A penalty-free algorithm for solving nonlinear equality constrained optimization

Yonggang Pei[*]
Henan Engineering Laboratory for Big Data Statistical Analysis and Optimal Control
College of Mathematics and Information Science
Henan Normal University
Xinxiang 453007
China
peiyonggang@htu.edu.cn

Lanting Dong
College of Mathematics and Information Science
Henan Normal University
Xinxiang 453007
China
1789190577@qq.com

Weiyue Kong
College of Mathematics and Information Science
Henan Normal University
Xinxiang 453007
China
465668259@qq.com

Abstract. In contrast to standard penalty type methods for nonlinear optimization, penalty-free type methods do not have to determine penalty parameters and have promising numerical results. So they have attracted the attention of many researchers. Filter method is a classical penalty-free method. In this paper, a different filter technique is introduced for solving nonlinear equality constrained optimization. The optimality condition of the nonlinear optimization problem is regarded as a new filter pair which is built in the backtracking line search framework. Then, the trial step size is accepted if one of the two measures in filter is improved after the search direction is determined. Under some reasonable assumptions, the global convergence of the algorithm is proved. Some preliminary numerical results are presented to show the usefulness of the proposed algorithm.

Keywords: nonlinear programming, equality constrained optimization, line search, filter method, global convergence.

1. Introduction

Nonlinear constrained optimization is one of the most fundamental models in scientific calculation because of the wide use of optimization in science, engineer-

[*]. Corresponding author

ing, economics, industry and so on [20, 17]. Furthermore, nonlinear optimization is a basic building block of more complex design and optimization paradigms. Many impressive methods have been established to tackle nonlinear constrained optimization [20, 5, 16].

Line search and trust region are two common methods for constrained programming. In these frameworks, it has to balance the twin goals of reducing the objective function and satisfying the constraints. Merit functions are typical tools for achieving this balance. However, the penalty parameters in merit functions are not easy to be determined for good performance. Penalty-free methods use different techniques to handle the balance problem. Penalty-free methods include filter methods and filter-free methods.

Fletcher and Leyffer [7] originally proposed filter methods which can be embedded in line search or trust region framework and replace the traditional merit function to ensure global convergence for nonlinear programming. Wächter and Biegler [30, 29] gave a filter line search method and proved its global convergence and local convergence. Leyffer and Vanaret [17] presented an augmented Lagrangian filter method. Nonmonotone filter SQP methods were proposed in [11,25]. As for primal-dual interior-point algorithms with a filter line search method, their effectiveness has been proved in [31, 15]. Chen and Sun [3] give a dwindling filter line search method for unconstrained optimization. Some dwindling filter line search methods were also introduced in [1, 13] for constrained optimization. Gu and Zhu [12] proposed a line search filter method with reduced Hessian for nonlinear programming. Zhu and Pu [35] presented a line search filter algorithm with inexact step computations for equality constrained optimization. A line search filter secant method was proposed in [33]. Based on trust-region framework, Fletcher et al. [6] and Wächter et al. [32] propose trust-region SQP-filter algorithms for nonlinear programming. Pei and Zhu [21] propose some algorithms combining line search filter technique. Li and Zhu [18] propose and analyzed two interior trust-region methods with line search filter technique. Recently, Sun et al. [26] introduced a trust region sequential quadratic programming approach for nonlinear systems based on nonlinear model predictive control.

There are also many researchers who proposed filter-free methods. For example, Ulbrich [27] proposed a non-monotone trust region methods for nonlinear equality constrained optimization without a penalty function. Bielschowsky [2] presented a type of trust-region algorithm for solving nonlinear programming problems with equality constraints. Xue, Shen and Pu [34] proposed a penalty-function-free line search SQP method for nonlinear programming. Liu and Yuan [19] presented a sequential quadratic programming method without a penalty function or a filter for nonlinear equality constrained optimization. Qiu and Chen [23] presented a class of trust region algorithms without using a penalty function or a filter for nonlinear inequality constrained optimization and analyze their global and local convergence. Chen and Dai constructed a new filter-free method in [4].

In this paper, we introduce a line search filter method which aims to obtain the stationary point of nonlinear equality constrained optimization. The search direction is computed based on the null space method in each iteration. Then, the step size is determined by the filter mechanism and the backtracking line search. The proof of global convergence is given. The numerical results show that our algorithm has achieved good results for some problems.

The most important difference from other filter methods is that the composition of the filter. Most of filter methods employ the objective function (or the Lagrange function) and the violation of the constraints as the filter pair, while the first-order optimality conditions of the nonlinear programming are used as the filter pair in this paper. So, it is not required to compute the value of objective function (or the Lagrange function) in every iteration to obtain a first order optimal point, while the objective function (or the Lagrange function) has to be evaluated in other filter methods. This is the main feature of the proposed filter method.

The outline of this paper is as follows. In section 2, we state the line search filter approach. The global convergence of the algorithm is proved in section 3. Finally, we report some numerical experiments in section 4.

The notation of this paper is as follows. Norms $\|\cdot\|$ denote a fixed vector norm and its compatible matrix norm unless otherwise noted. For brevity, we use the convention $(x, y) = (x^T, y^T)^T$ for vectors x, y. Given two vectors $v, w \in \mathbb{R}^n$, we define the convex segment $[v, w] := \{v + t(w - v) : t \in [0, 1]\}$. Finally, we denote by $O(t_k)$ a sequence $\{v_k\}$ satisfying $\|v_k\| \leq \beta t_k$ for some constant $\beta > 0$ independent of k, and by $o(t_k)$ a sequence $\{v_k\}$ satisfying $\|v_k\| \leq \beta_k t_k$ for some positive sequence $\{\beta_k\}$ with $\lim_k \beta_k = 0$.

2. A line search filter approach.

Consider the following nonlinear programming (NLP)

$$\text{(1a)} \qquad \underset{x}{\text{minimize}} \quad f(x)$$

$$\text{(1b)} \qquad \text{subject to} \quad c(x) = 0,$$

where the objective function $f(x): \ \mathbb{R}^n \to \mathbb{R}$ and the equality constraints $c(x): \ \mathbb{R}^n \to \mathbb{R}^m$ with $m < n$ are sufficiently smooth.

The Karush-Kuhn-Tucker (KKT) conditions for the NLP (1) are

$$\text{(2a)} \qquad g(x) - A(x)y = 0,$$

$$\text{(2b)} \qquad c(x) = 0,$$

where we denote with $A(x) := \nabla c(x)$ the transpose of the Jacobian of the constraints c, and with $g(x) := \nabla f(x)$ the gradient of the objective function. The vector y corresponds to the Lagrange multipliers for the equality constraints (1b). Under certain constraint qualifications, such as linear independence of the constraint gradients, the KKT conditions are the first order optimality conditions for NLP (1).

Given an initial estimate x_0, the proposed algorithm of this paper generates a sequence of improved estimates x_k of the solution for the NLP(1). In every iteration k, the next iterate point can be determined if the search direction p_k and step size α_k have worked out, i.e, $x_{k+1} := x_k + \alpha_k p_k$, where $\alpha_k \in (0, 1]$.

As for the search direction, we have following system (see [20]):

$$\begin{bmatrix} H_k & -A_k \\ A_k^T & 0 \end{bmatrix} \begin{pmatrix} p_k \\ y_{k+1} \end{pmatrix} = - \begin{pmatrix} g_k \\ c_k \end{pmatrix}. \tag{3}$$

Here, $A_k := A(x_k)$, $g_k := g(x_k)$, and $c_k := c(x_k)$. The symmetric matrix H_k denotes the Hessian $\nabla^2_{xx} L(x_k, y_{k+1})$ of the Lagrangian

$$L(x, y) := f(x) - y^T c(x)$$

of the NLP (1), or an approximation to this Hessian. Here, we require that

$$\left(\nabla^2_{xx} L(x_k, y_{k+1}) - H_k\right) d = O(\|d\|^2). \tag{4}$$

The vector y_{k+1} is some estimate of the optimal multipliers corresponding to the equality constrains (1b), and its updating formula will be given later (see (6)).

Assume that the projection of the Hessian approximation H_k onto the null space of A_k^T is uniformly positive definite. In other words, there exists a constant λ_H so that

$$\lambda_{\min}(N_k^T H_k N_k) > \lambda_H,$$

where $\lambda_{\min}(N_k^T H_k N_k)$ is the smallest eigenvalue of matrix $N_k^T H_k N_k$ and the columns of matrices $N_k \in \mathbb{R}^{n\times(n-m)}$ form an orthonormal basis matrix of the null space of A_k^T.

Using the methods in [20], p_k can be decomposed into

$$p_k = Y_k p_Y + N_k p_N, \tag{5}$$

where columns of $Y_k \in \mathbb{R}^{n\times m}$ span the range space of A_k.

Substituting (5) into (3), the following system can be obtained to solve p_Y and p_N:

$$(A_k^T Y_k) p_Y = -c_k,$$
$$(N_k^T H_k N_k) p_N = -N_k^T H_k Y_k p_Y - N_k^T g_k.$$

We can choose $Y_k = A_k$, which is valid for Y_k when A_k^T has full row rank. Then a multiplier updating formula is obtained as follows:

$$y_{k+1} = [A_k^T A_k]^{-1} A_k^T (H_k p_k + g_k). \tag{6}$$

Along this direction p_k, the largest step size should be found to satisfy descent condition. Finally, the sequence $\{x_k\}$ of iterates converge to a stationary point of the NLP(1). In this paper we consider a backtracking line search procedure to compute the largest step size. A decreasing sequence of step sizes $\alpha_{k,l} \in (0,1] (l = 0, 1, 2, \ldots)$ is tried until some acceptance criterion is satisfied. In the remainder of this section we describe how the filter acceptance criterion is applied to the line search framework.

According to the first-order optimality condition, the filter pair is defined as $(\theta(x), \omega(x,y))$, where

$$\theta(x) = \|c(x)\|,$$

$$\omega(x,y) = \frac{1}{2}\|\nabla_x L(x,y)\|^2 = \frac{1}{2}\|g(x) - A(x)y\|^2.$$

The underlying idea is to interpret the NLP(1) as a biobjective optimization problem with two goals: minimizing the constraint violation $\theta(x)$ and minimizing the first order criticality measure $\omega(x,y)$ until they tend to zero. In the filter mechanism, the constraint violation measure is more important, because the optimal solution of NLP problem must be feasible. For both measurements, the trail point $x_k(\alpha_{k,l})$ can be accepted if it reduces either of the filter pair, i.e., if $\theta(x_k(\alpha_{k,l})) < \theta(x_k)$ or $\omega(x_k(\alpha_{k,l}), y_{k+1}) < \omega(x_k, y_{k+1})$. For convenience, we denote $\omega(x_k) = \omega(x_k, y_{k+1})$. Note that this criterion is not very demanding and therefore generally allows for a larger step size. However, this simple concept is not enough to ensure global convergence. Based on previous experience (see [6] and [8]), we have following descent criterion.

A trial step size $\alpha_{k,l}$ provides sufficient reduction with respect to the current iterate x_k, if

(7a) $$\theta(x_k(\alpha_{k,l})) \le (1 - \gamma_\theta)\theta(x_k)$$

or

(7b) $$\omega(x_k(\alpha_{k,l})) \le \omega(x_k) - \gamma_w \theta(x_k)$$

holds for fixed constants $\gamma_\theta, \gamma_\omega \in (0,1)$. In a practical implementation, the constants $\gamma_\theta, \gamma_\omega$ typically are chosen to be small. Sometimes, this descent criterion may cause the algorithm to converge to a feasible but not an optimal point. In order to prevent this, the following switching conditions should be given:

(8) $$m_k(\alpha_{k,l}) < 0 \quad \text{and} \quad [-m_k(\alpha_{k,l})]^\tau [\alpha_{k,l}]^{1-\tau} > \delta[\theta(x_k)]^\phi$$

with fixed constants $\delta > 0, \phi > 1, \tau \ge 1$, where

(9) $$m_k(\alpha) \overset{\text{def}}{=} \alpha(g_k - A_k y_{k+1})^T H_k p_k.$$

If the condition (8) holds, the step p_k is a descent direction. In order to make the current step size $\alpha_{k,l}$ acceptable, we require that $\alpha_{k,l}$ satisfies the Armijo-type condition

(10) $$\omega(x_k(\alpha_{k,l})) \le \omega(x_k) + \eta_\omega m_k(\alpha_{k,l}).$$

Here, $\eta_\omega \in \left(0, \frac{1}{2}\right)$ is a fixed constant. If this Armijo-type condition (10) is not met, aiming to be accepted, we should decrease the step size and find the next trial point. But if the switching condition (8) does not hold, we can take the method revert to the acceptance criterion (7).

As we have known, the switching condition ensures that for the optimality enforced by the Armijo condition (10) is sufficiently large compared to the current constraint violation. Therefore, it prevents a trial point from making only a small progress when it is far away from the feasible region.

For two measures θ and ω, a trial point may improve one and worsen the other one. It happens between two such trial points. For preventing this cycle (like methods in [7, 30]), a filter is defined as a set $\mathcal{F}_k \subseteq [0, \infty) \times \mathbb{R}$ containing all $(\theta, \omega)-$ pairs that are prohibited in iteration k. We say that a trial point $x_k(\alpha_{k,l})$ is acceptable to the current filter $\mathcal{F}_k$ if

$$(\theta(x_k(\alpha_{k,l})), \omega(x_k(\alpha_{k,l}))) \notin \mathcal{F}_k. \tag{11}$$

During the algorithm we require that the current iterate x_k is always acceptable to the current filter $\mathcal{F}_k$. After some new iterates x_{k+1} have been accepted, the current filter can be augmented by

$$\mathcal{F}_{k+1} := \mathcal{F}_k \cup \left\{(\theta, \omega) \in \mathbb{R}^2 : \theta \geq (1-\gamma_\theta)\theta(x_k) \quad \text{and} \quad \omega \geq \omega(x_k) - \gamma_\omega \theta(x_k)\right\}. \tag{12}$$

If the filter is not augmented, it remains unchanged, i.e., $\mathcal{F}_{k+1} := \mathcal{F}_k$. Then $\mathcal{F}_k \subseteq \mathcal{F}_{k+1}$ holds for all k. In algorithm, if a trial point $x_k(\alpha_{k,l})$ does not satisfy the switching condition (8) but satisfies acceptance criterion (7), we need use (12) to augment filter. If a trial point makes the switching condition (8) hold and then makes the Armijo-type condition (10) hold, the filter remains unchanged. If the filter has been augmented in iteration k, x_k ensures to provide sufficient reduction for one of the measures.

The computed trial step $\alpha_{k,l}$ may be smaller than

$$\alpha_k^{\min} := \gamma_\alpha \begin{cases} \min\left\{\gamma_\theta, \frac{\gamma_\omega \theta(x_k)}{-(g_k - A_k y_{k+1})^T H_k p_k}, \frac{\delta[\theta(x_k)]^\phi}{[-(g_k - A_k y_{k+1})^T H_k p_k]^\tau}\right\}, & \text{if } (g_k - A_k y_{k+1})^T H_k p_k < 0, \\ \gamma_\theta, & \text{otherwise.} \end{cases} \tag{13}$$

where $\gamma_\alpha \in (0, 1]$. Then there is no admissible step size can be found. So the algorithm switches to a feasibility restoration phase, whose purpose is to find a new iterate x_{k+1} that satisfies (7) and is also acceptable to the current filter by trying to decrease the constraint violation. Our approach is to compute a new iterate x_{k+1} by decreasing the infeasibility measure θ such that x_{k+1} satisfies the sufficient decrease conditions (7) and is acceptable to the filter, i.e., $(\theta(x_{k+1}), \omega(x_{k+1})) \notin \mathcal{F}_k$. This process can be called the feasibility restoration phase. In (13), γ_α is a safety factor. It is useful to avoid invoking the feasibility restoration phase unnecessarily in a practical implementation.

We are now ready to state the overall algorithm for solving the NLP (1) formally.

```
**Algorithm I**

  **Given**

      Set constants $\theta_{\max} \in (\theta(x_0), \infty]; \gamma_\theta, \gamma_\omega \in (0,1); \delta > 0; \gamma_\alpha \in (0,1]; \phi > 1; \tau \geq 1; \eta_\omega \in \left(0, \frac{1}{2}\right); 0 < \tau_1 \leq \tau_2 < 1$.

  **Initialize**

      Initialize the starting point $x_0$, the initial filter $\mathcal{F}_0:=\{(\theta,\omega)\in\mathbb{R}^2 : \theta \geq \theta_{\max}\}$, the multiplier $y_0$ and the iteration counter $k=0$.

  **Loop** until $x_k$ satisfies the KKT conditions (2) for some $y_{k+1} \in \mathbb{R}^m$

      Compute the search direction $p_k$ from (5);
      Compute multiplier $y_{k+1}$ with (6).

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

      **Loop** until find $x_k(\alpha_{k,l})$ is accepted by current filter $\mathcal{F}_k$
          **if** $\alpha_{k,l} < \alpha_k^{\min}$ with $\alpha_k^{\min}$ defined by (13), go to the feasibility restoration phase;
          **else** compute the new trial point $x_k(\alpha_{k,l}) = x_k + \alpha_{k,l} p_k$;
              **if** $x_k(\alpha_{k,l})$ is acceptable to the current filter $\mathcal{F}_k$, **break**.
              **else** choose $\alpha_{k,l+1} \in [\tau_1\alpha_{k,l}, \tau_2\alpha_{k,l}]$, set $l = l+1$.
              **end**
          **end**
      **end**
      **If** (8) holds,
          **if** (10) holds, set $\alpha_k := \alpha_{k,l}$ and $x_{k+1} := x_k(\alpha_k)$, $\mathcal{F}_{k+1} = \mathcal{F}_k$;
          **else** choose $\alpha_{k,l+1} \in [\tau_1\alpha_{k,l}, \tau_2\alpha_{k,l}]$, set $l = l+1$ and go to the inner "Loop";
          **end**
      **else**
          **if** (7) holds, set $\alpha_k := \alpha_{k,l}$, $x_{k+1} := x_k(\alpha_k)$, and augment the filter using (12);
          **else** choose $\alpha_{k,l+1} \in [\tau_1\alpha_{k,l}, \tau_2\alpha_{k,l}]$, set $l = l+1$ and go to the inner "Loop".
          **end**
```

```
        end if
        Set $k = k + 1$;

    End

    Feasibility restoration phase

        Compute a new iterate $x_{k+1}$. Augment the filter using (12) (for $x_k$) and
        increase the iteration counter $k = k + 1$. Go to the outer "Loop".
```

Lemma 2.1. *Algorithm I is well defined.*

Proof. In the inner "Loop", it is clear that $\lim_l \alpha_{k,l} = 0$. If $\theta(x_k) > 0$, it can be seen from (13) that $\alpha_k^{\min} > 0$. So, the algorithm either accepts a new iterate or switches to the feasibility restoration phase. On the other hand, if $\theta(x_k) = 0$ and the algorithm does not stop at a KKT point, then $(g_k - A_k y_{k+1})^T H_k p_k < 0$ (see, Lemma 3.3 below). In that case, $\alpha_k^{\min} = 0$, and the Armijo condition (10) is satisfied, i.e., a new iterate is accepted. □

3. Global convergence

In the remainder of this paper we denote some index sets.

$\mathcal{Z} \subseteq \mathbb{N}$: the filter is augmented in the corresponding iterations, i.e.,

$$\mathcal{F}_k \subsetneqq \mathcal{F}_{k+1} \quad \Leftrightarrow \quad k \in \mathcal{Z}.$$

$\mathcal{R} \subseteq \mathcal{Z}$: the feasibility restoration phase is invoked in the corresponding iterations.

As is common for most line search methods, some reasonable assumptions are stated down below, which is necessary for the global convergence analysis of Algorithm I.

Assumptions G

Let $\{x_k\}$ be the sequence generated by Algorithm I. Assume that the feasibility restoration phase always terminates successfully.

(G1) There exists an open set $\mathcal{C} \subseteq \mathbb{R}^n$ with $[x_k, x_k + p_k] \subseteq \mathcal{C}$, for all k such that f and c are twice continuously differentiable over $\mathcal{C}$.

(G2) The matrices H_k approximating the Hessian of the Lagrangian in (3) are uniformly bounded for all k, that is, there exists a constant $M_H > 0$ so that

$$\|H_k\| \leq M_H$$

for all k.

(G3) There exists a constant $\lambda_H > 0$ such that

$$\lambda_{\min}(N_k^T H_k N_k) \geq \lambda_H, \tag{14}$$

where $\lambda_{\min}(N_k^T H_k N_k)$ is the smallest eigenvalue of matrix $N_k^T H_k N_k$.

(G4) There exists a constant $M_A > 0$ such that

$$\sigma_{\min}(A_k) \geq M_A, \tag{15}$$

where $\sigma_{\min}(A_k)$ is the smallest singular value of A_k.

In every iteration, it is possible to make sure that (14) is valid by monitoring and possibly modifying the eigenvalues of the reduced Hessian of (3) (see [28]). Similarly, we can guarantee that the entire sequence $\{H_k\}$ is uniformly bounded.

In the convergence analysis of the filter method, we employ

$$\chi(x_k) := \|g_k - A_k y_{k+1}\|_2 \tag{16}$$

as the criticality measure. If $\theta_k \to 0$ and $\chi(x_{k_i}) \to 0$, then there exists a y_* such that the KKT conditions (2) are satisfied for (x_*, y_*).

Let us prove the global convergence of Algorithm I in detail.

Firstly, we give some preliminary results.

Lemma 3.1. *Suppose Assumptions G hold. Then there exist constants M_p, M_y, $M_m > 0$, such that*

$$\|p_k\| \leq M_p, \quad \|y_{k+1}\| \leq M_y, \quad |m_k(\alpha)| \leq M_m\alpha,$$

for all k and $\alpha \in (0, 1]$.

Proof. From (G1) we have that the right-hand side of (3) is uniformly bounded. Additionally, Assumptions (G2), (G3), and (G4) guarantee that the inverse of the matrix in (3) exists and is uniformly bounded for all k. Consequently, the solution of (3) $(p_k, y_{k+1})^T$ is uniformly bounded, i.e., $\|p_k\| \leq M_p$ and $\|y_{k+1}\| \leq M_y$. From (9), we can get $m_k(\alpha)/\alpha = (g_k - A_k y_{k+1})^T H_k p_k$. So, $|m_k(\alpha)| \leq M_m\alpha$. □

Lemma 3.2. *Suppose Assumption (G1) holds. Then there exist constants $C_\theta, C_\omega > 0$ such that for all k and $\alpha \in (0, 1]$*

$$|\theta(x_k + \alpha p_k) - (1 - \alpha)\theta(x_k)| \leq C_\theta \alpha^2 \|p_k\|^2, \tag{17a}$$

$$|\omega(x_k + \alpha p_k) - \omega(x_k) - m_k(\alpha)| \leq C_\omega \alpha^2 \|p_k\|^2. \tag{17b}$$

Proof. The proof of (17a) can be found in Lemma 3.3 from [22].

Next, we give the proof of (17b). From second order Taylor expansions,

$$\begin{aligned}
& |\omega(x_k+\alpha p_k)-\omega(x_k)-m_k(\alpha)| \\
\overset{(9)}{=} & \left|\frac{1}{2}\|\nabla_x L(x_k+\alpha p_k,y_{k+1})\|^2-\frac{1}{2}\|\nabla_x L(x_k,y_{k+1})\|^2\right. \\
& \left. -\alpha(g_k-A_k y_{k+1})^T H_k p_k\right| \\
= & \left|\frac{1}{2}\nabla_x L(x_k+\alpha p_k,y_{k+1})^T\nabla_x L(x_k+\alpha p_k,y_{k+1})\right. \\
& \left. -\alpha(g_k-A_k y_{k+1})^T H_k p_k-\frac{1}{2}\nabla_x L(x_k,y_{k+1})^T\nabla_x L(x_k,y_{k+1})\right| \\
\le & \left|\frac{1}{2}\nabla_x L(x_k,y_{k+1})^T\nabla_x L(x_k,y_{k+1})+\alpha\nabla_x L(x_k,y_{k+1})^T\nabla^2_{xx}L(x_k,y_{k+1})p_k\right. \\
& \left. +o(\alpha^2\|p_k\|^2)-\frac{1}{2}\nabla_x L(x_k,y_{k+1})^T\nabla_x L(x_k,y_{k+1})-\alpha(g_k-A_k y_{k+1})^T H_k p_k\right| \\
= & \left|\alpha(g_k-A_k y_{k+1})^T(\nabla^2_{xx}L(x_k,y_{k+1})-H_k)p_k+o(\alpha^2\|p_k\|^2)\right| \\
\overset{(4)}{=} & \left|O(\alpha^2\|p_k\|^2)+o(\alpha^2\|p_k\|^2)\right| \\
\le & C_\omega\alpha^2\|p_k\|^2.
\end{aligned}$$

So, the conclusion is proved. □

Lemma 3.3. *Suppose Assumptions G hold. If $\{x_{k_i}\}$ is a subsequence of iterates for which $\chi(x_{k_i})\ge\epsilon_1$ with constants $\epsilon_1>0$ independent of i. Then there exists $\epsilon_2>0$ independent of i, we have*

$$\theta(x_{k_i})=0 \quad\Longrightarrow\quad m_{k_i}(\alpha)<-\alpha\epsilon_2<0, \tag{18}$$

for all i and $\alpha\in(0,1]$. Then,

$$\Theta_k\overset{\text{def}}{=}\min\{\theta:(\theta,\omega)\in\mathcal{F}_k\}>0, \tag{19}$$

for all k and $\alpha\in(0,1]$.

Proof. If $\theta(x_k)=0$, but $\chi(x_k)>0$, the Algorithm I would not terminate. From (3) and (16),

$$m_k(\alpha)/\alpha=(g_k-A_k y_{k+1})^T H_k p_k=-\|g_k-A_k y_{k+1}\|_2^2=-\chi_k^2<-\epsilon_1^2,$$

so

$$m_k(\alpha)<-\alpha\epsilon_1^2=-\alpha\epsilon_2<0,$$

where $\epsilon_2=\epsilon_1^2$, i.e., (18) holds.

The proof of (19) is by induction. From initialize of Algorithm I, it is clear that the statement is valid for $k = 0$ because $\theta_{\max} > 0$. Suppose the statement is true for k. If $\theta(x_k) > 0$ and the filter is augmented in iteration k, it is clear from the update rule (12) that $\Theta_{k+1} > 0$, since $\gamma_\theta \in (0, 1)$. On the other hand, if $\theta(x_k) = 0$, we have got $m_k(\alpha) < 0$, for all $\alpha \in (0, 1]$. So, the switching condition (8) is true for all trial step sizes. Therefore, algorithm always makes α_k have been accepted then it must have that α_k satisfies (10). Consequently, the filter is not augmented. Hence, $\Theta_{k+1} = \Theta_k > 0$. □

Under Assumptions G, the sequence $\theta(x_k)$ converges to zero can be proved. That is to say, all limit points of $\{x_k\}$ are feasible (see, Lemma 3.4, Lemma 3.5 and Theorem 3.1). Consider from whether the filter is augmented a finite number of times or not.

Lemma 3.4. *Suppose that Assumptions G hold. If the filter is augmented only a finite number of times, i.e.,* $|\mathcal{Z}| < \infty$. *Then*

$$\lim_{k\to\infty} \theta(x_k) = 0. \tag{20}$$

Proof. Choose K such that for all iterations $k \geq K$ the filter is not augmented in iteration k. From the filter augmenting in Algorithm I we then have that for all $k \geq K$ both conditions (8) and (10) are satisfied for α_k. We now distinguish two cases, where $k \notin \mathcal{Z}$.

Case 1 ($\tau > 1$). From (8) it follows with M_m from Lemma 3.1 that

$$\delta[\theta(x_k)]^\phi < [-m_k(\alpha_k)]^\tau [\alpha_k]^{1-\tau} \leq M_m^\tau \alpha_k.$$

Hence, we can get

$$(\alpha_k)^{\tau-1} > \delta^{\tau-1}[\theta(x_k)]^{\phi(\tau-1)} M_m^{\tau(1-\tau)}$$

and

$$-m_k(\alpha_k) > \delta[\theta(x_k)]^\phi M_m^{1-\tau}.$$

This implies

$$\begin{aligned}\omega(x_k) - \omega(x_k(\alpha_k)) &\geq \eta_\omega(-m_k(\alpha_k))\\ &> \eta_\omega \delta[\theta(x_k)]^\phi M_m^{1-\tau}\\ &= c_1[\theta(x_k)]^\phi\end{aligned}$$

with $c_1 = \eta_\omega \delta M_m^{1-\tau}$.

Case 2 ($\tau = 1$). From (8) we have $\delta[\theta(x_k)]^\phi < -m_k(\alpha_k)$ such that from (10) we immediately obtain

$$\begin{aligned}\omega(x_k) - \omega(x_k(\alpha_k)) &\geq \eta_\omega(-m_k(\alpha_k))\\ &> \eta_\omega \delta[\theta(x_k)]^\phi.\end{aligned}$$

In either case, we have for all $k \notin \mathcal{Z}$ that

$$\omega(x_k(\alpha_k)) - \omega(x_k) < -\tilde{c}[\theta(x_k)]^{\phi} \tag{21}$$

for some $\tilde{c} > 0$. Hence, for all $i = 1, 2, ...,$

$$\begin{aligned} \omega(x_{K+i}) &= \omega(x_K) + \sum_{k=K}^{K+i-1} (\omega(x_{k+1}) - \omega(x_k)) \\ &< \omega(x_k) - \tilde{c} \sum_{k=K}^{K+i-1} [\theta(x_k)]^{\phi}. \end{aligned}$$

Since $\omega(x_{K+i})$ is bounded below as $i \to \infty$, the series on the right-hand side in the last line is bounded, which in turn implies (20). □

The following lemma considers a subsequence $\{x_{k_i}\}$ with $k_i \in \mathcal{Z}$, for all i, and its proof is borrowed from the method of [6].

Lemma 3.5. *Let $\{x_{k_i}\}$ be a subsequence of iterates generated by Algorithm I such that the filter is augmented in iteration k_i, i.e., $k_i \in \mathcal{Z}$, for all i. Furthermore, assume that there exist constants $c_\omega \in \mathbb{R}$ and $C_\theta > 0$ such that*

$$\omega(x_{k_i}) \geq c_\omega \quad \textit{and} \quad \theta(x_{k_i}) \leq C_\theta,$$

for all i (for example, if Assumption (G1) holds). It then follows that

$$\lim_{i\to\infty} \theta(x_{k_i}) = 0.$$

Proof. Suppose, for the purpose of obtaining a contradiction, that there exists an infinite subsequence $\{k_i\} \subseteq \mathcal{Z}$ such that

$$\theta_{k_i} \geq \epsilon, \tag{22}$$

for all i and for some $\epsilon > 0$. At each iteration k_i, the $(\theta, \omega) - pair$ associated with x_{k_i}, that is $(\theta_{k_i}, \omega_{k_i})$, is added to the filter. This means that no other (θ, ω)-pair can be added to the filter at a later stage within the square

$$[\theta_{k_i} - \gamma_\theta \epsilon, \theta_{k_i}] \times [\omega_{k_i} - \gamma_\omega \epsilon, \omega_{k_i}]$$

or with the intersection of this square with $\mathcal{F}_0$. Note that this holds, even if $(\theta_{k_i}, \omega_{k_i})$ is later removed from the filter, since the rule for removing entries, ensures that the envelope never shrinks. Now observe that the area of each of theses squares is $\gamma_\theta \gamma_\omega \epsilon^2$. As a consequence, the set $\mathcal{F}_0 \bigcap \{(\theta, \omega) | \omega \leq \kappa_\omega\}$ is completely covered by at most a finite number of such squares, for any choice of $\kappa_\omega \geq c_\omega$. Since the pairs $(\theta_{k_i}, \omega_{k_i})$ keep on being added to the filter, this implies that ω_{k_i} tends to infinity when i tends to infinity. Let us assume, without loss of generality, that $\omega_{k_{i+1}} \geq \omega_{k_i}$, for all i sufficiently large. But (7) and (22) imply that

$$\theta_{k_{i+1}} \leq (1-\gamma_\theta)\theta_{k_i} \leq \theta_{k_i} - \gamma_\theta \epsilon,$$

and therefore that θ_{k_i} converges to zero, which contradicts (22). Hence this latter assumption is impossible and the conclusion follows. $\square$

Based on the previous two lemmas, we can prove the following theorem similar to the proof in [21].

Theorem 3.1. *Suppose Assumptions G hold. Then*

$$\lim_{k\to\infty} \theta(x_k) = 0. \tag{23}$$

The next lemma shows that the Armijo condition (10) is satisfied under certain condition that there exists a step length bounded away from zero.

Lemma 3.6. *Suppose Assumptions G hold. Let $\{x_{k_i}\}$ be a subsequence. There exists certain constant $\bar{\alpha} > 0$ such that for all k_i and $\alpha \leq \bar{\alpha}$*

$$\omega(x_{k_i} + \alpha p_{k_i}) - \omega(x_{k_i}) \leq \eta_\omega m_{k_i}(\alpha). \tag{24}$$

Proof. Let M_p and C_ω be the constants from Lemmas 3.1 and 3.2. It then follows for all $\alpha \leq \bar{\alpha}$ with $\bar{\alpha} := \frac{(1-\eta_\omega)\epsilon_2}{C_\omega M_p^2}$ and (18) that

$$\begin{aligned} & \omega(x_{k_i} + \alpha p_{k_i}) - \omega(x_{k_i}) - m_{k_i}(\alpha) \\ \leq\ & C_\omega \alpha^2 \|p_{k_i}\|^2 \\ \leq\ & \alpha(1-\eta_\omega)\epsilon_2 \\ \leq\ & -(1-\eta_\omega) m_{k_i}(\alpha), \end{aligned}$$

which implies (24). $\square$

The previous Theorem 3.1 has proved that a series of points generated by the Algorithm I make the constraint violation tend to zero. Next, we prove that Assumptions G guarantee that the optimality measure $\chi(x_k)$ is not bounded away from zero, i.e., if $\{x_k\}$ is bounded, at least one limit point is a first order optimal point for the NPL (1).

Lemma 3.7. *Suppose that Assumptions G hold and that the filter is augmented only a finite number of times, i.e., $|\mathcal{Z}| < \infty$. Then*

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

Proof. Since $|\mathcal{Z}| < \infty$, there exists $K \in \mathbb{N}$ such that $k \notin \mathcal{Z}$, for all $k \geq K$. Suppose the claim is not true, i.e., there exist a subsequence $\{x_{k_i}\}$ and a constant

$\epsilon > 0$ such that $\chi(x_{k_i}) \geq \epsilon$, for all i. From (10), (18) and (23), there exist $\tilde{K} \geq K$, for $k_i \geq \tilde{K}$

$$\omega(x_{k_i+1}) - \omega(x_{k_i}) \leq \eta_\omega m_{k_i}(\alpha_{k_i}) \leq -\alpha_{k_i}\eta_\omega\epsilon_2.$$

Reasoning as in the proof of Lemma 3.4, one can conclude that $\lim_i \alpha_{k_i} = 0$, since $\omega(x_{k_i})$ is bounded below and $\omega(x_k)$ is monotonically decreasing (from (21)) for all $k \geq \tilde{K}$. We can now assume without loss of generality that $\tilde{K}$ is sufficiently large such that $\alpha_{k_i} < 1$. This means that for $k_i \geq \tilde{K}$ the first trial step $\alpha_{k,0} = 1$ has not been accepted. The last rejected trial step size

$$\alpha_{k_i,l_i} \in [\alpha_{k_i}/\tau_2, \alpha_{k_i}/\tau_1] \tag{25}$$

during the backtracking line search procedure then satisfies (8) since $k_i \notin \mathcal{Z}$ and $\alpha_{k_i,l_i} > \alpha_{k_i}$. Thus, it must have been rejected because it violates (10), i.e., it satisfies

$$\omega(x_{k_i} + \alpha_{k_i,l_i}p_{k_i}) - \omega(x_{k_i}) > \eta_\omega m_{k_i}(\alpha_{k_i,l_i}), \tag{26}$$

or it has been rejected because it is not acceptable to the current filter, i.e.,

$$(\theta(x_{k_i} + \alpha_{k_i,l_i}p_{k_i}), \omega(x_{k_i} + \alpha_{k_i,l_i}p_{k_i})) \in \mathcal{F}_{k_i} = \mathcal{F}_K. \tag{27}$$

We conclude the proof by showing that neither (26) nor (27) can be true for sufficiently large k_i.

As for (26), since $\lim_i \alpha_{k_i} = 0$, so $\lim_i \alpha_{k_i,l_i} = 0$ (see (25)). In particular, for sufficiently large k_i, $\alpha_{k_i,l_i} \leq \bar{\alpha}$ with $\bar{\alpha}$ from Lemma 3.6, i.e., (26) can not be satisfied for those k_i.

As for (27), from Lemma 3.3, $\Theta_K > 0$. Using Lemma 3.1 and Lemma 3.2, we see that

$$\theta(x_{k_i} + \alpha_{k_i,l_i}p_{k_i}) \leq (1 - \alpha_{k_i,l_i})\theta(x_{k,i}) + C_\theta M_p^2(\alpha_{k_i,l_i})^2.$$

Since $\lim_i \alpha_{k_i,l_i} = 0$ and from Theorem 3.1 also $\lim_i \theta(x_{k_i}) = 0$, it follows that $\theta(x_{k_i} + \alpha_{k_i,l_i}p_{k_i}) < \Theta_K$ for k_i sufficiently large, which contradicts (27). □

In order to find the acceptable step size of the current filter (see (11)), we establish a bound of step size as follows.

Lemma 3.8. *Suppose Assumptions G hold. Let* $\{x_{k_i}\}$ *be a subsequence and* $m_{k_i}(\alpha) \leq -\alpha\epsilon_2$ *for a constant* $\epsilon_2 > 0$ *independent of* k_i *and for all* $\alpha \in (0, 1]$. *Then there exist constants* $c_2, c_3 > 0$ *such that*

$$(\theta(x_{k_i} + \alpha p_{k_i}), \omega(x_{k_i} + \alpha p_{k_i})) \notin \mathcal{F}_{k_i},$$

for all k_i *and* $\alpha \leq \min\{c_2, c_3\theta(x_{k_i})\}$.

Proof. Let M_p, C_θ, and C_ω be the constants from Lemma 3.1 and Lemma 3.2. Define $c_2 := \min\{1, \epsilon_2/(M_p^2 C_\omega)\}$ and $c_3 := 1/(M_p^2 C_\theta)$.

Now choose an iterate x_{k_i}. The mechanisms of Algorithm I ensure that

$$(\theta(x_{k_i}), \omega(x_{k_i})) \notin \mathcal{F}_{k_i}.$$

For $\alpha \leq c_2$ we have $\alpha^2 \leq \frac{\alpha\epsilon_2}{M_p^2 C_\omega} \leq \frac{-m_{k_i}(\alpha)}{C_\omega \|p_{k_i}\|^2}$ or, equivalently,

$$m_{k_i}(\alpha) + C_\omega \alpha^2 \|p_{k_i}\|^2 \leq 0$$

and it follows with (17b) that

$$\omega(x_{k_i} + \alpha p_{k_i}) \leq \omega(x_{k_i}).$$

Similarly, for $\alpha \leq c_3\theta(x_{k_i}) \leq \frac{\theta(x_{k_i})}{\|p_{k_i}\|^2 C_\theta}$, we have $-\alpha\theta(x_{k_i}) + C_\theta \alpha^2 \|p_{k_i}\|^2 \leq 0$ and thus from (17a)

$$\theta(x_{k_i} + \alpha p_{k_i}) \leq \theta(x_{k_i}).$$

The update rule (12) imply that for all k, the filter has the following property:

$$(\bar{\theta}, \bar{\omega}) \notin \mathcal{F}_k \;\Rightarrow\; (\theta, \omega) \notin \mathcal{F}_k \qquad \text{if} \qquad \theta \leq \bar{\theta} \;\text{ and }\; \omega \leq \bar{\omega}.$$

Then we obtain the conclusion. □

The last lemma explains that the filter is eventually not augmented when the iteration corresponds to a subsequence with only nonoptimal limit points. This result is used in the proof of the main global convergence theorem to yield a contradiction.

Lemma 3.9. *Suppose Assumption G hold. Let $\{x_{k_i}\}$ be a subsequence of $\{x_k\}$ with $\chi(x_{k_i}) \geq \epsilon$ for a constant $\epsilon > 0$ independent of k_i. The there exists $K \in \mathbb{N}$ such that for all $k_i \geq K$ the filter is not augmented in iteration k_i, i.e., $k_i \notin \mathcal{Z}$.*

Proof. Since by Theorem 3.1 we have $\lim_i \theta(x_{k_i}) = 0$, it follows from Lemma 3.3 that there exist constants $\epsilon_1, \epsilon_2 > 0$ such that

$$\theta(x_{k_i}) \leq \epsilon_1 \quad \textit{and} \quad m_{k_i}(\alpha) \leq -\epsilon_2\alpha \tag{28}$$

for k_i sufficiently large and $\alpha \in (0, 1]$. Without loss of generality we can assume that (28) is valid for all k_i. We can now apply Lemma 3.6 and Lemma 3.8 to obtain the constants $\bar{\alpha}, c_2, c_3 > 0$. Choose $K > \mathbb{N}$ such that for all $k_i \geq K$

$$\theta(x_{k_i}) < \min\left\{\theta_{\text{inc}}, \frac{\bar{\alpha}}{c_3}, \frac{c_2}{c_3}, \left[\frac{\tau_1 c_3 \epsilon_2^{\tau}}{\delta}\right]^{\frac{1}{\phi-1}}\right\} \tag{29}$$

with τ_1 from Algorithm I. For all $k_i \geq K$ with $\theta(x_{k_i}) = 0$ we can argue as in the proof of Lemma 3 that both (8) and (10) hold in iteration k_i such that $k_i \notin \mathcal{Z}$.

For the remaining iterations $k_i \geq K$ with $\theta(x_{k_i}) > 0$ we note that (29) implies that

$$\frac{\delta[\theta(x_{k_i})]^{\phi}}{\epsilon_2^{\tau}} < \tau_1 c_3 \theta(x_{k_i})$$

(since $\phi > 1$), as well as

$$c_3\theta(x_{k_i}) < \min\{\bar{\alpha}, c_2\}.$$

Now choose an arbitrary $k_i \geq K$ with $\theta(x_{k_i}) > 0$ and define

$$\beta_{k_i} = c_3\theta(x_{k_i}) = \min\{\bar{\alpha}, c_2, c_3\theta(x_{k_i})\}.$$

Lemma 3.6 and Lemma 3.8 then imply that a trial step size $\alpha_{k,l} \leq \beta_{k_i}$ satisfies both

$$\omega(x_{k_i}(\alpha_{k_i,l})) - \omega(x_{k_i}) \leq \eta_\omega m_k(\alpha_{k_i,l}) \tag{30}$$

and

$$(\theta(x_{k_i}(\alpha_{k_i,l}), \omega(x_{k_i}(\alpha_{k_i,l})))) \notin \mathcal{F}_{k_i}. \tag{31}$$

If we now denote with $\alpha_{k_i,L}$ the first trial step size satisfying both (30) and (31), then the backtracking line search procedure implies that for $\alpha \geq \alpha_{k_i,L}$

$$\alpha \geq \tau_1\beta_{k_i} = \tau_1 c_3\theta(x_{k_i}) > \frac{\delta[\theta(x_{k_i})]^{\phi}}{\epsilon_2^{\tau}}$$

and therefore for $\alpha \geq \alpha_{k_i,L}$

$$\delta[\theta(x_{k_i})]^{\phi} < \alpha\epsilon_2^{\tau} = \alpha^{1-\tau}(\alpha\epsilon_2)^{\tau} \leq \alpha^{1-\tau}[-m_{k_i}(\alpha)]^{\tau}.$$

This means that $\alpha_{k_i,L}$ and all previous trial step sizes satisfy (8) and (10). Consequently, for all trial step sizes $\alpha_{k_i,l} \geq \alpha_{k_i,L}$, we have $\alpha_{k_i,L} \geq \alpha_{k_i}^{\min}$. Hence, the method dose not switch to the feasibility phase for those trial step sizes. Therefore, $\alpha_{k_i,L}$ is indeed the accepted step size α_{k_i}. Since it satisfies both (8) and (30), the filter is not augmented in iteration k_i. □

Based on the lemmas and theorem above, we prove the main conclusion of this paper, that is, the global convergence result of the Algorithm I.

Theorem 3.2. *Suppose Assumptions G hold. Then*

$$\lim_{k\to\infty} \theta(x_k) = 0 \tag{32a}$$

and

$$\liminf_{k\to\infty} \chi(x_k) = 0. \tag{32b}$$

In other words, all limit points are feasible, and if $\{x_k\}$ is bounded, then there exists a limit point x_* of $\{x_k\}$ which is a first order optimal point for the equality constrained NLP(1).

Proof. (32a) follows from Theorem 3.1. In the case of the filter is augmented only a finite number of times, (32b) has been proved by Lemma 3.7. Otherwise, there exists a subsequence $\{x_{k_i}\}$ such that $k_i \in \mathcal{Z}$, for all i. Now, suppose that $\limsup_i \chi(x_{k_i}) > 0$. Then, there exist a subsequence $\{x_{k_{i_j}}\}$ of $\{x_{k_i}\}$ and a constant $\epsilon > 0$ such that $\lim_j \theta(x_{k_{i_j}}) = 0$ and $\chi(x_{k_{i_j}}) > \epsilon$, for all k_{i_j}. Applying Lemma 3.9 to $\{x_{k_{i_j}}\}$, we see that there is an iteration k_{i_j}, in which the filter is not augmented, i.e. $k_{i_j} \notin \mathcal{Z}$. This contradicts the choice of $\{x_{k_i}\}$ such that $\lim_i \chi(x_{k_i}) = 0$, which proves (32b). □

4. Numerical results

In this section, we present the numerical results of Algorithm I which have been performed on a desktop with Intel(R) Core(TM) i5-8250 CPU. Algorithm I was implemented as a MATLAB code and run under MATLAB version 9.2.0.518641(R2017a).

The selected parameter values are:

$$\begin{aligned}
&\epsilon = 10^{-6}, \gamma_\theta = 10^{-5},\\
&\gamma_\omega = 10^{-5}, \delta = 10^{-2},\\
&\gamma_\alpha = 10^{-4}, \phi = 2.01, \tau = 1.1,\\
&\eta_\omega = 0.25, \tau_1 = 0.25; \tau_2 = 0.75.
\end{aligned}$$

The computation terminates when

$$\max\{\|g(x_k) - A(x_k)y_{k+1}\|_2, \|c(x_k)\|\} \le \epsilon$$

is satisfied. The update of B_k is implemented by using the methods recommended in Section 4.4 and Section 4.5 in [16].

The results are reported in Table 1 where the test problems are numbered in the same way as in [14] and in [24], respectively. For example, HS27 is the problem 27 in [14] and S252 is the problem 252 in [24]. For higher dimension problems, we compute a few versions of different dimensions from CUTEst collection [10] to see the efficiency of Algorithm I (see Table 2). The CPU times in Table 1 and Table 2 are counted in seconds. In these tables, NIT and NG are the numbers of iterations.

In addition, we compared Algorithm I in this paper with SNOPT [9] (Version 5.3). The results are listed in Table 3. We also compared Algorithm I with the algorithm in [13] and in [12]. Furthermore, we show their comparison figures (see, figure 1).

Table 1: Numerical results of Algorithm I

Problem	NIT	NG	CPU Time	$\|c(x_k)\|$	$\|g_k - A_k y_{k+1}\|_2$
byrdsphr	5	5	0.0469	1.7585e-13	1.9254e-09
himmelba	2	2	0.0781	0	7.8531e-08
himmelbc	2	2	0.0313	2.8424e-13	9.3043e-08
hs046	73	73	0.1094	1.3668e-11	6.1298e-07
hypcir	2	2	0.0156	1.1055e-13	6.4251e-08
maratos	6	6	0.0156	5.6520e-07	5.6520e-07
mwright	48	48	0.0469	9.4930e-07	2.3720e-07
powellbs	24	24	0.0313	2.1871e-08	1.5785e-12
supersim	2	2	0.0156	0	3.2034e-10
tame	2	2	0.0313	0	2.6859e-10
booth	2	2	0.0469	8.8818e-16	2.3054e-07
hong	7	7	0.0156	2.7704e-07	1.8421e-09
gottfr	5	5	0.0156	2.7446e-07	5.0986e-09
hatfldf	10	10	0.0313	5.3390e-07	6.9068e-11
hs051	6	6	0.0156	0	1.0400e-08
haifas	14	14	0.0625	7.0891e-07	1.7048e-08
recipe	2	2	0.0313	3.0307e-15	2.2964e-07
robot	4	4	0.0469	3.3815e-13	8.1110e-07
try-b	4	4	0.0156	9.2922e-08	4.8773e-07
cluster	9	9	0.0313	3.0766e-07	1.2776e-07
zangwil3	4	4	0.0625	7.9331e-09	4.2125e-10
bt2	20	20	0.0156	6.7502e-14	4.6037e-08
bt3	7	7	0.0156	1.5701e-16	9.4202e-08
bt5	6	6	0.0156	2.0367e-09	2.6323e-07
bt6	13	13	0.0313	8.8776e-07	4.3058e-07
bt7	12	12	0.0313	9.1551e-16	2.0754e-08
bt8	7	7	0.0156	5.1592e-10	3.7845e-07
bt9	13	13	0.0156	2.1202e-09	4.2764e-08
bt10	7	7	0.0156	2.5066e-10	7.4824e-09
bt11	16	16	0.0156	5.6726e-13	9.0841e-08
bt12	9	9	0.0156	8.2212e-10	6.7032e-10
HS07	10	10	0.0156	1.6101e-10	3.4446e-08
HS08	6	6	0.0313	4.6483e-07	2.1096e-10
HS09	6	6	0.0156	0	2.1181e-07
HS26	16	16	0.0156	8.7215e-07	3.4365e-08
HS27	30	30	1.2813	1.4955e-10	6.5668e-07
HS28	8	8	0.0156	2.2204e-16	1.2406e-07
HS39	13	13	0.0156	5.8040e-12	5.1408e-07
HS40	6	6	0.0156	8.5750e-10	2.2161e-08
HS42	8	8	0.0156	1.9229e-12	7.3685e-07
HS47	30	30	0.1406	7.0971e-09	4.7927e-07
HS49	29	29	0.0313	0	5.0716e-07
HS61	7	7	0.0156	3.9662e-13	7.3996e-11
HS77	12	12	0.0313	9.6157e-12	2.4020e-07
HS78	12	12	0.0469	8.4495e-07	3.7625e-08
HS79	15	15	0.0156	2.2486e-12	9.3840e-08
S216	8	8	0.1563	0	7.0458e-07
S219	18	18	0.1250	2.5963e-13	5.6842e-10
S252	43	43	0.0313	2.8200e-13	3.5788e-09
S254	13	13	0.0156	1.6703e-09	9.2910e-07
S269	8	8	0.0469	1.1102e-16	1.1357e-07
S316	3	3	0.0156	5.5120e-12	9.3910e-07
S317	4	4	0.0156	3.5741e-07	6.3540e-08
S335	24	24	0.0469	1.1196e-13	1.1547e-07
S336	7	7	0.0313	1.7902e-15	3.0976e-08
S338	11	11	0.0156	1.5776e-10	2.8493e-07
S344	11	11	0.0313	4.1567e-13	3.2734e-08
S345	18	18	0.0469	4.0980e-10	5.9741e-07
S378	61	61	0.1094	6.9241e-10	4.3425e-07

Table 2: Numerical results of Algorithm I for higher dimension problems

Problem	n	m	NIT	NG	CPU-time	$\|c(x_k)\|$	$\|g_k - A_k y_{k+1}\|_2$
bdvalue	50	50	3	3	0.0625	1.2862e-16	9.4625e-09
	100	100	3	3	0.1875	1.8105e-16	1.4843e-08
	200	200	3	3	0.9844	2.0619e-16	2.2062e-08
	300	300	4	4	2.8594	2.8180e-16	6.9444e-09
	500	500	4	4	10.3750	3.2170e-16	1.6849e-08
	600	600	4	4	9.0625	3.8500e-16	1.8518e-08
	700	700	4	4	13.0781	3.9056e-16	2.6413e-08
	800	800	4	4	27.7031	4.5237e-16	2.8287e-08
	900	900	4	4	29.7812	4.7032e-16	3.0043e-08
	1000	1000	4	4	44.3594	4.9907e-16	3.1703e-08
broydn3d	10	10	2	2	0.0156	3.9823e-14	3.0160e-08
	100	100	2	2	0.9375	2.4878e-12	4.2629e-07
	200	200	2	2	2.2969	1.5215e-12	3.5042e-07
	300	300	2	2	4.7500	2.0474e-12	5.1705e-07
	400	400	2	2	6.3750	7.2633e-12	9.6331e-07
	500	500	2	2	16.3594	1.4098e-12	4.5545e-07
	600	600	2	2	19.9688	3.6116e-12	8.1693e-07
	700	700	2	2	19.7813	2.2992e-12	6.2797e-07
	800	800	2	2	64.1250	3.0570e-12	8.2897e-07
	900	900	2	2	96.1875	1.3670e-12	5.0918e-07
	1000	1000	2	2	153.5156	1.8805e-12	4.9751e-07
gilbert	50	1	23	23	0.0625	6.7002e-12	9.5829e-07
	100	1	24	24	0.2344	3.6336e-12	7.5592e-07
	200	1	25	25	0.6719	2.1927e-12	5.7521e-07
	300	1	25	25	1.5938	3.3153e-12	7.0874e-07
	400	1	25	25	3.3438	4.4363e-12	8.2086e-07
	500	1	25	25	4.7500	5.5415e-12	9.1941e-07
	600	1	26	26	7.2188	2.1458e-12	5.3157e-07
	700	1	26	26	10.9688	2.4893e-12	5.7464e-07
	800	1	26	26	15.0000	2.9293e-12	6.1470e-07
	900	1	26	26	19.5781	3.2849e-12	6.5231e-07
	1000	1	26	26	27.4219	3.4908e-12	6.8786e-07
	1300	1	26	26	68.7344	4.5803e-12	7.8493e-07
	1500	1	26	26	121.8906	5.5477e-12	8.4345e-07
	1800	1	26	26	240.9063	6.2157e-12	9.2432e-07
	2000	1	26	26	355.7345	7.1335e-12	9.7451e-07

Table 3: Comparison of Algorithm I with SNOPT

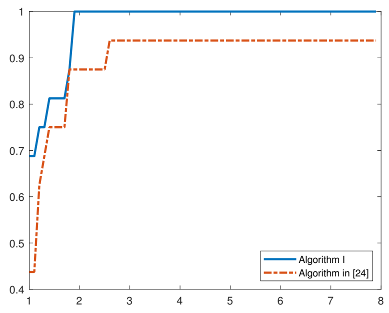
Problem	Algorithm I		SNOPT	
	NIT	NG	NIT	NG
bt2	20	20	16	20
bt3	7	7	5	5
bt5	8	8	10	10
bt6	15	15	17	16
bt7	12	12	21	36
bt8	7	7	13	13
bt9	18	18	20	30
bt10	7	7	23	23
bt11	19	19	15	14
bt12	9	9	9	9
hs07	10	10	18	30
hs08	5	5	2	2
hs09	7	7	9	8
hs26	16	16	25	24
hs27	30	30	22	23
hs28	8	8	4	4
hs39	13	13	20	30
hs40	6	6	7	7
hs42	8	8	8	8
hs47	29	29	24	31
hs49	29	29	37	32
hs61	7	7	69	168
hs77	12	12	15	14
hs78	12	12	9	7
hs79	15	15	14	14

Table 4: Comparison of Algorithm I with Algorithm in [13]

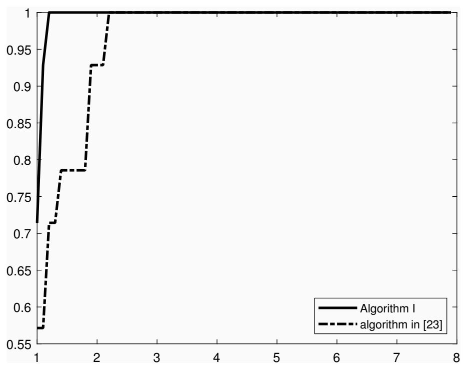
Problem			Algorithm I		Algorithm [13]	
	n	m	NIT	NG	NIT	NG
hs07	2	1	10	10	9	9
hs09	2	1	7	7	8	8
hs27	3	1	30	30	24	24
hs28	3	1	8	8	10	10
hs39	4	2	13	13	15	15
hs40	4	3	6	6	6	6
hs42	4	2	8	8	9	9
hs47	5	3	29	29	30	30
hs61	3	2	7	7	10	10
hs77	5	2	12	12	22	22
hs78	5	3	12	12	8	8
hs79	5	3	15	15	10	10
s219	4	2	18	18	26	26
s254	3	2	13	13	9	9
s269	5	3	8	8	9	9
s316	2	1	3	3	14	14

Table 5: Comparison of Algorithm I with Algorithm in [12]

Problem			Algorithm I		Algorithm in [12]	
	n	m	NIT	NG	NIT	NG
hs07	2	1	10	10	10	10
hs08	2	2	5	5	5	5
hs09	2	1	6	6	6	6
hs26	3	1	16	16	26	26
hs28	3	1	8	8	9	9
hs40	4	3	6	6	6	6
hs42	4	2	8	8	10	10
hs47	5	3	29	29	27	27
hs49	5	2	29	29	26	26
hs77	5	2	12	12	18	18
hs78	5	3	12	12	11	11
hs79	5	3	15	15	14	14
s216	2	1	8	8	12	12
s269	5	3	8	8	9	9



(a) Comparison of Algorithm I with [13].



(b) Comparison of Algorithm I with [12].

Figure 1: Comparison figure.

5. Conclusion

We have introduced a line search method combining filter technique for solving nonlinear equality constrained optimization. The global convergence analysis for this algorithm is presented under some reasonable assumptions. The preliminary numerical experiments show its practical performance. The comparison of numerical experiment data show that the algorithm in this paper has good effect. The main feature of the proposed filter method is that it is not required to compute the value of objective function (or the Lagrange function) in every iteration to obtain a first order optimal point. However, the analysis of local convergence rate is absent. That is what we are working on.

Acknowledgements

The authors are particularly grateful to the anonymous reviewers for their valuable suggestions, insightful comments, and encouragement regarding this paper, which greatly help us to improve the presentation of this paper.

The authors gratefully acknowledge the supports of the National Natural Science Foundation of China (12071133,11801152), Key Scientific Research Project for Colleges and Universities in Henan Province (21A110012).

References

[1] F. Arzani, M.R. Peyghami, *A new dwindling nonmonotone filter method without gradient information for solving large-scale systems of equations*, Iranian Journal of Numerical Analysis and Optimization, 8 (2018), 19-39.

[2] Roberto H. Bielschowsky, Francisco A. M. Gomes, *Dynamic control of infeasibility in equality constrained optimization*, SIAM J. Optim., 19 (2008), 1299-1325.

[3] Yannan Chen, Wenyu Sun, *A dwindling filter line search method for unconstrained optimization*, Math. Comp., 84 (2015), 187-208.

[4] Zhongwen Chen, Yu-Hong Dai, Jiangyan Liu, *A penalty-free method with superlinear convergence for equality constrained optimization*, Computational Optimization and Applications, 1-33.

[5] J. E. Dennis, Jr., Mahmoud El-Alem, Maria C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, SIAM J. Optim., 7 (1997), 177-207.

[6] Roger Fletcher, Nicholas I. M. Gould, Sven Leyffer, Philippe L. Toint, Andreas Wachter, *Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming*, SIAM J. Optim., 13 (2002), 635-659 (2003).

[7] Roger Fletcher, Sven Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), 239-269.

[8] Roger Fletcher, Sven Leyffer, Philippe L. Toint, *On the global convergence of a filter-SQP algorithm*, SIAM J. Optim., 13 (2002), 44-59 (electronic).

[9] Philip E. Gill, Walter Murray, Michael A. Saunders, *SNOPT: an SQP algorithm for large-scale constrained optimization*, SIAM Rev., 47 (2005), 99-131 (electronic).

[10] Nicholas I. M. Gould, Dominique Orban, Philippe L. Toint, *CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization*, Comput. Optim. Appl., 60 (2015), 545-557.

[11] Nicholas IM Gould, Yueling Loh, Daniel P Robinson, *A nonmonotone filter sqp method: Local convergence and numerical results*, SIAM Journal on Optimization, 25 (2015), 1885-1911.

[12] Chao Gu, Detong Zhu, *A nonmonotone line search filter method with reduced Hessian updating for nonlinear optimization*, J. Syst. Sci. Complex., 26 (2013), 534-555.

[13] Chao Gu, Detong Zhu, *A dwindling filter line search algorithm for nonlinear equality constrained optimization*, J. Syst. Sci. Complex., 28 (2015), 623-637.

[14] Willi Hock, Klaus Schittkowski, *Test examples for nonlinear programming codes,*, volume 187 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin-New York, 1981.

[15] Renke Kuhlmann, Christof B¨uskens, *A primal-dual augmented Lagrangian penalty-interior-point filter line search algorithm*, Math. Methods Oper. Res., 87 (2018), 451-483.

[16] Marucha Lalee, Jorge Nocedal, *Todd Plantenga. On the implementation of an algorithm for large-scale equality constrained optimization*, SIAM J. Optim., 8 (1998), 682-706.

[17] Sven Leyffer, Charlie Vanaret, *Augmented lagrangian filter method*, 2016.

[18] Dan Li, Detong Zhu, *An affine scaling interior trust-region method combining with line search filter technique for optimization subject to bounds on variables*, Numer. Algorithms, 77 (2018), 1159-1182.

[19] Xinwei Liu, Yaxiang Yuan, *A sequential quadratic programming method without a penalty function or a filter for nonlinear equality constrained optimization*, SIAM J. Optim., 21, 545-571.

[20] Jorge Nocedal, Stephen J. Wright, *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second edition, 2006.

[21] Yonggang Pei, Detong Zhu, *A trust-region algorithm combining line search filter method with Lagrange merit function for nonlinear constrained optimization*, Appl. Math. Comput., 247 (2014), 281-300.

[22] Yonggang Pei, Detong Zhu, *A trust-region algorithm combining line search filter technique for nonlinear constrained optimization*, Int. J. Comput. Math., 91 (2014), 1817-1839.

[23] Songqiang Qiu, Zhongwen Chen, *Global and local convergence of a class of penalty-free-type methods for nonlinear programming*, Appl. Math. Model., 36 (2012), 3201-3216.

[24] Klaus Schittkowski, *More test examples for nonlinear programming codes*, volume 282 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, 1987.

[25] Chungen Shen, Sven Leyffer, Roger Fletcher, *A nonmonotone filter method for nonlinear optimization*, Comput. Optim. Appl., 52 (2012), 583-607.

[26] Z.B. Sun, Y.Y. Sun, Y. Li, K.P. Liu, *A new trust region-sequential quadratic programming approach for nonlinear systems based on nonlinear model predictive control*, Eng. Optim., 51 (2019), 1071-1096.

[27] M. Ulbrich, S. Ulbrich, *Non-monotone trust region methods for nonlinear equality constrained optimization without a penalty function*, volume 95 (2003), 103-135. ISMP 2000, Part 3 (Atlanta, GA).

[28] Robert J. Vanderbei and David F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, volume 13 (1999), pages 231-252, Computational optimization—a tribute to Olvi Mangasarian, Part II.

[29] Andreas Wachter, Lorenz T. Biegler, *Line search filter methods for nonlinear programming: local convergence*, SIAM J. Optim., 16 (2005), 32-48 (electronic).

[30] Andreas Wachter, Lorenz T. Biegler, *Line search filter methods for nonlinear programming: motivation and global convergence*, SIAM J. Optim., 16 (2005), 1-31 (electronic).

[31] Andreas Wachter, Lorenz T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., 106 (2006), 25-57.

[32] Andrea Walther, Lorenz Biegler, *On an inexact trust-region SQP-filter method for constrained nonlinear optimization*, Comput. Optim. Appl., 63 (2016), 613-638.

[33] Zhujun Wang, Li Cai, and Detong Zhu, *Line search filter inexact secant methods for nonlinear equality constrained optimization*, Appl. Math. Comput., 263 (2015), 47-58.

[34] Wenjuan Xue, Chungen Shen, Dingguo Pu, *A penalty-function-free line search SQP method for nonlinear programming*, J. Comput. Appl. Math., 228 (2009), 313-325.

[35] Xiaojing Zhu Dingguo Pu, *A line search filter algorithm with inexact step computations for equality constrained optimization*, Appl. Numer. Math., 62 (2012), 212-223.

Accepted: January 29, 2021