Applied Mathematical Modelling 35 (2011) 1696–1708

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A modified QP-free feasible method ☆

Hua Wang [a,*], Dingguo Pu [b]

[a] School of Mathematics and Information, Shanghai Lixin University of Commerce, Shanghai 201620, PR China
[b] Department of Mathematics, Tongji University, Shanghai 200092, PR China

ARTICLE INFO

Article history:
Received 25 November 2008
Received in revised form 15 September 2010
Accepted 4 October 2010
Available online 13 October 2010

Keywords:
Constrained optimization
QP-free algorithm
NCP function
Filter
Global convergence
Superlinear convergence

ABSTRACT

In this paper, we presented a modified QP-free filter method based on a new piecewise linear NCP functions. In contrast with the existing QP-free methods, each iteration in this algorithm only needs to solve systems of linear equations which are derived from the equality part in the KKT first order optimality conditions. Its global convergence and local superlinear convergence are obtained under mild conditions.

1. Introduction

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

$$\begin{aligned} &\min f(x), \\ &\text{s.t.} \quad g_j(x) \leqslant 0, \quad j \in I = \{1,2,\ldots,m\}, \end{aligned} \tag{1}$$

where f: $R^n \to R$ and $g(x) = (g_1(x), g_2(x), \ldots, g_m(x))^T$: $R^n \to R^m$ are continuously differentiable functions. We denote by $D = \{x \in R^n | g(x) < 0\}$ and $\bar{D} = cl(D)$ the strictly feasible set and the feasible set of the Problem (NLP), respectively.

The Lagrangian function associated with the Problem (NLP) is the function

$$L(x,\lambda) = f(x) + \lambda^T g(x),$$

where $\lambda = (\lambda^1, \lambda^2, \ldots, \lambda^m)^T \in R^m$ is the multiplier vector. For simplicity, we use (x,λ) to denote the column vector $(x^T, \lambda^T)^T$.

A point $(\bar{x},\bar{\lambda}) \in R^n \times R^m$ is called a *Karush–Kuhn–Tucker* (KKT) point or a KKT pair of Problem (NLP), if it satisfies the following conditions:

$$\nabla_x L(\bar{x},\bar{\lambda}) = 0, \quad g(\bar{x}) \leqslant 0, \quad \bar{\lambda} \geqslant 0, \quad g_i(\bar{x})\bar{\lambda}_i = 0, \quad \forall i \in I, \tag{2}$$

where $I := \{1 \leqslant i \leqslant m\}$. We also say $\bar{x}$ is a KKT point if there exists a $\bar{\lambda}$ such that $(\bar{x},\bar{\lambda})$ satisfies (2).

☆ This research is supported by the National Science Foundation of China (No. 10771162) and Shanghai Budget Project (No. 1130IA15, No. 1130IA18).
* Corresponding author.
E-mail address: chengyj819@sina.com (H. Wang).

0307-904X/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.apm.2010.10.002

There are many practical methods for solving problem (NLP). Among these methods, as we know, the sequential quadratic programming (SQP) method is one of the most efficient methods to solve problem (NLP). Because its superlinear convergence rate, it has been widely studied [1–8]. 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 that the related quadratic programming subproblems to be solvable per iteration, but it is difficult to be satisfied. Moreover, the solution of the sequential quadratic subproblem may be unbounded, which leads to the sequence generated by the method is divergence.

Based on the above reasons, Panier et al. [9] gave a feasible QP-free algorithm for overcoming the difficulties encountered in the SQP methods. Their method needs to solve two linear systems and a quadratic subproblem at each iteration. In addition, in the global convergence theorem, there is a restrictive condition which requires that the number of stationary points is finite. By the means of the Fisher–Burmeister function, Qi and Qi [10] proposed a QP-free algorithm for solving problem (NLP). It need to solve three linear systems and one least-square problem at each iteration. Using a new piecewise linear NCP functions, Zhou and Pu [11] proposed a QP-free method which need to solve three linear systems and one least-square problem at each iteration, and they only proved the global convergence of the algorithm.

In this paper, we presented a modified QP-free filter method based on the new piecewise linear NCP functions proposed by Zhou and Pu [11]. This algorithm has the following merits: it requires to solve only systems of linear equations. In order to overcome the Maratos effect, a high order direction is computed by solving a system of linear equations with small scale. Moreover, we adapt the filter technique, which is proposed by Fletcher and Leyffer [12] in 2002, and it saves the computational cost largely. In the end, its global convergence and local superlinear convergence are obtained under mild conditions.

2. Preliminaries

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

2.1. Some definitions and propositions

Definition 2.1 (*NCP pair and SNCP pair*). We call a pair $(a,b) \in R^2$ to be an NCP pair if $a \geqslant 0$, $b \geqslant 0$ and $ab = 0$; and call (a,b) to be an SNCP pair if (a,b) is an NCP pair and $a^2 + b^2 \neq 0$.

Definition 2.2 (*NCP function*). A function ϕ: $R^2 \to R$ is called an NCP function if $\phi(a,b) = 0$ if and only if (a,b) is an NCP pair.

In this paper, we use a new 3-piecewise linear NCP function $\psi(a,b)$ as follows:

$$\psi(a,b) = \begin{cases} 3a - a^2/b & \text{if} \quad b \geqslant a > 0, \text{ or } 3b > -a \geqslant 0, \\ 3b - b^2/a & \text{if} \quad a > b > 0 \text{ or } 3a > -b \geqslant 0, \\ 9a + 9b & \text{if} \quad 0 \geqslant a \text{ and } -a \geqslant 3b, \text{ or } -3a \leqslant b \leqslant 0, \end{cases} \tag{3}$$

If $(a,b) \neq (0,0)$, then

$$\nabla\psi(a,b) = \begin{cases} \begin{pmatrix} 3 - 2a/b \\ a^2/b^2 \end{pmatrix} & \text{if} \quad b \geqslant a > 0, \text{ or } 3b > -a \geqslant 0, \\ \begin{pmatrix} b^2/a^2 \\ 3 - 2b/a \end{pmatrix} & \text{if} \quad a > b > 0 \text{ or } 3a > -b \geqslant 0, \\ \begin{pmatrix} 9 \\ 9 \end{pmatrix} & \text{if} \quad 0 \geqslant a \text{ and } -a \geqslant 3b, \quad \text{or } -3a \leqslant b \leqslant 0 \end{cases} \tag{4}$$

and

$$A_\psi = \partial_B\psi(0,0) = \left\{ \begin{pmatrix} 3 - 2t \\ t^2 \end{pmatrix} : -3 \leqslant t \leqslant 1 \right\} \cup \left\{ \begin{pmatrix} t^2 \\ 3 - 2t \end{pmatrix} : -3 \leqslant t \leqslant 1 \right\}. \tag{5}$$

It is easy to check the following proposition.

Proposition 2.1. *For the function $\psi(a,b)$ the following holds.*

(I) $\psi(a, b) = 0 \iff a \geqslant 0,\ b \geqslant 0,\ ab = 0$;
(II) *the square of ψ is continuously differentiable;*
(III) *ψ is twice continuously differentiable everywhere except at the origin, but it is strongly semismooth at the origin;*
(IV) *for any $(\alpha,\beta) \in \partial_B\psi(a,b)$, $(a,b) \neq (0,0)$, or any $(\alpha,\beta) \in \partial_B\psi(0,0)$, $\alpha^2 + \beta^2 \geqslant 1 > 0$.*

Now we construct the semismooth equation $\Phi(x,\lambda) = 0$, which is equivalently reformulated as the KKT point conditions. Let

$$\Phi(x,\lambda) = (\phi_1(x,\lambda),\ldots,\phi_{n+m}(x,\lambda))^T,$$

where

$$\phi_i(x,\lambda) = \nabla_{x_i} L(x,\lambda), \quad 1 \leqslant i \leqslant n,$$

and

$$\phi_i(x,\lambda) = \psi(-g_j(x),\lambda_j), \quad n+1 \leqslant i = n+j \leqslant n+m.$$

If $(g_j(x),\lambda_j) \neq (0,0)$ and $n+1 \leqslant i = j+n \leqslant n+m$, then $\phi_i(x,\lambda)$ is continuously differentiable at $(x,\lambda) \in R^{n+m}$. We have

$$\nabla\phi_i = \begin{cases} \begin{pmatrix} -(3+2g_j(x)/\lambda_j)\nabla g_j(x), \\ (g_j^2(x)/\lambda_j^2)e_j \end{pmatrix} & \begin{matrix} \text{if} \quad \lambda_j \geqslant -g_j(x) > 0, \\ \text{or} \quad 3\lambda_j > g_j(x) \geqslant 0, \end{matrix} \\ \begin{pmatrix} -(\lambda_j/g_j(x))^2\nabla g_j(x), \\ (3+2\lambda_j/g_j(x))e_j \end{pmatrix} & \begin{matrix} \text{if} \quad -g_j(x) > \lambda_j > 0, \\ \text{or} \quad -3g_j(x) > -\lambda_j \geqslant 0, \end{matrix} \\ \begin{pmatrix} -9\nabla g_j(x) \\ 9e_j \end{pmatrix} & \begin{matrix} \text{if} \quad 0 \geqslant -g_j(x) \text{ and } g_j(x) \leqslant -3\lambda_j, \\ \text{or} \quad -3g_j(x) \leqslant \lambda_j \leqslant 0, \end{matrix} \end{cases} \tag{6}$$

where $e_j = (0,\ldots,0,1,0,\ldots,0)^T \in R^m$ is the *jth* column of the unit matrix, its *jth* element is 1, and other elements are 0.

If $(g_j(x),\lambda_j) = (0,0)$ and $n+1 \leqslant i = j+n \leqslant n+m$, then $\Phi_i(x,\lambda)$ is strongly semismooth and directionally differentiable at $(x,\lambda)) \in R^{n+m}$. We have

$$A_\psi = \partial_B\psi(0,0) = \left\{ \begin{pmatrix} (-3-2t)\nabla g_j(x) \\ t^2 e_j \end{pmatrix} : -3 \leqslant t \leqslant 1 \right\} \cup \left\{ \begin{pmatrix} -t^2\nabla g_j(x) \\ (3+2t)e_j \end{pmatrix} : -3 \leqslant t \leqslant 1 \right\}, \tag{7}$$

2.2. The notion of filter

To avoid using the classical merit function with penalty term, in which the penalty parameter is difficult to obtain, we adopt the filter technique, which is proposed by Fletcher and Leyffer [12]. The acceptability of step is determined by comparing the constraint violation and objective function value with previous iterates collected in a filter. The new iterate is acceptable for the filter if either feasibility or the objective function value is sufficiently improved in comparison to all iterates bookmarked in the current filter. The promising numerical results lead to a growing interest in filter methods in recent years. In [12], they define the constraint violation by

$$h(x) = \|g(x)^+\|_\infty = \sum_{j=1}^m \max\{0, g_j(x)\}.$$

It is easy to see that $h(x) = 0$ if and only if x is a feasible point. So a trial point should reduce either the value of constraint violation h or the objective function f. To ensure sufficient decrease of at least one of the two criteria, we say that a point x^1 dominates a point x^2 whenever

$$h(x^1) \leqslant h(x^2) \text{ and } f(x^1) \leqslant f(x^2). \tag{8}$$

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

$$h(x^i) \leqslant h(x^j) \text{ or } f(x^i) \leqslant f(x^j), \tag{9}$$

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

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

$$h(x) \leqslant (1-\gamma)h^j \text{ or } f(x) \leqslant f^j - \gamma h^j, \tag{10}$$

for all $(h^j,f^j) \in \mathcal{F}$, where γ is close to zero. So, there is negligible difference in practice between (10) and (9). As the algorithm progresses, we may want to add a (h,f)-pair to the filter. If $x^k + d^k$ is acceptable for $\mathcal{F}$, then $x^{k+1} = x^k + d^k$, and

$$D_{k+1} = \left\{ (h^j,f^j) | h^j \geqslant h^k \text{ and } f^j - \gamma h^j \geqslant f^k - \gamma h^k, \ \forall (h^j,f^j) \in \mathcal{F} \right\}.$$

Filter set is update as the following rule

$$(\mathcal{F}_{k+1}) \quad \mathcal{F}_{k+1} = \mathcal{F}_k \bigcup \left\{ (h^{k+1},f^{k+1}) \right\} \setminus D_{k+1}. \tag{11}$$

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

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

$$h(x) \leqslant (1-\gamma)h^k \text{ and } f(x) \leqslant f^k - \gamma h^k \tag{12}$$

is also acceptable for the filter and x^k. Let

$$\Phi_1(x,\lambda) = (\phi_{n+1}(x,\lambda),\ldots,\phi_{n+m}(x,\lambda))^T. \tag{13}$$

Replacing the violation constrained function $h(x) = \sum_{j=1}^m \max\{0, g_j(x)\}$ in the filter F of Fletcher and Leyffer's method, we use the violation constrained function

$$p(g(x),\lambda) = \|\Phi_1(x,\lambda)\|_\infty. \tag{14}$$

For convenience, we use $\|\cdot\|$ to instead of $\|\cdot\|_\infty$ in this paper.

3. Description of algorithm

For the sake of simplicity, we denote

$$I_1(x,\lambda) = \{i|(g_i(x),\lambda_i) \neq (0,0)\}, \quad I_2(x,\lambda) = \{j|(g_j(x),\lambda_j) \neq (0,0)\}. \tag{15}$$

Let $(\xi_j(x,\lambda),\gamma_j(x,\lambda)) = (-1,1)$, if $j \in I_2$, otherwise let

$$(\xi_j(x,\lambda),\gamma_j(x,\lambda)) = \nabla\psi(a,b)|_{a=-g_j(x),b=\lambda_j}.$$

We have $\xi_j(x,\lambda) < 0, \gamma_j(x,\lambda)) > 0$,

$$\xi_j(x,\lambda) = \begin{cases} -(3+2g_j(x)/\lambda_j), \ \textit{if} \ \lambda_j \geqslant -g_j(x) > 0 \ \textit{or} \ 3\lambda_j > g_j(x) \geqslant 0, \\ -(\lambda_j/g_j(x))^2, \ \textit{if} \ -g_j(x) > \lambda_j > 0 \ \textit{or} \ -3g_j(x) > -\lambda_j \geqslant 0, \\ -9, \ \textit{if} \ 0 \geqslant -g_j(x) \ \textit{and} \ g_j(x) \geqslant 3\lambda_j \ \textit{or} \ 3g_j(x) \leqslant \lambda_j \leqslant 0 \end{cases} \tag{16}$$

and

$$\gamma_j(x,\lambda) = \begin{cases} (\lambda_j/g_j(x))^2, \ \textit{if} \ \lambda_j \geqslant -g_j(x) > 0 \ \textit{or} \ 3\lambda_j > g_j(x) \geqslant 0, \\ (3+2\lambda_j/g_j(x)), \ \textit{if} \ -g_j(x) > \lambda_j > 0 \ \textit{or} \ -3g_j(x) > -\lambda_j \geqslant 0, \\ 9, \ \textit{if} \ 0 \geqslant -g_j(x) \ \textit{and} \ g_j(x) \geqslant 3\lambda_j \ \textit{or} \ 3g_j(x) \leqslant \lambda_j \leqslant 0 \end{cases} \tag{17}$$

In the following algorithm, let $\xi_j^k = \xi_j(x^k,\lambda^k)$ and $\gamma_j^k = \gamma_j(x^k,\lambda^k)$, $\eta_j^k = \sqrt{2\gamma_j^k}$,

$$V^k = \begin{pmatrix} V_{11}^k & V_{12}^k \\ V_{21}^k & V_{22}^k \end{pmatrix} = \begin{pmatrix} H^k + c_1^k I_n & \nabla g_{L_k}(x^k) \\ diag_{L_k}(\xi^k)\nabla g_{L_k}(x^k)^T & diag_{L_k}(\eta^k) \end{pmatrix}, \tag{18}$$

where I_n is the n order unit matrix, $c_1^k = c_1 \min\left\{1, \|\overline{\Phi}^k\|^\nu\right\}$ $\nu > 1$, $c_1 \in (0,1)$, $\overline{\Phi}^k = \Phi(x^k,\bar{\lambda}^k)$, $\bar{\lambda}^k$ is obtained in Algorithm 3.1, $diag_{L_k}(\xi^k)$ or $diag_{L_k}(\eta^k)$ denotes the diagonal matrix whose jth diagonal element is $\xi_j(x,\lambda)$ or $\eta_j(x,\lambda)$, respectively.

Algorithm 3.1

Step 0. Initialization.
Parameters: ε_0, M_0, $\theta \in (0,1)$, $\tau \in (2,3)$, $\bar{\mu} > 0$, η, α_1, $\alpha_2 \in (0,1)$;
Data: $x^0 \in D$, $H^0 \in R^{n\times n}$, an initial symmetric positive definite matrix, $(\bar{\mu}, f(x^0)) \in F_0$.
Set $k = 0$;
Step 1. Computation of an approximate active constraints set L_k:
For the current point x^k and the parameter $\mu(x^k) = \left(\mu_j^k, j \in I\right) > 0$
1.1. Let $i = 0$, $\varepsilon_{k,i} = \varepsilon_0$, $M_{k,i} = M_0$;
1.2. Set

$$L_{k,i} = \left\{j \in I| -\varepsilon_{k,i}\mu_j^k \leqslant g_j(x^k) \leqslant 0\right\}, \tag{19}$$
$$A_{k,i} = (\nabla g_j(x^k), j \in L_{k,i}),$$
$$V_{k,i} = V\left(x^k, H^k, L_{k,i}\right).$$

If $A_{k,i}$ is of full rank and $\|(V_{k,i})^{-1}\| < M_{k,i}$, let $L_k = L_{k,i}$, $A_k = A_{k,i}$, $V_k = V_{k,i}$, $i_k = i$, and go to Step 2;
1.3. Set $i = i+1$, $\varepsilon_{k,i} = \frac{1}{2}\varepsilon_{k,i-1}$, $M_{k,i} = \frac{1}{2}M_{k,i-1}$, and go to Step 1.2 (inner loop A);
Step 2. Computation of the direction d_0^k:
Computation d_0^k and $\lambda_0^k = \left(\lambda_{0j}^k, j \in L_k\right)$ by solving the following linear system in (d,λ)

$$V_k\begin{pmatrix} d \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f(x^k) \\ 0 \end{pmatrix}. \tag{20}$$

If $d_0^k = 0,\ \lambda_0^k \geqslant 0$, STOP. If $d_0^k \neq 0$, let $J_k = L_k,\ \hat{d}^k = d_0^k$, and go to Step 4, otherwise, let $j_k \in L_k$, such that

$$\lambda_{0j_k}^k = \min\left\{\lambda_{0j}^k, j \in L_k\right\} < 0 \tag{21}$$

and set $J_k = L_k \backslash \{j_k\}$;

Step 3. Computation of the direction $\bar{d}_0^k$:

3.1. Compute $(\bar{d}_0^{k1}, \bar{\lambda}_0^{k1})$ by solving the following linear system in (d, λ):

$$\overline{V}\begin{pmatrix} d \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f(x^k) \\ diag_{k_k}\left(\xi^k\right)\left(\lambda_{0j_k}^k\right)^3 \end{pmatrix}, \tag{22}$$

where $\bar{L}_k = (\nabla g_j(x^k), j \in J_k),\ \overline{V} = V\left(x^k, H^k, \bar{L}_k\right)$;

3.2. Compute $(\bar{d}_0^{k2}, \bar{\lambda}_0^{k2})$ by solving the following linear system in (d, λ):

$$\overline{V}\begin{pmatrix} d \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f(x^k) \\ diag_{k_k}\left(\xi^k\right)\left(\lambda_{0j_k}^k\right)^3 - \|\bar{d}_0^{k1}\|^{\nu} diag_{L_k}\left(\xi^k\right) e_{L_k} \end{pmatrix}, \tag{23}$$

where $e_{L_k} = (1, \ldots, 1)^T \in R^{|L_k|}$;

3.3. Let

$$\begin{pmatrix} \bar{d}_0^k \\ \bar{\lambda}_0^k \end{pmatrix} = b^k \begin{pmatrix} \bar{d}_0^{k1} \\ \bar{\lambda}_0^{k1} \end{pmatrix} + \rho_k \begin{pmatrix} \bar{d}_0^{k2} \\ \bar{\lambda}_0^{k2} \end{pmatrix}, \tag{24}$$

where $b^k = 1 - \rho^k$ and $\rho^k = (\theta - 1)\frac{(\bar{d}_0^{k1})^T \nabla f^k}{1 + |\sum_{j=1}^{n} \lambda_{0j}^k| \|\bar{d}_0^k\|^{\nu}}$;

3.4. Set $\hat{d}^k = \bar{d}^k$;

Step 4. Computation of the high-order revised direction d_1^k:

4.1. 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 as well as denote $A_k = \begin{pmatrix} A_k^1 \\ A_k^2 \end{pmatrix}$.

4.2. Compute s_1^k by solving the following linear system in s

$$\left(A_k^1\right)^T s = -\psi_k e - \tilde{f}^k, \tag{25}$$

where

$$\begin{aligned} &\psi_k = \max\left\{\left\|\hat{d}^k\right\|^{\tau}, \max_{j \in L_k, \lambda_{0j}^k \neq 0}\left|\frac{\xi_i^k}{-\eta_j^k \lambda_{0j}^k} - 1\right|^{\theta}\left\|\hat{d}^k\right\|^2\right\}, \quad e = (1, \ldots, 1)^T \in R^{|L_k|}, \\ &\tilde{f}^k = \left(\tilde{f}_j^k, j \in L_k\right), \quad \tilde{f}_j^k = \begin{pmatrix} g_j\left(x^k + \hat{d}^k\right) & j \in J_k \\ 0 & j \in L_k \setminus J_k \end{pmatrix}. \end{aligned} \tag{26}$$

4.3. Denote $0 = (0, \ldots, 0)^T \in R^{n - |L_k|}$. Define d_1^k to be the vector formed by s_k and 0 such that

$$A_k^T d_1^k = \left(A_k^1\right)^T s_k + \left(A_k^2\right)^T 0 = \left(A_k^1\right)^T s_k \tag{27}$$

and set $d^k = \hat{d}^k + d_1^k$.

Step 5. Test to accept the trial step:

If $x^k + d^k$ is not acceptable for the filter.

If $\left\|\Phi_1\left(x^k, d^k\right)\right\| > \left\|d^k\right\| \min\left\{\eta, \alpha_1\left\|d^k\right\|^{\alpha_2}\right\}$, call Restoration Algorithm (Algorithm 3.2) to obtain $x_r^k = x^k + s_r^k$, and go to Step 2. Otherwise go to Step 6;

If $x^k + d^k$ is acceptable for the filter, let $x^{k+1} = x^k + d^k$, and add x^{k+1} to the filter, go to Step 9;

Step 6. Computation of the direction q^k:

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

$$\begin{aligned} &\rho^k = -\nabla f(x^k)^T d^k, \quad \pi^k = -\left(A_k^1\right)^{-1} \nabla f_1(x^k), \\ &\tilde{d}^k = \frac{-\rho_k\left(\left(A_k^1\right)^{-1}\right)^T e}{1 + 2|e^T \pi_k|}, \quad q^k = \rho^k(d^k + \bar{d}^k), \end{aligned} \tag{28}$$

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

Step 7. $\alpha_{k,0} = 1$, $l = 0$;

Step 8. If $x^k + \alpha_{k,l}q^k$ is not acceptable for the filter, then go to Step 8. Otherwise let $\alpha_k = \alpha_{k,l}$, $x^{k+1} = x^k + \alpha_k q^k$ and add x^{k+1} to the filter, go to Step 1;

Step 9. $\alpha_{k,l+1} = \alpha_{k,l}/2$, $l = l + 1$, go to Step 6 (inner loop B);

Step 10. Update:

Choose $H^{k+1} \in \sum$, $\sigma_{k+1} \in [\sigma_l, \sigma_r]$, $k = k+1$. Set $\bar{\lambda}^k = \min\left\{\lambda_0^{k-1}, \bar{\mu}e\right\}$, $\overline{\Phi} = \Phi(x^k, \bar{\lambda}^k)$. If $\left\|\Phi_1(x^k, d^k)\right\| > \left\|d^k\right\|\left\{\eta, \alpha_1\left\|d^k\right\|^{\alpha_2}\right\}$, call Restoration Algorithm (Algorithm 3.2) to obtain $x_r^k = x^k + s_r^k$, and go to Step 2. Or else, go to Step 1.

If $\left\|\Phi_1(x^k, d^k)\right\| > \left\|d^k\right\| \min\left\{\eta, \alpha_1\left\|d^k\right\|^{\alpha_2}\right\}$, we give the restoration algorithm (Algorithm 3.2) to compute the x_r^k such that $\left\|\Phi_1\left(x_r^k, \lambda_r^k\right)\right\| \leqslant \eta \min\left\{\left\|\Phi_1^k\right\|^I, \alpha_1\left\|d^k\right\|^{\theta}\right\}$, where $2 < \theta \leqslant 3$, $\left\|\Phi_1^k\right\|^I = \min\left\{p^i | p^i > 0, (p^i, f^i) \in F\right\}$.

In a restoration algorithm, it is therefore desired to decrease the value of $\|\Phi_1\|$. The direct way is utilized Newton method or the similar ways to attack $g(x+s)^+ = 0$. We now give the restoration algorithm.

Algorithm 3.2

Step 1. Let $x_0^k = x^k$, $\Delta_0^k = \sigma^k$, $j = 0$, η, $\bar{\eta} \in (0,1)$, $2 < \theta \leqslant 3$;

Step 2. If $\left\|\Phi_1\left(x_j^k, \lambda_j^k\right)\right\| \leqslant \eta \min\left\{\left\|\Phi_1^k\right\|^I, \alpha_1\left\|d^k\right\|^{\theta}\right\}$, then let $x_r^k = x_j^k$, STOP;

Step 3. Compute

$$\begin{aligned} \min \quad & \left\|\Phi_1\left(x_j^k, \lambda_j^k\right)\right\| - \left\|\Psi_1\left(-g_j^k - A_j^k d, \lambda_j^k\right)\right\|, \\ \text{s.t.} \quad & \|d\| \leqslant \Delta_j^k, \end{aligned} \tag{29}$$

to get s_j^k, where

$$\Psi_1(-g(x), \lambda) = (\psi(-g_{n+1}(x), \lambda_{n+1}), \ldots, \psi(-g_{n+m}(x), \lambda_{n+m})).$$

Let $r_j^k = \dfrac{\left\|\Phi_1\left(x_j^k, \lambda_j^k\right)\right\| - \left\|\Phi_1\left(x_j^k + d, \lambda_j^k\right)\right\|}{\left\|\Phi_1\left(x_j^k, \lambda_j^k\right)\right\| - \left\|\Psi_1\left(-g_j^k - A_j^k d, \lambda_j^k\right)\right\|}$;

Step 4. If $r_j^k \leqslant \bar{\eta}$, then let $x_{j+1}^k = x_j^k$, $\Delta_{j+1}^k = \frac{1}{2}\Delta_j^k$, $j = j+1$ and go to Step 3. Otherwise, let $x_{j+1}^k = x_j^k + s_j^k$, $\Delta_{j+1}^k = 2\Delta_j^k$, get A_{j+1}^k, $j = j+1$ and go to Step 2.

The above restoration algorithm is a Newton method for $\|\Phi_1\| = 0$. This method is utilized frequently [13]. Of course, there are other restoration algorithm, such as interior point restoration algorithm, SLP restoration algorithm and so on.

4. Global convergence of algorithm

Assumptions

A1: The set $\overline{D}$ is bounded.

A2: The strictly feasible set D is nonempty. The level set $S = \{x | f(x) \leqslant f(x_0) \text{ and } x \in \overline{D}\}$ is bounded.

A3: f and g_i, $(i = 1, \ldots, m)$ are Lipschitz continuously differentiable and for all $y, z \in R^{n+m}$, $\|L(y) - L(z)\| \leqslant c_2\|y - z\|$.

A4: H_k is positive definite and there exists a positive number m_1 such that $0 < d^T H_k d \leqslant m_1\|d\|^2$ for all $d \in R^n$, $d \neq 0$.

Lemma 4.1. *If $\overline{\Phi}^k \neq 0$, then V^k is nonsingular. Furthermore, assume that (x^*, λ^*) is an accumulation point of $\{(x^k, \lambda^k)\}$, $(x^k, \lambda^k) \to (x^*, \lambda^*)$, $\overline{\Phi}^k \to \Phi^*$ and $V^k \to V^*$. If $\Phi^* \neq 0$, then $\|(V^k)^{-1}\|$ is bounded and V^* is nonsingular.*

Proof. If $V^k \begin{pmatrix} u \\ v \end{pmatrix} = 0$ for some $\begin{pmatrix} u \\ v \end{pmatrix} \in R^{n+|L_k|}$, where $u = (u_1, \ldots, u_n)^T$, $v = (v_1, \ldots, v_{|L_k|})^T$, then we have

$$\left(H^k + c_1^k I_n\right)u + \nabla g_{L_k}(x^k)v = 0 \tag{30}$$

and

$$diag_{L_k}(\xi^k)\nabla g_{L_k}(x^k)^T u + diag_{L_k}(\eta^k)v = 0. \tag{31}$$

Assume $\overline{\Phi}^k \neq 0$, obviously we have $c_1^k \neq 0$. From the definition of ξ_j^k and η_j^k, we have that $\xi_j^k < 0$ and $\eta_j^k > 0$, $j = 1, 2, \ldots, m$. Thus, $diag_{L_k}(\eta^k)$ is nonsingular. We have

$$v = -\left(diag_{L_k}(\eta^k)\right)^T diag_{L_k}\left(\xi^k\right)\nabla g_{L_k}(x^k)^T u. \tag{32}$$

Putting (32) into (30), we have

$$u^T\left(H^k + c_1^k I_n\right)u - u^T\nabla g_{L_k}(x^k) diag_{L_k}\left(\xi^k\right)(diag_{L_k}(\eta^k))^{-1}\left(\nabla g_{L_k}(x^k)\right)^T u = 0. \tag{33}$$

$u^T\left(H^k + c_1^k I_n\right)u = 0$ and $u = 0$ are implied by the fact that $H^k + c_1^k I_n$ is positive definite and $-\nabla g_{L_k}(x^k) diag_{L_k}\left(\xi^k\right)\left(diag_{L_k}(\eta^k)\right)^{-1}\nabla g_{L_k}(x^k)^T$ is positive semi-definite, then $v = 0$ by (31). The first part of this lemma holds.

On the other hand, without loss of generality we may assume that $c_1^{k(i)} \to c_1^* \neq 0$, $diag_{L_k}\left(\xi^{k(i)}\right) \to diag_{L_k}(\xi^*)$, $diag_{L_k}(\eta^{k(i)}) \to diag_{L_k}(\eta^*)$ and $H^{k(i)} \to H^*$. We know that $\eta_j^* > 0$ for all $j = 1,2,\ldots,m$. $H^{k(i)} \to H^*$ imply that H^* is positive semi-definite. By replacing index k by $*$ in the above proof, it is easy to check that V^* is nonsingular. Assumption $V^k \to V^*$ imply that $\|(V^k)^{-1}\|$ is bounded. This lemma holds. □

From Lemma 4.1 we have $\|(V^k)^{-1}\|$ is also uniformly bounded. It is then not difficult to see from Step 1 of Algorithm 3.1 that the inner loop A terminates in finite number of times, i.e. the parameter $\varepsilon_{k,i}$ will return to be fixed after finitely many iterations and $M_{k,i}$ will also be constant after many iterations.

If $\Phi(x^k, \lambda^k) = 0$, then (x^k, λ^k) is a KKT point of Problem (NLP). Without loss of generality, in the sequel, we may assume that $\Phi(x^k, \lambda^k) \neq 0$ for all k.

Because V^k is nonsingular, (20) (21) or (24) always has unique solution.

V^k is nonsingular, so $B^k = (V^k)^{-1}$ exist. Let

$$B^k = \begin{pmatrix} H^k + c_1^k I_n & \nabla g_{L_k}(x^k) \\ diag_{L_k}\left(\xi^k\right)\nabla g_{L_k}(x^k)^T & diag_{L_k}(\eta^k) \end{pmatrix}^{-1} = \begin{pmatrix} B_{11}^k & B_{12}^k \\ B_{21}^k & B_{22}^k \end{pmatrix}. \tag{34}$$

By calculating directly, we have

$$B_{11}^k = \left(H^k + c_1^k I_n\right)^{-1} + \left(H^k + c_1^k I_n\right)^{-1}\nabla g_{L_k}(x^k)(Q^k)^{-1} \cdot diag_{L_k}\left(\xi^k\right)\nabla g_{L_k}(x^k)^T\left(H^k + c_1^k I_n\right)^{-1} \tag{35}$$

$$B_{12}^k = -\left(H^k + c_1^k I_n\right)^{-1}\nabla g_{L_k}(x^k)(Q^k)^{-1} \tag{36}$$

$$B_{21}^k = -(Q^k)^{-1} diag_{L_k}\left(\xi^k\right)\nabla g_{L_k}(x^k)^{-1}\left(H^k + c_1^k I_n\right)^{-1} \tag{37}$$

$$B_{22}^k = (Q^k)^{-1} \tag{38}$$

where $Q^k = diag_{L_k}(\eta^k) - diag_{L_k}\left(\xi^k\right)\nabla g_{L_k}(x^k)^T\left(H^k + c_1^k I_n\right)^{-1}\nabla g_{L_k}(x^k)$.

Lemma 4.2. *If $\Phi^k \neq 0$, then $d_0^k = 0$ if and only if $\nabla f(x^k) = 0$, and $d_0^k = 0$ implies $\lambda_0^k = 0$ and (x^k, λ_0^k) is a KKT point of the Problem (NLP).*

Proof. If $\nabla f(x^k) = 0$, then $d_0^k = 0$ and $\lambda_0^k = 0$ by (20). If $d_0^k = 0$, then (20) implies $\nabla g_{L_k}(x^k)\lambda_{0j_k}^k = -\nabla f(x^k)$ and $diag_{L_k}(\eta^k)\lambda_{0j_k}^k = 0$. From the definition of η_j^k, we have $\eta_j^k > 0$, $j = 1,2,\ldots,m$. Thus, $diag_{L_k}(\eta^k)$ is nonsingular. So, $\lambda_0^k = 0$ and $\nabla f(x^k) = 0$. □

Lemma 4.3. *If $d_0^k \neq 0$, then*

1. $c_1^k\left\|d_0^k\right\|^2 \leqslant \left(d_0^k\right)^T\left(H^k + c_1^k I_n\right)d_0^k \leqslant -\left(d_0^k\right)^T\nabla f(x^k)$;
2. $(\bar{d}_0^{k1})^T\nabla f(x^k) = \left(d_0^k\right)^T\nabla f(x^k) - \sum_{j:\lambda_{0j_k}^k<0}\left(\lambda_{0j_k}^k\right)^4$;
3. $(\bar{d}_0^k)^T\nabla f(x^k) \leqslant \theta(\bar{d}_0^{k1})^T\nabla f(x^k)$;
4. $\nabla f(x^k)^T q^k \leqslant -\frac{1}{2}(\rho^k)^2 < 0$.

Proof. (20) implies

$$\left(H^k + c_1^k I_n\right)d_0^k + \nabla g_{L_k}(x^k)\lambda_0^k = -\nabla f(x^k) \tag{39}$$

and

$$diag_{L_k}\left(\xi^k\right)\left(\nabla g_{L_k}(x^k)\right)^T d_0^k + diag_{L_k}(\eta^k)\lambda_0^k = 0 \tag{40}$$

we have

$$\lambda_0^k = -\left(diag_{L_k}(\eta^k)\right)^{-1} diag_{L_k}\left(\xi^k\right)\left(\nabla g_{L_k}(x^k)\right)^T d_0^k \tag{41}$$

Putting (41) into (39), we have

$$\begin{aligned}-\left(d_0^k\right)^T\nabla f(x^k) &= \left(d_0^k\right)^T\left(\left(H^k + c_1^k I_n\right) + \nabla g_{L_k}(x^k)\lambda_0^k\right)\\ &= (d_0^k)^T\left(H^k + c_1^k I_n\right)d_0^k - \left(d_0^k\right)^T\nabla g_{L_k}(x^k)\cdot\left(diag_{L_k}(\eta^k)\right)^{-1} diag_{L_k}\left(\xi^k\right)\left(\nabla g_{L_k}(x^k)\right)^T d_0^k\end{aligned} \tag{42}$$

$\left(d_0^k\right)^T\nabla g_{L_k}(x^k)\left(diag_{L_k}(\eta^k)\right)^{-1} diag_{L_k}(\xi^k)\left(\nabla g_{L_k}(x^k)\right)^T d_0^k \leqslant 0$ implies

$$c_1^k\left\|d_0^k\right\|^2 \leqslant \left(d_0^k\right)^T\left(H^k + c_1^k I_n\right)d_0^k \leqslant -\left(d_0^k\right)^T\nabla f(x^k) \tag{43}$$

The first part of the lemma holds. (20) and (34) imply

$$\left(d_0^k\right)^T = -B_{11}^k\nabla f(x^k), \quad \lambda_0^k = -B_{21}^k\nabla f(x^k) \tag{44}$$

The property of the matrix implies

$$\begin{aligned}(Q^k)^{-1} diag_{L_k}(\xi^k) &= \left(\left(diag_{L_k}(\xi^k)\right)^{-1} Q^k\right)^{-1}\\ &= \left\{\left(diag_{L_k}(\xi^k)\right)^{-1}\left[diag_{L_k}(\eta^k) - diag_{L_k}(\xi^k)\cdot(\nabla g_{L_k}(x^k))^T\left(H^k + c_1^k I_n\right)^{-1}\nabla g_{L_k}(x^k)\right]\right\}^{-1}\\ &= \left\{\left[diag_{L_k}(\eta^k) - \left(\nabla g_{L_k}(x^k)\right)^T\left(H^k + c_1^k I_n\right)^{-1}\cdot\nabla g_{L_k} diag_{L_k}(\xi^k)\right]\left(diag_{L_k}(\xi^k)\right)^{-1}\right\}^{-1}\\ &= \left[(Q^k)^T\left(diag_{L_k}(\xi^k)\right)^{-1}\right]^{-1} = diag_{L_k}(\xi^k)\left((Q^k)^T\right)^{-1}\end{aligned} \tag{45}$$

(21), (34) and (45) imply $B_{12}^k diag_{L_k}(\xi^k) = \left(B_{21}^k\right)^T$ and

$$(\bar{d}_0^{k1})^T\nabla f(x^k) = -\left(B_{11}^k\nabla f(x^k)\right)^T\nabla f(x^k) - \left[\left(B_{12}^k\right)^T diag_{L_k}(\xi^k)\right]^T\nabla f(x^k)\left(\lambda_0^k\right)^3 = \left(d_0^k\right)^T\nabla f(x^k) - \sum_{j:\lambda_{0j_k}^k<0}\left(\lambda_{0j_k}^k\right)^4. \tag{46}$$

The second part of this lemma holds. (24)–(26) and (46) imply

$$(\bar{d}_0^{k2} - \bar{d}_0^{k1})^T\nabla f(x^k) = \|\bar{d}_0^{k1}\|^{\upsilon}\left[B_{12}^k diag_{L_k}(\xi^k)e_{L_k}\right]^T\nabla f(x^k) = \|\bar{d}_0^{k1}\|^{\upsilon}\sum_{j=1}^m\lambda_{0j_k}^k$$

and

$$(\bar{d}_0^k)^T\nabla f(x^k) = (1-\rho_k)(\bar{d}_0^{k1})^T\nabla f(x^k) + \rho_k(\bar{d}_0^{k2})^T\nabla f(x^k) \leqslant \theta(\bar{d}_0^{k1})^T\nabla f(x^k). \tag{47}$$

The third part of this lemma holds. Finally, from (28), we obtain

$$\begin{aligned}\nabla f(x^k)^T q^k &= \rho^k\nabla f(x^k)^T(d^k + \bar{d}^k) = \rho^k(\nabla f(x^k)^T d^k + \nabla f(x^k)^T\bar{d}^k) = \rho^k(-\rho^k + \nabla f_1(x^k)^T\tilde{d}^k) = \rho^k\left(-\rho^k + \frac{\rho^k(\pi^k)^T e}{1+2|e^T\pi^k|}\right)\\ &\leqslant -\frac{1}{2}(\rho^k)^2 < 0\end{aligned} \tag{48}$$

This lemma holds. □

Lemma 4.4. *The inner loop B terminates in finite number of times.*

Proof. By contradiction, if the conclusion is false, then the Algorithm 3.1 will run infinitely between Step 8 and Step 9, so we have

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

and $x^k + \alpha_{k,l}q^k$ is not acceptable for the filter.

since x^k is acceptable for the filter, we have

$$\|\Phi_1(x^k,\mu^k)\| \leqslant \theta\left\|\Phi_1^l\right\| \text{ or } f(x^k)-f(x^l) \leqslant -\alpha_{k-1}\theta\left\|\Phi_1^k\right\| \ \forall\left(f^l,\left\|\Phi_1^l\right\|\right)\in F^k.$$

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

$$\left\|\Phi_1\left(x^k+\alpha_{k,l}q^k,\mu^k+\alpha_{k,l}^2\lambda^k\right)\right\| > \theta\left\|\Phi_1^l\right\| \tag{49}$$

and

$$f(x^k+\alpha_{k,l}q^k)-f(x^l) > -\alpha_k\theta\left\|\Phi_1(x^k+\alpha_{k,l}q^k)\right\|. \tag{50}$$

For the point x^k, if it holds that $\|\Phi_1(x^k,\mu^k)\| \leqslant \theta\left\|\Phi_1^l\right\|$, then by $\alpha_{k,l}\to 0$,

$$\left\|\Phi_1\left(x^k+\alpha_{k,l}q^k,\mu^k+\alpha_{k,l}^2\lambda^k\right)\right\| \leqslant \theta\left\|\Phi_1^l\right\| \tag{51}$$

which contradicts (49).

If it holds $f(x^k)-f(x^l) \leqslant -\alpha_{k-1}\theta\left\|\Phi_1^k\right\|$, then by $\alpha_{k,l}\to 0$, we get

$$\begin{aligned} f(x^k+\alpha_{k,l}q^k) &= f(x^k)+\alpha_{k,l}\nabla f_1(x^k)^Tq^k+O\left(\left\|\alpha_{k,l}q^k\right\|^2\right) \\ &\leqslant f(x^k) \leqslant f(x^l)-\alpha_{k,l}\theta\left\|\Phi_1^k\right\| \end{aligned} \tag{52}$$

which contradicts (50).

Based on the above analysis, this lemma holds. □

Lemma 4.4 means that there exists a constant $\bar{\alpha} > 0$, such that $\alpha_k \geqslant \bar{\alpha}$ for large enough k.

By the above statement, we see that Algorithm 3.1 is implementable. Now we turn to prove the global convergence of Algorithm 3.1. We assume that assumptions A1-A4 holds.

Lemma 4.5. *Assume $x^k \to x^*$ and $\Phi^k > \varepsilon > 0$ for some ε, then the sequence of $\left\{d_0^k,\lambda_0^k\right\}$, $\{\bar{d}_0^{k1},\bar{\lambda}_0^{k1}\}$ and $\{\bar{d}_0^{k2},\bar{\lambda}_0^{k2}\}$ are all bounded on $k = 0,1,\ldots$*

Proof. If $x^k \to x^*$ and $\Phi^k > \varepsilon > 0$ for some ε, then the matrix sequence $\{(V^k)^{-1}\}$ is uniformly bounded form Lemma 4.1. $\{x^k\}$ is bounded due to the assumption A3. The solubility of system (20) implies that $\left\{d_0^k,\lambda_0^k\right\}$ is bounded, which implies the boundedness of $\{\bar{d}_0^{k1}\}$ of the right-hand side of (21). Hence $\{\bar{d}_0^{k1},\bar{\lambda}_0^{k1}\}$ is also bounded. Finally, the boundedness of $\{\bar{d}_0^{k1},\bar{\lambda}_0^{k1}\}$ implies the boundedness of the right-hand side of (22). Hence $\{\bar{d}_0^{k2},\bar{\lambda}_0^{k2}\}$ is also bounded. □

Lemma 4.6. *Assume $x^k \to x^*$ and $\overline{\Phi}^k > \varepsilon > 0$ for some ε. There is a $c_3 > 0$ such that, for all $k = 1,2,\ldots$,*

$$\|\bar{d}_0^k-\bar{d}_0^{k1}\| \leqslant c_3\left\|d_0^k\right\|.$$

Proof. It is from the Lemma 4.1 that there exists a $c_3 > 0$ such that,for all $k = 0,1,\ldots,\ c_3 \geqslant 2m\rho^k\|(V^k)^{-1}\|$.

Let $\Delta d^k = \bar{d}_0^k-\bar{d}_0^{k1}$ and $\Delta\lambda^k = \bar{\lambda}_0^k-\bar{\lambda}_0^{k1}$. Then by (21)–(25), $(\Delta d^k,\Delta\lambda)$ is the solution of

$$V\begin{pmatrix}\Delta d^k\\ \Delta\lambda^k\end{pmatrix} = \begin{pmatrix}0\\ -\rho^k\left\|d_0^k\right\|^{\nu}diag_{L_k}(\xi^k)e_{L_k}\end{pmatrix}. \tag{53}$$

It is easy to see that

$$\left\|(\Delta d^k,\Delta\lambda^k)\right\| \leqslant c_3\left\|d_0^k\right\|^{\nu},\quad \left\|\Delta d^k\right\| \leqslant c_3\left\|d_0^k\right\|^{\nu}$$

the lemma holds. □

Lemma 4.7. *Assume $x^k \to x^*$, and $\lambda^k \to \lambda^*$,*

1. *If $\bar{d}_0^k \to 0$, then $\lambda_j^* \geqslant 0$ for any $1 \leqslant j \leqslant m$;*
2. *If $d_0^k \to 0$, then x^* is a KKT point of the Problem (NLP);*
3. *If $\bar{d}_0^k \to 0$ and $\overline{\Phi}^k > \varepsilon > 0$ for some ε, then x^* is a KKT point of the Problem (NLP).*

Proof. It follows from the Lemma 4.3 that

$$(\bar{d}_0^k)^T \nabla f(x^k) \leqslant -c_1^k \theta \left\| d_0^k \right\|^2 - \theta \sum_{j:\lambda_{0j_k}^k<0} \left(\lambda_{0j_k}^k\right)^4. \tag{54}$$

Hence $\{\bar{d}_0^k\} \to 0$ implies that

$$\sum_{j:\lambda_{0j_k}^k<0} \left(\lambda_{0j_k}^k\right)^4 \to 0 \text{ and } \lambda_j^* \geqslant 0, \quad 1 \leqslant j \leqslant m.$$

The first part of this lemma holds.

Because $\bar{\lambda}^k \leqslant \bar{\mu}$ and $\{\lambda^k\}$ are bounded, there is an accumulation point λ^* of $\{\bar{\lambda}^k\}$. Without loss of generality we assume that $c^k \to c^*,\ \lambda^k \to \lambda^*,\ \bar{\lambda}^k \to \bar{\lambda}^*$ and $L_k \to L^*$. (54) implies that, for any accumulation point λ^* of $\{\lambda^k\},\ \lambda_i^* \geqslant 0,\ 1 \leqslant i \leqslant m$. Taking the limitations in both side of (20), by noting $d_0^k \to 0$, we obtain $\lambda_{L_*}^{*\,T} \nabla g_{L_*}(x^*) = -\nabla f(x^*)$ and $diag_{L_*}(\eta^*)\lambda_{L_*}^* = 0$. If $-g_i(x^*) > 0$, for some $1 \leqslant i \leqslant m$, then $-\eta_i^* \geqslant \delta > 0$ and $\lambda_i^* = 0$, that is, for any $1 \leqslant i \leqslant m$, $g_i(x^*)\lambda_i^* = 0$. The second part of this lemma holds.

If $\bar{d}_0^k \to 0$, and $\overline{\Phi}^k > \varepsilon > 0$ for some ε, then (54) implies $d_0^k \to 0$. So, x^* is a KKT point of the Problem (NLP). This lemma holds. □

Lemma 4.8. *The Restoration Algorithm terminates in a finite number of iteration.*

Proof. It is similar to lemma 1 in [13]. □

By the above statement, we see that Algorithm 3.1 is implementable. Now, we turn to prove the global convergence of Algorithm 3.1.

Lemma 4.9. *Suppose that infinite points are added to the filter, then* $\lim_{k\to\infty,k\in K} p^k = 0$*, where K is an infinite set.*

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

$$p^k \geqslant \varepsilon > 0.$$

At each iteration k, (p^k,f^k) is added to the filter. By (11), we can deduce that (p,f)-pair be added to the filter at a large stage within the square

$$[p^k - \gamma\varepsilon, p^k] \times [f^k - \gamma\varepsilon, f^k],$$

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

$$p^{k+1} \leqslant (1-\gamma)p^k \leqslant p^k - \gamma\varepsilon.$$

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

Lemma 4.10 [11]. *Assume $x^k \to x^*$ and $\overline{\Phi}^k > \varepsilon > 0$ for some ε. If $d^k \to 0$, then (x^*,λ^*) is a KKT point of the Problem (NLP), where λ^* is an accumulation point of $\{\lambda^k\}$.*

Lemma 4.11 [11]. *Assume $x^k \to x^*$ and $\overline{\Phi}^k > \varepsilon > 0$ for some ε. If liminf $\{\|(d^k)^{-1}\|\} > 0$, then (x^*,λ^*) is a KKT point of the Problem (NLP), where λ^* is an accumulation point of $\{\lambda^k\}$. From above lemmas, the following global convergence theorem holds.*

Theorem 4.1. *If x^* is a limit point of $\{x^k\}$, (x^*,λ^*) is a KKT point of the Problem (NLP).*

5. Superlinear convergence of algorithm

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

Assumptions

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

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

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

where $L(x,\lambda) = f(x) + \sum_{j=1}^m \lambda_j g_j(x),\ I(x^*) = \{j|g_j(x^*) = 0\}$.
B3: At x^*, strict complementarity slackness and linear independence of the gradients of the active constraints hold.
B4: Matrices H^k, $k = 1,2,\ldots$ are symmetric positive definite and satisfy the following condition

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

Lemma 5.1. *It holds, for $k \to \infty$, that*

$$L_k \equiv I(x^*) = I_*, \quad d^k \to 0, \quad \lambda^k \to \left(\lambda_j^*, j \in I_*\right).$$

Proof. By Lemma 4.9, $H^k \to H^*$, it holds that $d^k \to 0$ as $k \to \infty$. According to Lemma 4.1, it follows that $I^* \subset L \equiv L_k$. First, we prove that

$$\lambda^k \to (\lambda_j^*, j \in L),$$

since x^* is the KKT point of Problem (NLP), we have

$$\nabla f(x^*) + A_*\lambda_L^* = 0, \quad \lambda_L^* \geqslant 0, \quad \lambda_j^* = 0 \quad j \in I \setminus L,$$

where $\lambda_L^* = (\lambda_j^*, j \in L),\ A_* = (\nabla g_j(x^*), j \in L)$.
From Lemma 4.1, it following that

$$A_*^T A_* \text{ is nonsingular, and } (A_k^T A_k)^{-1} \to (A_*^T A_*)^{-1}.$$

So $\lambda_L^* = -(A_*^T A_*)^{-1} A_*^T \nabla f(x^*)$
Moreover, by KKT condition of Problem (NLP), we have

$$\nabla f(x^k) + H^k d^k + A_k \lambda^k = 0.$$

Hence, $\lambda^k = -\left(A_k^T A_k\right)^{-1} A_k^T(\nabla f(x^k) + H^k d^k) \to -\left(A_*^T A_*\right)^{-1} A_*^T \nabla f(x^*) = \lambda_L^*$.
Second, we prove that $L \subset I^*$.
For $j_0 \bar{\in} L$, if $j_0 \bar{\in} I_*$ by contradiction, there must be a constant $\xi_0 > 0$ such that $g_{j_0}(x^*) \leqslant -\xi_0 < 0$. Again, since $g_{j_0}(x)$ is continuously differentiable, and $d^k \to 0 (k \to \infty)$, we have for k large enough

$$g_{j_0}(x^*) + \nabla g_{j_0}(x^*)^T d^k \leqslant -\frac{\xi_0}{2} < 0,$$

which means $j_0 \bar{\in} L$, contradicts the above assumption. Hence $L \equiv L_k \equiv I^*$. □

Lemma 5.2. *Suppose A1–A4, B1–B4 hold, then $x^{k+1} = x^k + d^k$ for k sufficiently large.*

Proof. Suppose x^k is acceptable for the filter, we will show that for k sufficiently large, $x^k + d^k$ is acceptable for the filter. From Lemma 4.9 and Lemma 5.1, we know that $d^k \to 0$, $\left\|\Phi_1^k\right\| \to 0$ as $k \to \infty$. Also, by the construction of Algorithm 3.1, we have $\|\Phi_1(x^k)\| = o(\|d^k\|^2)$. So, we just need to show that $f(x^k + d^k) \leqslant f(x^k) + \gamma\|\Phi_1(x^k)\|$. Let $\delta^k = f(x^k + d^k) - f(x^k) - \gamma\|\Phi_1(x^k)\|$, we have

$$\delta^k = \nabla f(x^k)^T d^k + \frac{1}{2} d_k^T \nabla^2 f(x^k) d^k + o(\|d^k\|^2).$$

While by the KKT condition of Problem (NLP) and $\|\Phi_1^k\| \to 0$, we have

$$\nabla f(x^k)^T d^k = -(d^k)^T H^k d^k - \sum_{j=1}^m \lambda_j^k \nabla g_j(x^k)^T d^k,$$

$$g_j(x^k) + \nabla g_j(x^k)^T d^k + \frac{1}{2}(d^k)^T \nabla^2 g_j(x^k) d^k = o(\|d^k\|^2).$$

Then it holds

$$\delta^k = -(d^k)^T H^k d^k - \sum_{j=1}^{m} \lambda_j^k \nabla g_j(x^k) + \frac{1}{2}(d^k)^T \nabla^2 L\left(x^k, \lambda_0^k\right) d^k + o(\|d^k\|^2)$$
$$= -\frac{1}{2}(d^k)^T H^k d^k + \sum_{j=1}^{m} \lambda_j^k g_j(x^k) + \frac{1}{2}(d^k)^T (\nabla^2 L\left(x^k, \lambda_0^k\right) - H^k) d^k + o(\|d^k\|^2). \quad (55)$$

According to $\lambda_k \to \lambda_j^* > 0,\ g_j(x^k) \to g_j(x^*) < 0,\ j \in I_*$ and Assumption A4, we have

$$\delta_k \leqslant -\frac{a}{2}\|d^k\|^2 + \frac{1}{2}(d^k)^T (\nabla_{xx}^2 L\left(x^k, \lambda_0^k\right) - \nabla_{xx}^2 L(x^*, \lambda^*)) d_k + \frac{1}{2}(d^k)^T (\nabla_{xx}^2 L(x^*, \lambda^*) - H^k) d^k + o(\|d^k\|^2). \quad (56)$$

Since $x^k \to x^*,\ \lambda_0^k \to \lambda^*$, then

$$(d^k)^T \left(\nabla_{xx}^2 L(x^*, \lambda^*) - H^k\right) d^k = o(\|d^k\|^2).$$

Therefore, while k is sufficiently large, it holds

$$\delta^k \leqslant -\frac{a}{2}\|d^k\|^2 + o(\|d^k\|^2) \leqslant 0.$$

Hence, for all k large enough, $x^k + d^k$ is acceptable for the filter. □

In view of Lemma 4.2, assumption B4 and the way of Theorem 3.1 in [14], it is easy to get the convergence theorem as follows.

Theorem 5.1. *Under all stated assumptions, the algorithm is superlinearly convergent, i.e., the sequence $\{x^k\}$ generated by the algorithm satisfies $\|x^{k+1} - x^*\| = o(\|x^k - x^*\|)$.*

6. Numerical tests

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

(1) Updating of H^k is done by

$$H^{k+1} = \begin{cases} H^k, & \text{if} \quad s_k^T y_k \leqslant 0, \\ H^k + \frac{y_k^T y_k}{y_k^T s_k} - \frac{H^k s_k s_k^T H^k}{s_k^T H^k s_k}, & \text{if} \quad s_k^T y_k > 0, \end{cases} \quad (57)$$

(2) The stop criteria is $\|d^k\|$ sufficiently small;
(3) If an equality constraint $g(x) = 0$ exists in the original problem, it is most easily handled as two corresponding inequalities $g(x) \leqslant 0$ and $g(x) \geqslant 0$, and we can apply the above algorithm.

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

Table 1
Numerical results for Algorithm 3.1.

Problem	x^0	IT	$\|\Phi_1\|$	FV
4	1.125, 0.125	3	1.0e−08	2.6327e+00
5	0, 0	6	6.14e−06	−1.9751e+00
9	0, 0	4	1.72e−06	−0.5012e+00
11	4.9, 0.1	8	5.84e−06	−8.4985e+00
12	0, 0	12	4.07e−06	1.0054e+00
24	1, 0.5	5	3.19e−06	−1.0000e+00
26	−2.6, 2, 2	15	2.33e−06	0.0000e+00
28	−4, 1, 1	5	7.22e−06	0.0000e+00
29	1, 1, 1	10	1.11e−07	−22.6274e+00
30	1, 1, 1	16	9.01e−06	1.0000e+00
33	0, 0, 3	5	5.62e−06	−4.5876e+00
34	0, 1.05, 2.9	7	2.18e−06	−0.8342e+00
35	0, 1.05, 2.9	19	1.19e−06	1.1082e−01
41	2, 2, 2, 2	5	2.33e−06	1.9259e+00
44	0, 0, 0, 0	6	1.06e−06	−15.0000e+00
51	2.5, 0.5, 2, −1, 0.5	5	4.25e−06	0.0000e+00
66	0, 1.05, 2.9	6	8.47e−06	0.5166e+00
71	1, 5, 5, 1	4	5.85e−07	17.0140e+00

Problem: the number of problems in [15], x^0: the starting vector, IT: the number of iterations, $\|\Phi_1\|$: the value of $\|\Phi_1(\cdot)\|$ at the final iterate (x^k, λ^k), FV: the objective function value at the final iteration.

We can see that the numerical results indicate that this method is quite promising.

7. Conclusions

The proposed algorithm combines a QP-free method with a 3-piecewise linear NCP function to globalize the process. Each step is obtained only through systems of linear equations, and a higher order step is computed in order to overcome the Maratos effect. The algorithm makes use of filter technique so that the computational cost is decreased largely. The convergent results and the preliminary numerical tests in this paper shows that the method is interesting and of significance. However, to prove the superlinearly convergence of our algorithm, we suppose some rigorous conditions such as the strict complementarity condition and so on. We hope that we can get rid of them in our future work.

Acknowledgement

We are deeply indebted to the editors and the anonymous referees whose insightful comments and suggestions helped us a lot to improve the quality of the paper.

References

[1] P.T. Boggs, J.W. Tolle, P. Wang, On the local convergence of qusi-newton methods for constrained optimization, SIAM J. Control Optim. 20 (1982) 161–171.
[2] J.F. Bonnons, E.R. Painer, A.L. Titts, J.L. Zhou, Avoiding the Maratos effect by means of nonmontone linesearch Inequality constrained problems-feasible iterates, SIAM J. Numer Anal. 29 (1922) 1187–1202.
[3] F. Facchinei, S. Lucidi, Quadraticly and superlinearly convergent for the solution of inequality constrained optimization problem, J. Optim. Theory Appl. 85 (1995) 265–289.
[4] R. Fletcher, S. Leyffer, Nonlinear programming without a penalty function, Math. Program. 91 (2002) 239–269.
[5] S.P. Han, Superlinearly convergence variable metric algorithm for general nonlinear programming problems, Math. Program. 11 (1976) 263–282.
[6] M.J.D. Powell, A fast algorithm for nonlinear constrained optimization calculations, in: G.A. Waston (Ed.), Numerical Analysis, Springer-Verlag, Berlin, 1978, pp. 144–157.
[7] M.J.D. Powell, Variable metric methods for constrained optimization, in: A. Bachen et al. (Eds.), Mathematical Programming-The state of Art, Springer-Verlag, Berlin, 1982.
[8] Z.B. Zhu, K.C. Zhang, J.B. Jian, An improved SQP algorithm for inequality constrained optimization, Math. Methods Oper. Res. 58 (2003) 271–282.
[9] E.R. Panier, A.L. Tits, J.N. Herskovits, A QP-free, globally, locally superlinear convergent method for the inequality constrained optimizaiton problems, SIAM J. Control Optim. 36 (1988) 788–811.
[10] H. Qi, L. Qi, A new QP-free, globally V, locally superlinear convergent feasible method for the solution of inequality constrained optimization problems, SIAM J. Optim. 11 (2000) 113–132.
[11] Y. Zhou, D.G. Pu, A new QP-free feasible method for inequality constrained optimization, Math. Methods Oper. Res. 11 (2007) 31–43.
[12] R. Fletcher, S. Leyffer, P.L. Toint, On the global convergence of a filter-SQP algorithm, SIAM J. Optim. 13 (2002) 44–59.
[13] P.Y. Nie, C.F. Ma, A trust region filter method for general nonlinear programming, Appl. Math. Comput. 172 (2006) 1000–1017.
[14] F. Facchinei, S. Lucidi, Quadraticly and superlinearly convergent for the solution of inequality constrained optimization problem, JOTA 85 (1995) 265–289.
[15] W. Hock, K. Schittkowski, Test examples for nonlinear programming codes, Lecture Notes in Econom and Math Systems, vol. 87, Springer-Verlag, Berlin, 1981.