A SUPERLINEARLY CONVERGENT HYBRID ALGORITHM
FOR SOLVING NONLINEAR PROGRAMMING

CHUANHAO GUO
School of Economics and Management, Zhejiang Sci-Tech University
Hangzhou, Zhejiang 310018, China

ERFANG SHAN
School of Management, Shanghai University
Shanghai 200444, China

WENLI YAN
School of Mathematics and Physics, Huaiyin Institute of Technology
Huaian, Jiangsu 223003, China

(Communicated by Gianni Di Pillo)

ABSTRACT. In this paper, a superlinearly convergent hybrid algorithm is proposed for solving nonlinear programming. First of all, an improved direction is obtained by a convex combination of the solution of an always feasible quadratic programming (QP) subproblem and a mere feasible direction, which is generated by a reduced system of linear equations (SLE). The coefficient matrix of SLE only consists of the constraints and their gradients corresponding to the working set. Then, the second-order correction direction is obtained by solving another reduced SLE to overcome the Maratos effect and obtain the superlinear convergence. In particular, the convergence rate is proved to be locally one-step superlinear under a condition weaker than the strong second-order sufficient condition and without the strict complementarity. Moreover, the line search in our algorithm can effectively combine the initialization processes with the optimization processes, and the line search conditions are weaker than the previous work. Finally, some numerical results are reported.

1. Introduction. In this paper, we consider the constrained optimization problem to minimize an objective function $f_0(x)$ with inequality constraints

$$(\text{NIO}) \quad \min f_0(x) \quad \text{s.t.} \quad f_i(x) \leq 0, \ i \in I := \{1, 2, \ldots, m\},$$

where $x \in \mathbb{R}^n$ and $f_i(x) : \mathbb{R}^n \to \mathbb{R} (i \in \{0\} \cup I)$ are nonlinear smooth functions. Denote the feasible set and gradients for problem (NIO) by

$$\Theta := \{x \in \mathbb{R}^n : f_i(x) \leq 0, i \in I\}, \quad g_i(x) = \nabla f_i(x), \ i \in \{0\} \cup I.$$
moreover, a great deal of effort has been devoted to developing efficient SQP algorithms \[5, 8, 17\] during the past several decades. Given an estimate \(x\) of the solution for problem (NIO) and a symmetric matrix \(H\) that approximates the Hessian of the Lagrangian function \(L(x, \lambda) := f_0(x) + \sum_{i \in I} \lambda_i f_i(x)\), where \(\lambda\) is a vector of nonnegative Lagrange multiplier estimates, the standard SQP search direction is obtained by solving QP subproblem as follows

\[
\begin{align*}
\text{min} & \quad g_0(x)^T d + \frac{1}{2} d^T H d \\
\text{s.t.} & \quad f_i(x) + g_i(x)^T d \leq 0, \quad i \in I.
\end{align*}
\]

Under some suitable conditions, SQP algorithms are known to be globally and locally superlinearly convergent.

Note that a limitation in the application of SQP methods to problem (NIO) is the possible inconsistency of the constraints in subproblem (QP1). To address this difficulty, many techniques have been proposed \[1, 22\]. Jian \[12\] presented a new regularization technique for dealing with the inconsistent QP subproblem in SQP methods, the modified QP subproblem is as follows

\[
\begin{align*}
\text{min} & \quad g_0(x)^T d + \frac{1}{2} d^T H d \\
\text{s.t.} & \quad f_i(x) + g_i(x)^T d \leq \Upsilon(x), \quad i \in I^+(x), \\
& \quad f_i(x) + g_i(x)^T d \leq 0, \quad i \in I^-(x),
\end{align*}
\]

where \(\Upsilon(x) := \max\{0, f_i(x), i \in I\}\), \(I^+(x) := \{i \in I : f_i(x) > 0\}\), \(I^-(x) := \{i \in I : f_i(x) \leq 0\}\). The subproblem (QP2) has the following advantages:

\[
\begin{itemize}
\item subproblem (QP2) is always feasible with \(d = 0\).
\item subproblem (QP2) is strictly convex when \(H\) is positive definite, so it has a unique solution.
\item \(d\) is a solution of subproblem (QP2) if and only if it is a KKT point of subproblem (QP2).
\end{itemize}
\]

With the help of subproblem (QP2), Jian et al. proposed some superlinear convergent algorithms in \[11\].

In \[13\], Jian et al. present a new SQP algorithm for solving problem (NIO) by using of subproblem (QP2). The correctional directions are obtained by explicit formulas in which the generalized projection technique is applied. With the help of curve search technique and some suitable conditions, the algorithm is proven to be global and superlinear convergent. Recently, Jian et al., proposed a new algorithm, namely method of quasi-strongly sub-feasible directions \[14\], for solving problem (NIO). This new algorithm can ensure that the feasibility of constraints is nondecreasing, i.e., \(I^-(x^k) \subseteq I^-(x^{k+1})\), and can also ensure that the iteration points always enter into the feasible set \(\Theta\) after a finite number of iterations. From the above relationship for \(I^-(x)\), it is obviously that \(|I^-(x^k)| \leq |I^-(x^{k+1})|\), where \(|I^-(x)|\) means that the number of elements in set \(I^-(x)\). Moreover, the numerical results in \[14\] show that the calculational performance of this algorithm is more promising.

In this paper, motivated by the algorithm ideas in \[13\] and the method of quasi-strongly sub-feasible directions in \[14\], a new algorithm is proposed for solving problem (NIO). The main search direction is obtained by a convex combination of \(d_0\) and a mere feasible direction \(d_2\) which are generated by solving a QP subproblem and a SLE, respectively. In order to overcome the Maratos effect and obtain the superlinear convergence, the second-order correction direction is obtained by solving another SLE. To reducing the possible computational cost, the working set is introduced. Moreover, the coefficient matrix of SLEs only consists of the constraints and
their gradients corresponding to the working set. Finally, the convergence rate is proved to be locally one-step superlinear under a condition weaker than the strong second-order sufficient condition and without the strict complementarity. Some elementary numerical results are reported to show that the proposed algorithm is promising.

The paper is organized into six sections. In Section 2 we present our algorithm and its properties. In Sections 3 and 4 we show that this algorithm possesses global and superlinear convergence, respectively. In Section 5 we present the results of some elementary numerical experiments. Finally, Section 6 contains some conclusions for the proposed algorithm.

For notational simplicity, for a point \( x \in \mathbb{R}^n \) and a set \( J \subseteq I \), denote
\[
\bar{f}_i(x) := \begin{cases} f_i(x), & \text{if } i \in I^-(x), \\ f_i(x) - \Upsilon(x), & \text{otherwise}, \end{cases}, \quad I(x) := \{ i \in I : \bar{f}_i(x) = 0 \},
\]
\[
f_J(x) := \{ f_i(x), i \in J \}, \quad g_J(x) := \{ g_i(x), i \in J \}.
\]

2. Description of algorithm. In this section, based on a multiplier function defined below, we first introduce the working set, and then present a new algorithm with arbitrary initial point for solving problem (NIO).

First of all, the following basic assumption for problem (NIO) is assumed to be satisfied throughout the paper.

**Assumption 1.** The functions \( f_i(x) (i \in \{0\} \cup I) \) are all continuously differentiable and gradient vectors \( \{ g_i(x) : i \in I(x) \} \) are linearly independent for each \( x \in \mathbb{R}^n \).

In order to define the working set, we make use of the following function \( \varphi : \mathbb{R}^{n+m} \to [0, +\infty) \)
\[
\varphi(x, \lambda) = \| \phi(x, \lambda) \|^{\nu}, \nu \in (0, 1),
\]
which is named as multiplier function when \( \lambda \) is the corresponding multiplier vector, and where the function \( \phi : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m} \) is defined by
\[
\phi(x, \lambda) := \left( \nabla_x \mathcal{L}(x, \lambda) \min \{ -f_i(x), \lambda_i \} \right).
\]

It is easy to verify that \( (x, \lambda) \) is the KKT pair of problem (NIO) if and only if \( \varphi(x, \lambda) = 0 \). Moreover, similar to the proof of Theorem 3.7 in [4], we have \( \varphi(x, \lambda) \) is an accurate identification function for \( I(x) \), where \( (x, \lambda) \) is the KKT pair of problem (NIO).

The corresponding multiplier vector \( \lambda^k \) associated with iteration point \( x^k \) is defined as follows
\[
\lambda^0 := z^0, \quad \lambda^k := \lambda^{k-1}, \quad k > 0,
\]
where \( z^0 \) and \( \lambda^{k-1} \) are quoted in the following Algorithm 1. Then, the working set by use of (1) and (2) is defined as follows
\[
I_k := \{ i \in I : \bar{f}_i(x^k) + \varphi(x^k, \lambda^k) \geq 0 \}.
\]

Jian [11] showed that \( I_k \equiv I(x^*) \) when the iteration point \( (x^k, \lambda^k) \) is sufficiently close to the KKT pair \( (x^*, \lambda^*) \) of problem (NIO) under some suitable conditions and the Mangasarian-Fromovitz constraint qualification holds at \( (x^*, \lambda^*) \). In Lemma 4.4(i), we will show that \( I_k \) yielded by (3) can also exact identification of \( I(x^*) \) when the iteration point is sufficiently close to \( x^* \).

Now, we present our new algorithm for solving problem (NIO) as follows.
Algorithm 1
Parameters: \(0 < \gamma, \eta, \varepsilon, \theta, \nu, \sigma < 1, \theta < \sigma, \alpha \in (0, 0.5), \rho > 1, \tau \in (2, 3), z_{\text{max}} > z_{\text{min}} > 0\).
Data: \(x^0 \in \mathbb{R}^n\), a symmetric positive definite matrix \(H_0 \in \mathbb{R}^{n \times n}\) and \(z^0\) with \(z^0_i \in (z_{\text{min}}, z_{\text{max}}], i \in I\).
Set \(k := 0\).

**Step 1.** Compute \(\varphi(x^k, \lambda^k)\) from (1) and (2), and then generate the working set \(I_k\) by (3).

**Step 2.** Solve the following QP subproblem

\[
\begin{align*}
\min_{d^k} &\quad g_0(x^k)^T d^k + \frac{1}{2} d^T H_k d \\
\text{s.t.} &\quad \bar{f}_i(x^k) + g_i(x^k)^T d^k \leq 0, \quad i \in I_k
\end{align*}
\]

(4) to get a unique solution \(d^k_0\) and the corresponding KKT multiplier vector \(\lambda_{I_k}\), and let \(\lambda^k = (\lambda_{I_k}, 0_{I_k'})\), where \(0\) denotes zero vector with appropriate dimension throughout the following sections. If \((d^k_0, \Upsilon(x^k)) = (0, 0)\), then \(x^k\) is a KKT point of problem (NIO) and stop, otherwise, go to Step 3.

**Step 3.** Solve the following SLE

\[
M_k \left( \begin{array}{c} d \\ h \end{array} \right) := \begin{pmatrix} H_k & N_k \\ N_k^T & -D_k \end{pmatrix} \left( \begin{array}{c} d \\ h \end{array} \right) = \begin{pmatrix} 0 \\ -\left(|d^k_0|^\tau + \Upsilon(x^k)^\sigma\right)\varpi - F(x^k + d^k_0) \end{pmatrix}
\]

(5)
to get a solution \((d^k_1, h^k_1)\), where \(\varpi = (1, \ldots, 1)^T \in \mathbb{R}^{|I_k|}\), and

\[
\left\{ \begin{array}{l}
N_k = (g_i(x^k), i \in I_k), \\
D_k = \text{diag}(D^k_i, i \in I_k), \\
D^k = ||\bar{f}_i(x^k)||/||\bar{f}_i(x^k) + g_i(x^k)^T d^k_0|| + ||d^k_0||, i \in I_k, \\
F(x^k + d^k_0) = (f_i(x^k + d^k_0) - f_i(x^k) - g_i(x^k)^T d^k_0, i \in I_k).
\end{array} \right.
\]

(6)

Then, let \(d^k := d^k_0 + d^k_1\).

**Step 4.** Let \(t = 1\). (a) If the inequalities

\[
\left\{ \begin{array}{l}
\varepsilon_k f_0(x^k + td^k) \leq \varepsilon_k f_0(x^k) + \alpha t g_0(x^k)^T d^k_0 + \rho(1 - \alpha) t \Upsilon(x^k)^\rho, \\
f_i(x^k + td^k) \leq \max(0, \Upsilon(x^k) - \alpha t (||d^k_0||^\tau + \Upsilon(x^k)^\tau)), \quad i \in I, \\
|\tau^{-}(x^k + td^k)| \geq |\tau^{-}(x^k)|,
\end{array} \right.
\]

(7)
are satisfied, where \(\varepsilon_k > 0\) if \(\Upsilon(x^k) = 0\), and \(\varepsilon_k \geq 0\) if \(\Upsilon(x^k) > 0\), then let \(t_k = t\), and go to Step 7, otherwise, go to (b).

(b) Let \(t := \frac{1}{k} t\).

If \(t < \varepsilon\), then go to Step 5, otherwise, go to (a).

**Step 5.** Solve SLE

\[
M_k \left( \begin{array}{c} d \\ h \end{array} \right) = \begin{pmatrix} 0 \\ -||d^k_0|| + \Upsilon(x^k)^\sigma\varpi \end{pmatrix}
\]

(8)
to get a solution \((d^k_2, h^k_2)\), and compute \(\beta_k \in (0, 1]\) by the following formula

\[
\beta_k g_0(x^k)^T d^k_2 + (1 - \beta_k) g_0(x^k)^T d^k_0 \leq \theta g_0(x^k)^T d^k_0 + \Upsilon(x^k)^\theta.
\]

(9)
Let

\[
\bar{d}^k := (1 - \beta_k) d^k_0 + \beta_k d^k_2.
\]

(10)
Step 6. Compute the steplength $t_k$ which is the first number $t$ in the sequence \{1, \eta, \eta^2, \ldots\} that satisfies the following inequalities
\[
\begin{align*}
\varepsilon_k f_0(x^k + t\bar{d}^k) &\leq \varepsilon_k f_0(x^k) + \gamma t g_0(x^k)^T \bar{d}^k + \rho(1 - \gamma)t \Upsilon(x^k)\|\bar{d}^k\|_2^2, \\
f_i(x^k + t\bar{d}^k) &\leq \max\{0, \Upsilon(x^k) - \gamma t / \beta_k (\|\bar{d}^k\|_2^2 + \Upsilon(x^k)\})\}, \quad i \in I,
\end{align*}
\] (11)
and let $d^k = \bar{d}^k$.

Step 7. Compute a new symmetric positive definite matrix $H_{k+1}$ by some suitable techniques, set $x^{k+1} := x^k + t_k d^k$, $k := k + 1$, and go to Step 1.

Remark 1. (i) Let $d_0^k$ be an optimal solution of subproblem (4), combining the fact that subproblem (4) always has a feasible solution $d = 0$, it holds that
\[
g_0(x^k)^T d_0^k \leq \frac{1}{2} (d_0^k)^T H_k d_0^k < 0
\] (12)
under the conditions of $d_0^k \neq 0$ and the positive definiteness of $H_k$. Thus, $d_0^k$ is a descent direction for problem (NIO).

(ii) Based on the structure of Algorithm 1, it is obviously that our algorithm is very different from the one of [13]. First of all, we make use of the working set $I_k$ to remove those constraints that are locally irrelevant, and thus the scale of subproblem (4) and SLEs (5) as well as (8) are largely decreased, accordingly, the cost of solving subproblem and SLEs could be reduced to some extent per each iteration. Secondly, instead of an explicit formula, a high-order direction is generated by solving a system of linear equations which also only includes the estimates of the active constraints and their gradients, and this is generally more efficient than computing an inverse matrix. Finally, instead of arc search, the line search used in our new algorithm can also well combine the initialization and optimization processes, and meanwhile the line search conditions of Algorithm 1 are weaken than that of conditions in [13].

(iii) In each iteration of Algorithm 1, if the condition (a) in Step 4 is satisfied for a suitable steplength $t$, then we do not need to perform Step 5 and Step 6, and can go to Step 7 directly. Thus, SLE (8) and $\beta_k$ as well as line search inequalities (11) do not need to be computed, the computational cost of per iteration could be reduced greatly. Moreover, we can also prove that the condition (a) in Step 4 is always hold for $t = 1$ after a finite number of iterations (see Theorem 3), and the proposed algorithm is to be shown superlinearly convergent with this condition (see Theorem 5).

(iv) The technique of convex combination of the form (10) was first introduced by Mayne and Polak [18] as “bending” technique to ensure the feasibility of the iteration points. Subsequently, Bonnans, et.al., propose a new feasible SQP algorithm, by modifying this convex combination technique, for solving problem (NIO) in [2]. Recently, Jian, et.al., present a strongly sub-feasible SQP algorithm for solving problem (NIO) [10] with the help of this convex combination formula.

(v) In Algorithm 1, we set $\varepsilon_k > 0$ if $\Upsilon(x^k) = 0$, and $\varepsilon_k \geq 0$ if $\Upsilon(x^k) > 0$. It does not affect any of the analysis if $\varepsilon_k$ in the first inequality of (7) and (11) is always set to a positive constant. The goal is to allow $\varepsilon_k = 0$ in the case $\Upsilon(x^k) > 0$ to reduce (if possible) the computational cost of the line search and obtain a satisfactory numerical performance, since in this case the first inequality of (7) or (11) does not work.

The following lemmas shows some important properties of the proposed Algorithm 1.
Lemma 2.1. Suppose that Assumption 1 holds and matrix \( H_k \) is positive definite. Then,
(i) if Algorithm 1 stops at Step 2 with \((d_0^k, \Upsilon(x^k)) = (0, 0)\), then \( x^k \) is a KKT point of problem \((\text{NIO})\);
(ii) the matrix \( M_k \) defined in (9) is nonsingular, therefore, SLEs (6) and (8) have a unique solution, respectively;
(iii) \( g_0(x^k)^T \bar{d}^k \leq -\frac{1}{2} \theta(d_0^k)^T H_k d_0^k + \Upsilon(x^k)\theta \), \( g_i(x^k)^T \bar{d}^k \leq -\theta_k(\|d_0^k\| + \Upsilon(x^k)\sigma) \), \( \forall i \in I(x^k) \).

Proof. (i) In view of the KKT conditions of subproblem (11) and \((d_0^k, \Upsilon(x^k)) = (0, 0)\), it follows that this conclusion holds immediately.

(ii) The proof of this part is similar to that of Lemma 2.4 in [5], and is omitted here.

(iii) First, it holds that
\[ g_0(x^k)^T \bar{d}^k \leq \theta g_0(x^k)^T d_0^k + \Upsilon(x^k)\theta \leq -\frac{1}{2} \theta(d_0^k)^T H_k d_0^k + \Upsilon(x^k)\theta \]
from (9), (11) and (12). Then, combining the constraints of subproblem (11) with (5), it follows that
\[ g_i(x^k)^T \bar{d}^k \leq 0, \quad g_i(x^k)^T d_0^k = -\|d_0^k\| + \Upsilon(x^k)\sigma, \quad \forall i \in I(x^k). \]
Again from (11) and for all \( i \in I(x^k) \), we have
\[ g_i(x^k)^T \bar{d}^k = (1 - \beta_k)g_i(x^k)^T d_0^k + \beta_k g_i(x^k)^T d_0^k \leq -\beta_k(\|d_0^k\| + \Upsilon(x^k)\sigma). \]

From Lemma 2.1(iii), we know that \( \bar{d}^k \) is an improved direction in a sense. Therefore, if Algorithm 1 does not stop at Step 2, i.e., \((d_0^k, \Upsilon(x^k)) \neq (0, 0)\), we are always able to obtain the next iteration point \( x^{k+1} \) from \( x^k \). According to the structure of Algorithm 1 and Lemma 2.1, the following lemma holds obviously.

Lemma 2.2. Suppose that Assumption 1 holds. Then,
(i) Algorithm 1 is well-defined;
(ii) if there exists an iteration index \( k_0 \) such that \( x^{k_0} \in \Theta \), then \( x^k \in \Theta \) and \( f_0(x^{k+1}) \leq f_0(x^k) \) for all \( k \geq k_0 \);
(iii) if \( x^k \not\in \Theta \) and \( x^{k+1} \not\in \Theta \), then \( \Upsilon(x^{k+1}) < \Upsilon(x^k) \).

3. Global convergence. In this section, we will establish the global convergence of Algorithm 1. If Algorithm 1 stops at \( x^k \), it follows that \( x^k \) is a KKT point of problem \((\text{NIO})\) from Lemma 2.1(i) and Step 2 of Algorithm 1. Now, we assume that Algorithm 1 generates an infinite iteration sequence \( \{x^k\} \), and prove that any accumulation point \( x^* \) of \( \{x^k\} \) is a KKT point of problem \((\text{NIO})\) under some assumptions. For this purpose, the following assumption is necessary.

Assumption 2. The matrix sequence \( \{H_k\} \) is uniformly positive definite, i.e., there exists two positive constants \( a \leq b \) such that
\[ a\|d\|^2 \leq d^T H_k d \leq b\|d\|^2, \quad \forall d \in \mathbb{R}^n, \forall k. \tag{13} \]
Denote the active set of subproblem (11) by \( \Lambda_k := \{i \in I_k : \tilde{f}_i(x^k) + g_i(x^k)^T d_0^k = 0\} \), and suppose that \( x^* \) is a given accumulation point of \( \{x^k\} \). Then, in view of
$I^+(x^k), I^-(x^k), \Lambda_k$ and $I_k$ are all subsets of the finite set $I$, we can assume that there exists an infinite index set $K$ such that

$$\begin{align*}
&I^+(x^k), I^-(x^k), \Lambda_k \text{ and } I_k \text{ are all subsets of the finite set } I, \\
&k \in K,
&I^+(x^k) = I^-, I^+(x^k) = I^+, \Lambda_k = \Lambda, I_k = I, \forall k \in K
\end{align*}$$

holds from Lemma 2.2(ii). The results established in the following lemma are important in the later discussions.

**Lemma 3.1.** Suppose that Assumptions 7 and 11 hold. Then

(i) the sequence $\{d_0^k\}_K$ is bounded;

(ii) there exists a constant $\bar{\epsilon} > 0$ such that $||M_k^{-1}|| \leq \bar{\epsilon}$ for all $k \in K$;

(iii) the sequence $\{d_0^k\}_K, \{d_0^2\}_K, \{d^k\}_K$ and $\{h_i^k\}_K$ are all bounded;

(iv) $\lim_{k \in K}(d_0^k, \bar{\nabla}x(x_k)) = (0, 0), \lim_{k \in K}d_0^k = \lim_{k \in K}d^k = \lim_{k \in K}d_1^k = 0$.

**Proof.** (i) First of all, by $\lim_{k \in K}x^k = x^*$ and the continuous of $g_0(x)$, there exists a constant $\zeta > 0$ such that $||g_0(x_k)|| \leq \zeta$ for all $k \in K$. Then, from (13) and $d = 0$ is a feasible solution of subproblem (3), it follows that $\{d_0^k\}_K$ is bounded.

(ii) Suppose by contradiction that there exists an infinite index set $K' \subseteq K$ such that (14) holds and

$$\begin{align*}
&||M_k^{-1}|| \rightarrow \infty, k \in K', \\
&M_k \rightarrow M_* := \begin{pmatrix} H_* & N_* \\ N_*^T & -D_* \end{pmatrix}, d_0^k \rightarrow d_0^*, I(x_k) \rightarrow I(x^*),
\end{align*}$$

where

$$\begin{align*}
N_* &= (g_i(x^*), i \in \bar{I}), \\
D_* &= \text{diag}(D_i^*, i \in \bar{I}), \\
D_i^* &= \|f_i(x^*)\| (\|f_i(x^*)\| + g_i(x^*)^T d_0^* + ||d_0^*||), i \in \bar{I}.
\end{align*}$$

From the definition of $I(x^*)$ and $\bar{I}$, it follows that $I(x^*) \subseteq \bar{I}$. Then, it is easy to get $D_i^* \geq 0, \forall i \in \bar{I}$ and $D_i^* > 0, \forall i \in \bar{I} \backslash I(x^*)$. Therefore, in view of Lemma 2.2(ii), we can get that $M_k$ is nonsingular, so $||M_k^{-1}|| \rightarrow ||M_*^{-1}||, k \in K'$, which contradicts with (15), thus the conclusion (ii) holds.

(iii) Combining (3), (8), (10) with parts (i) and (ii), part (iii) holds immediately.

(iv) The proof of part (iv) is similar to Lemma 3.2 in [13] and is omitted here. \qed

Based on Lemma 6.1, the global convergence theorem of Algorithm 1 is presented as follows.

**Theorem 3.2.** Suppose that Assumptions 7 and 11 hold. Then Algorithm 1 either stops at a KKT point $x^k$ of problem (NIO) after a finite number of iterations or generates an infinite sequence $\{x^k\}$ such that each accumulation point $x^*$ of $\{x^k\}$ is a KKT point of problem (NIO). Furthermore, there exists an index set $K$ such that $\{(x^k, \lambda^k)\}_K$ converges to the KKT pair $(x^*, \lambda^*)$ of problem (NIO).

**Proof.** By choosing an infinite index set $K$ such that (13) holds, and denoting matrix $A_k = (g_i(x_k), i \in \Lambda)$. In view of $(x^k, d_0^k, \bar{\nabla}x(x_k)) \rightarrow (x^*, 0, 0), k \in K$, we can conclude that $\Lambda \subseteq I(x^*) = \{i \in I : \bar{f}_i(x^*) = 0\} \subseteq \bar{I}$, which together with Assumption 1 shows that $A_k^T A_k$ is nonsingular (see Remark 2) for $k \in K$ large enough, since $A_k \stackrel{K}{\rightarrow} A_*,$ where $A_* := (g_i(x^*)), i \in \Lambda$.

Denote $\lambda^k_A := (\lambda^k, i \in \Lambda), \lambda^k = (\lambda^k, 0_{I \backslash \Lambda}),$ from the KKT condition of subproblem (4), we have

$$\begin{align*}
&g_0(x^k) + H_k d_0^k + \sum_{i \in \bar{I}} \lambda_i^k g_i(x^k) = 0, \\
&\bar{f}_i(x^k) + g_i(x^k)^T d_0^k \leq 0, \lambda_i^k \geq 0, \lambda_i^k (\bar{f}_i(x^k) + g_i(x^k)^T d_0^k) = 0, \forall i \in \bar{I},
\end{align*}$$

(16)
which further implies that
\[
\lambda_k^i = -(A_k^T A_k)^{-1} A_k^T (g_0(x^k) + H_k d_0^k) \rightarrow -(A_*^T A_*)^{-1} A_*^T g_0(x^*) \quad \text{def} \quad \lambda_A^*
\]
holds for \(k\) large enough.

Let the multiplier vector \(\lambda^* = (\lambda_A^*, 0)\), then \(\lim_{k \to K} \lambda_k = \lambda^*\). Passing to the limit \(k \in K\) and \(k \to \infty\) in (10), it follows that
\[
g_0(x^*) + N_* \lambda^* = 0, \quad f_i(x^*) \leq 0, \quad \lambda_i^* \geq 0, \quad f_i(x^*) \lambda_i^* = 0, \quad i \in I,
\]
which shows that \((x^*, \lambda^*)\) is a KKT pair of problem (NIO).

\begin{remark}
From the definitions of \(N_k\) and \(D^k\) in (6), it holds that the matrix \((N_k^T N_k + D^k)\) is nonsingular and positive definite under Assumption 4.
\end{remark}

4. Rate of convergence. In order to show that the entire sequence \(\{x^k\}\) converges to a KKT point \(x^*\) of problem (NIO), the following additional assumption is necessary.

**Assumption 3.** (i) The functions \(f_i(x)\) (\(i \in \{0\} \cup I\)) are all second-order continuously differentiable.

(ii) The KKT pair \((x^*, \lambda^*)\) is a quasi-regular point of problem (NIO), i.e., the matrix \(M(J)\) is nonsingular at \((x^*, \lambda^*)\), \(\forall J \subseteq I(x^*)\) \(I'_+\) (empty set include), where
\[
M(J) := \begin{bmatrix}
\nabla^2 x_L(x^*, \lambda^*) & -N_I^+_i \\
N_I^+_i & 0
\end{bmatrix}, \quad I'_+ := \{i \in I(x^*) : \lambda_i^* > 0\}.
\]

**Remark 3.** The definition of quasi-regular point comes from the Definition 4.3 in [23], which includes the detailed analysis about it.

In general, the local superlinear convergence of SQP or SSLE (i.e., Sequential system of linear equations) algorithms is established under the assumptions that both the strict complementary and the second-order sufficient condition (shorted by SOSC) hold, in which case, SOSC is equivalent to the stronger second-order sufficient condition (shorted by SSOSC). In this paper, we do not need to assume that the strict complementary condition holds, what’s more, we could get the relationship among Assumption (ii), SOSC and SSOSC in terms of the definition of quasi-regular point as follows

\[
\text{SSOSC} \Rightarrow \text{Assumption (ii)} \Rightarrow \text{SOSC}.
\]

The main purpose of Assumption (ii) is to ensure that \((x^*, \lambda^*)\) is an isolated KKT point of problem (NIO). In order to show that the entire sequence \(\{x^k\}\) converges to \(x^*\), the following important result [19] is cited here.

**Lemma 4.1.** Assume that \(\omega^* \in \mathbb{R}^n\) is an isolated accumulation point of the sequence \(\{\omega^k\} \subseteq \mathbb{R}^n\), and for every subsequence \(\{\omega^k\}_K\) converging to \(\omega^*\), there exists an infinite subset \(K' \subseteq K\) such that \(\|\omega^{k+1} - \omega^k\|_{K'} \to 0\). Then the entire sequence \(\{\omega^k\}\) converges to \(\omega^*\).

**Lemma 4.2.** Suppose that Assumptions [1] and [3] hold. If \(\{x^k\}_K \to x^*\), then

(i) \(I_k = I(x^*)\) for all \(k \in K\) large enough;

(ii) \(\{d_0^k, d_1^k, d_2^k, d_3^k\} \to \{0, 0, 0, 0\}, \quad k \in K\).

**Proof.** (i) According to Theorem 3.3 and the uniqueness of the KKT multiplier, it is easy to get \((x^k, \lambda^k) \to (x^*, \lambda^*), \quad k \in K\). From Assumption (ii) and Theorem 4.4
in [23], we have \( \varphi(x, \lambda) \) is the active-set identification function. So, it holds that \( I_k = I(x^*) \) for all \( k \in K \) large enough.

(ii) In view of Lemma 5.1(ii) and (iv), part (ii) holds immediately. \( \square \)

Together with the lemmas above, the strong convergence of Algorithm 1 is presented as follows.

**Theorem 4.3.** Suppose that Assumptions 1, 2, and 3 hold. Then the entire sequence \( \{x^k\} \) generated by Algorithm 1 converges to \( x^* \), i.e, Algorithm 1 is strongly convergent.

**Proof.** From Assumption 3(ii) and Theorem 4.4 [23], it follows that \( x^* \) is an isolated accumulation point of \( \{x^k\} \). Then combining with Lemma 4.2(ii) and \( d_k = d_0^k + d_1^k \) or \( d_k = d^k \), we have

\[
\|x^{k+1} - x^k\| = \|t_k d_k\| \to 0, \quad k \in K, \quad k \to \infty,
\]

which further implies that the entire sequence \( \{x^k\} \) converges to \( x^* \) in view of Lemma 4.1. \( \square \)

The following lemma is a direct result of Theorem 4.3, Lemmas 4.1 and 4.2 and it plays an important role in the later analysis of the convergence rate.

**Lemma 4.4.** Suppose that Assumptions 1, 2, and 3 hold. Then

(i) \( I^*_k \subseteq \Lambda_k \subseteq I(x^*) \equiv I_k \) for \( k \) large enough;

(ii) \( \lim_{k \to +\infty} (d^k_0, d^k_1, d^k_2, d^k, d^k_k) = (0, 0, 0, 0) \);

(iii) \( \lim_{k \to +\infty} \Upsilon(x^k) = \Upsilon(x^*) = 0, \lim_{k \to +\infty} \lambda^k = \lambda^* \).

**Lemma 4.5.** Suppose that Assumptions 1, 2, and 3 hold. Then

\[
||d^k_1|| = O(||d^k_0||^\tau + \Upsilon(x^k)^\tau) + O(||d^k_0||^4 + o(\Upsilon(x^k)^4)), \quad ||h^k_1|| = O(||d^k_0||^2) + O(\Upsilon(x^k)^2).
\]

**Proof.** In view of (53), (6), \( \tau \in (2, 3) \) and Lemma 4.1(ii) as well as Lemma 4.5, we have by Taylor expansion

\[
\left\| \left( \begin{array}{c} d^k_k \\ h^k_1 \\ \end{array} \right) \right\| = O(||d^k_0||^\tau + \Upsilon(x^k)^\tau + F(x^k + d^k_0))
\]

\[
= O(||d^k_0||^\tau) + O(\Upsilon(x^k)^\tau) + O(\Upsilon(x^k)^\tau) + O(\Upsilon(x^k)^\tau)
\]

which implies that Lemma 4.5 holds immediately. \( \square \)

To ensure that the steplength \( t_k \equiv 1 \) for \( k \) large enough without the strict complementary assumption, an additional assumption as follows is necessary.

**Assumption 4.** Suppose that the KKT pair \((x^*, \lambda^*)\) and the matrix \( H_k \) satisfy

\[
||\langle \nabla^2_x \mathcal{L}(x^*, \lambda^*) - H_k \rangle d^k_0 || = o(||d^k_0||).
\]

**Theorem 4.6.** Suppose that Assumptions 1, 2, and 3 hold. Then the unit steplength is always accepted in Step 4 for \( k \) large enough.

**Proof.** First of all, we prove that the second and the last inequalities of (7) hold for \( t = 1 \) and \( k \) large enough.

For \( i \not\in I(x^*) \), i.e., \( f_i(x^*) < 0 \), in view of \( (x^k, d^k_0, d^k_1, \Upsilon(x^k)) \to (x^*, 0, 0, 0) \) \((k \to +\infty)\), it holds that \( d^k = d^k_0 + d^k_1 \to 0 \) \((k \to +\infty)\). Thus we can conclude that the second and the third inequalities of (7) are satisfied for \( t = 1 \) and \( k \) large enough.
For \( i \in I(x^*) \), since \( \lim_{k \to +\infty} f_i(x^k) = f_i(x^*) = 0 \) and \( \lim_{k \to +\infty} \Upsilon(x^k) = 0 \) it holds that from (17) \[
abla_i f(x^k) \to 0, \quad D_k^i \to 0, \quad D_k^i = o(||\nabla_i f(x^k) + g_i(x^k)^T d_k^0||) + o(||d_k^0||).
\]

On one hand, from (4) and Lemma 4.5 we have
\[
g_i(x^k)^T D_k^i = -||d_k^0||^r - \Upsilon(x^k)^r - f_i(x^k) + g_i(x^k)^T d_k^0 + D_k^ih_k^i \\
= -||d_k^0||^r - \Upsilon(x^k)^r - f_i(x^k) + g_i(x^k)^T d_k^0 + o(||\nabla_i f(x^k) + g_i(x^k)^T d_k^0||) + O(||d_k^0||^3) + o(\Upsilon(x^k)^r).
\] (17)

On the other hand, it holds from Lemma 4.5 and (17) that
\[
f_i(x^k + d_k^0) = f_i(x^k) + g_i(x^k)^T d_k^0 + O(||d_k^0||^2) \\
= f_i(x^k) + g_i(x^k)^T d_k^0 + O(||d_k^0||^3) + o(\Upsilon(x^k)^r)
\] and
\[
f_i(x^k + d_k^0) = -||d_k^0||^r - \Upsilon(x^k)^r - ||\nabla_i f(x^k) + g_i(x^k)^T d_k^0|| + o(||\nabla_i f(x^k) + g_i(x^k)^T d_k^0||) + o(\Upsilon(x^k)^r), \forall i \in I(x^*) \cap I^-.
\] (18)

In view of (18) and \( \tau \in (2, 3) \), it follows that \( f_i(x^k + d_k^0) \leq 0 \), for \( i \in I(x^*) \cap I^- \) and \( k \) large enough, i.e., the third and the second inequalities of (4) for \( i \in I(x^*) \cap I^- \) hold.

For \( i \in I(x^*) \cap I^+ \), it follows from (19), \( \tau \in (2, 3) \), \( \alpha \in (0, \frac{1}{2}) \) and the second inequality of (7) that
\[
f_i(x^k) + d_k^0 - \max \{0, \Upsilon(x^k) - \alpha(||d_k^0||^r + \Upsilon(x^k)^r)\} \\
= -(1 - \alpha)(||d_k^0||^r + \Upsilon(x^k)^r) - ||\nabla_i f(x^k) + g_i(x^k)^T d_k^0|| + o(||\nabla_i f(x^k) + g_i(x^k)^T d_k^0||) + O(||d_k^0||^3) \\
\leq 0,
\] (20)
i.e., the second inequality of (4) holds for \( i \in I(x^*) \cap I^+ \) and \( k \) large enough.

Summarizing the analysis above, the second and the last inequalities of (4) are satisfied for \( t = 1 \) and \( k \) large enough.

Then we show that the first inequality of (4) holds for \( t = 1 \) and \( k \) large enough.

By Taylor expansion and Lemma 4.5, we have
\[
\Gamma_k := f_0(x^k + d_k^0) - f_0(x^k) - \alpha g_0(x^k)^T d_k^0 - \rho(1 - \alpha)\Upsilon(x^k)^r \\
= g_0(x^k)^T d_k^0 + \frac{1}{2}(d_k^0)^T \nabla^2 f_0(x^k) d_k^0 - \alpha g_0(x^k)^T d_k^0 \\
- \rho(1 - \alpha)\Upsilon(x^k)^r + o(||d_k^0||^2) \\
= g_0(x^k)^T (d_k^0 + d_k^b) + \frac{1}{2}(d_k^0)^T \nabla^2 f_0(x^k) d_k^0 - \alpha g_0(x^k)^T d_k^0 \\
- \rho(1 - \alpha)\Upsilon(x^k)^r + o(||d_k^0||^2) + o(\Upsilon(x^k)^r).
\] (21)

Again from the KKT conditions of subproblem (4) and Lemma 4.5 it follows that
\[
g_0(x^k)^T d_k^0 = -(d_k^0)^T H_k d_k^0 - \sum_{i \in \Lambda_k} \lambda_i^k g_i(x^k)^T d_k^0 \\
and
\[
g_0(x^k)^T (d_k^0 + d_k^b) = -(d_k^0)^T H_k d_k^0 - \sum_{i \in \Lambda_k} \lambda_i^k g_i(x^k)^T (d_k^0 + d_k^b) + o(||d_k^0||^2) + o(\Upsilon(x^k)^r)
\] (22)
hold. For \( i \in \Lambda_k \subseteq I(x^*) \), it holds that \( \tilde{f}_i(x^k) + \nabla f_i(x^k)^T d^k_i = 0 \), which together with (14) and (19) as well as \( Y(x^k) = o(Y(x^k)^\sigma) \) implies that

\[
f_i(x^k + d^k) = -||d^k_i||^2 + Y(x^k)^\sigma + o(||d^k_i||^2) + o(Y(x^k)^\sigma), \quad i \in \Lambda_k.
\]

(24)

Again from Lemma 4.5 and \( Y(x^k) = o(Y(x^k)^\sigma) \), we have

\[
f_i(x^k + d^k) = f_i(x^k) + g_i(x^k)^T (d^k_0 + d^k_i) + \frac{1}{2}(d^k_0)^T \nabla^2 f_i(x^k) d^k_i + o(||d^k_i||^2) + o(Y(x^k)^\sigma).
\]

(25)

Combining (24) with (25), it follows that

\[
- \sum_{i \in \Lambda_k} \lambda_i^k g_i(x^k)^T (d^k_0 + d^k_i) = \sum_{i \in \Lambda_k} \lambda_i^k \tilde{f}_i(x^k) + \frac{1}{2} \sum_{i \in \Lambda_k} \lambda_i^k (d^k_0)^T \nabla^2 f_i(x^k) d^k_i + o(||d^k_i||^2) + O(Y(x^k)^\sigma).
\]

(26)

Thus, substituting (26) into (21), we have

\[
g_0(x^k)^T (d^k_0 + d^k_i) = -(d^k_0)^T H_k d^k_0 + \sum_{i \in \Lambda_k} \lambda_i^k \tilde{f}_i(x^k) + \frac{1}{2} (d^k_0)^T \nabla^2 f_i(x^k, \lambda^k) (d^k_0) + o(||d^k_i||^2) + O(Y(x^k)^\sigma).
\]

(27)

Now, substituting (27) and (28) into (21), we have

\[
\Gamma_k = -(d^k_0)^T H_k d^k_0 + \sum_{i \in \Lambda_k} \lambda_i^k \tilde{f}_i(x^k) + \frac{1}{2} (d^k_0)^T \nabla^2 f_i(x^k, \lambda^k) (d^k_0) - \alpha g_0(x^k)^T d^k_0
\]

\[
- \rho(1 - \alpha) Y(x^k)^\theta + o(||d^k_0||^2) + O(Y(x^k)^\sigma)
\]

\[
= (\alpha - \frac{1}{2}) (d^k_0)^T H_k d^k_0 + (1 - \alpha) \sum_{i \in \Lambda_k} \lambda_i^k \tilde{f}_i(x^k) + \frac{1}{2} (d^k_0)^T \nabla^2 f_i(x^k, \lambda^k) - H_k
\]

\[
d^k_0 - \rho(1 - \alpha) Y(x^k)^\theta + o(||d^k_0||^2) + O(Y(x^k)^\sigma),
\]

which further show that from Assumptions 4.4 \( \lambda_i^k \tilde{f}_i(x^k) \leq 0 \) and \( \alpha \in (0, \frac{1}{2}) \) as well as \( \theta < \sigma \)

\[
\Gamma_k \leq (\alpha - \frac{1}{2}) a||d^k_0||^2 + o(||d^k_0||^2) - \rho(1 - \alpha) Y(x^k)^\theta + o(Y(x^k)^\theta) \leq 0
\]

holds for \( k \) large enough. Hence, the first inequality of (7) holds for \( t = 1 \) and \( k \) large enough. The whole proof is completed.

At the end of this section, based on Theorem 4.6, we can easily obtain that the steplength \( t_k = 1 \) and \( x^{k+1} \in \Theta \) for \( k \) large enough, respectively. Then according to Theorem 4.7 and \( ||d^k_i|| = o(||d^k_0||) \), similar to the proof of Theorem 4.5 in 13, we can prove the rate of convergence of the Algorithm 1 as follows.

**Theorem 4.7.** Under all above mentioned assumptions, we have for \( k \) large enough:

(i) the steplength \( t_k = 1 \);

(ii) \( Y(x^{k+1}) \equiv 0 \) after a finite number of iterations, i.e., \( x^{k+1} \in \Theta \).

**Theorem 4.8.** Suppose that Assumptions 4.4, 4.5 and 4.6 are all satisfied. Then Algorithm 1 is locally one-step superlinearly convergent, i.e., \( ||x^{k+1} - x^*|| = o(||x^k - x^*||) \).
5. Numerical experiments. In this section, some preliminary numerical results are reported, and the computing results show that Algorithm 1 (shorted by ALGO 1) is promising and effective. The algorithm is implemented by using MATLAB R2008a on Windows XP platform, and on a PC with 2.53GHz CPU. For each test problem, we choose $H_0 = E_n$ (i.e., the identity matrix) as the initial guess of the Lagrangian Hessian, and the approximation Hessian matrix $H_k$ is updated by the BFGS formula described in [20].

During the numerical experiments, the parameters are selected as follows:

$$
\gamma = \theta = \alpha = 0.4, \quad \eta = \nu = 0.5, \quad \sigma = 0.55, \quad \rho = 2, \quad \tau = 2.5, \quad \varepsilon = 0.5^3.
$$

The algorithm stops if and only if one of the following termination criterion is satisfied:

(i) $\phi(x^k, \lambda^k) \leq 1.0e - 6$;
(ii) $\|d_0^k\| \leq 1.0e - 6$ and $\Upsilon(x) = 0$,

where $\epsilon$ is a given small constant. The corresponding results are reported by using the performance profiles as described in Dolan and Moré’s paper [3]. Additionally, according to Remark 1(v), we know that the value of parameter $\varepsilon_k$ may influence the computational cost of ALGO 1, especially for some problems with complicated objective functions. In order to more intuitive illustrate this conclusion, we report and compare the numerical results of ALGO 1 in two cases

Case I: $\varepsilon_k \equiv 1$; Case II: $\varepsilon_k = \begin{cases} 0, & \text{if } \Upsilon(x^k) > 0; \\ 1, & \text{if } \Upsilon(x^k) = 0. \end{cases}$

First of all, in order to show the calculation results of ALGO 1 under above two cases, some problems [9, 21] which are listed in the following Table 1 are tested. The corresponding parameters for the test problems are show in Table 1, where "Prob" denotes the number of the test problem in [9, 21], $n/m$ is displayed as the number of variables of the problem and the number of inequality constraints, respectively.

According to the value of $\Upsilon(x^0)$, we note that all initial iteration points are infeasible for the corresponding problems.

Table 1. The corresponding parameters for the test problems (1)

| Prob | n/m | $x^0$       | $\Upsilon(x^0)$ | Prob | n/m | $x^0$       | $\Upsilon(x^0)$ |
|------|-----|------------|----------------|------|-----|------------|----------------|
| 013  | 2/3 | $(-5, -5)^T$ | 5.00          | 098  | 6/16| $(20, 0, \ldots, 0)^T$ | 19.69          |
| 018  | 2/6 | $(1, 5)^T$  | 20.00          | 100  | 7/4 | $(3, \ldots, 3)^T$ | 188.00         |
| 023  | 2/9 | $(-5, -5)^T$ | 11.00          | 108  | 9/14| $(0.1, 5, 0.5, 1.0, 1, 0.5, 0.1)^T$ | 23.26         |
| 030  | 3/7 | $(5, 10, 15)^T$ | 124.00     | 224  | 2/8 | $(-2, -1)^T$ | 5.00           |
| 033  | 3/6 | $(1, 4, 7)^T$ | 2.00          | 225  | 2/5 | $(1, 4)^T$ | 3.00           |
| 038  | 4/8 | $(0, 3, 1, 2)^T$ | 3.00        | 234  | 2/5 | $(1, 3)^T$ | 9.00           |
| 043  | 4/3 | $(-10, 2, -8, 5)^T$ | 236.00     | 250  | 3/8 | $(-3, -3, -3)^T$ | 15.00         |
| 044  | 4/10 | $(-20, \ldots, -20)^T$ | 20.00     | 264  | 4/3 | $(0, 0, 0, 3)^T$ | 6.00           |
| 066  | 3/8 | $(0, 0, 100)^T$ | 90.00        | 337  | 3/3 | $(1, 1, 0)^T$ | 1.00           |
| 076  | 4/7 | $(1, 2, 3, 4)^T$ | 7.00         | 354  | 4/5 | $(1, -1, 1, -1)^T$ | 1.00           |

The corresponding numerical results are given in Figure 1. The profile of figure is based on the number of objective function evaluations. The $\rho(t)$ is the (cumulative)
distribution function for the performance ratio within a factor \( t \in \mathbb{R} \). The value of \( \rho(t) \) is the probability that the solver will over the rest of the solvers. The comparative performance results are shown in Figure 1. In view of the results in Figure 1, it is obviously that the performance of ALGO 1 under Case II is always better than that of ALGO 1 under Case I, i.e., the number of objective function evaluations under Case II is always less than that under Case I. For instance, the number of objective function evaluations of problem 043 under Case I is 108, but under Case II the number of objective function evaluations of problem 043 is only 17, and so on. Moreover, we further note that the optimal solutions of each test problem are the same under two cases during the numerical experiments. The parameter \( \varepsilon_k \) only influence the number of objective function evaluations and further influence the CPU time, which will affect the total computational cost of ALGO 1. Therefore, ALGO 1 has some potential value and advantage for solving some problems with more complicated objective functions.

![Figure 1. The performance of ALGO 1 for solving problems in Table 1 under Case I and Case II, respectively.](image)

Note that the test problems in Table 1 are relatively small. In order to show the more clearly effectiveness of ALGO 1 for solving some large scale problems under Case II, the Svanberg [7] problem (in different dimensions and different infeasible initial points), and the series of problems S394 which have adjustable dimension are modified from the problem S394 in [21], are further tested. The

\[
\text{(S394)} \quad \begin{align*}
\min_{x} & \quad f_0(x) = \sum_{i=1}^{n} (x_i^4 + x_i^2 - 0.01x_i) \\
\text{s.t.} & \quad f_1(x) = \sum_{i=1}^{n} x_i^2 - 1 \leq 0.
\end{align*}
\]

parameters of the test problems are given in Table 2. The profile of figure is also based on the number of objective function evaluations, and the performance results are shown in Figure 2. In view of the results in Figure 2, it is obviously that the number of objective function evaluations under Case II is much less than the number of objective function evaluations under Case I. Moreover, the number of iterations and the optimal solutions for each test problem are also the same under two cases. Summarizing the results in Figure 1 and Figure 2, we can conclude that ALGO 1
Table 2. The corresponding parameters for the test problems (2)

| Prob         | n/m | $x_0$          | $\Upsilon(x_0)$ | Prob     | n/m | $x_0$          | $\Upsilon(x_0)$ |
|--------------|-----|----------------|-----------------|----------|-----|----------------|-----------------|
| Svanberg-20  | 20/60 | $(5, \ldots, 5)^T$ | 4.20           | S394-40  | 40/1  | $(0.4, \ldots, 0.4)^T$ | 5.40            |
|              |      | $(0.5, \ldots, 0.5)^T$ | 2.42           |          |       | $(0.3, \ldots, 0.3)^T$ | 5.30            |
| Svanberg-30  | 30/90 | $(5, \ldots, 5)^T$ | 4.20           | S394-70  | 70/1  | $(0.4, \ldots, 0.4)^T$ | 10.20           |
|              |      | $(0.5, \ldots, 0.5)^T$ | 2.50           |          |       | $(0.3, \ldots, 0.3)^T$ | 5.30            |
| Svanberg-40  | 40/120 | $(5, \ldots, 5)^T$ | 4.20           | S394-100 | 100/1 | $(0.4, \ldots, 0.4)^T$ | 15.00           |
|              |      | $(0.5, \ldots, 0.5)^T$ | 2.54           |          |       | $(0.3, \ldots, 0.3)^T$ | 8.00            |
| Svanberg-50  | 50/150 | $(5, \ldots, 5)^T$ | 4.20           | S394-150 | 150/1 | $(0.4, \ldots, 0.4)^T$ | 23.00           |
|              |      | $(0.5, \ldots, 0.5)^T$ | 2.57           |          |       | $(0.3, \ldots, 0.3)^T$ | 12.50           |
| Svanberg-200 | 200/600 | $(5, \ldots, 5)^T$ | 4.20           | S394-200 | 200/1 | $(0.4, \ldots, 0.4)^T$ | 31.00           |
|              |      | $(0.5, \ldots, 0.5)^T$ | 2.64           |          |       | $(0.3, \ldots, 0.3)^T$ | 17.00           |
| Svanberg-300 | 300/900 | $(5, \ldots, 5)^T$ | 4.20           | S394-300 | 300/1 | $(0.4, \ldots, 0.4)^T$ | 31.00           |
|              |      | $(0.5, \ldots, 0.5)^T$ | 2.65           |          |       | $(0.3, \ldots, 0.3)^T$ | 17.00           |

with the parameter $\varepsilon_k$ selected as Case II is encouraging, and also has some great potential value and advantage for solving some practical problems with complicated objective functions.

Then, for further showing the performance of our algorithm, we compare the number of iterations required by ALGO 1 under Case II to those required by Wang-Chen-He’s algorithm [23] (shorted by ALGO-WCH) and algorithm in [10] (denoted by ALGO 2.1), and the corresponding results are shown in the following Figure 3. The optimality tolerance is the same as in [23] and [10], respectively. The left figure of Figure 3 gives the performance of ALGO 1 under Case II and ALGO-WCH, the test problems are the same as in [23]. For some large problems in Table 4 in [10], the corresponding performance of ALGO 1 under Case II compared to ALGO 2.1 are shown in the right figure of Figure 3. From the results in Figure 3, it is quite clear that the performance of ALGO 1-II is better than that of ALGO-WCH, ALGO 2.1, respectively; i.e., ALGO 1-II has the most wins compare with ALGO-WCH and ALGO 2.1. This further implies that our new Algorithm 1 is is much more promising and competitive with ALGO-WCH, ALGO 2.1, respectively.
Summarizing the above analysis of numerical results, we can conclude that our new Algorithm 1 is more promising and competitive with some algorithms. Furthermore, by using the selection way (29) of parameter $\varepsilon_k$, we could reduce a certain amount of computation of Algorithm 1, especially for solving some problems which objective functions are complicated and initial iteration points are far from the feasible region of the problem. These also implies that Algorithm 1 has some potential advantage and value.

6. Conclusions. A one-step superlinearly convergent hybrid algorithm is proposed for solving nonlinear programming. First, an improved direction is obtained by a convex combination of the solution of an always feasible QP subproblem and a mere feasible direction. Second, the second-order correction direction is obtained by solving an reduced SLE to overcome the Maratos effect and obtain the superlinear convergence, the coefficient matrix of SLE only consists of the constraints and their gradients corresponding to the new working set. In particular, the convergence rate is proved to be locally one-step superlinear under a condition weaker than the strong second-order sufficient condition and without the strict complementarity. Moreover, the line search in our algorithm can effectively combine the initialization processes with the optimization processes. Finally, some elementary numerical results and comparisons show that the proposed algorithm is promising and competitive.

Acknowledgments. We are very grateful to the associate editor and two anonymous referees for their helpful comments.

REFERENCES

[1] R. H. Byrd, F. E. Curtis and J. Nocedal, Infeasibility detection and SQP methods for nonlinear optimization, *SIAM Journal on Optimization*, 20 (2010), 2281–2299.

[2] J. F. Bonnans, E. R. Panier, A. L. Tits and J. L. Zhou, Avoiding the Maratos effect by means of a nonmonotone line search. II. Inequality constrained problems-feasible iterates, *SIAM Journal on Numerical Analysis*, 29 (1992), 1187–1202.

[3] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming*, 91 (2002), 201–213.

[4] F. Facchinei, A. Fischer and C. Kanzow, On the accurate identification of active constraints, *SIAM Journal on Optimization*, 9 (1998), 14–32.

Figure 3. The left figure shows the performance of ALGO 1-II and ALGO-WCH, the right figure shows the performance of ALGO 1-II and ALGO 2.1.
C. H. Guo, Y. Q. Bai and J. B. Jian, An improved sequential quadratic programming algorithm for solving general nonlinear programming problems, *Journal of Mathematical Analysis and Applications*, **409** (2014), 777–789.

N. I. M. Gould and D. P. Robinson, A second derivative SQP method: Local convergence and practical issues, *SIAM Journal on Optimization*, **20** (2010), 2049–2079.

N. I. M. Gould, D. Orban and Ph. L. Toint, A constrained and unconstrained testing environment, revisited, *ACM Transactions on Mathematical Software*, **29** (2003), 353–372.

M. Heinkenschloss and D. Ridzal, A matrix-free trust-region SQP method for equality constrained optimization, *SIAM Journal on Optimization*, **24** (2014), 1507–1541.

W. Hock and K. Schittkowski, Test examples for nonlinear programming codes, *Journal of Optimization Theory and Applications*, **30** (1980), 127–129.

J. B. Jian, C. H. Guo, C. M. Tang and Y. Q. Bai, A new superlinearly convergent algorithm of combining QP subproblem with system of linear equations for nonlinear optimization, *Journal of Computational and Applied Mathematics*, **273** (2015), 88–102.

J. B. Jian, Fast algorithms for smooth constrained optimization—theoretical analysis and numerical experiments, Science Press, Beijing, 2010.

J. B. Jian, A strong subfeasible direction algorithm with superlinear convergence, *Journal of Systems Science and Systems Engineering*, **5** (1996), 287–295.

J. B. Jian, C. M. Tang, Q. J. Hu and H. Y. Zheng, A new superlinearly convergent strongly subfeasible sequential quadratic programming algorithm for inequality-constrained optimization, *Numerical Functional Analysis and Optimization*, **29** (2008), 376–409.

J. B. Jian, Y. H. Chen and C. H. Guo, A strongly convergent method of quasi-strongly subfeasible directions for constrained optimization, *Pacific Journal of Optimization*, **7** (2011), 339–351.

J. B. Jian, X. Y. Ke, H. Y. Zheng and C. M. Tang, A method combining norm-relaxed QP subproblems with systems of linear equations for constrained optimization, *Journal of Computational and Applied Mathematics*, **223** (2009), 1013–1027.

P. Morin, R. H. Nochetto, M. S. Pauletti and M. Verani, Adaptive SQP Method for Shape Optimization, *Numerical Mathematics and Advanced Applications 2009*, (2010), 663–673.

J. L. Morales, J. Nocedal and Y. Wu, A sequential quadratic programming algorithm with an additional equality constrained phase, *IMA Journal of Numerical Analysis*, **32** (2012), 553–579.

D. Q. Mayne and E. Polak, A superlinearly convergent algorithm for constrained optimization problems, *Algorithms for Constrained Minimization of Smooth Nonlinear Functions*, (1982), 45–61.

E. R. Panier and A. L. Tits, On combining feasibility, descent and superlinear convergence in inequality constrained optimization, *Mathematical Programming*, **59** (1993), 261–276.

J. F. A. Pantoja and D. Q. Mayne, Exact penalty function algorithm with simple updating of the penalty parameter, *Journal of Optimization Theory and Applications*, **69** (1991), 441–467.

K. Schittkowski, *More Test Examples for Nonlinear Programming Codes*, Springer-Verlag New York, Inc., 1987.

P. Spellucci, A new technique for inconsistent QP problems in the SQP methods, *Mathematical Methods of Operations Research*, **47** (1998), 355–400.

Y. L. Wang, L. F. Chen and G. P. He, Sequential systems of linear equations method for general constrained optimization without strict complementarity, *Journal of Computational and Applied Mathematics*, **182** (2005), 447–471.

Received April 2015; 1st revision December 2015; 2nd revision June 2016.

E-mail address: guo-ch@163.com
E-mail address: efshan@shu.edu.cn
E-mail address: wenliyy@163.com