A kind of QP-free feasible method

Haiyan Zhang\textsuperscript{a,b,*}, Guojun Li\textsuperscript{a}, Hongluan Zhao\textsuperscript{c}

\textsuperscript{a} School of Mathematics, Shandong University, Jinan 250100, PR China
\textsuperscript{b} Department of Mathematics and Physics, Shandong Jiaotong University, Jinan 253023, PR China
\textsuperscript{c} School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, PR China

\textbf{A B S T R A C T}

This paper is concerned with a kind of QP-free feasible algorithm which solves an inequality constrained nonlinear optimization problem. Under some weaker conditions than those in [H. Qi, L. Qi, A New QP-free, globally convergent, locally superlinear convergent algorithm for inequality constrained optimization, SIAM J. Optim. 11 (2000) 113–132], we prove that the algorithm is implementable and globally convergent. Moreover, some numerical test results are given to indicate that the algorithm is quite promising.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

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

\[
\min f(x)
\]
\[
s.t. \ x \in D = \{x \in \mathbb{R}^n | G(x) \leq 0\},
\]

where \( f(x) \) : \( \mathbb{R}^n \rightarrow \mathbb{R} \) and \( G(x) = (g_1(x), g_2(x), \ldots, g_m(x))^T : \mathbb{R}^n \rightarrow \mathbb{R}^m \) are continuously differentiable functions. For simplicity, we denote the above optimization problem by Problem (NLP).

The Lagrange function associated with Problem (NLP) is

\[
\mathcal{L}(x, \lambda) = f(x) + \lambda^T G(x),
\]

where \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T \in \mathbb{R}^m \) is a multiplier vector.

A point \( (x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m, x \in D \) is called a Karush–Kuhn–Tucker (KKT in short) point of Problem (NLP) if it satisfies

\[
\nabla \mathcal{L}(x, \lambda) = 0, \quad G(x) \leq 0, \quad \lambda \geq 0, \quad \lambda_i g_i(x) = 0, \ i = 1, 2, \ldots, m.
\]

(1)

For the sake of convenience, let us first introduce some existing results. In 1988, Panier et al. [3] proposed a QP-free method for Problem (NLP). Their algorithm first calculates a descent direction \( d^0 \), by solving the following linear system, which is derived from (1)

\[
\begin{bmatrix}
H_k & \nabla G(x^k) \\
\text{diag}(\mu^k) \nabla G(x^k)^T & \text{diag}(G(x^k))
\end{bmatrix}
\begin{bmatrix}
d \\
\lambda
\end{bmatrix} = \begin{bmatrix}
-\nabla f(x^k) \\
0
\end{bmatrix},
\]

(2)
where $H_k \in \mathbb{R}^{n \times n}$ is an estimate of the Hessian of $L(x^k, \lambda^k)$, $x^k$ the current estimate of a solution $x^*$, $x^{k+1}$ the next estimate, $\mu^k$ the current estimate of the KKT multiplier vector associated with $x^*$, and $\lambda^{10}$ the next estimate of this vector. To guarantee the feasibility of the next iterate, they continue to calculate a direction $d_{k+1}$ by solving the following perturbed system of (2)

$$
\begin{bmatrix}
H_k & \nabla G(x^k) \\
\text{diag}(\mu^k) \nabla G(x^k) & \text{diag}(G(x^k))
\end{bmatrix}
\begin{bmatrix}
d_k \\
\lambda_n
\end{bmatrix}
= 
\begin{bmatrix}
-f(x^k) \\
-\|d^0\| \text{diag}(\mu^k)e
\end{bmatrix},
$$

(3)

where $\nu > 2, \nu \in \mathbb{R}^n$ is the vector of all ones. The search direction is a convex combination of the two directions, namely,

$$
d_k = (1 - \rho) d_{k+1} + \rho d_{k+1},
$$

where $\rho$ is calculated explicitly. However, if some multiplier $\mu^k$ corresponding to a nearly active constraint $g_i(x^k)$ becomes very small, as has been noted in [3], systems (2) and (3) may become very ill-conditioned. Thus, if $x^k$ is close to the solution of Problem (NLP), the strict complementarity conditions are not satisfied.

Recently, nonlinear complementarity problems (NCPs in short) have attracted much attention due to their various applications (see e.g. [2, 5, 7, 9]). In [8], Qi and Qi proposed a new QP-free feasible method based on the following Fischer–Burmeister (F–B in short) NCP function

$$
\phi_0(a, b) = a^2 + b^2 - a - b, \quad a, b \in \mathbb{R}.
$$

The function has some important properties, among which are:

- $\phi_0(a, b) = 0 \iff a \geq 0, b \geq 0, ab = 0$.
- The square of $\phi_0$ is continuously differentiable.
- $\phi_0$ is twice continuously differentiable everywhere except at the origin, but it is strongly semi-smooth at the origin.

Clearly, the KKT point conditions (1) can be equivalently reformulated as the following

$$
\phi_0(x, \lambda) = \begin{bmatrix}
\nabla L(x, \lambda) \\
\phi_0(-g_1(x), \lambda_1) \\
\vdots \\
\phi_0(-g_m(x), \lambda_m)
\end{bmatrix} = 0.
$$

With the F–B NCP function, Qi and Qi [8] proved that their algorithm was globally convergent without isolatedness of the accumulation point and the strict complementarity condition. They also proved that their algorithm was superlinearly convergent under some mild conditions. But, to prove global convergence, Qi and Qi [8] used some stronger conditions. One is the linear independence of gradients of active constrained functions at the solution, another is the uniformly positive definiteness of $H_k$, which is obtained by the Quasi-Newton update.

In our setting, we will study Problem (NLP). The target is to propose a kind of QP-free feasible method which solves Problem (NLP) and prove that our algorithm is implementable under some suitable conditions. We also prove that our algorithm is globally convergent under some weaker conditions than those of Qi and Qi [8], i.e., without assuming the linear independence of gradients of active constrained functions at the solution and the uniformly positive definiteness of $H_k$. This is a difference from the work of Qi and Qi [8]. However, for the sake of technique, we use similar conditions to those of Qi and Qi [8] to prove the superlinear convergence of our algorithm.

To design our algorithm for solving Problem (NLP), we replace the F–B NCP function $\phi_0(a, b)$ used in [8] with the following smoothing function for the F–B NCP function

$$
\phi(a, b, \epsilon) = \sqrt{a^2 + b^2 + 2 \epsilon^2} - a - b, \quad \epsilon > 0,
$$

which has the following properties used later on.

- $\phi(a, b, \epsilon) = 0 \iff a > 0, b > 0, ab = \epsilon^2$.
- For any fixed $\epsilon > 0, \phi(a, b, \epsilon)$ is continuously differentiable for all $(a, b) \in \mathbb{R}^2$.
- For any fixed $(a, b) \in \mathbb{R}^2$, the following limit

$$
\lim_{\epsilon \to 0} \left( \frac{\partial \phi(a, b, \epsilon)}{\partial a}, \frac{\partial \phi(a, b, \epsilon)}{\partial b} \right)
$$

exists.

Regarding the smoothing function $\phi(a, b, \epsilon)$, we refer the reader to Chen [4] for a detailed introduction.

Based on the above properties of $\phi(a, b, \epsilon)$, we construct a smoothing function of $\phi_0(x, \lambda)$ by replacing $\phi_0(a, b)$ with $\phi(a, b, \epsilon)$, that is

$$
\phi(x, \lambda, \epsilon) = \begin{bmatrix}
\nabla L(x, \lambda) \\
\phi(-g_1(x), \lambda_1, \epsilon) \\
\vdots \\
\phi(-g_m(x), \lambda_m, \epsilon)
\end{bmatrix}.
$$

Then we easily check the following conclusions about $\phi(x, \lambda, \epsilon)$:
• For any fixed $\epsilon > 0$, $\phi(x, \lambda, \epsilon)$ is continuously differentiable for all $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m$.
• The error of $\phi(x, \lambda, \epsilon)$ to $\phi_0(x, \lambda)$ is bounded by the smooth parameter $\epsilon$, namely,
\[
\| \phi(x, \lambda, \epsilon) - \phi_0(x, \lambda) \|_\infty \leq \kappa \epsilon, \quad \forall (x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m, \text{ where } \kappa > 0.
\]

The paper is organized as follows. In Section 2, a kind of QP-free feasible method is proposed. In Section 3, we will show that our algorithm is well defined. The conditions of the global convergence and superlinear convergence of the algorithm are discussed in Sections 4 and 5, respectively. In Section 6, we give some numerical test results which show that our algorithm is better than that of Qi and Qi [8]. In the last Section, some brief conclusions are drawn.

2. Algorithm

Let $(x^k, \mu^k) \in \mathbb{R}^n \times \mathbb{R}^m$ be given with $x^k$ being a strictly feasible point which ensures $g_i(x^k) < 0$ for all $i = 1, 2, \ldots, m$, and let $e^k = \| \phi_0(x^k, \lambda^k) \|_2$, where $\nu > 2$, and $\lambda^k$ will be obtained in our algorithm (Algorithm 2.1). Taking advantage of the conclusions about $\phi(x, \lambda, \epsilon)$, for any fixed $\epsilon > 0$, $\phi(x, \lambda, \epsilon)$ is continuously differentiable for all $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m$, then at the point $(x^k, \mu^k)$, $\phi(x, \lambda, \epsilon)$ exists a Jacobian matrix, namely,
\[
\begin{bmatrix}
\nabla^2_{xx} L(x^k, \mu^k) & \nabla G(x^k) \\
\text{diag}(\xi^k) & \text{diag}(\eta^k)
\end{bmatrix}
\]
(6)
where $\text{diag}(\xi^k)$ and $\text{diag}(\eta^k)$ denote diagonal matrixes whose $i$th diagonal elements are the following $\xi^k_i$ and $\eta^k_i$, respectively,
\[
\xi^k_i = \xi_i(x^k, \mu^k) = \frac{\partial \phi(-g_i(x^k), \mu^k, e^k)}{\partial g_i(x^k)} = \frac{g_i(x^k)}{\sqrt{g_i^2(x^k) + (\mu^k)^2 + 2(e^k)^2}} + 1,
\]
\[
\eta^k_i = \eta_i(x^k, \mu^k) = \frac{\partial \phi(-g_i(x^k), \mu^k, e^k)}{\partial \mu^k} = \frac{\mu^k}{\sqrt{g_i^2(x^k) + (\mu^k)^2 + 2(e^k)^2}} - 1.
\]
Moreover, set $\eta^k_0 = -\sqrt{-2\eta^k}$ and
\[
\theta^k_0 = \left( \frac{1}{\sqrt{g_i^2(x^k) + (\mu^k)^2 + 2(e^k)^2} + \sqrt{g_i^2(x^k) + (\mu^k)^2 + 2(e^k)^2 + \mu^k}} \right)^2,
\]
where $\eta^k_i$ and $\theta^k$ will be used in the proof of the following lemmas (Lemmas 4.3 and 5.5). In order to apply Quasi-Newton methods to solve Problem (NLP) and achieve superlinear convergence of this algorithm, we use the following matrix
\[
V_k = \begin{bmatrix}
H_k + c^k I_n & \nabla G(x^k) \\
\text{diag}(\xi^k) & \text{diag}(\eta^k)
\end{bmatrix}
\]
instead of $\Phi'(x, \lambda, \epsilon)$ in (6), where $H_k$ is a Lagrangian Hessian estimate and $H_k$ is updated by the BFGS method, $c^k = c_1 \min\{1, \| \phi_0(x^k, \lambda^k) \|_2 \}$, $c_1 \in (0, 1)$, and $I_n$ is the $n$-order unit matrix.

We will now give our algorithm.

Algorithm 2.1. Step 0. Initialization.
Choose $\alpha \in (0, \frac{1}{2})$, $\beta \in (0, 1)$, $\theta \in (0, 1)$, $\nu > 2$, $\tau \in (2, 3)$, $\kappa \in (0, 1)$, and $\tilde{\mu} > 0$ is a large enough number. Given $(x^0, \mu^0)$ and a symmetric positive definite matrix $H_0 \in \mathbb{R}^{n \times n}$, where $\tilde{x}^0$ is a strictly feasible point, $\mu^0 \in \mathbb{R}^m$, $0 < \mu^0_i \leq \tilde{\mu}$, $i = 1, 2, \ldots, m$, $e^0 = \| \phi_0(x^0, \mu^0) \|_2^2$. Set $k = 0$.

Step 1. Compute the search direction.
(1) Compute $d^{(0)}$ and $\lambda^{(0)}$ by solving the following linear system in $(d, \lambda)$
\[
V_k \begin{bmatrix}
d \\
\lambda
\end{bmatrix} = \begin{bmatrix}
-\nabla f(x^k) \\
0
\end{bmatrix}.
\]
If $d^{(0)} = 0$, then stop. Otherwise go to (2) below.
(2) Compute $d^{(1)}$ and $\lambda^{(1)}$ by solving the following linear system in $(d, \lambda)$
\[
V_k \begin{bmatrix}
d \\
\lambda
\end{bmatrix} = \begin{bmatrix}
-\nabla f(x^k) \\
\text{diag}(\xi^k)(\lambda^{(0)})^3
\end{bmatrix},
\]
(8)
where the $i$th element of $(\lambda^{(0)})^3$ is denoted by $(\min(0, \lambda^{(0)}))_i^3$.
(3) Compute $d^{(2)}$ and $\lambda^{(2)}$ by solving the following linear system in $(d, \lambda)$
\[
V_k \begin{bmatrix}
d \\
\lambda
\end{bmatrix} = \begin{bmatrix}
-\nabla f(x^k) \\
\text{diag}(\xi^k)(\lambda^{(1)})^3 - \text{diag}(\xi^k)e\|d^{(1)}\|_v
\end{bmatrix},
\]
where $v = (1, 1, \ldots, 1)^T \in \mathbb{R}^m$.
Compute the search direction \(d^k\) and the approximate multiplier vector \(\lambda^k\) as follows
\[
\begin{bmatrix}
  d^k \\
  \lambda^k
\end{bmatrix} = (1 - \rho^k) \begin{bmatrix}
  d^{k1} \\
  \lambda^{k1}
\end{bmatrix} + \rho^k \begin{bmatrix}
  d^{k2} \\
  \lambda^{k2}
\end{bmatrix},
\]
where
\[
\rho^k = \frac{(d^{k1})^T \nabla f(x^k)}{1 + \sum_{i=1}^{k} \lambda_i^k \|d^{k1}\|^2}.
\]
(5) Compute a correction \(\hat{d}^k\) by solving the following least square problem in \(d\)
\[
\min_d d^T H_d d \quad \text{s.t.} \quad g_i(x^k + d) + d^T \nabla g_i(x^k) = -\varphi^k, \quad i \in I_k,
\]
where \(I_k = \{i | g_i(x^k) \geq -\lambda_i^k\}\) and
\[
\varphi^k = \max \left\{ \|d^k\|^2, \max_{i \in I_k} \left| -\frac{\xi_i^k}{\sqrt{2\lambda_i^k}} - 1 \right| \|d^k\|^2 \right\}.
\]
If (12) has no solution or if \(\|\hat{d}^k\| > \|d^k\|\), then set \(\hat{d}^k = 0\).

Step 2. Line search.
Let \(t^j = \beta^j\), where \(j\) is the smallest non-negative integer satisfying
\[
\begin{align*}
  f(x^k + t^j d^k + (t^j)^2 \hat{d}^k) &\leq f(x^k) + \alpha t^j d^T \nabla f(x^k), \\
g_i(x^k + t^j d^k + (t^j)^2 \hat{d}^k) &< 0.
\end{align*}
\]
(13)

Step 3. Update.
Update \(H_k\) and obtain a symmetric positive definite matrix \(H_{k+1}\) whose expression will be given in Section 6. Set
\[
\begin{align*}
x^{k+1} &= x^k + t^j d^k + (t^j)^2 \hat{d}^k, \quad \lambda^{k+1} = \min \{\lambda^k, \bar{\mu}\}, \\
\mu^{k+1} &= \min \{\max \{\lambda^k, \|d^k\|\}, \bar{\mu}\}, \quad \epsilon^{k+1} = \|\Phi_0(x^{k+1}, \lambda^{k+1})\|.
\end{align*}
\]
(14) \(15\)
If \(\Phi_0(x^{k+1}, \mu^{k+1}) = 0\) or \(\Phi_0(x^{k+1}, \lambda^{k+1}) = 0\), then stop. Otherwise, set \(k = k + 1\), then go back to step 1.

Remark 2.2. In our algorithm, we first calculate a descent direction \(d^0\) by solving the system (7) which is derived from (1). However, this direction \(d^0\) can converge to zero with a negative multiplier. In order to avoid this point, we continue to calculate a deep descent direction \(d^1\) by solving the system (8), which is obtained by adding a slight perturbation \(\text{diag}(\xi^k)(\lambda^0)\) in the right-hand side of the system (7). The purpose of the system (9) is to ensure the feasibility of the next iterate. To avoid the Maratos effect, a further bending of the search direction can be obtained by solving the system (12).

3. Implementation of the algorithm

In this section, we will check that our algorithm (Algorithm 2.1) is implemented. To do this, we first give some preliminary results (Lemma 3.1, Corollary 3.2 and Lemmas 3.3–3.5). We need the following assumptions.

(A1) The strictly feasible set \(E = \{x \in R^n \mid G(x) < 0\}\) is nonempty, and the set \(D \cap \{x \mid f(x) \leq f(x^0)\}\) is compact.

(A2) \(H_k\) is positive definite and there exists a positive number \(m\) such that
\[
0 < d^T H_k d \leq M \|d\|^2
\]
for every \(k\) and \(d \in R^n\), \(d \neq 0\).

If \(\Phi_0(x^k, \lambda^k) = 0\) or \(\Phi_0(x^k, \mu^k) = 0\), then \((x^k, \lambda^k)\) or \((x^k, \mu^k)\) is a KKT point of Problem (NLP). Without loss of generality, in the following, for all \(k\), we assume that \(\Phi_0(x^k, \lambda^k) \neq 0\) and \(\Phi_0(x^k, \mu^k) \neq 0\).

Lemma 3.1. If \(\Phi_0(x^k, \lambda^k) \neq 0\), then \(V_k\) is nonsingular for all \(k\).

Proof. If \(V_k(u) = 0\) for some \((u, v) \in R^n \times R^m\), then
\[
u = (v_1, v_2, \ldots, v_m)^T,
\]
then we have
\[
\begin{align*}
(H_k + c_i^k d_i) u + \nabla G(x^k) v &= 0, \\
\text{diag}(\xi^k)(\nabla G(x^k))^2 u + (\text{diag}(\eta^k)) v &= 0.
\end{align*}
\]
Let \( \phi_0(x^*, \tilde{\lambda}^*) \neq 0 \). Obviously, \( e^i \neq 0 \) and \( c^i \neq 0 \). Moreover, the definitions of \( \eta^i \) and \( \xi^i \) imply that \( \text{diag}(\eta^i) < 0 \) and \( \text{diag}(\xi^i) > 0 \). By (17), we derive

\[
\nu = -(\text{diag}(\eta^i))^{-1} \text{diag}(\xi^i)(\nabla G(x^i))^T u.
\]

Substituting (18) into (16) and multiplying (16) by \( \mu^T \), we get

\[
u^T (H_k + c^i I_n) u + u^T \nabla G(x^i)(-\text{diag}^{-1}(\eta^i))\text{diag}(\xi^i)(\nabla G(x^i))^T u = 0.
\]

Since \( H_k + c^i I_n \) is positive definite and \( \nabla G(x^i)(-\text{diag}^{-1}(\eta^i))\text{diag}(\xi^i)(\nabla G(x^i))^T \) is positive semi-definite, we get \( u = 0 \) from (19), then \( \nu = 0 \) from (18). This lemma holds. \( \square \)

In terms of (A1) and (A2), without loss of generality, we may assume that \( \{x^i\}_K \to x^* \in D \) and \( \{H_k\}_K \to H_* \), where \( K \subset N \) is a subset of indices and \( H_* \) is a positive definite matrix in \( \mathbb{R}^{n \times n} \). Since \( \{\eta^i\} \), \{\xi^i\}, \{c^i\} and \{\tilde{\lambda}^i\} are bounded, without loss of generality, we also assume that \( \text{diag}(\eta^i)_K \to \text{diag}(\eta^*) \), \( \text{diag}(\xi^i)_K \to \text{diag}(\xi^*) \), \( \{c^i\}_K \to c^* \), \( \{(x^i, \tilde{\lambda}^i)\}_K \to (x^*, \tilde{\lambda}^*) \) and \( \{\mu^i\}_K \to \mu^* \). Putting all the limits together, we get \( \{\phi_0(x^i, \tilde{\lambda}^i)\}_K \to \phi_0(x^*, \tilde{\lambda}^*) \), \( \{V_k\}_K \to V_* \) and \( \{e^i\}_K \to e^* \). By \textbf{Lemma 3.1}, we can easily draw the following conclusion.

**Corollary 3.2.** If \( \phi_0(x^*, \tilde{\lambda}^*) \neq 0 \), then \( V_* \) is nonsingular and \( \|V_*^{-1}\|_K \) is bounded, where \( K \subset N \) is a subset of indices.

**Proof.** From \( \phi_0(x^*, \tilde{\lambda}^*) \neq 0 \), it is easy to see \( e^* \neq 0 \) and \( \eta^* < 0 \), \( i = 1, 2, \ldots, m \). By replacing the index \( k \) by \( \ast \) in the proof of \textbf{Lemma 3.1}, we can check that \( V_* \) is nonsingular, which implies that \( \|V_*\|_K \) is bounded. Then this proof is completed. \( \square \)

Noticing \textbf{Lemma 3.1}, we know that \( V_* \) is nonsingular. Let

\[
V_k^{-1} = A_k = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}.
\]

Through some computations, we get

\[
A_{11}^k = (H_k + c^i I_n)^{-1}(I_n + \nabla G(x^i)(Q_k)^{-1} \text{diag}(\xi^i)(\nabla G(x^i))^T(H_k + c^i I_n)^{-1}),
\]

\[
A_{12}^k = -(H_k + c^i I_n)^{-1} \nabla G(x^i)(Q_k)^{-1},
\]

\[
A_{21}^k = -(Q_k)^{-1} \text{diag}(\xi^i)(\nabla G(x^i))^T(H_k + c^i I_n)^{-1},
\]

\[
A_{22}^k = (Q_k)^{-1},
\]

where

\[
Q_k = \text{diag}(\eta^i) - \text{diag}(\xi^i)(\nabla G(x^i))^T(H_k + c^i I_n)^{-1} \nabla G(x^i).
\]

**Lemma 3.3.** If \( d^{i0} = 0 \), then \( \gamma f(x^i) = 0 \), and \( (x^i, \tilde{\lambda}^{i0}) \) is a KKT point of Problem (NLP).

**Proof.** If \( \phi_0(x^i, \tilde{\lambda}^i) = 0 \), \( (x^i, \tilde{\lambda}^i) \) is a KKT point of Problem (NLP). Now, we let \( \phi_0(x^i, \tilde{\lambda}^i) \neq 0 \). If \( d^{i0} = 0 \), the system (7) reduces to

\[
\nabla G(x^i)\lambda^{i0} = -\nabla f(x^i), \quad \text{diag}(\eta^i)\lambda^{i0} = 0.
\]

Note \( \phi_0(x^i, \tilde{\lambda}^i) \neq 0 \), it follows that \( \text{diag}(\eta^i) < 0 \). From the above equations, we easily get \( \lambda^{i0} = 0, \nabla f(x^i) = 0 \), and \( (x^i, 0) \) satisfies the condition of (1). So the proof is completed. \( \square \)

Without loss of generality, hereinafter, we assume that the algorithm never terminates at Step 1 (1), i.e., \( d^{i0} \neq 0 \).

**Lemma 3.4.** If \( d^{i0} \neq 0 \), then

1. \( c^i \|d^{i0}\| \leq (d^{i0})^T(H_k + c^i I_n)d^{i0} \leq -(d^{i0})^T \nabla f(x^i)^T \)
2. \( (d^{i1})^T \nabla f(x^i) = (d^{i0})^T \nabla f(x^i) - \sum_{\lambda^{i0} > 0}\lambda^{i0} c^i \lambda^{i0} \)
3. \( (d^{i1})^T \nabla f(x^i) \leq \theta(d^{i1})^T \nabla f(x^i) \)

**Proof.** (7) implies

\[
(H_k + c^i I_n)d^{i1} + \nabla G(x^i)\lambda^{i0} = -\nabla f(x^i),
\]

\[
\text{diag}(\eta^i)(\nabla G(x^i))^T d^{i0} + (\text{diag}(\eta^i))\lambda^{i0} = 0.
\]

Noting (27), we can get

\[
\lambda^{i0} = -(\text{diag}(\eta^i))^{-1} \text{diag}(\xi^i)(\nabla G(x^i))^T d^{i0}.
\]
Substituting (28) into (26) and multiplying (26) by \((d^0)^T\), we have
\[
(d^0)^T((H_k + c^T_k) d^0 + \nabla G(x^k) \lambda^0) = -(d^0)^T \nabla G(x^k) \lambda^0 = -(d^0)^T \nabla f(x^k).
\]
From \((d^0)^T \nabla G(x^k) \lambda^0 = -(d^0)^T \nabla f(x^k)\), we know
\[
c^k ||d^0||^2 \leq (d^0)^T (H_k + c^T_k) d^0 \leq - ||d^0||^2 \nabla f(x^k).
\]
So the first part of the lemma holds.
(7) and (20) imply
\[
d^0 = -A_{11}^T \nabla f(x^k), \quad \lambda^0 = -A_{12}^T \nabla f(x^k).
\]
Using property of the matrix, we obtain
\[
(Q_k)^{-1} \text{diag}(\xi_k) = (\text{diag}(\xi_k))^{-1} = \text{diag}(\xi_k)(Q_k)^{-1}.
\]
From (7), (22), (23) and (30), we get
\[
\lambda^0 = -A_{21}^T \nabla f(x^k) = - \text{diag}(\xi_k)(A_{12}^T)^T \nabla f(x^k).
\]
By virtue of (8), (22) and (31), we have
\[
(d^1)^T \nabla f(x^k) = -(\nabla f(x^k))^T (A_{11}^T)^T \nabla f(x^k) + ((\lambda^0)^T)^T \text{diag}(\xi_k)(A_{12}^T)^T \nabla f(x^k)
\]
\[
= (d^0)^T \nabla f(x^k) - \sum_{i : \lambda^0_i < 0} (\lambda^0_i)^4.
\]
So the second part of this lemma holds.
Now we prove the last part of the lemma. (8), (9) and (31) imply
\[
(d^2 - d^1)^T \nabla f(x^k) = - ||d^1||^2 (A_{12}^T \text{diag}(\xi_k) e)^T \nabla f(x^k) = ||d^1||^2 e^T \lambda^0.
\]
Finally, according to (8)–(11), (32) and (33), we get
\[
(d^3)^T \nabla f(x^k) = (1 - \rho^3)(d^1)^T \nabla f(x^k) + \rho^3(d^2)^T \nabla f(x^k)
\]
\[
= (d^1)^T \nabla f(x^k) + \rho^3 ||d^1||^2 e^T \lambda^0
\]
\[
= (d^1)^T \nabla f(x^k) - (\theta - 1) \frac{(d^1)^T \nabla f(x^k)}{1 + \sum_{i = 1}^m \lambda^0_i ||d^1||^2} \sum_{i = 1}^m \lambda^0_i
\]
\[
\leq \theta(d^1)^T \nabla f(x^k).
\]
Thus, this lemma holds.
\[
\square
\]
**Lemma 3.5.** If \(d^0 \neq 0\), then there exists a \(\widetilde{t}\) such that (13) holds for all \(t \in (0, \widetilde{t})\).

**Proof.** If \(d^0 \neq 0\), we have from the continuous differentiability of \(f(x)\)
\[
f(x^k) - f(x^k + t^k d^k + t^2 \hat{d}) = f(x^k) - f(x^k + t^k d^k + t^2 \hat{d}) - f(x^k + t^k d^k + t^2 \hat{d})
\]
\[
= -t(d^1)^T \nabla f(x^k) + O(||t^2||) - t^2(d^2)^T \nabla f(x^k + t^k d^k) + O(||t^4||)
\]
\[
\geq -t(d^1)^T \nabla f(x^k).
\]
Similarly, from \(g_i(x) < 0\) and the continuous differentiability of \(g_i(x)\), there exists \(\widetilde{t} > 0\) such that the second formula of (13) holds for \(t \in (0, \widetilde{t})\). Then we complete the proof.

Therefore, Lemmas 3.3–3.5 indicate that our algorithm (Algorithm 2.1) is implemented.

### 4. Global convergence

In this section, the global convergence of the algorithm will be proved without assuming the linear independence of the gradients of active constrained function and the uniformly positive definiteness of the submatrix \(H_k\) obtained by the Newton or Quasi-Newton methods. This is different from the procedure followed in Qi and Qi [8]. To get Theorem 4.5, we first introduce some preliminary results (Lemmas 4.1–4.4). We also suppose assumptions (A1) and (A2) hold in this section.

**Lemma 4.1.** Let \(x^*\) be an accumulation point of \(\{x^k\}\) and let \(K \subset N\) be a subset of indices such that \([x^k]_K \to x^*\) and \([\phi_0(x^k, \bar{x}^k)]_K \to \phi_0(x^*, \bar{x}^*)\) for some \(\epsilon > 0\). Then the sequences of \(\{(d^0, \lambda^0)_K\}, \{(d^1, \lambda^1)_K\}\) and \(\{(d^2, \lambda^2)_K\}\) are all bounded on \(k = 0, 1, 2, \ldots\).
Proof. From Corollary 3.2, if $|x^k_k| \to x^*$ and $(\theta_0(x^k, \lambda_k^k))_k > \epsilon > 0$ for some $\epsilon$, then the matrix sequence $(V_k^{-1})_k$ is uniformly bounded. Also, $(x_k^k)_k$ is bounded due to assumption (A1). The solution of (7) implies that $(d_{k, 0}, \lambda_{k, 0})_k$ is bounded, which then implies the boundedness of the right-hand side of (8). Hence $(d_{k, 1}, \lambda_{k, 1})_k$ is also bounded. Finally, the boundedness of $(d_{k, 2}, \lambda_{k, 2})_k$ implies the boundedness of the right-hand side of (9) and hence that of $(d_{k, 3}, \lambda_{k, 3})_k$. The proof is completed. □

Lemma 4.2. Let $x^*$ be an accumulation point of $|x^k_k|$ and let $K \subset N$ be a subset of indices such that $|x^k_k| \to x^*$ and $(\theta_0(x^k, \lambda_k^k))_k > \epsilon > 0$ for some $\epsilon$. If $(d_k)_k \to 0$, then $x^*$ is a KKT point of Problem (NLP). Moreover, $(\lambda_{k, 0})$ converges to the unique multiplier vector $\lambda^*$ which is associated with $x^*$.

Proof. It follows from Lemma 3.4 that

$$(d_k^k)^T \nabla f(x^k_k) \leq -c^T \theta d_{k, 0}^2 - \theta \sum_{i, \lambda_{k, 0} < 0} (\lambda_{k, 0})^4, \quad k \in K.$$ 

Let $(d_k)_k \to 0$ for $k \in K$, we have

$$\sum_{i, \lambda_{k, 0} < 0} (\lambda_{k, 0})^4 \to 0, \quad (d_k^0) \to 0. \tag{34}$$

Since $(\lambda_{k, 0})_k, (\mu_k)_k, (\lambda_k)_k$ and $(\eta_k)_k$ are bounded, there exist $\lambda^*, \mu^*, \lambda^*, c^*$ and a subset $K'$ of $K$ such that

$$(\lambda_{k, 0})_k \to \lambda^*, \quad (\mu_k)_k \to \mu^*, \quad (\lambda_k)_k \to \lambda^*, \quad (\eta_k)_k \to c^*.$$ 

(34) implies that $\lambda_i^* > 0, i = 1, 2, \ldots, m$, and by the definition of $\lambda_k^*$, we know $\lambda_i^* > 0, i = 1, 2, \ldots, m$. Noting the definition of $\eta_i^*$, we have

$$\lim_{k \to \infty} \eta_i^k = \eta_i^* = -\sqrt{2} \left(1 - \frac{\mu^*}{\sqrt{g_i^2(x^*) + (\mu_i^*)^2 + 2||\phi_0(x^*, \lambda^*)||^2}}\right)^{1/2} < 0. \tag{35}$$

Taking limits on both sides of (7), and noting $(d_k)_k \to 0$, we derive

$$(\lambda^*)^T \nabla G(x^*) = -\nabla f(x^*), \tag{36}$$

$$(\mu^*) = 0. \tag{37}$$

If $g_i(x^*) = 0$, then we have $\lambda_i^* g_i(x^*) = 0, i = 1, 2, \ldots, m$. If $g_i(x^*) \neq 0$, then we get $\lambda_i^* = 0$ by (35) and (37). In conclusion, for any $1 \leq i \leq m$, $(x^*, \lambda^*)$ satisfies

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) = 0,$$

$$\lambda_i^* g_i(x^*) = 0, \quad i = 1, 2, \ldots, m.$$ 

From (1), we know that $x^*$ is a KKT point of Problem (NLP). This lemma holds. □

Lemma 4.3. Let $x^*$ be an accumulation point of $|x^k_k|$ and let $K \subset N$ be a subset of indices such that $|x^k_k| \to x^*$ and $(\theta_0(x^k, \lambda_k^k))_k > \epsilon > 0$ for some $\epsilon$. If $(d_k)_k \to 0$, then $(x^*, \lambda^*)$ is a KKT point of Problem (NLP), where $\lambda^*$ is an accumulation point of $(\lambda_k)_k$.

By Lemma 4.2, the proof of Lemma 4.3 is similar to that of Lemma 3.8 in [3], so we omit it. The following result is the same as that given by Lemma 3.6 in [8].

Lemma 4.4. Let $x^*$ be an accumulation point of $|x^k_k|$ and let $K \subset N$ be a subset of indices such that $|x^k_k| \to x^*$ and $(\theta_0(x^k, \lambda_k^k))_k > \epsilon > 0$ for some $\epsilon$. If $\inf ||d_k^k||_k > 0$, then $(x^*, \lambda^*)$ is a KKT point of Problem (NLP).

Therefore we get the following global convergence theorem.

Theorem 4.5. If $x^*$ is a limit point of $|x^k_k|$, then $x^*$ is a KKT point of Problem (NLP).

Proof. Let $(x_k, \lambda_k)_k \to (x^*, \lambda^*)$. If $\theta_0(x^*, \lambda^*) = 0$, then $(x^*, \lambda^*)$ is a KKT point of Problem (NLP). If $\theta_0(x^*, \lambda^*) \neq 0$, then $(\lambda_k)_k$ is bounded by Lemma 4.1. By Lemmas 4.2–4.4, we can prove that $(x^*, \lambda^*)$ is a KKT point, where $\lambda^*$ is an accumulation point of $(x_k)_k$. The proof is completed. □
5. Superlinear convergence

In this section, to prove the superlinear convergence of Algorithm 2.1, we also need the following assumptions (A3)–(A7) except for (A1) and (A2) given in Section 2. Let \((x^*, \lambda^*)\) be a limit point of the sequence \(\{(x^k, \lambda^k)\}\). According to Theorem 4.5, then \((x^*, \lambda^*)\) is a KKT point. We also let \(I(x^*) = \{ig(x^*) = 0\}\).

(A3) \([V_g(x)]\) are linearly independent, where \(i \in I(x^*)\).

(A4) \(H_k\) is uniformly positive definite and there exist two positive numbers \(m_1\) and \(m_2\) such that

\[
0 < m_2 \|d\|^2 \leq \|\nabla H_k d\| \leq m_1 \|d\|^2, \quad \forall d \in \mathbb{R}^n, k = 1, 2, \ldots.
\]

(A5) The strict complementarity condition is satisfied at each KKT point \((x^*, \lambda^*)\), namely \(\lambda^+_i - \mu_i > 0, i = 1, 2, \ldots, m\).

(A6) The second-order sufficiency condition for Problem (NLP) holds at each KKT point \((x^*, \lambda^*)\), i.e., the Hessian \(V_{xx}^2 (x^*, \lambda^*)\) is positive definite on the subspace \([h | \bar{h}^T V_g(x^*) = 0, i \in I(x^*)]\).

(A7) The sequences of \(\{H_k\}\) satisfy

\[
\frac{\|P_k((H_k - V_{xx}^2 (x^*, \lambda^*))d)\|}{\|d\|} \to 0,
\]

where \(P_k = I - N_k (N_k^T N_k)^{-1} N_k^T\) and \(N_k = [V_g(x) : i \in I(x^*)]\).

Similarly to the proof of Lemma 3.1 and Corollary 3.2, we have:

Lemma 5.1. Assume that (A1)–(A4) hold. Then \(\|V_c^{-1}\|\) is bounded. Furthermore, if \(\{V_k\} \to V_0\), then \(V_0\) is nonsingular.

Clearly, from Lemma 5.1 and the proof of Lemma 4.1, we can easily check the following result.

Lemma 5.2. Assume that (A1)–(A4) hold. Then the sequences of \(\{(a^{i0}, \lambda^{i0})\}, \{(a^{i1}, \lambda^{i1})\}\) and \(\{(a^{i2}, \lambda^{i2})\}\) are all bounded on \(k = 0, 1, 2, \ldots\)

To prove Theorem 5.4, we also need the following result.

Lemma 5.3. There exists \(c_1 > 0\) such that \(\|d^k - d^{i0}\| \leq c_1 \|d^{i0}\|\) for all \(k = 0, 1, 2, \ldots\).

Proof. According to Lemmas 5.1 and 5.2, we know that \(\{d^{i0}\}, \{\lambda^{i0}\}\) and \(\{V^{-1}\}\) are bounded. Furthermore, the definition of \(\rho^k\) yields the boundedness of the sequence \(\|\rho^k\|^2\). Let \(c_1 = 2 \sup \|\rho^k\| \sup \|\{(V_k)^{-1}\}\|\). Then \(\|d^k - d^{i0}\|\) and \(\|\lambda^k - \lambda^{i0}\|\) are all bounded on \(k = 0, 1, 2, \ldots\).

This lemma holds. \(\square\)

The proof of the following theorem (Theorem 5.4) is similar to that of Lemma 4.2 and Corollary 4.3 in [8], so we omit it.

Theorem 5.4. Assume that (A1)–(A6) hold. Then we have:

1. The whole sequence \((x^k, \lambda^{i0}) \to (x^*, \lambda^*)\), where \(\lambda^*\) is the unique multiplier of \(\{\lambda^{i0}\}\).
2. \(d^{i0} \to 0\), \(d^{i1} \to 0\) and \(d^{i2} \to 0\).

Since \(H_k\) is displaced by \(H_k + c^k I_n\) in the coefficient matrix \(V_k\) of our algorithm, we need to prove that the following limit (39) holds, which will be used in Lemma 5.5,

\[
\frac{\|P_k((H_k + c^k I_n) - V_{xx}^2 (x^*, \lambda^*))d)\|}{\|d\|} \to 0.
\]

It is easy to see that \(\lim_{k \to \infty} \Phi_0(x^k, \lambda^k) = \lim_{k \to \infty} \Phi_0(a^{i0}, \lambda^{i0}) = 0\). From the definition of \(c^k\), then we know that \(c^k \to 0\) as \(k \to \infty\). By (38) and \(c^k \to 0\) as \(k \to \infty\), we get

\[
\frac{\|P_k((H_k + c^k I_n) - V_{xx}^2 (x^*, \lambda^*))d)\|}{\|d\|} = \frac{\|d\| \|P_k((H_k + c^k I_n) - V_{xx}^2 (x^*, \lambda^*))d)\|}{\|d\|} \\
\leq \frac{\|d\| \|P_k((H_k + c^k I_n) - V_{xx}^2 (x^*, \lambda^*))d)\|}{\|d\|} \\
= \|P_k((H_k + c^k I_n) - V_{xx}^2 (x^*, \lambda^*))d)\| \\
\leq \frac{\|P_k((H_k + c^k I_n) - V_{xx}^2 (x^*, \lambda^*))d)\|}{\|d\|} + \frac{\|P_k c^k I_n d\|}{\|d\|} \to 0.
\]
Let \( \Phi \) be implemented and generate a sequence \( \{ (x^k, \lambda^k) \} \) with an accumulation point \( (x^*, \lambda^*) \). Then \( (x^*, \lambda^*) \) is a KKT point of Problem (NLP) and \( (x^k, \lambda^k) \) superlinearly converges to \( (x^*, \lambda^*) \).

### Lemma 5.5
Let (A1)–(A7) hold. For a large enough \( k \), the step \( t_k = 1 \) is accepted by the line search.

### Theorem 5.6
Assume that (A1)–(A7) hold. Let Algorithm 2.1 be implemented and generate a sequence \( \{ (x^k, \lambda^k) \} \) with an accumulation point \( (x^*, \lambda^*) \). Then \( (x^*, \lambda^*) \) is a KKT point of Problem (NLP) and \( (x^k, \lambda^k) \) superlinearly converges to \( (x^*, \lambda^*) \).

### Remark 5.7
From Lemma 5.2, we know that \( \lambda^{k_0} \) is bounded. If there exists a sufficiently large \( \bar{\mu} \) such that \( \lambda^{k_0} < \bar{\mu} \), furthermore, \( \bar{\lambda}^k = \lambda^{k_0} \) for large enough \( k \), then we can prove that the algorithm is also superlinearly convergent without (A4) and (A5).

### 6. Numerical tests
Algorithm 2.1 was implemented in MATLAB and tested over a set of problems from Hock and Schittkowski [6]. The details about the implementation are described as follows.

1. Test problems and parameters. A total of 30 problems were selected from [6]. These problems only have inequality constraints and the starting points provided are strictly feasible. From here 20 problems were tested in [8], and the numerical results of their Algorithm are presented in the following table (Table 1), which can also be found in [8].

2. It is worth pointing out that our algorithm is relatively insensitive to the parameter values of the algorithm. To illustrate this, we respectively use the following three groups of different parameters to test our algorithm:

   (i) \( c_1 = 0.5, \alpha = 0.3, \beta = 0.5, \theta = 0.8, v = 3, \tau = 2.5, \kappa = 0.9, \bar{\mu} = +\infty \),

   (ii) \( c_1 = 0.4, \alpha = 0.25, \beta = 0.5, \theta = 0.7, v = 3, \tau = 2.5, \kappa = 0.8, \bar{\mu} = +\infty \),

   (iii) \( c_1 = 0.6, \alpha = 0.3, \beta = 0.45, \theta = 0.8, v = 4, \tau = 2.4, \kappa = 0.9, \bar{\mu} = +\infty \), where the first group of parameters, (i), was used in [8]. The corresponding numerical test results can be seen in Tables 1–4, where we adopt the following notations.

   Problem: Number of problems in Hock and Schittkowski [6],

   NIT: Number of iterations,

   NF: The number of function \( f(x) \) evaluations,

   NG: The number of function \( G(x) \) evaluations,

   \( \| \phi \| \): Value of \( \| \phi(\cdot, \cdot) \| \) at the final iterate \( (x^k, \lambda^{k_0}) \),

   FV: Objective function value at the final iterate,

   Group i: The ith group of parameter.

3. The termination criterion. It follows from the properties of the Fischer–Burmeister function that the final iterate must be an approximate KKT point of NLP. Hence we use the termination criterion \( \| \phi_0(x^k, \lambda^{k_0}) \| \leq 10^{-5} \), where \( \phi_0 \) is defined in (4). This termination criterion works quite well for our test problems.

4. BFGS update. In our algorithm (Algorithm 2.1), the initial Lagrangian Hessian estimate is \( H_0 = I \), and \( H_k \) is updated by the BFGS method. In particular, we set

   \[
   H_{k+1} = H_k - \frac{H_k s^k (s^k)^T H_k}{(s^k)^T H_k s^k} + \frac{y^k (y^k)^T}{(s^k)^T y^k},
   \]
| Problem | NIT | NF | NG | $||\phi||$ | FV |
|---------|-----|----|----|-------------|----|
| 1       | 17  | 30 | 46 | 1.2e-07     | 0.0000000000e-02 |
| 2       | 11  | 18 | 19 | 2.3e-08     | 0.000000000171e-02 |
| 3       | 5   | 11 | 13 | 7.2e-07     | 2.6666666667e+00 |
| 4       | 5   | 10 | 12 | 1.6e-08     | -1.913229550e+00 |
| 5       | 5   | 10 | 18 | 1.2e-06     | 3.0000000000e+01 |
| 6       | 12  | 17 | 19 | 3.5e-10     | -1.0000000000e+00 |
| 7       | 9   | 12 | 13 | 4.3e-07     | -2.2627416998e+01 |
| 8       | 7   | 9  | 12 | 5.2e-08     | 1.0000000000e+00 |
| 9       | 9   | 20 | 23 | 2.2e-07     | 6.0000000000e+00 |
| 10      | 12  | 15 | 20 | 1.8e-09     | -4.5857864377e+00 |
| 11      | 19  | 40 | 43 | 4.9e-07     | -8.3403244525e-01 |
| 12      | 8   | 11 | 13 | 8.3e-07     | 1.1111111111e+00 |
| 13      | 15  | 39 | 49 | 6.5e-11     | -3.3000000000e+03 |
| 14      | 16  | 41 | 47 | 2.6e-07     | -3.4560000000e+03 |
| 15      | 23  | 57 | 69 | 3.9e-09     | 0.0000000023e-02 |
| 16      | 11  | 25 | 29 | 1.1e-06     | -4.4000000000e+01 |
| 17      | 14  | 21 | 29 | 7.4e-09     | -1.5000000000e+01 |
| 18      | 27  | 68 | 92 | 3.6e-06     | 2.845969723e-02 |
| 19      | 25  | 47 | 73 | 2.7e-07     | 5.1816327415e-01 |
| 20      | 92  | 231| 715| 5.9e-06     | -1.1620365071e+03 |
| 21      | 78  | 169| 432| 4.8e-07     | 7.4984632582e-03 |
| 22      | 9   | 16 | 24 | 1.1e-08     | -4.6818181812e+00 |
| 23      | 38  | 84 | 101| 8.2e-07     | -5.2803351332e+06 |
| 24      | 69  | 135| 321| 3.9e-06     | -3.2348678973e+01 |
| 25      | 45  | 64 | 157| 5.3e-07     | 1.3507596129e+02 |
| 26      | 13  | 27 | 37 | 1.3e-08     | 6.806305732e+02 |
| 27      | 16  | 29 | 201| 2.1e-07     | -4.5778469710e+01 |
| 28      | 18  | 36 | 37 | 7.8e-06     | 2.4390264939e+01 |
| 29      | 98  | 157| 583| 9.6e-06     | 3.2348679344e+01 |
| 30      | 101 | 135| 632| 7.5e-07     | 6.6482045009e+02 |

| Problem | NIT | NF | NG | $||\phi||$ | FV |
|---------|-----|----|----|-------------|----|
| 1       | 16  | 31 | 47 | 1.2e-07     | 0.0000000000e-02 |
| 2       | 11  | 19 | 21 | 2.7e-08     | 0.00000000126e-02 |
| 3       | 4   | 8  | 12 | 6.8e-07     | 2.6666666667e+00 |
| 4       | 5   | 11 | 11 | 3.6e-08     | -1.9132295498e+00 |
| 5       | 12  | 12 | 16 | 4.2e-06     | 3.0000000000e+01 |
| 6       | 11  | 16 | 18 | 3.9e-10     | -0.0999999978e+01 |
| 7       | 8   | 14 | 17 | 5.2e-07     | -2.2627416997e+01 |
| 8       | 7   | 12 | 14 | 6.4e-08     | 1.0000000000e+00 |
| 9       | 8   | 21 | 25 | 2.8e-07     | 6.0000000000e+00 |
| 10      | 11  | 16 | 21 | 1.7e-09     | -4.5857863978e+00 |
| 11      | 19  | 41 | 45 | 5.9e-07     | -8.3403244526e-01 |
| 12      | 18  | 10 | 13 | 7.8e-07     | 1.1111111111e+01 |
| 13      | 13  | 48 | 59 | 8.9e-11     | -3.3000000000e+03 |
| 14      | 17  | 43 | 48 | 3.3e-07     | -3.4560000000e+03 |
| 15      | 25  | 69 | 77 | 4.5e-09     | 0.0000000109e+02 |
| 16      | 12  | 25 | 29 | 2.0e-06     | -4.4000000000e+01 |
| 17      | 15  | 23 | 28 | 6.8e-09     | -1.5000000000e+01 |
| 18      | 29  | 76 | 98 | 3.5e-06     | 2.8459696979e-02 |
| 19      | 25  | 48 | 77 | 2.9e-07     | 5.1816327409e-01 |
| 20      | 95  | 246| 689| 7.9e-06     | -1.1620365069e+03 |
| 21      | 79  | 176| 452| 4.6e-07     | 7.4984643498e-03 |
| 22      | 10  | 17 | 25 | 2.3e-08     | -4.6818181809e+00 |
| 23      | 39  | 87 | 105| 8.7e-07     | -5.2803351329e+06 |
| 24      | 72  | 145| 330| 4.2e-06     | -3.2348678969e+01 |
| 25      | 43  | 60 | 146| 6.8e-07     | 1.3507596098e+02 |
| 26      | 14  | 31 | 40 | 2.3e-08     | 6.8063057312e+02 |
| 27      | 18  | 37 | 219| 3.1e-07     | -4.5778405842e+01 |
| 28      | 17  | 27 | 35 | 7.4e-06     | 2.4306210163e+01 |
| 29      | 103 | 168| 593| 9.5e-06     | 3.2348680368e+01 |
| 30      | 97  | 137| 642| 6.3e-07     | 6.6482045237e+02 |
Table 4 (Group (iii)) Numerical test results for Algorithm 2.1

<table>
<thead>
<tr>
<th>Problem</th>
<th>NIT</th>
<th>NF</th>
<th>NG</th>
<th>$|\Phi|$</th>
<th>FV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17</td>
<td>34</td>
<td>48</td>
<td>1.5e−07</td>
<td>0.00000000001e−02</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>18</td>
<td>20</td>
<td>2.3e−08</td>
<td>0.00000000102e−02</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>12</td>
<td>14</td>
<td>7.2e−07</td>
<td>2.6669566666e+00</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>4.7e−08</td>
<td>−1.9132225550e+00</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>10</td>
<td>17</td>
<td>2.2e−06</td>
<td>3.0000000000e+00</td>
</tr>
<tr>
<td>24</td>
<td>10</td>
<td>14</td>
<td>17</td>
<td>3.0e−10</td>
<td>−1.0000000000e+00</td>
</tr>
<tr>
<td>29</td>
<td>9</td>
<td>15</td>
<td>18</td>
<td>5.6e−08</td>
<td>−2.2627416986e+01</td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>10</td>
<td>15</td>
<td>9.4e−09</td>
<td>1.0000000000e+00</td>
</tr>
<tr>
<td>31</td>
<td>9</td>
<td>24</td>
<td>29</td>
<td>4.2e−07</td>
<td>6.0000000000e+00</td>
</tr>
<tr>
<td>33</td>
<td>12</td>
<td>17</td>
<td>23</td>
<td>3.7e−09</td>
<td>−4.5857864381e+00</td>
</tr>
<tr>
<td>34</td>
<td>20</td>
<td>49</td>
<td>56</td>
<td>6.2e−07</td>
<td>−8.3402344498e+01</td>
</tr>
<tr>
<td>35</td>
<td>7</td>
<td>12</td>
<td>15</td>
<td>8.4e−07</td>
<td>1.1111111111e−01</td>
</tr>
<tr>
<td>36</td>
<td>16</td>
<td>41</td>
<td>52</td>
<td>9.1e−10</td>
<td>−3.3000000000e+03</td>
</tr>
<tr>
<td>37</td>
<td>16</td>
<td>42</td>
<td>48</td>
<td>1.3e−07</td>
<td>−3.4560000000e+03</td>
</tr>
<tr>
<td>38</td>
<td>24</td>
<td>61</td>
<td>70</td>
<td>3.4e−09</td>
<td>0.0000000000e+02</td>
</tr>
<tr>
<td>43</td>
<td>12</td>
<td>25</td>
<td>30</td>
<td>2.6e−06</td>
<td>−4.4000000000e+01</td>
</tr>
<tr>
<td>44</td>
<td>14</td>
<td>21</td>
<td>31</td>
<td>5.7e−09</td>
<td>−1.5000000000e+01</td>
</tr>
<tr>
<td>57</td>
<td>30</td>
<td>85</td>
<td>107</td>
<td>6.3e−06</td>
<td>2.8459669719e+02</td>
</tr>
<tr>
<td>66</td>
<td>25</td>
<td>49</td>
<td>79</td>
<td>3.9e−07</td>
<td>5.1816327503e−01</td>
</tr>
<tr>
<td>67</td>
<td>94</td>
<td>239</td>
<td>713</td>
<td>5.4e−06</td>
<td>−1.1620365073e+03</td>
</tr>
<tr>
<td>70</td>
<td>82</td>
<td>187</td>
<td>473</td>
<td>7.6e−07</td>
<td>7.498463519e−03</td>
</tr>
<tr>
<td>76</td>
<td>10</td>
<td>19</td>
<td>26</td>
<td>2.2e−08</td>
<td>−4.6818181814e+00</td>
</tr>
<tr>
<td>84</td>
<td>37</td>
<td>80</td>
<td>96</td>
<td>2.7e−06</td>
<td>−5.2803351330e+06</td>
</tr>
<tr>
<td>86</td>
<td>70</td>
<td>143</td>
<td>341</td>
<td>5.2e−07</td>
<td>−3.2348678974e+01</td>
</tr>
<tr>
<td>93</td>
<td>48</td>
<td>60</td>
<td>162</td>
<td>4.3e−07</td>
<td>1.5057961161e+02</td>
</tr>
<tr>
<td>100</td>
<td>15</td>
<td>37</td>
<td>42</td>
<td>1.9e−08</td>
<td>6.8063005732e+02</td>
</tr>
<tr>
<td>110</td>
<td>20</td>
<td>42</td>
<td>237</td>
<td>2.7e−07</td>
<td>−4.5778453612e+01</td>
</tr>
<tr>
<td>113</td>
<td>18</td>
<td>29</td>
<td>45</td>
<td>6.8e−06</td>
<td>2.4305280816e+01</td>
</tr>
<tr>
<td>117</td>
<td>106</td>
<td>173</td>
<td>597</td>
<td>8.7e−06</td>
<td>2.3243678905e+01</td>
</tr>
<tr>
<td>118</td>
<td>108</td>
<td>152</td>
<td>659</td>
<td>7.6e−07</td>
<td>6.6482045108e+02</td>
</tr>
</tbody>
</table>

where

$$y^k = \begin{cases} \hat{y}, \\ (\hat{y}^k)^T s^k \geq 0.2(s^k)^T H_k s^k, \end{cases}$$

and

$$s^k = \begin{cases} x^{k+1} - x^k, \\ \lambda^{10}, \end{cases}$$

$$\bar{d} = 0.8(s^k)^T H_k s^k / ((s^k)^T H_k s^k - (s^k)^T \hat{y}^k).$$

(4) Computing the correction direction. In order to save computation, evaluation of the correction $\bar{d}$ should be calculated only when the iterate is close to a solution of Problem (NLP). In our implementation, $\bar{d}$ is calculated when $\|\Phi(x^k, \lambda^k)\| \leq 1$ and $\|d^k\| \leq 0.1$.

By numerical experiments, it is worth pointing out the following case. For any $\epsilon^k > 0$, we have proved that the matrix $V_k$ is nonsingular. In practical implementation, if $\epsilon^k$ is small enough, then we find that the matrix $V_k$ is near singular unless the linear independence of the active constraints is present. However, the results of the numerical experiments in Tables 1–4 indicate that our algorithm (Algorithm 2.1) is better than that of Qi and Qi [8].

7. Conclusions

In our algorithm (Algorithm 2.1), the matrix $H_k$ is replaced by $H_k + \epsilon^k I_n$. This idea is stimulated in [1]. What is more, we replace $\Phi(x, \lambda)$ used in [8] with $\Phi(x, \lambda, \epsilon)$, then we can prove the global convergence of the algorithm without the linear independence of the gradients of active constrained functions at the solution and the uniformly positive definiteness of $H_k$ which is obtained by the Quasi-Newton update. Tables 1–4 show that our algorithm is better than that in [8]. However, to prove that our algorithm is superlinearly convergent, 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.

Acknowledgements

The authors would like to thank the anonymous referee for the careful reading and helpful comments and suggestions that led to an improved version of this paper. The authors also thank Professor D. PU for many useful advices.
References