Journal of Computational Mathematics, Vol.15, No.2, 1997, 179–192.

AN SQP ALGORITHM WITH NONMONOTONE LINE SEARCH FOR GENERAL NONLINEAR CONSTRAINED OPTIMIZATION PROBLEM*[1]

G.P. He[2]
(*Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing China*)
B.Q. Diao
(*Shandong Institute of Mining and Technology, Taian, Shandong, China*)
Z.Y. Gao
(Northern Jiaotong University, Beijing, China)

Abstract

In this paper, an SQP type algorithm with a new nonmonotone line search technique for general constrained optimization problems is presented. The new algorithm does not have to solve the second order correction subproblems for each iterations, but still can circumvent the so-called Maratos effect. The algorithm's global convergence and superlinear convergent rate have been proved. In addition, we can prove that, after a few iterations, correction subproblems need not be solved, so computation amount of the algorithm will be decreased much more. Numerical experiments show that the new algorithm is effective.

1. Introduction

In this paper, we consider the following optimization problem (P):

$$\text{(P)} \qquad \begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_j(x) = 0 \quad (j = 1, \cdots, m'), \\ & g_j(x) \le 0 \quad (j = m'+1, \cdots, m), \end{aligned}$$

where $x = (x_1, \cdots, x_n)^T \in E^n$, $f(x)$, $g_j(x)$ $(j = 1, \cdots, m)$ are all real-valued smooth functions.

In recent years, Sequential Quadratic Programming (SQP) algorithms have been extensively used for the solution of such problems, and they have been widely investigated by many authors (see, e.g. [1-5]).

An attractive feature of the SQP method is that, under some suitable conditions, a superlinear convergence can be obtained, provided that the unit stepsize is eventually accepted along the direction computed by solving the quadratic programming subproblem. In order to enforce global convergence towards Kuhn-Tucker points of

* Received April 20, 1995.
1) The Project was supported by the National Natural Science Foundation.
2) New address: Shandong Institute of Mining and Technology, Taian, Shandong 271019, China.

the original problem, a general approach is to define a merit function that measures progress towards the solution, and to choose a stepsize that yields a sufficient decrease in the merit function.

A standard merit function is following nondifferentiable penalty function:

$$\Phi(x) = f(x) + r\sum_{j=1}^{m'} |g_j(x)| + r\sum_{j=m'+1}^{m} \max(0, g_j(x)), \tag{1.1}$$

where r is a positive penalty parameter.

The main difficulty encountered in SQP methods with this merit function is that the line search can truncate the steplength near a solution, thus destroying the superlinear convergence. This is so-called Maratos effect. Two types of techniques have been proposed to overcome this undesirable problem. One method is the so called watchdog technique [6] in which the step of one is tentatively accepted if sufficient decrease was achieved at the previous iteration, compared to the lowest value of the merit function obtained so far. If this lowest value is not improved upon within a given finite number of iterations, the algorithm restarts from the iterate at which this value was achieved. Under some conditions it is shown that a step of one is always accepted in the vicinity of a local solution of (P). However, in the early iterations, numerous function and gradient evaluations may be wasted due to "backtracking". Another approach is that of modifying the search direction when near to a solution to avoid the Maratos effect. In this type approach, a correction subproblem must be solved at each iteration and an arc search is performed. Many additional function or gradient evaluations of constraints at auxiliary points are also needed to be performed per iteration (see, e.g., Ref. [7-8]).

In Ref.[9], by combining the arc search technique with nonmonotone line search scheme [10], Panier and Tits presented an algorithm for avoiding the Maratos effect. Their algorithm has an advantage that, after a few iterations, function evaluations are no longer performed at any auxiliary point. For early iterations, however, this algorithm still has to solve two quadratic programming subproblems to determine a search direction per step. This is not necessary for most of optimization problems.

The aim of this paper is to propose a new algorithm for improving the work of Ref.[9]. This algorithm uses a linear search scheme which is different from that of Ref.[9], and only needs to solve a quadratic programming subproblem and, if sometimes necessary, a system of linear equations at each iteration. The advantage mentioned above for the algorithm of Ref.[9] still can be preserved by the new algorithm, but the total computation amounts of new algorithm is less than that of the algorithm of Ref.[9] per iteration. Numerical experiments show that the new algorithm is very effective.

This paper is organized as follows. Algorithm A is stated in Section 2. In Section 3, under some mild assumptions, we prove that Algorithm A is global convergent. Rate of convergence is analyzed in Section 4. In Section 5, some numerical results are reported.

2. Algorithm

In the following, we let $L = \{1, \cdots, m'\}$, $M = \{m'+1, \cdots, m\}$. For any iteration point x_k, in order to compute a search direction, we will make use of a quadratic programming $QP_0(x_k, H_k)$ defined for a symmetric positive definite matrix $H_k \in E^{n\times n}$ by

$$
(\mathrm{QP}_0) \qquad \begin{aligned} \min \quad & \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d \\ \text{s.t.} \quad & g_j(x_k) + \nabla g_j(x_k)^T d = 0, \quad (j \in L), \\ & g_j(x_k) + \nabla g_j(x_k)^T d \le 0, \quad (j \in M). \end{aligned}
$$

Denote its solution by d_k and associated multiplier vector by λ^k. Then d_k and λ^k satisfy the following K-T conditions:

$$\nabla f(x_k) + H_k d_k + \sum_{j=1}^{m} \lambda_j^k \nabla g_j(x_k) = 0, \tag{2.1a}$$

$$g_j(x_k) + \nabla g_j(x_k)^T d_k = 0, \qquad (j \in L), \tag{2.1b}$$

$$g_j(x_k) + \nabla g_j(x_k)^T d_k \le 0, \qquad (j \in M), \tag{2.1c}$$

$$\lambda_j^k [g_j(x_k) + \nabla g_j(x_k)^T d_k] = 0, \qquad (j \in M), \tag{2.1d}$$

$$\lambda_j^k \ge 0, \qquad (j \in M). \tag{2.1e}$$

In the early iterations, for x_k, a correction direction $\hat{d}_k$ may be computed by solving following system of linear equations $LS(x_k, d_k)$ defined by

$$
(\mathrm{LS}) \qquad \begin{cases} H_k d + \nabla f(x_k) + \displaystyle\sum_{i=1}^{m} \lambda_i \nabla g_i(x_k) = 0, \\ \nabla g_j(x_k)^T d + g_j(x_k) = \beta_{kj}, \quad (j \in \overline{M} = \{j \in M \mid \lambda_j^k > 0\} \cup L), \end{cases}
$$

where $\beta_{kj} = -g_j(x_k + d_k) + \|d_k\|^\tau$, $(j \in \overline{M})$, $\tau > 2$ is a constant.

Now, we define a function $\Psi(x_k + d)$ for x_k and $d \in E^n$ by

$$
\begin{aligned} \Psi(x_k + d) = & f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + r \sum_{i=1}^{m'} |g_j(x_k) + \nabla g_j(x_k)^T d| \\ & + r \sum_{i=m'+1}^{m} \max(0, g_j(x_k) + \nabla g_j(x_k)^T d) \end{aligned} \tag{2.2}
$$

The line search test involves a maximum past value Φ_k defined over the last four iteration points by $\Phi_k = \max\limits_{l=0,\cdots,3} \Phi(x_{k-l})$, where negative indices that may appear in the early iterations are discarded.

Algorithm A.

Parameters. $r > 0$, $\mu_1, \mu_2 \in (0,1)$ and $\mu_2 < \mu_1$, $\overline{\alpha}_0 = 1$, $\tau > 2$.

Data. $x_0 \in E^n$, $H_0 \in E^{n\times n}$ is a symmetric positive definite matrix.

Step 0. Initialization. Set $k = 0$.

Step 1. Compute d_k solution of the quadratic programming $QP_0(x_k, H_k)$. Associated multiplier vector is λ^k. If d_k=0, stop.

Step 2. Let $\overline{x}_{k+1} = x_k + \overline{\alpha}_k d_k$, and compute

$$\rho_k^{(1)} = \frac{\Phi_k - \Phi(\overline{x}_{k+1})}{\Psi(x_k) - \Psi(\overline{x}_{k+1})}.$$

Step 3. If $\rho_k^{(1)} < \mu_1$, then go to Step 4; Otherwise, set $\alpha_k = \overline{\alpha}_k$, $x_{k+1} = x_k + \alpha_k d_k$,

$$\overline{\alpha}_{k+1} = \begin{cases} 2\alpha_k, & \text{if} \quad 2\alpha_k \leq 1, \\ \alpha_k, & \text{if} \quad 2\alpha_k > 1, \end{cases}$$

then go to Step 6.

Step 4. Compute $\hat{d}_k$ and $\hat{\lambda}^k$, solution of $LS(x_k, d_k)$. If there is no solution or $\|\hat{d}_k - d_k\| > \|d_k\|$, set $\hat{d}_k = d_k$. Let $x(\overline{\alpha}_k) = x_k + \overline{\alpha}_k d_k + \overline{\alpha}_k^2(\hat{d}_k - d_k)$, and compute

$$\rho_k^{(2)} = \frac{\Phi_k - \Phi(x(\overline{\alpha}_k))}{\Psi(x_k) - \Psi(\overline{x}_{k+1})}.$$

Step 5. If $\rho_k^{(2)} \geq \mu_2$, then set $\alpha_k = \overline{\alpha}_k$, $x_{k+1} = x(\alpha_k)$, and let

$$\overline{\alpha}_{k+1} = \begin{cases} 2\alpha_k, & \text{if} \quad 2\alpha_k \leq 1 \quad \text{and} \quad \rho_k^{(2)} \geq \mu_1, \\ \alpha_k, & \text{otherwise}, \end{cases}$$

go to Step 6; If $\rho_k^{(2)} < \mu_2$, then set $\overline{\alpha}_k := \frac{1}{2}\overline{\alpha}_k$, return back to Step 2.

Step 6. Updates. Compute a new symmetric positive definite approximation H_{k+1} to the Hessian of the Lagrangian. Set $k = k + 1$, go back to Step 1.

In next part of this paper, we make the following assumptions:

A1. The functions f, g_j $(j = 1, \cdots, m)$ are all continuously differentiable.

A2. The feasible sets of quadratic programming subproblems $QP_0(x, H)$ are always nonempty.

Now we can easily prove following two lemmas:

Lemma 2.1. *If d_k is a K-T point of (QP_0) and $d_k \neq 0$, λ^k is associated multiplier vector satisfying $r > |\lambda_j^k|$, $(1 \leq j \leq m)$, then functions $\Phi(x)$ and $\Psi(x)$ all have directional derivatives along d_k at point x_k, and they are equal to each other, i.e., $D_{d_k}\Phi(x_k) = D_{d_k}\Psi(x_k) < 0$, where*

$$D_{d_k}\Phi(x_k) = \lim_{\alpha \to 0} \frac{\Phi(x_k + \alpha d_k) - \Phi(x_k)}{\alpha},$$
$$D_{d_k}\Psi(x_k) = \lim_{\alpha \to 0} \frac{\Psi(x_k + \alpha d_k) - \Psi(x_k)}{\alpha}.$$

Lemma 2.2. *If d_k is a K-T point of (QP_0) and $d_k \neq 0$, λ^k is associated multiplier vector satisfying $r > |\lambda_j^k|$, $(1 \leq j \leq m)$, then, for $\forall \alpha \in (0, 1)$, we have*

$$\Psi(x_k) - \Psi(x_k + d_k) \geq \frac{1}{2} d_k^T H_k d_k \quad \textit{and} \quad \Psi(x_k) - \Psi(x_k + \alpha d_k) \geq \frac{\alpha}{2} d_k^T H_k d_k.$$

With these lemmas, we know that

$$\lim_{\alpha\to 0}\frac{\Phi(x_k)-\Phi(x_k+\alpha d_k)}{\Psi(x_k)-\Psi(x_k+\alpha d_k)}=1.$$

In addition, we have $\Phi_k-\Phi(x_k+\alpha d_k)\geq \Phi(x_k)-\Phi(x_k+\alpha d_k)$. Hence, if $\overline{\alpha}_k$ is small enough, we always have $\rho_k^{(1)}>\mu_1$. This means that the algorithm does not cycle between Step 2 and Step 5, and it is well defined.

3. Global Convergence

In order to establish the global convergence of Algorithm A, in addition to A1 and A2, we need the following three assumptions furthermore.

A3. There exists a positive constant $\kappa>0$ such that, for all k and for all $y\in E^n$,

$$\frac{1}{\kappa}\|y\|^2\leq y^T H_k y\leq \kappa\|y\|^2.$$

A4. The penalty parameter r satisfies the following condition:

$$r>\sup_k\max_{1\leq j\leq m}|\lambda_j^k|.$$

A5. The sequences $\{d_k\}$, $\{\hat{d}_k\}$, $\{x_k\}$ are all bounded.

Lemma 3.1. **(i)** $\lim_{k\to\infty}[\Psi(x_k)-\Psi(x_k+d_k)]=0$; **(ii)** $\lim_{k\to\infty}H_kd_k=0$.

Proof. (i) Suppose by contradiction the assertion is false. Then, without loss of any generality, we can suppose that there exists a small enough constant $\epsilon>0$ and an integer constant K_0 such that, for all $k\geq K_0$, we have

$$\Psi(x_k)-\Psi(x_k+d_k)>\epsilon. \tag{3.1}$$

By the definition of $\rho_k^{(2)}$ and the convex property of $\Psi(x_k+d)$ for d, we always have

$$\Phi_k-\Phi(x_{k+1})\geq\mu_2[\Psi(x_k)-\Psi(\overline{x}_{k+1})]\geq\mu_2\overline{\alpha}_k[\Psi(x_k)-\Psi(x_k+d_k)] \tag{3.2}$$

So, for all $k\geq K_0$, we obtain

$$\Phi_k-\Phi(x_{k+1})\geq\mu_2\overline{\alpha}_k\epsilon. \tag{3.3}$$

Now, without loss of generality, we assume that $k>3$ and define a positive integer constant by $l(k)$ such that

$$k-3\leq l(k)\leq k, \tag{3.4}$$

and

$$\Phi(x_{l(k)})=\max_{0\leq j\leq 3}[\Phi(x_{k-j})]. \tag{3.5}$$

Since $\Phi(x_{k+1})\leq\Phi(x_{l(k)})$, we have following relation

$$\Phi(x_{l(k+1)})=\max_{0\leq j\leq 3}[\Phi(x_{k+1-j})]\leq\max\,[\max_{0\leq j\leq 3}[\Phi(x_{k-j})],\quad \Phi(x_{k+1})]$$

$$= \max[\Phi(x_{l(k)}),\ \Phi(x_{k+1})] = \Phi(x_{l(k)}), \tag{3.6}$$

and we know that

$$\begin{aligned}\Phi(x_{l(k)}) &\le \Phi_{l(k)-1} + \mu_2[\Psi(\overline{x}_{l(k)-1+1}) - \Psi(x_{l(k)-1})] \\ &\le \Phi(x_{l(l(k)-1)}) + \mu_2 \alpha_{l(k)-1}[\Psi(x_{l(k)-1} + d_{l(k)-1}) - \Psi(x_{l(k)-1})].\end{aligned} \tag{3.7}$$

Since the sequence $\{\Phi(x_{l(k)})$ is monotone decrease and bounded, it must have limit, so

$$\lim_{k\to\infty} \alpha_{l(k)-1}[\Psi(x_{l(k)-1} + d_{l(k)-1}) - \Psi(x_{l(k)-1})] = 0, \tag{3.8}$$

that is

$$\lim_{k\to\infty} \alpha_{l(k)-1} H^{\frac{1}{2}}_{l(k)-1} d_{l(k)-1} = 0. \tag{3.9}$$

By assumption A3, we have $\lim\limits_{k\to\infty} \alpha_{l(k)-1}\|d_{l(k)-1}\|^2 = 0$. Hence

$$\lim_{k\to\infty} \alpha_{l(k)-1}\|d_{l(k)-1}\| = 0. \tag{3.10}$$

Now, we prove that

$$\lim_{k\to\infty} \alpha_k\|d_k\| = 0. \tag{3.11}$$

Let $\hat{l}(k) = l(k+4)$. Then, using a induction scheme, we can prove that, for $\forall j \ge 1$, if $k \ge j-1$, we always have

$$\lim_{k\to\infty} \alpha_{\hat{l}(k)-j}\|d_{\hat{l}(k)-j}\| = 0, \tag{3.12}$$

and

$$\lim_{k\to\infty} \Phi(x_{\hat{l}(k)-j}) = \lim_{k\to\infty} \Phi(x_{l(k)}). \tag{3.13}$$

First assume $j = 1$. Since $\{\hat{l}(k)\} \subset \{l(k)\}$, by (3.10), (3.12) is obviously correct. In addition, notting that $\|\hat{d}_k - d_k\| \le \|d_k\|$, we have $\|x_{\hat{l}(k)} - x_{\hat{l}(k)-1}\| \longrightarrow 0$. Hence we have $\lim\limits_{k\to\infty} \Phi(x_{\hat{l}(k)-1}) = \lim\limits_{k\to\infty} \Phi(x_{\hat{l}(k)}) = \lim\limits_{k\to\infty} \Phi(x_{l(k)})$. This means (3.13) holds.

Now suppose that (3.12) and (3.13) hold for some $j > 1$. Then, we prove that these formulas are still correct for $j+1$.

Obviously, we have

$$\Phi(x_{\hat{l}(k)-j}) \le \Phi(x_{l(\hat{l}(k)-j-1)}) + \mu_2 \alpha_{\hat{l}(k)-j-1}[\Psi(x_{\hat{l}(k)-j-1} + d_{\hat{l}(k)-j-1}) - \Psi(x_{\hat{l}(k)-j-1})].$$

Let $k \longrightarrow \infty$, by (3.13), we obtain

$$\lim_{k\to\infty} \alpha_{\hat{l}(k)-j-1}[\Psi(x_{\hat{l}(k)-j-1} + d_{\hat{l}(k)-j-1}) - \Psi(x_{\hat{l}(k)-j-1})] = 0.$$

That is $\lim\limits_{k\to\infty} \alpha_{\hat{l}(k)-j-1} H^{\frac{1}{2}}_{\hat{l}(k)-j-1} d_{\hat{l}(k)-j-1} = 0$, hence $\lim\limits_{k\to\infty} \alpha_{\hat{l}(k)-j-1}\|d_{\hat{l}(k)-j-1}\| = 0$. And we have

$$\|x_{\hat{l}(k)-j} - x_{\hat{l}(k)-j-1}\| \longrightarrow 0,$$

$$\lim_{k\to\infty}\Phi(x_{\hat{l}(k)-j-1})=\lim_{k\to\infty}\Phi(x_{\hat{l}(k)-j})=\lim_{k\to\infty}\Phi(x_{l(k)}).$$

This shows that (3.12) and (3.13) are correct for $j+1$.

For $\forall k$, we know

$$x_{k+1}=x_{\hat{l}(k)}-\sum_{j=0}^{\hat{l}(k)-k}\alpha_{\hat{l}(k)-j}\overline{d}_{\hat{l}(k)-j},$$

where $\overline{d}_{\hat{l}(k)-j}=d_{\hat{l}(k)-j}$ or $\overline{d}_{\hat{l}(k)-j}=d_{\hat{l}(k)-j}+\alpha_{\hat{l}(k)-j}(\hat{d}_{\hat{l}(k)-j}-d_{\hat{l}(k)-j})$, they are determined respectively by Step 1 and Step 4 of Algorithm A.

Since $\alpha_{\hat{l}(k)-j}\|d_{\hat{l}(k)-j}\|\longrightarrow 0$, and $\alpha_{\hat{l}(k)-j}\|\hat{d}_{\hat{l}(k)-j}-d_{\hat{l}(k)-j}\|\longrightarrow 0$ so we have

$$\lim_{k\to\infty}\|x_{k+1}-x_{\hat{l}(k)}\|=0.$$

and

$$\lim_{k\to\infty}\Phi(x_{k+1})=\lim_{k\to\infty}\Phi(x_{\hat{l}(k)}).$$

But since

$$\Phi(x_{k+1})\le\Phi(x_{l(k)})+\mu_2[\Psi(\overline{x}_{k+1})-\Psi(x_k)],$$

we know $\lim\limits_{k\to\infty}[\Psi(\overline{x}_{k+1})-\Psi(x_k)]=0$. Therefore, for k large enough, $\Psi(x_k)-\Psi(x_k+d_k)\le\frac{1}{2}\epsilon$, which contradicts to (3.1).

(ii) The proof is obvious.

Theorem 3.2. *Algorithm A either stops at a K-T point of (P) or generates a sequence $\{x_k\}$ for which each limit point is a K-T point of (P).*

Proof. Algorithm A can stop only at Step 1 and the conclusion is trivial in this case. Now we suppose that $\{x_k\}$ is an infinite sequence and $\{x_k\}_K\longrightarrow x^*$. By the boundedness of $\{\lambda^k\}$, without loss of any generality, we can suppose that the sequence $\{\lambda^k\}_K$ has a limit, say λ^*. Considering (2.1a) and (2.1e), by Lemma 3.1 (ii), we have

$$\nabla f(x^*)+\sum_{i=1}^{m}\lambda_i^*\nabla g_i(x^*)=0, \tag{3.14}$$

and

$$\lambda_j^*\ge 0,\quad (j\in M). \tag{3.15}$$

In addition, from following relation

$$\begin{aligned}&\Psi(x_k)-\Psi(x_k+d_k)\ge\Psi(x_k)-\Psi(x_k+d_k)-\frac{1}{2}d_k^TH_kd_k\\&=r\sum_{j=1}^{m'}|g_j(x_k)|+r\sum_{j=m'+1}^{m}\max(0,\ g_j(x_k))-\sum_{j=1}^{m}\lambda_j^kg_j(x_k)\\&\ge\sum_{j=1}^{m'}(r-|\lambda_j^k|)|g_j(x_k)|+\sum_{j\in M,\ g_j(x_k)>0}(r-\lambda_j^k)|g_j(x_k)|-\sum_{j\in M,g_j(x_k)<0}\lambda_j^kg_j(x_k)\ge 0\end{aligned}$$

and Lemma 3.1 (i), we have

$$\sum_{j=1}^{m'}(r-|\lambda_j^*|)|g_j(x^*)| + \sum_{j\in M, g_j(x^*)>0}(r-\lambda_j^*)|g_j(x^*)| - \sum_{j\in M,\ g_j(x^*)<0}\lambda_j^* g_j(x^*) = 0.$$

So, we obtain $g_j(x^*)=0$, $(j\in L)$, $g_j(x^*)\le 0$, $(j\in M)$, and $\lambda_j^* g_j(x^*)=0$. This shows that x^* is a K-T point of (P).

4. Superlinear Convergence

In this section, we will prove superlinear convergence of Algorithm A. First, we assume some more conditions on the functions involved. Assumption (A1) is replaced by the following.

A1'. The functions f, $g_j (j=1,\cdots,m)$ are at least twice continuously differentiable.

We also make the following assumptions:

A6. The sequence $\{x_k\}$ generated by Algorithm A possesses a limit point x^* (in view of Theorem 3.2, a K-T point of (P)) at which the gradients of active constraints are linearly independent, and second order sufficient conditions and strict complementary slackness hold, i.e., the Lagrange multiplier vector $\lambda^*\in E^m$ at x^* satisfies $\lambda_j^*>0 \quad (\forall j\in J_0(x^*)=\{j\,|\,g_j(x^*)=0,\ j=m'+1,\cdots,m\})$, $\lambda_j^*\ne 0 \quad (\forall j\in L)$ and the Hessian matrix $\nabla_{xx}^2 L(x^*,\lambda^*)$ of the Lagrangian function $L(x,\lambda)=f(x)+\sum_{j=1}^{m}\lambda_j g_j(x)$ of (P) is positive definite on the subspace $\{y\,|\,y^T\nabla g_j(x^*)=0,\ (\forall j\in J_0(x^*)\cup L)\}$.

By Lemma 3.1 (ii) and Assumption A3, we know that $\|d_k\|\to 0$. So with a similar method as that of Ref.[9], we can easily show the following lemma.

Lemma 4.1. *The entire sequence $\{x_k\}$ converges to x^*, $\|d_k\|\to 0$ and $\lambda^k\to\lambda^*$.*

Now suppose that the active set of constraints at x^* is already determined, without loss of generality, we denote it by $I^*=\{1,2,\cdots,m',m'+1,\cdots,l\}$. By assumption A6, when we turn our attention to the convergence rate of the algorithm, we only need to consider the following problem with equality constraints:

$$\text{(EP)} \qquad \begin{cases} \min & f(x) \\ \text{s.t.} & g_j(x)=0 \quad (j=1,\cdots,l). \end{cases}$$

Now let

$$g_k=(g_1(x_k),\cdots,g_l(x_k))^T, \quad N_k=(\nabla g_1(x_k),\cdots,\nabla g_l(x_k)),$$
$$N_*=(\nabla g_1(x^*),\cdots,\nabla g_l(x^*)), \quad P_k=I-N_k[N_k^T N_k]^{-1}N_k^T,$$

where $I\in E^{n\times n}$ is a unit matrix. From assumptions A3 and A6, we know that, for k large enough, the matrices N_k are of full rank, and we have

Lemma 4.2. *The matrices*

$$\begin{bmatrix} H_k & N_k \\ N_k^T & 0 \end{bmatrix}$$

are uniformly nonsingular.

Lemma 4.3. *For k large enough, we have $\beta_{kj} = \bigcirc(\|d_k\|^2)$, $(j = 1, \cdots, l)$.*

Lemma 4.4. *For k large enough, we have $\|\hat{d}_k - d_k\| = \bigcirc(\|d_k\|^2)$ and $\|\hat{\lambda}^k - \lambda^k\| = \bigcirc(\|d_k\|^2)$.*

Proof. Since $\lambda^k \to \lambda^*$, by A6, we know $\overline{M} = \{1, 2, \cdots, l\}$. From (2.1) and the definition of (LS), we have

$$\begin{pmatrix} H_k & N_k \\ N_k^T & 0 \end{pmatrix} \begin{pmatrix} \hat{d}_k - d_k \\ \hat{\lambda}^k - \lambda^k \end{pmatrix} = \begin{pmatrix} 0 \\ \beta_k \end{pmatrix},$$

from Lemma 4.2 and 4.3, we can easily know the result is correct.

In order to obtain superlinear convergence, we furthermore assume

A7. The matrices H_k are positive definite and satisfy

$$\lim_{k\to\infty} \frac{\|P_k(H_k - \nabla^2_{xx}L(x^*, \lambda^*)d_k\|}{\|d_k\|} = 0. \tag{4.1}$$

Theorem 4.5. *For k large enough, the step size $\alpha_k = 1$.*

Proof. We denote the first trial value of α^k for every iteration by $\overline{\alpha}_k$. In the following proof, the iteration index k is generally dropped to simplify the notation.

Since $x(\overline{\alpha}) = x + \overline{\alpha}(d + \overline{\alpha}(\hat{d} - d))$ and the definition of $\Psi(x + d)$, we have

$$|\Psi(x + \overline{\alpha}d) - \Phi(x(\overline{\alpha}))| \le \omega_1 + \omega_2 + \omega_3, \tag{4.2}$$

where

$$\begin{aligned}
\omega_1 &= |f(x) + \overline{\alpha}\nabla f(x)^T d + \frac{1}{2}\overline{\alpha}^2 d^T H d - f(x + \overline{\alpha}(d + \overline{\alpha}(\hat{d} - d)))|, \\
\omega_2 &= r\sum_{i=1}^{m'} |g_i(x) + \overline{\alpha}\nabla g_i(x)^T d - g_i(x + \overline{\alpha}(d + \overline{\alpha}(\hat{d} - d)))|, \\
\omega_3 &= r\sum_{i=m'+1}^{l} |\max(0,\ g_i(x) + \overline{\alpha}\nabla g_i(x)^T d) - \max(0,\ g_i(x + \overline{\alpha}(d + \overline{\alpha}(\hat{d} - d))))|.
\end{aligned}$$

First, since

$$\begin{aligned}
&|g_i(x) + \overline{\alpha}\nabla g_i(x)^T d - g_i(x + \overline{\alpha}(d + \overline{\alpha}(\hat{d} - d)))| \\
=&|g_i(x) + \overline{\alpha}\nabla g_i(x)^T d - [g_i(x) + \overline{\alpha}\nabla g_i(x)^T(d + \overline{\alpha}(\hat{d} - d)) \\
&+ \frac{1}{2}\overline{\alpha}^2(d + \overline{\alpha}(\hat{d} - d))^T\nabla^2 g_i(x)(d + \overline{\alpha}(\hat{d} - d))]| + \circ(\|d + \overline{\alpha}(\hat{d} - d)\|^2) \\
=&|\overline{\alpha}^2\nabla g_i(x)^T(\hat{d} - d) + \frac{1}{2}\overline{\alpha}^2(d + \overline{\alpha}(\hat{d} - d))^T\nabla^2 g_i(x)(d + \overline{\alpha}(\hat{d} - d))| + \circ(\|d\|^2) \\
=&\overline{\alpha}^2|\nabla g_i(x)^T(\hat{d} - d) + \frac{1}{2}d^T\nabla^2 g_i(x)d| + \circ(\|d\|^2),
\end{aligned}$$

we have

$$\omega_2 = r\sum_{i=1}^{m'}(\overline{\alpha}^2|\nabla g_i(x)^T(\hat{d} - d) + \frac{1}{2}d^T\nabla^2 g_i(x)d| + \circ(\|d\|^2)).$$

Now from (LS) and K-T conditions of (EP), we can know that

$$\nabla g_i(x)^T(\hat{d}-d) = -g_i(x+d) + \|d\|^\tau, \tag{4.3}$$

and since

$$\begin{aligned} g_i(x+\hat{d}) &= g_i(x+d+\hat{d}-d) = g_i(x+d) + \nabla g_i(x+d)^T(\hat{d}-d) + \bigcirc(\|d\|^4) \\ &= g_i(x+d) + \nabla g_i(x)^T(\hat{d}-d) + \bigcirc(\|d\|^3), \end{aligned} \tag{4.4}$$

so we obtain

$$g_i(x+\hat{d}) = \bigcirc(\|d\|^\tau). \tag{4.5}$$

In addition, it is obvious that

$$\begin{aligned} g_i(x+\hat{d}) &= g_i(x) + \nabla g_i(x)^T\hat{d} + \frac{1}{2}\hat{d}^T\nabla^2 g_i(x)\hat{d} + \circ(\|\hat{d}\|^2) \\ &= g_i(x) + \nabla g_i(x)^T d + \nabla g_i(x)^T(\hat{d}-d) + \frac{1}{2}d^T\nabla^2 g_i(x)d + \circ(\|d\|^2), \end{aligned} \tag{4.6}$$

hence we have

$$\nabla g_i(x)^T(\hat{d}-d) + \frac{1}{2}d^T\nabla^2 g_i(x)d = \circ(\|d\|^2). \tag{4.7}$$

So it follows that

$$\omega_2 = \overline{\alpha}^2 \cdot \circ(\|d\|^2). \tag{4.8}$$

Similarly, notting that $\max(0,s) = \dfrac{|s|+s}{2}$, we can easily prove that

$$\omega_3 = \overline{\alpha}^2 \cdot \circ(\|d\|^2). \tag{4.9}$$

Now we make an estimation of ω_1. First we have

$$\begin{aligned} \omega_1 &= |f(x) + \overline{\alpha}\nabla f(x)^T d + \frac{1}{2}\overline{\alpha}^2 d^T H d - f(x) - \overline{\alpha}\nabla f(x)^T(d+\overline{\alpha}(\hat{d}-d)) \\ &\quad - \frac{1}{2}\overline{\alpha}^2(d+\overline{\alpha}(\hat{d}-d))^T\nabla^2 f(x)(d+\overline{\alpha}(\hat{d}-d))| + \overline{\alpha}^2 \cdot \circ(\|d+\overline{\alpha}(\hat{d}-d)\|^2) \\ &= \left|\frac{1}{2}\overline{\alpha}^2 d^T H d - \overline{\alpha}^2\nabla f(x)^T(\hat{d}-d) - \frac{1}{2}\overline{\alpha}^2 d^T\nabla^2 f(x)d\right| + \overline{\alpha}^2 \cdot \circ(\|d\|^2) \end{aligned}$$

From (2.1) and (4.7), we obtain

$$\begin{aligned} \nabla f(x)^T(\hat{d}-d) &= -d^T H(\hat{d}-d) - \sum_{i=1}^{l}\lambda_i\nabla g_i(x)^T(\hat{d}-d) \\ &= \bigcirc(\|d\|^3) - \sum_{i=1}^{l}\lambda_i\nabla g_i(x)^T(\hat{d}-d) = \frac{1}{2}\sum_{i=1}^{l}\lambda_i d^T\nabla^2 g_i(x)d + \circ(\|d\|^2), \end{aligned} \tag{4.10}$$

so we have

$$\omega_1 = \frac{1}{2}\overline{\alpha}^2|d^T H d - d^T\nabla^2 f(x)d - \sum_{i=1}^{l}\lambda_i d^T\nabla^2 g_i(x)d| + \circ(\|d\|^2)$$

$$= \frac{1}{2}\overline{\alpha}^2 |d^T(H - \nabla^2 f(x) - \sum_{i=1}^{l} \lambda_i \nabla^2 g_i(x))d| + \circ(\|d\|^2)$$
$$= \frac{1}{2}\overline{\alpha}^2 |d^T(H - \nabla^2_{xx} L(x,\ \lambda))d| + \circ(\|d\|^2).$$

If we let $\Gamma = H - \nabla^2_{xx} L(x^*,\ \lambda^*)$, then from the definition of P, we have

$$\begin{aligned} d^T\Gamma d &= d^T P\Gamma d + d^T(I-P)\Gamma d = d^T P\Gamma d + d^T N(N^TN)^{-1}N^T\Gamma d \\ &= d^T P\Gamma d - g(x)^T(N^TN)^{-1}N^T\Gamma d = \circ(\|d\|^2) + \circ(\|g(x)\|), \end{aligned}$$

where $g(x) = (g_1(x), \cdots, g_l(x))^T$. Hence it follows that

$$\begin{aligned} \omega_1 &= \frac{1}{2}\overline{\alpha}^2 |d^T(H - \nabla^2_{xx}L(x^*,\lambda^*) + \nabla^2_{xx}L(x^*,\lambda^*) - \nabla^2_{xx}L(x,\lambda))d| + \circ(\|d\|^2), \\ &= \overline{\alpha}^2(\circ(\|d\|^2) + \circ(\|d\|\|g(x)\|)). \end{aligned}$$

This implies

$$|\Psi(x+\overline{\alpha}d) - \Phi(x(\overline{\alpha}))| \le \overline{\alpha}^2[\circ(\|d\|^2) + \circ(\|g(x)\|)]. \tag{4.11}$$

On the other hand, let

$$\xi = \min\{\inf_k [\min_{m'+1\le i\le l} \lambda_i^k], \quad r - \sup_k [\max_{1\le i\le l} |\lambda_i^k|]\},$$

from Lemma 3 in [8], we know the following matrices are uniformly positive definite. $\overline{H}_k = H_k + \sigma N_k N_k^T$, where $\sigma = \dfrac{\xi}{\sup \|g(x_k)\|}$. So we have

$$d^T H d \ge \overline{\sigma}\|d\|^2 - \sigma\|g(x)\|^2 \ge \overline{\sigma}\|d\|^2 - \xi\|g(x)\| \tag{4.12}$$

for some $\overline{\sigma} > 0$. Therefore, it follows that

$$\begin{aligned} &\Psi(x) - \Psi(x+d) = r\sum_{j=1}^{m'} |g_j(x)| + r\sum_{j=m'+1}^{l} \max(0, g_j(x)) - \sum_{j=1}^{l} \lambda_j g_j(x) + \frac{1}{2}d^T H d \\ &\ge \sum_{j=1}^{m'} (r - |\lambda_j^k|)|g_j(x_k)| + \sum_{j\in M, g_j(x_k)>0} (r - \lambda_j^k)|g_j(x_k)| - \sum_{j\in M, g_j(x_k)<0} \lambda_j^k g_j(x_k) + \frac{1}{2}d^T H d \\ &\ge \sum_{j=1}^{m'} \Big(r - \sup_k(\max_{1\le j\le l} |\lambda_j^k|)|g_j(x^*)| + \sum_{j\in M, g_j(x_k)>0} \Big(r - \sup_k(\max_{1\le j\le l} |\lambda_j^k|)|g_j(x_k)| \\ &\quad - \inf_k(\min_{m'+1\le j\le l} \lambda_j^k) \sum_{j\in M, g_j(x_k)<0} g_j(x_k) + \frac{1}{2}d^T H d \\ &\ge \frac{1}{2}d^T H d + \xi\|g(x)\| \ge \frac{1}{2}\overline{\sigma}\|d\|^2 + \frac{1}{2}\xi\|g(x)\|. \end{aligned}$$

From (3.2), we have

$$\Psi(x) - \Psi(x+\overline{\alpha}d) \ge \overline{\alpha}[\Psi(x) - \Psi(x+d)] \ge \frac{1}{2}\overline{\alpha}(\overline{\sigma}\|d\|^2 + \xi\|g(x)\|). \tag{4.13}$$

Therefore it holds that

$$\Big|\frac{\Psi(x+\overline{\alpha}d)-\Phi(x(\overline{\alpha}))}{\Psi(x)-\Psi(x+\overline{\alpha}d)}\Big| \le \frac{\overline{\alpha}^2(\circ(\|d\|^2)+\circ(\|g(x)\|))}{\frac{1}{2}\overline{\alpha}(\overline{\sigma}\|d\|^2+\xi\|g(x)\|)} \to 0$$

and

$$\rho^{(2)} = \frac{\Phi-\Phi(x(\overline{\alpha}))}{\Psi(x)-\Psi(x+\overline{\alpha}d)} \ge \frac{\Phi(x)-\Phi(x(\overline{\alpha}))}{\Psi(x)-\Psi(x+\overline{\alpha}d)} = 1+\frac{\Psi(x+\overline{\alpha}d)-\Phi(x(\overline{\alpha}))}{\Psi(x)-\Psi(x+\overline{\alpha}d)} \to 1.$$

Thus, we know that there exists a large K_0 such that, for any $k \ge K_0$, $\rho_k^{(2)} > \mu_1 > \mu_2$. Since $\overline{\alpha}_k \ge \alpha_{k-1}$, for $k \ge K_0$, we always have $\alpha_k = \overline{\alpha}_k \ge \alpha_{k-1}$, and if $2\alpha_k \le 1$, then $\overline{\alpha}_{k+1} = 2\alpha_k > \alpha_{k-1}$. Hence α_k is monotonously increased, it must be equal to one in finite steps from some $k_1 \ge K_0$.

With a similar method as that of Ref.[9], we can prove following conclusion:

Theorem 4.6. *Under the stated assumptions, we have* $\lim\limits_{k\to\infty} \dfrac{\|x_{k+2}-x^*\|}{\|x_k-x^*\|} = 0$. *Moreover,* $\|x_{k+1}-x^*\| = \bigcirc(\|x_k-x^*\|)$.

Finally, we show that Algorithm A can deal with Maratos effect well in such a way that the constraints are evaluated at auxiliary points in the early iterations only.

Theorem 4.7. *For k large enough, second order correction $\hat{d}_k$ is not computed.*

Proof. We only need to show that, for k large enough, we always have

$$\Phi(x_k+d_k)-\mu_1(\Psi(x_k+d_k)-\Psi(x_k)) \le \Phi(x_{k-3}). \tag{4.14}$$

Since

$$g_j(x_k)+\nabla g_j(x_k)^T d_k = 0, \qquad (j=1,\cdots,l), \tag{4.15}$$

from the definition of Φ and Ψ, we get

$$\begin{aligned}
&\Phi(x_k+d_k)-\mu_1(\Psi(x_k+d_k)-\Psi(x_k))\\
=&f(x_k+d_k)+r\sum_{j=1}^{m'}|g_j(x_k+d_k)|+r\sum_{j=m'+1}^{l}\max(0,g_j(x_k+d_k))\\
&-\mu_1\Big(\nabla f(x_k)^T d_k+\frac{1}{2}d_k^T H_k d_k - r\sum_{j=1}^{m'}|g_j(x_k)| - r\sum_{j=m'+1}^{l}\max(0,g_j(x_k))\Big).
\end{aligned}$$

For given k, if $x_k = x_{k-1}+\hat{d}_{k-1}$, then

$$\begin{aligned}
g_j(x_k) &= g_j(x_{k-1}+\hat{d}_{k-1}) = g_j(x_{k-1})+\nabla g_j(x_{k-1})^T\hat{d}_{k-1}+\bigcirc(\|\hat{d}_{k-1}\|^2)\\
&= g_j(x_{k-1})+\nabla g_j(x_{k-1})^T d_{k-1}+\nabla g_j(x_{k-1})^T(\hat{d}_{k-1}-d_{k-1})+\bigcirc(\|\hat{d}_{k-1}\|^2)\\
&= \bigcirc(\|d_{k-1}\|^2).
\end{aligned}$$

If $x_k = x_{k-1}+d_{k-1}$, it is obvious that

$$g_j(x_{k-1}+d_{k-1}) = \bigcirc(\|d_{k-1}\|^2). \tag{4.16}$$

In addition, we have

$$g_j(x_k + d_k) = \bigcirc(\|d_k\|^2), \tag{4.17}$$

therefore,

$$\begin{aligned}&\Phi(x_k + d_k) - \mu_1(\Psi(x_k + d_k) - \Psi(x_k))\\ &\quad = f(x_k + d_k) - \mu_1 \nabla f(x_k)^T d_k + \bigcirc(\|d_{k-1}\|^2) + \bigcirc(\|d_k\|^2).\end{aligned} \tag{4.18}$$

With the same method as that of Ref.[9], we can easily obtain

$$\Phi(x_k + d_k) - \mu_1(\Psi(x_k + d_k) - \Psi(x_k)) = f(x^*) + \circ(\|x_{k-3} - x^*\|^2) \le \Phi(x_{k-3}).$$

And the conclusion follows.

5. Numerical Experiments

In order to test the given algorithm, an efficient implementation of the algorithm has been completed. In this implementation, we select $\mu_1 = 0.8$, $\mu_2 = 0.3$, $\tau = 2.99$, and $H_0 = I$ (an $n \times n$ unit matrix). H_k is updated by means of the $BFGS$ formula with Powell's modifications as described in Ref.[3].

Table 1.

No	n	m_1	m_2	NIT	NF	NC	FV	EPS	ACT
4	2	2	0	2	2	0	2.66666666	0.1D−8	0
6	2	0	1	10	14	11	0.00000000	0.1D−8	3
8	2	0	2	4	4	10	−1.0000000	0.1D−8	1
12	2	1	0	7	10	8	−30.000000	0.1D−8	1
24	2	5	0	7	9	21	−1.0000000	0.1D−8	0
26	3	0	1	20	26	22	0.00000000	0.1D−8	2
27	3	0	1	24	28	25	0.03999999	0.1D−8	4
32	3	4	1	3	5	6	1.00000000	0.1D−8	1
33	3	6	0	7	8	16	−4.5857864	0.1D−8	0
39	4	0	2	12	12	24	−1.0000000	0.1D−8	0
47	5	0	3	25	37	75	0.00000000	0.1D−6	14
49	5	0	2	16	20	32	0.00000000	0.1D−8	1
50	5	0	3	15	25	48	0.00000000	0.1D−8	1
60	3	6	1	9	10	10	0.03256820	0.1D−8	0
61	3	0	2	9	14	18	−143.64614	0.1D−8	1
78	5	0	3	8	10	24	−2.9197004	0.1D−8	1
79	5	0	3	10	11	33	0.07877682	0.1D−8	0
80	5	10	3	6	7	21	0.05394985	0.1D−8	0
81	5	10	3	10	11	33	0.05394985	0.1D−8	0
119	16	32	8	15	15	120	249.294724	0.1D−6	0

The stopping criterion in Step 1 is unsuitable for implementation. Instead, execution is terminated if the norm of d_k is less than a constant $\epsilon > 0$. Generally, we take $\epsilon = 1.0D - 8$. In addition, if the norm of gradient of the Lagrange function at the current point with the multipliers obtained in solving QP_0 is less than ϵ , the execution is also terminated.

Experiments were conducted on all test problems from Ref.[12], where an initial point is provided for each problem. The results are summarized in Table 1. In that table, for each test problem (No), n is the number of variables in the problem, m_1 the number of inequality constrains, m_2 the number of equality constrains, NIT the number of iterations, NF the number of evaluations of the objective function, NC the number of evaluations of scalar constraint functions, FV the final value of the objective function, EPS the stopping criterion threshold and ACT the number of computing second order correction steps. From Table 1 we can see that, using the new nonmonotone line search, our algorithm do not almost compute the second order correction steps for the most of problems, the behavior of this algorithm appears to be competitive with that of the most effective techniques presently available.

References

[1] S.-P. Han, Superlinearly convergent variable metric algorithms for general nonlinear programming problems, *Math. Prog.,* 11(1976), 263–282.

[2] S.-P. Han, A globally convergent method for nonlinear programming, *J.O.T.A.,* 22:3 (1977), 297–309.

[3] M.J.D. Powell, A fast algorithm for nonlinearly constrained optimization calculations, in Numerical Analysis, Proceedings, Biennial conference, Dundee 1977, Lecture Notes In Math. 630, G. A. Waston, ed., Springer-Verlag, Berlin, New York, 1978, 144–157.

[4] M.J.D. Powell, Variable metric methods for constrained optimization, in A. Bachem, M. Grotschel and B. Korte, eds., Math. Prog.: the State of Art, Bonn, 1982. (Springer-Verlag, Berlin, 1983).

[5] P.T. Boggs, J.W. Tolle and P. Wang, On the local convergence of quasi-Newton methods for constrained optimization, *SIAM J.,* Control and Optimization, 20:1(1982), 161–171.

[6] R.M. Chamberlain, M.J.D. Powell, C. Lemarechal and H.C. Pedersen, The watchdog technique for forcing convergence in algorithms for constrained optimization, *Math. Prog. Study,* 16(1982), 1–17.

[7] D.Q. Mayne and E. Polak, A superlinearly convergent algorithm for constrained optimization problems, *Math. Prog. Study,* 16(1982), 45–61.

[8] M. Fukushima, A successive quadratic programming algorithm with global and superlinear convergence properties, *Math. Prog.,* 35(1986), 253–264.

[9] E.R. Panier and A.L. Tits, Avoiding the Maratos effect by means of a nonmonotone line search, I. general constrained problems, *SIAM J. Numer. Anal.,* 28(4) (1991), 1183–1195.

[10] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM J. Numer. Anal.,* 23(1986), 707–716.

[11] Y.-L. Lai and G.-P. He, A variable metric method for constrained optimization with second order correction, *J. Sys. Sci.* & *Math. Scis.,* 10(3) (1990), (in Chinese), 216–227.

[12] W. Hock and K. Schittkowski, Test examples for nonlinear programming codes, in Lecture Notes in Econom. and Math. Systems, 187, Springer-Verlag, Berlin, New York, 1981.