^{-6}$, $\sigma_l = 1$, $\sigma_r = 1.5$. The computation terminates when the stopping criterion $\|d_k\| + h(x_k) \leq \epsilon$ is satisfied.

Table 1 Numerical results.

	Algorithm 2.1		IPOPT	
Problem	Nt	Nf	Nt	Nf
HS01	55	79	25	33
HS02	13	31	12	14
HS03	10	10	5	6
HS04	2	2	6	7
HS05	7	9	8	9
HS11	8	8	9	10
HS12	9	9	9	10
HS13	33	33	32	39
HS15	6	14	17	22
HS16	6	8	11	12
HS17	12	12	18	19
HS18	10	10	15	19
HS20	17	39	14	15
HS21	4	4	8	9
HS22	6	7	6	7
HS23	6	6	10	12
HS24	5	5	12	14
HS29	10	12	9	10
HS30	20	20	8	12
HS31	8	13	8	9
HS33	5	5	13	16
HS35	7	8	7	8
HS36	2	2	13	14
HS37	10	13	12	13
HS43	12	16	9	10
HS45	8	8	7	8
HS57	22	23	21	23
HS59	18	18	34	54
HS65	8	9	15	16
HS66	8	8	7	8
HS70	40	40	30	46
HS76	7	7	7	8
HS83	4	4	17	18
HS84	3	3	15	16
HS95	2	2	22	32
HS96	2	2	18	19
HS97	7	7	17	18
HS98	7	7	22	23
HS100	17	34	11	22
HS104	19	19	8	9

All the forty nonlinear inequality constrained problems are numbered in the same way as in [5]. Nt and Nf stand for the numbers of iterations and function evaluations, respectively. IPOPT is an interior-point filter line-search algorithm for nonlinear optimization (see [18]). To compare the performance of Algorithm 2.1 and IPOPT, we use the performance profiles as described in [2]. Our profiles are based on function evaluations and the numbers of iterations. The benchmark results are generated by running a solver on a set $\mathcal{P}$ of problems and recording information of interest such as the number of function evaluations. Let $\mathcal{S}$ be the set of solvers in comparison. For each problem p and solver s, we define

$$t_{p,s} = \text{the number of function evaluations required to solve problem } p \text{ by solver } s.$$

Accordingly, set $t_{p,s}$ to be the number of gradient evaluations or iterations. The comparison is based on the performance ratio defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in \mathcal{S}\}}.$$

If we define

$$\rho_s(\tau) = \frac{1}{n_p}\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\},$$

then $\rho_s(\tau)$ is the probability for the solver $s \in \mathcal{S}$. The value of $\rho_s(1)$ is the probability that the solver will win over the rest of the solvers. From Figures 1–2, it is clear that Algorithm 2.1 wins over IPOPT on the given problems.

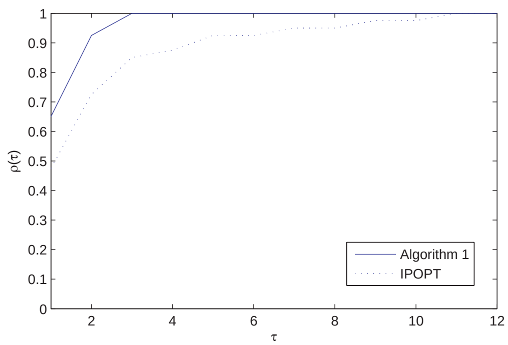

Figure 1 Performance profile on the numbers of iterations.

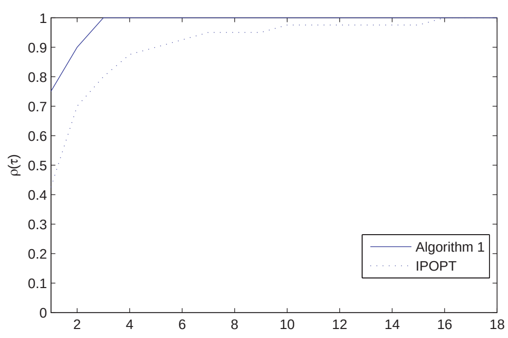

Figure 2 Performance profile on function evaluations.

6 Conclusion

In this paper, we propose a dwindling filter algorithm with Zhou's modified subproblem for nonlinear inequality constrained optimization, which requires less computational costs compared with the traditional filter algorithm (see [18]) on the given problems. The feasibility restoration phase is not used in the algorithm. Under mild conditions, global convergence and local superlinear convergence rates are obtained. Moreover, the global convergence result of Theorem 3.1 is stronger than that in [16]. How to choose the dwindling parameters $\phi(\alpha)$ to get better numerical experience deserves further research.

Acknowledgement The authors would like to thank the two anonymous referees for their comments and suggestions that helped to improve the presentation of this paper.

References

[1] Chen, Y. and Sun, W., A dwindling filter line search method for unconstrained optimization, Technical Report of Optimization No. 2010-09-01, School of Mathematical Science, Nanjing Normal University, Nanjing.

[2] Dolan, E. D. and Moré, J. J., Benchmarking optimization software with performance profiles, *Math. Program.*, **91**, 2002, 201–213.

[3] Fletcher, R. and Leyffer, S., Nonlinear programming without a penalty function, *Math. Program.*, **91**, 2002, 239–269.

[4] Gu, C. and Zhu, D., A secant algorithm with line search filter method for nonlinear optimization, *Appl. Math. Modell.*, **35**, 2011, 879–894.

[5] Hock, W. and Schittkowski, K., Test examples for nonlinear programming codes, Lecture Notes in Economics and Mathematics System, Vol. 187, Springer-Verlag, New York, 1981.

[6] Jian, J. B. and Liu, Y., A superlinearly convergent method of quasi-strongly sub-feasible directions with active set identifying for constrained optimization, *Nonlinear Anal. Real World Appl.*, **12**, 2011, 2717–2729.

[7] Long, J., Ma, C. F. and Nie, P. Y., A new filter method for solving nonlinear complementarity problems, *Appl. Math. Comput.*, **185**, 2007, 705–718.

[8] Li, C. and Sun, W., On filter-successive linearization methods for nonlinear semidefinite programming, *Sci. China Ser. A*, **52**, 2009, 2341–2361.

[9] Nie, P. Y., Sequential penalty quadratic programming filter methods for nonlinear programming, *Nonlinear Anal. Real World Appl.* **8**, 2007, 118–129.

[10] Nie, P. Y., A derivative-free method for the system of nonlinear equations, *Nonlinear Anal. Real World Appl.*, **7**, 2006, 378–384.

[11] Nie, P. Y. and Ma, C. F., A trust region filter mehtod for general nonlinear programming, *Appl. Math. Comput.*, **172**, 2006, 1000-1017.

[12] Nocedal, J. and Wright, S., Numerical Optimization, Springer-Verlag, New York, 1999.

[13] Shen, C. G., Xue, W. J. and Pu, D. G., A filter SQP algorithm without a feasibility restoration phase, *Comp. Appl. Math.*, **28**, 2009, 167–194.

[14] Su, K. and Yu, Z., A modified SQP method with nonmonotone technique and its global convergence, *Comput. Math. Appl.*, **57**, 2009, 240–247.

[15] Su, K., A globally and superlinearly convergent modified SQP-filter method, *J. Global Optim.*, **41**, 2008, 203–217.

[16] Wächter, A. and Biegler, L. T., Line search filter methods for nonlinear programming: Motivation and global convergence, *SIAM J. Optim.*, **6**, 2005, 1–31.

[17] Wächter, A. and Biegler, L. T., Line search filter methods for nonlinear programming: Local convergence, *SIAM J. Optim.*, **6**, 2005, 32–48.

[18] Wächter, A. and Biegler, L. T., On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Math. Program.*, **106**, 2006, 25–57.

[19] Zhou, G. L., A modified SQP method and its global convergence, *J. Global Optim.*, **11**, 1997, 193–205.

[20] Zhu, Z. B. and Wang, S., A superlinearly convergent numerical algorithm for nonlinear programming, *Nonlinear Anal. Real World Appl.*, **13**, 2012, 2391–2402.

[21] Zhu, Z. B. and Jian, J. B., An efficient feasible SQP algorithm for inequality constrained optimization, *Nonlinear Anal. Real World Appl.*, **10**, 2009, 1220–1228.