Journal of Global Optimization **21:** 201–218, 2001.

A Modified SQP Method with Nonmonotone Linesearch Technique

JU-LIANG ZHANG and XIANG-SUN ZHANG
Institute of Applied Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing, 100080, China (e-mail: xszhang@amath2.amt.ac.cn)

(Received 6 September 2000; accepted in revised form 20 March 2001)

Abstract. In this paper, a modified SQP method with nonmonotone line search technique is presented based on the modified quadratic subproblem proposed in Zhou (1997) and the nonmonotone line search technique. This algorithm starts from an arbitrary initial point, adjusts penalty parameter automatically and can overcome the Maratos effect. What is more, the subproblem is feasible at each iterate point. The global and local superlinear convergence properties are obtained under certain conditions.

Key words: Nonlinear optimization, SQP method, Nonmonotone line search technique, Global convergence, Superlinear convergence

1. Introduction

We consider the following constrained optimization problem:

$$\begin{aligned} \min_{x\in R^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) \leqslant 0 \end{aligned} \tag{1}$$

where $f : R^n \to R, g : R^n \to R^m$ are continuously differentiable functions. There are many practical methods for solving (1) such as gradient projection method, trust region method and SQP method. Among these methods, SQP method is an important one. SQP method is to generate iteratively a sequence $\{x_k\}$ which converges to a K–T point of the problem (1) by solving the following quadratic subproblem

$$\begin{aligned} \min_{d\in R^n} \quad & \nabla f(x_k)^T d + \tfrac{1}{2} d^T H_k d \\ \text{s.t.} \quad & g(x_k) + g'(x_k) d \leqslant 0 \end{aligned} \tag{2}$$

where $H_k \in R^{n\times n}$ is a symmetric positive definite matrix. The iterate formation is as follows

$$x_{k+1} = x_k + t_k d_k$$

where d_k is the solution of (2) and t_k is the step-size chosen by some line search to reduce the value of a merit function for (1).

SQP method is one of the most effective methods for solving nonlinear programming. Many papers contributed to this method, such as Boggs et al. (1982), Bonnons et al. (1992), Han (1976), Han (1977), Powell (1978) and Powell (1982) to name a few. But there are a lot of theoretic and practical problems which are still actively investigated. Especially if the quadratic subproblem (2) is infeasible or the solutions of the sequential quadratic subproblem are unbounded, the SQP method fails or generates a sequence which diverges. Because of this, Burke and Han (1989), Zhou (1997) modified the quadratic subproblem respectively to ensure that the subproblem is feasible at each iterate point, and proved that their methods are globally convergent. However, Han and Burke's method is only a conceptual method and can not be implementable practically. Zhou's method can be implemented in practice, but its global convergence is obtained under exact line search.

In 1978, Maratos (1978) pointed out that for SQP method, the unit step-size can not be accepted although the iterate points are close enough to the optimum of the problem (1) when the non-differentiable exact penalty function is used as the merit function and the solution of (2) is used as the search direction. This phenomenon is named as Maratos effect. For this difficulty, there are two techniques to circumvent it: Watchdog technique (Chamberlin et al., 1982) and Second–order correction technique (Mayne and Polak, 1982). Watchdog technique needs much estimation of the value of functions and their gradients, and Second–order technique needs to solve an additional quadratic subproblem or linear equation system at each iterate point. This is time-consuming.

Bonnons et al. (1992) and Panier and Titts (1991) proposed a SQP method with nonmonotone line search by using the nonmonotone line search technique proposed in Grippo et al. (1986) on the SQP method. This method needs only to solve an additional quadratic subproblem or linear equation system within finite number of iterates. Hence it overcomes the Maratos effect with less computation.

In this paper, a modified SQP method is proposed by combining the subproblem proposed in Zhou (1997) and nonmonotone line search technique. The method has the following merits: starts from an arbitrary initial point, automatically adjusts penalty parameter, the subproblem is feasible at each iterate point, and needs to solve an additional linear equation system within finite number of iterates hence overcomes the Maratos effect with less computation. Under very mild conditions, its global convergence and local superlinear convergence are obtained.

This paper is organized as follows. In Section 2, some definitions and lemmas are given. Section 3 states the algorithm model. The global convergence of the proposed algorithm is presented in Section 4. In Section 5, we study the local superlinear convergence of the proposed algorithm, and some discussions and numerical examples are given in the last section.

The symbols we use in this paper are standard. For convenience, we list some of them as follows:

(1) $f'(x,d) = \lim_{\lambda \downarrow 0}(f(x+\lambda d) - f(x))/\lambda$;

(2) $g'(x)$ is Frechet derivative of g at x;
(3) $\|x\|_\infty = \max\{|x_j|, j = 1, 2, \ldots, n\}$;
(4) $M = \{1, 2, \ldots, m\}$, $N = \{1, 2, \ldots, n\}$, $e = (1, 1, \ldots, 1)^T \in R^n$.

2. Signs and Lemmas

Let

$$g_0(x) = 0,$$

$$\Phi(x) = \max\{g_j(x) : \ j \in M \cup \{0\}\}. \tag{3}$$

The direction derivative along $d \in R^n$ of $\Phi(x)$ is

$$\Phi'(x; d) = \max_{j \in I_0(x)} \{\nabla g_j(x)^T d\} \tag{4}$$

where $I_0(x) = \{j : g_j(x) = \Phi(x), j \in M \cup \{0\}\}$.

Generally speaking, $\Phi'(x, d)$ is not continuous. In Bazaraa and Goode (1982), Bazaraa et al. proposed a continuous approximation to $\Phi'(x; d)$, which is named as pseudo-direction derivative of $\Phi(x)$ along d at x :

$$\Phi^*(x; d) = \max_{j \in I_0(x)} \{g_j(x) + \nabla g_j(x)^T d\} - \Phi(x). \tag{5}$$

It is easy to prove that $\Phi^*(x; d)$ is continuous on $R^n \times R^n$.

LEMMA 2.1 Bazaraa and Goode (1982). $\forall x, d \in R^n$, *we have*

$$\Phi^*(x; d) \geqslant \Phi'(x; d), \tag{6}$$

and there exists $\delta > 0$ *such that*

$$\Phi^*(x; td) = \Phi'(x; td), \quad \forall t \in [0, \delta].$$

LEMMA 2.2 Bazaraa and Goode (1982). $\forall x \in R^n$, $\Phi^*(x, \cdot)$ *is a convex function on* R^n.

Let

$$\Psi(x) = \max\{g_j(x), j \in M\}. \tag{7}$$

For $\forall x, d \in R^n$, let $\Psi^*(x; d)$ be the first order approximation to $\Psi(x + d)$, namely

$$\Psi^*(x; d) = \max\{g_j(x) + \nabla g_j(x)^T d, j \in M\}. \tag{8}$$

For $\forall \sigma > 0$, functions $\Psi(x, \sigma)$, $\Psi^0(x, \sigma) : R^n \times R^+ \to R$ are defined as follows

$$\Psi(x, \sigma) = \min\{\Psi^*(x; d) : \|d\| \leqslant \sigma\}, \tag{9}$$

$$\Psi^0(x, \sigma) = \max\{\Psi(x, \sigma), 0\}. \tag{10}$$

REMARK: (9) equals to the following linear programming

$$LP(x,\sigma): \quad \min\{z : g_j(x) + \nabla g_j(x)^T d \leqslant z,\ j \in m,\ \|d\|_\infty \leqslant \sigma\}.$$

Denote

$$\theta(x,\sigma) = \Psi(x,\sigma) - \Psi(x), \tag{11}$$

$$\theta^0(x,\sigma) = \Psi^0(x,\sigma) - \Psi(x), \tag{12}$$

$$F = \{x : g_j(x) \leqslant 0,\ j \in M\} = \{x : \Psi(x) \leqslant 0\}, \tag{13}$$

$$F^c = \{x : \Psi(x) > 0\}. \tag{14}$$

DEFINITION 2.1 Burke and Han (1989) . Mangasarian–Fromotz constraint qualification (MFCQ) is said to be satisfied by $g(x) \leqslant 0$ at x if $\exists z \in R^n$ such that

$$\nabla g_j(x)^T z < 0 \quad \forall j \in \{j : g_j(x) \geqslant 0,\ j \in M\}.$$

LEMMA 2.3 Zhou (1997) . *$\forall x \in F^c$, if MFCQ is satisfied at x, then $\theta(x,\sigma) < 0$, $\forall \sigma > 0$.*

LEMMA 2.4 Zhou (1997) . *$\Psi(x,\sigma)$, $\Psi^0(x,\sigma)$, $\theta(x,\sigma)$, $\theta^0(x,\sigma)$ are continuous on $R^n \times R^+$.*

LEMMA 2.5 Zhou (1997) . *$\forall x \in F^c$, if $\theta(x,\sigma) < 0$, then $\theta^0(x,\sigma) < 0$.*

3. Algorithm Model

First, we modify the quadratic subproblem of SQP method. Given $x \in R^n$, $\sigma > 0$, $D(x,\sigma,\beta)$ is defined as the following set

$$D(x,\sigma,\beta) = \{d \in R^n : g_j(x) + \nabla g_j(x)^T d \leqslant \Psi^0(x,\sigma),\ j \in m,\ \|d\|_\infty \leqslant \beta\}$$

where $\beta > \sigma$. If $d^* \in R^n$ is the solution of $LP(x,\sigma)$, then $d^* \in D(x,\sigma,\beta)$ hence $D(x,\sigma,\beta)$ is nonempty. The quadratic subproblem (2) is replaced by the following convex programming problem

$$Q(x_k, H_k, \sigma_k, \beta_k) \quad \begin{array}{rl} \min\limits_{d \in R^n} & \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\ \text{s.t.} & g_j(x_k) + \nabla g_j(x_k)^T d \leqslant \Psi^0(x_k,\sigma_k),\ j \in M \\ & \|d\|_\infty \leqslant \beta \end{array}$$

Clearly, by the above statement, the convex programming $Q(x_k, H_k, \sigma_k, \beta_k)$ is feasible when $\sigma_k \leqslant \beta_k$. And if H_k is positive definite then the solution of $Q(x_k, H_k, \sigma_k, \beta_k)$ is unique and bounded. The convex programming problem has the following properties:

THEOREM 3.1 Zhou (1997) . *Suppose that* $x_k \in R^n$, $0 < \sigma_k < \beta_k$ *and* $H_k \in R^{n\times n}$ *is a symmetric positive definite matrix. If MFCQ is satisfied at* x_k*, then*

(1). The convex programming problem $Q(x_k, H_k, \sigma_k, \beta_k)$ *has a unique solution* d_k *which satisfies K–T conditions, i.e., there exist vectors* $U^k = (u_1^k, u_2^k, \dots, u_m^k)^T$, $V^k = (v_1^k, v_2^k, \dots, v_n^k)^T$ *and* $L^k = (l_1^k, l_2^k, \dots, l_n^k)$ *such that*
(a). $g_j(x_k) + \nabla g_j(x_k)^T d_k \leqslant \Psi^0(x_k, \sigma_k)$, $j \in m$, $\|d_k\|_\infty \leqslant \beta$;
(b). $U^k \geqslant 0$, $V^k \geqslant 0$, $L^k \geqslant 0$;
(c). $\nabla f(x_k) + H_k d_k + g'(x_k)^T U^k + V^k - L^k = 0$;
(d). $\sum_{j=1}^m u_j^k (g_j(x_k) + \nabla g_j(x_k)^T d_k - \Psi^0(x_k, \sigma)) = 0$,

$$V^{kT}(d_k - \beta_k e) = 0, \quad L^{kT}(-d_k - \beta_k e) = 0;$$

(2). If $d_k = 0$ *is the solution of* $Q(x_k, H_k, \sigma_k, \beta_k)$*, then* x_k *is a K–T point of problem (1).*

LEMMA 3.1. $\forall x \in F^c$, $0 < \sigma \leqslant \beta$, *if MFCQ is satisfied at* x *and* $d \in D(x, \sigma, \beta)$, *then* $\Phi^*(x; d) \leqslant \theta^0(x, \sigma) < 0$.

LEMMA 3.2. $\forall x \in F$, $0 < \sigma \leqslant \beta$, $d \in D(x, \sigma, \beta)$, *we have* $\Psi^*(x; d) = 0$.

Now we state our algorithm as follows.

Algorithm A:

Initial: Given $x_0 \in R^n$, $\alpha_0 > 0$, $\delta > 0$, $0 < \sigma_l < \sigma_r < \bar{\beta}$, $\sigma_0 \in [\sigma_l, \sigma_r]$, $\beta_0 \in [\sigma_0, \bar{\beta}]$, $0 < \mu < \frac{1}{2}$, $0 < \gamma < 1$, Σ is a compact set which consists of symmetric positive definite matrices, $H_0 \in \Sigma$. $k = 0$.

Step 1. Compute $\Psi(x_k, \sigma_k)$, $\Psi^0(x_k, \sigma_k)$.

Step 2. Let d_k be the solution of the convex programming problem $Q(x_k, H_i, \sigma_k, \beta_k)$. If $d_k = 0$, then x_k is a K–T point of problem (1).

Step 3. If $\nabla f(x_k)^T d_k + \alpha_k \Phi^*(x_k, d_k) \leqslant -d_k^T H_k d_k$, then $\alpha_{k+1} = \alpha_k$. Otherwise, let

$$\alpha_{k+1} = \max\left\{\frac{\nabla f(x_k)^T d_k + d_k^T H_k d_k}{-\Phi^*(x_k, d_k)}, 2\alpha_k\right\}.$$

Step 4. If $P_{\alpha_{k+1}}(x_k + d_k) \leqslant \max_{l=0,1}\{P_{\alpha_{k+1}}(x_{k-l})\} - \mu d_k^T H_k d_k$, let $x_{k+1} = x_k + d_k$ and go to Step 7.

Step 5. Let $\hat{d}_k$ be the least norm solution of the following linear equation system:

$$g_j(x_k + d_k) + \nabla g_j(x_k)^T d = 0, \quad j \in I(x_k) = \{j : j \in m, u_j^k > 0\}$$

and if the above linear equation system is inconsistent or $\|\hat{d}_k\| > \|d_k\|$, then let $\hat{d}_k = 0$.

Step 6. Let $x_{k+1} = x_k + t_k d_k + t_k^2 \hat{d}_k$, where t_k is the largest value of the sequence $\{1, \gamma, \gamma^2, \dots\}$ such that

$$P_{\alpha_{k+1}}(x_k + t_k d_k + t_k^2 \hat{d}_k) \leqslant \max_{l=0,1}\{P_{\alpha_{k+1}}(x_{k-l})\} + \mu t_k (\nabla f(x_k)^T d_k + \alpha_{k+1}\Phi^*(x_k; d_k)).$$

Step 7. Choose $H_{k+1} \in \Sigma$, $\sigma_{k+1} \in [\sigma_l, \sigma_r]$, $\beta_{k+1} \in (\sigma_{k+1}, \bar{\beta}]$. Let $k := k+1$, go to Step 1.

REMARK:.

(1) H_{k+1} can be obtained by iterative formula.

(2) The merit function in the algorithm is

$$P_\alpha(x) = f(x) + \alpha\Phi(x).$$

(3) At Step 7, σ_k, β_k can be obtained by iterative formula hence Step 7 is allowed to use to include trust region strategy.

(4) From the choice of α_k, we know

$$\begin{aligned} P'_{\alpha_k}(x_k; d_k) &= \nabla f(x_k)^T d_k + \alpha_k \Phi'(x_k; d_k) \\ &\leqslant \nabla f(x_k)^T d_k + \alpha \Phi^*(x_k; d_k) \\ &\leqslant -d_k^T H_k d_k \\ &< 0. \end{aligned}$$

thereby the choice of step-size is implementable.

(5) In Zhou's algorithm, the step-size is obtained by exact line search. In this paper, the step-size is obtained by $Armijo$ line search, which is implemented easily. Moreover, we also use the nonmonotone technique in our algorithm.

4. Global Convergence

In the sequel analysis, we always assume that the following conditions hold.

Assumption A:

(1) $f, g_j, j \in M$ are continuously differentiable functions;
(2) $\{x_k\}$ is a bounded sequence;
(3) There exist $0 < b_1 \leqslant b_2 < +\infty$ such that

$$b_1\|y\|^2 \leqslant y^T H_k y \leqslant b_2\|y\|^2, \quad \forall y \in R^n, k = 1, 2, \ldots$$

holds.

THEOREM 4.1 Zhou (1997). *Assume that MFCQ is satisfied at $x_0 \in R^n$. Let $\sigma_l > 0$ and $F = \{x : g(x) \leqslant 0\}$, then there exists a neighborhood $N(x_0)$ of x_0 such that*

(1) MFCQ is satisfied at any point in $N(x_0)$;

(2) If $x_0 \in F$, then $\Psi^0(x, \sigma) = 0$, for $\forall x \in N(x_0)$ and $\sigma \geqslant \sigma_l$, and

$$\frac{\theta^0(x, \sigma)}{\Phi^*(x; d)} \leqslant 1, \quad \forall x \in N(x_0)\backslash F, \sigma \geqslant \sigma_l,$$

where d is the solution of $Q(x, H, \sigma, \beta)$;

(3) If $x_0 \in F$, then

$$\sup\left\{\sum_{j=1}^{m} u_j : H \in \Sigma, x \in N(x_0), \sigma \in [\sigma_l, \sigma_r], \beta \imath n(\sigma, \bar{\beta}]\right\} < \infty,$$

where $\Sigma \subset R^{n\times n}$ is a compact set which consists of symmetric positive definite matrices and $0 < \sigma_l < \sigma_r < \bar{\beta}$.

COROLLARY 4.1. *Suppose that $x_0 \in R^n$ satisfies $g(x_0) \leqslant 0$ and MFCQ holds at x_0. Let $0 < \sigma_l < \sigma_r < \bar{\beta}$ and Σ be a compact set consisting of symmetric positive definite matrices, then there exists a neighborhood $N(x_0)$ of x_0 and a constant number $K \geqslant 0$ such that*

$$0 \leqslant \frac{\nabla f(x)^T d + \frac{1}{2} d^T H d}{-\Phi^*(x;d)} \leqslant \frac{(\sum_{j=1}^{m} u_j)\theta^0(x,\sigma)}{\Phi^*(x;d)} \leqslant K$$

where d is the solution of $Q(x, H, \sigma, \beta)$, $\forall (x, \sigma, \beta, H) \in N(x_0) \times \Gamma(\sigma_l, \sigma_r, \bar{\beta}) \times \Sigma$, where $\Gamma(\sigma_l, \sigma_r, \bar{\beta}) = \{(\sigma,\ beta) : \sigma \in [\sigma_l, \sigma_r], \beta \in (\sigma, \bar{\beta}]\}$.

LEMMA 4.1 Zhou (1997) . If MFCQ holds, suppose that $x_k \to \bar{x}$, $H_k \to \bar{H}$, $\sigma_k \to \bar{\sigma}$, $\beta_k \to \bar{\beta}$, then $d_k \to \bar{d}$ where d_k is the solution of $Q(x_k, H_k, \sigma_k, \beta_k)$ and $\bar{d}$ is the solution of $Q(\bar{x}, \bar{H}, \bar{\sigma}, \bar{\beta})$.

LEMMA 4.2. *Suppose that $\{x_k\}$ is an infinite sequence generated by Algorithm A. If $\alpha_k \to +\infty$, as $k \to \infty$, then any cluster point $\bar{x}$ of $\{x_k\}$ satisfies the constraint conditions of (1), i.e., $g(\bar{x}) \leqslant 0$.*

Proof. If $\bar{x}$ does not satisfy the constraint condition of (1), from Lemma 2.3 and Lemma 2.5, we know that $\theta^0(\bar{x}, \sigma) < 0$, $\forall \sigma > 0$.

Because $\bar{x}$ is a cluster point of $\{x_k\}$, there exists a subsequence $\{x_{k_i}\}$ such that

$$x_{k_i} \to \bar{x}, \quad i \to \infty.$$

Without loss of generality, we can assume that $d_{k_i} \to \bar{d}$ $H_{k_i} \to \bar{H}$ $\sigma_{k_i} \to \bar{\sigma}$ $\beta_{k_i} \to \bar{\beta}$, then from Lemma 4.1 we know that $\bar{d}$ is the solution of $Q(\bar{d}, \bar{H}, \bar{\alpha}, \bar{\beta})$. Lemma 3.1 implies that $\Phi^*(\bar{x}; \bar{d}) \leqslant \theta^0(\bar{x}, \sigma) < 0$.

Because $\Phi^*(x; d)$ is continuous in $R^n \times R^n$,

$$\Phi^*(x_{k_i}; d_{k_i}) \to \Phi^*(\bar{x}; \bar{d}), \quad i \to \infty.$$

On the other hand, by $\alpha_k \to +\infty$ and updating rule for α_k, we know that

$$\frac{\nabla f(x_k)^T d_k + d_k^T H_k d_k}{-\Phi^*(x_k; d_k)} \to +\infty, \quad k \to \infty. \tag{15}$$

Assumption (1) and the computation of d_k imply that the numerator of (15) is bounded, so $\Phi^*(x_k; d_k) \to 0$, as $k \to \infty$. We obtain a contradiction, which shows that the lemma is true. □

If MFCQ is satisfied at any point of R^n, from Lemma 4.2 and Corollary 4.1 we know that α_k is a constant when k is sufficiently large. So without lose of generality, we can assume in the sequel analysis that $\alpha_k = \alpha > 0$, $\forall k$.

LEMMA 4.3. *If x_k is not a K–T point of (1), then there exists a $t_k > 0$ such that*

$$P_\alpha(x_k + t_k d_k + t_k^2 \hat{d}_k) \leqslant P_\alpha(x_k) + \mu t_k (\nabla f(x_k)^T d_k + \alpha \Phi^*(x_k; d_k))$$

Proof. Let

$$\Omega_1 = f(x_k + t d_k + t^2 \hat{d}_k) - f(x_k)$$

and

$$\Omega_2 = \Phi(x_k + t d_k + t^2 \hat{d}_k) - \Phi(x_k),$$

then $P_\alpha(x_k + t d_k + t^2 \hat{d}_k) - P_\alpha(x_k) = \Omega_1 + \alpha \Omega_2$.

Note that for sufficiently small $t > 0$ and $\forall d \in R^n$, if $j \notin I_0(x_k)$, we have that

$$g_j(x_k) + t \nabla g_j(x_k)^T d_k \leqslant \max_{j \in I_0(x_k)} \{0, g_j(x_k) + \nabla g_j(x_k)^T d_k\}.$$

Hence for sufficiently small $t > 0$, we have

$$\begin{aligned} \Omega_2 &= \max_{j \in M} \{0, g_j(x_k + t d_k + t^2 \hat{d}_k)\} - \Phi(x_k) \\ &\leqslant \max_{j \in M} \{0, g_j(x_k) + t \nabla g_j(x_k)^T d_k\} - \Phi(x_k) + o(t) \\ &= \max_{j \in I_0(x_k)} \{0, g_j(x_k) + t \nabla g_j(x_k)^T d_k\} - \Phi(x_k) + o(t) \\ &= \Phi^*(x_k; t d_k) + o(t) \\ &\leqslant t \Phi^*(x_k; d_k) + o(t), \end{aligned} \tag{16}$$

and

$$\begin{aligned} \Omega_1 &= f(x_k + t d_k + t^2 \hat{d}_k) - f(x_k) \\ &= t \nabla f(x_k)^T d_k + o(t). \end{aligned} \tag{17}$$

(16) and (17) imply that

$$P_\alpha(x_k + t d_k + t^2 \hat{d}_k) - P_\alpha(x_k) \leqslant t(\nabla f(x_k)^T d_k + \alpha \Phi^*(x_k; d_k)) + o(t). \tag{18}$$

Note that Step 3 implies

$$\nabla f(x_k)^T d_k + \alpha \Phi^*(x_k; d_k) \leqslant -d_k^T H_k d_k. \tag{19}$$

On the other hand, it follows from Theorem 3.1 that

$$d_k \neq 0. \tag{20}$$

By (18), (19), (20) and Assumption (3), we have that there exists $t_k > 0$ such that

$$P_\alpha(x_k + t_k d_k + t_k^2 \hat{d}_k) \leqslant P_\alpha(x_k) + \mu t_k (\nabla f(x_k)^T d_k + \alpha \Phi^*(x_k; d_k)).$$

So the conclusion holds. □

Lemma 4.3 says that Algorithm A is well defined. Now we prove that Algorithm A is globally convergent. First we introduce two lemmas.

LEMMA 4.4. *Sequences* $\{t_k d_k\}$ *and* $\{x_{k+1} - x_k\}$ *converge to* 0.

Proof. Let $l(k)$ be integer number such that $k - 1 \leqslant l(k) \leqslant k$ and

$$P_\alpha(x_{l(k)}) = \max_{l=0,1}\{p_\alpha(x_{k-l})\}.$$

From Lemma 4.3, we obtain

$$\begin{aligned} P_\alpha(x_{l(k+1)}) &= \max_{l=0,1}\{P_\alpha(x_{k+1-l})\} \\ &\leqslant \max\{P_\alpha(x_{l(k)}), P_\alpha(x_{k+1})\} \\ &= P_\alpha(x_{l(k)}), \end{aligned} \tag{21}$$

i.e., sequence $\{P_\alpha(x_{l(k)})\}$ is a non-increasing sequence. Therefore, by Step 3 and Step 6, we have

$$\begin{aligned} P_\alpha(x_{l(k)}) &\leqslant \max_{l=0,1}\{P_\alpha(x_{l(k)-1-l})\} + \mu t_{l(k)-1}(\nabla f(x_{l(k)-1})^T d_{l(k)-1} \\ &\quad + \alpha \Phi(x_{l(k)-1}; d_{l(k)-1})) \\ &\leqslant P_\alpha(x_{l(l(k)-1)}) - \mu t_{l(k)-1} d_{l(k)-1}^T H_{l(k)-1} d_{l(k)-1}. \end{aligned} \tag{22}$$

From (22) and Assumption (1), we obtain

$$t_{l(k)-1} d_{l(k)-1} \to 0, \quad k \to \infty. \tag{23}$$

Since $\|\hat{d}_k\| \leqslant \|d_k\|$, then

$$\|x_{l(k)} - x_{l(k)-1}\| \to 0, \quad k \to 0. \tag{24}$$

Now set $\hat{l}(k) = l(k+3)$ and show, by induction, that for any $j \geqslant 1$,

$$\lim_{k\to\infty} t_{\hat{l}(k)-j} d_{\hat{l}(k)-j} = 0, \tag{25}$$

$$\lim_{k\to\infty} P_\alpha(x_{\hat{l}(k)-j}) = \lim_{k\to\infty} P_\alpha(x_{l(k)}). \tag{26}$$

In view of (21)–(24) and the fact $\{\hat{l}(k)\} \subset \{l(k)\}$, we have that

$$\begin{aligned} &|P_\alpha(x_{\hat{l}(k)-1}) - P_\alpha(x_{l(k)-1})| \\ &\quad \leqslant |P_\alpha(x_{\hat{l}(k)-1}) - P_\alpha(x_{\hat{l}(k)})| + |P_\alpha(x_{\hat{l}(k)}) - P_\alpha(x_{l(k)-1})| \to 0, \quad k \to \infty. \end{aligned}$$

So (25) and (26) hold for $j = 1$. Assume that (25) and (26) hold for a given j. By (22), we have that

$$P_\alpha(x_{\hat{l}(k)-j}) \leqslant P_\alpha(x_{l(\hat{l}(k)-j-1)}) - \mu t_{\hat{l}(k)-j-1} d^T_{\hat{l}(k)-j-1} H_{\hat{l}(k)-j-1} d_{\hat{l}(k)-j-1}.$$

By induction assumptions, we know that

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

So

$$t_{\hat{l}(k)-j-1} d_{\hat{l}(k)-j-1} \to 0, \quad k \to \infty,$$

and

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

and furthermore

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

Therefore (25) and (26) hold for $j + 1$.

For any j, since $\hat{l}(k) - k - 1 = l(k+3) - k - 1 \leqslant 2$, and

$$x_{k+1} = x_{\hat{l}(k)} - \sum_{j=1}^{\hat{l}(k)-k-1} [t_{\hat{l}(k)-j} d_{\hat{l}(k)-j} + t^2_{\hat{l}(k)-j} \hat{d}_{\hat{l}(k)-j}],$$

by (25) and (26),

$$\|x_{k+1} - x_{\hat{l}(k)}\| \to 0, \quad |P_\alpha(x_{k+1}) - P_\alpha(x_{\hat{l}(k)})| \to 0, \quad k \to \infty.$$

Consequently

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

Note that

$$P_\alpha(x_{k+1}) \leqslant P_\alpha(x_{l(k)}) - \mu t_k d_k^T H_k d_k,$$

we obtain

$$t_k d_k \to 0, \quad \|x_{k+1} - x_k\| \to 0, \quad k \to \infty. \qquad \square$$

LEMMA 4.5. *Let sequences* $\{x_k\}$ *and* $\{d_k\}$ *be generated by Algorithm A, then* $d_k \to 0$.

Proof. From Lemma 4.4, we know that

$$t_k d_k \to 0, \quad (k \to \infty). \tag{27}$$

Now we prove that $d_k \to 0$. Conversely, if $d_k \not\to 0$, then there exit a subsequence $\{d_i\}$ of $\{d_k\}$ and a positive constant number ϵ such that

$$\|d_i\| \geqslant \epsilon, \quad \forall i. \tag{28}$$

Now we prove that there exists $t' > 0$ such that

$$t_i \geqslant t', \quad \forall i. \tag{29}$$

Assume that (29) does not hold, then there exists a subsequence of $\{t_i\}$(without loss of generality, we can assume that the subsequence is $\{t_i\}$ itself) such that

$$t_i \to 0, \quad i \to \infty.$$

From Step 6, we have that

$$\begin{aligned} &P_\alpha\left(x_i + \frac{t_i}{\eta}d_i + \frac{t_i^2}{\eta}\hat{d}_i\right) \\ &\quad > \max_{l=0,1}\{P_\alpha(x_{i-l})\} + \mu\frac{t_i}{\eta}(\nabla f(x_i)^T d_i + \alpha\Phi^*(x;_i\,; d_i)) \qquad (30) \\ &\quad \geqslant P_\alpha(x_i) + \mu\frac{t_i}{\eta}(\nabla f(x_i)^T d_i + \alpha\Phi^*(x;_i\,; d_i)). \end{aligned}$$

By (18), $t_i \to 0$ and (30), we know that for i sufficiently large

$$\begin{aligned} &\frac{t_i}{\eta}(\nabla f(x_i)^T d_i + \alpha\Phi^*(x_i; d_i)) + o\left(\frac{t_i}{\eta}\right) \\ &\quad \geqslant P_\alpha\left(x_i + \frac{t_i}{\eta}d_i + \frac{t_i^2}{\eta}\hat{d}_i\right) - P_\alpha(x_i) \qquad (31) \\ &\quad \geqslant \mu\frac{t_i}{\eta}(\nabla f(x_i)^T d_i + \alpha\Phi^*(x_i; d_i)), \qquad (32) \end{aligned}$$

i.e.,

$$(1-\mu)\frac{t_i}{\eta}\left(\nabla f(x_i)^T d_i + \alpha\Phi^*(x_i; d_i)\right) + o\left(\frac{t_i}{\eta}\right) \geqslant 0.$$

It follows from the choice of α, (28) and Assumption A (3) that

$$-(1-\mu)b_1\epsilon^2 + o\left(\frac{t_i}{\eta}\right)/\frac{t_i}{\eta} \geqslant 0.$$

Let $i \to \infty$, and note that $t_i \to 0$, we obtain that

$$-(1-\mu)b_1\epsilon^2 \geqslant 0.$$

This contradicts $0 < \mu < \frac{1}{2}$. So (29) holds. (28) and (29) imply that $t_k d_k \not\to 0$. This contradicts to (27). The contradiction shows that the lemma is true. □

Combining Lemma 4.5, Lemma 4.1 and Theorem 3.1, we obtain

THEOREM 4.2. *If MFCQ holds at any $x \in R^n$, Algorithm A either stops at a K–T point of problem (1) or generates an infinite sequence $\{x_k\}$ whose cluster points are K–T points of problem (1).*

5. Superlinear Convergence

In this section, we prove that the algorithm is convergent superlinearly. For the analysis of the superlinear convergence of the algorithm, we need the following assumptions.

Assumption B:

(1′). Functions $f, g_j, j \in M$ are at least twice order continuously differentiable;

(4). Strong twice order sufficient conditions holds, i.e.,

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

where

$$L(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x), \hat{I}(x^*) = \{j \in I_0(x^*), \lambda_j^* > 0\}$$

and (x^*, λ^*) is a K–T pair of problem (1);

(5). At x^*, strict complementarity slackness and linear independence of the gradients of the active constraints hold;

(6). Matrices H_k, $k = 1, 2, \ldots$ are symmetric positive definite and satisfy the following condition

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

From the Assumption (1′) (2)–(6), we have the following lemma which is similar to Robinson (1982) and Bonnans and Launay (1995).

LEMMA 5.1. *Sequence $\{x_k\}$ converges to the solution x^* of problem (1).*

From Lemma 5.1 and Lemma 4.5, we know that $\|d_k\| \to 0$. So the constraint condition $\|d\|_\infty \leqslant \beta_k$ in $Q(x_k, H_k, \sigma_k, \beta_k)$ is redundant when k is sufficiently large. Theorem 4.1 (2) implies that $\Psi(x_k, \sigma_k) = 0$ for k sufficiently large. So the subproblem $Q(x_k, H_k, \sigma_k, \beta_k)$ is equivalent to the following quadratic programming subproblem when k is sufficiently large.

$$\begin{aligned} \min_{d \in R^n} \quad & \nabla f(x_k)^T d + \tfrac{1}{2} d^T H_k d \\ \text{s.t.} \quad & g_j(x_k) + \nabla g_j(x_k)^T d \leqslant 0, \ j \in M \end{aligned} \tag{33}$$

LEMMA 5.2. $(d_k, \lambda_k) \to (0, \lambda^*)$, *where* (x^*, λ^*) *is a K–T pair of problem (1) and* (d_k, λ_k) *is the K–T pair of the above quadratic programming subproblem.*

Proof. It is easy to verify. □

From Lemma 5.2 and the statements above, we have

$$U^k \to \lambda^*, \quad k \to \infty.$$

LEMMA 5.3.

$$\|\hat{d}_k\| = O(\|d_k\|^2) \quad \forall k \text{ sufficiently large.}$$

Proof. It is similar to Proposition 4.1 in De Q. Pantoja and Mayne (1991). □

LEMMA 5.4. *Suppose that* $\{x_k\}$ *is an infinite sequence generated by algorithm A, If Assumption (1′) and (2)–(6), then* $t_k = 1$, *for all* k *sufficiently large.*

Proof. From Lemma 5.3 and $\|d_k\| \to 0$, as $k \to \infty$, we know that for all k sufficiently large

$$\|\hat{d}_k\| < \|d_k\|.$$

Now we prove that the step-size $t_k = 1$ for all k sufficiently large. First we need to prove

$$P_\alpha(x_k + d_k + \hat{d}_k) - P_\alpha(x_k) \leqslant \mu(\nabla f(x_k)^T d_k + \alpha\Phi^*(x_k; d_k)). \tag{34}$$

From Assumption B(5), the gradients of the active constraints are linearly independent. Hence for all k sufficiently large $\hat{d}_k \neq 0$.

Now we prove that for all k sufficiently large, (34) holds. We need only to prove

$$T_k = P_\alpha(x_k + d_k + \hat{d}_k) - P_\alpha(x_k) - \mu(\nabla f(x_k)^T d_k + \alpha\Phi^*(x_k; d_k)) \leqslant 0. \tag{35}$$

Since $\|d_k\| \to 0$ as $k \to \infty$ and Lemma 5.3, then

$$P_\alpha(x_k + d_k + \hat{d}_k) - P_\alpha(x_k)$$
$$\leqslant \nabla f(x_k)^T d_k + \alpha\Phi^*(x_k; d_k) + \nabla f(x_k)^T \hat{d}_k + \frac{1}{2} d_k^T \nabla^2_{xx} f(x_k) d_k + o(\|d_k\|^2).$$

From Lemma 5.1 and the statement before Lemma 5.1, we have that

$$\nabla f(x_k) = -H_k d_k - g'(x_k)^T U^k.$$

From the boundness of H_k, the definition of $\hat{d}_k$ and Lemma 5.3, we know

$$\nabla f(x_k)^T \hat{d}_k = {U^k}^T g(x_k + d_k) + o(\|d_k\|^2).$$

Note that

$$u_k^j(g_j(x_k) + \nabla g_j(x_k)^T d_k) = 0, \quad \forall j \in m,$$

and

$$\nabla f(x_k)^T d_k + \alpha \Phi^*(x_k; d_k) \leqslant -d_k^T H_k d_k,$$

we have

$$\begin{aligned} T_k &\leqslant \left(\frac{1}{2} - \mu\right)(\nabla f(x_k)^T d_k + \alpha \Phi^*(x_k; d_k)) \\ &\quad + \frac{1}{2} d_k^T (\nabla_{xx}^2 L(x_k, U^k) - H_k) d_k + o(\|d_k\|^2) \\ &\leqslant -\left(\frac{1}{2} - \mu\right) b_1 \|d_k\|^2 + \frac{1}{2} d_k^T (L(x_k, U^k) - L(x^*, \lambda^*)) d_k \\ &\quad + \frac{1}{2} d_k^T (L(x^*, \lambda^*) - H_k) d_k + o(\|d_k\|^2). \end{aligned}$$

Since $x_k \to x^*$, $U^k \to \lambda^*$ and Assumption A (2), then

$$d_k^T (L(x_k, U^k) - L(x^*, \lambda^*)) d_k = o(\|d_k\|^2).$$

Assumption B (6) implies that

$$d_k^T (L(x^*, \lambda^*) - H_k) d_k = o(\|d_k\|^2).$$

Therefore when k is sufficiently large

$$T_k \leqslant -\left(\frac{1}{2} - u\right) b_1 \|d_k\|^2 + o\left(\|d_k\|^2\right) \leqslant 0.$$

Hence for all k sufficiently large, (35) holds.

From (34) and Step 6, we obtain

$$\begin{aligned} P_\alpha(x_k + d_k + \hat{d}_k) - \max_{l=0,1}\{P_\alpha(x_{k-l}) &\leqslant P_\alpha(x_k + d_k + \hat{d}_k) - P_\alpha(x_k) \\ &\leqslant \mu(\nabla f(x_k)^T d_k + \alpha \Phi^*(x_k; d_k)). \end{aligned} \tag{36}$$

Hence $t_k = 1$ for k sufficiently large. □

From Lemma 5.4 and the definition of the algorithm, we know that for all k sufficiently large, either $x_{k+1} = x_k + d_k$ or $x_{k+1} = x_k + d_k + \hat{d}_k$.

THEOREM 5.1. *If the conditions in Lemma 5.4 hold, then* $\{x_k\}$ *converges to* x^* *superlinearly, i.e.,*

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Proof. By assumptions and the results in Boggs et al. (1982), we have

$$\lim_{k\to\infty} \frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} = 0.$$

Since

$$\frac{\|x_k + d_k + \hat{d}_k - x^*\|}{\|x^k - x^*\|} \leqslant \frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} + \frac{\|d_k\|}{\|x_k - x^*\|} \cdot \frac{\|\hat{d}_k\|}{\|d_k\|},$$

let $k \to \infty$, note that $\frac{\|d_k\|}{\|x_k - x^*\|} \to 1$ and $\frac{\|\hat{d}_k\|}{\|d_k\|} \to 0$, then

$$\frac{\|x_k + d_k + \hat{d}_k - x^*\|}{\|x^k - x^*\|} \to 0.$$

Since for k sufficiently large, either $x_{k+1} = x_k + d_k$ or $x_{k+1} = x_k + d_k + \hat{d}_k$, so

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$ □

Now we state another principal result. It shows that the linear equation system in Step 5 needs only to be solved in finite number of iterates. This is the main reason that we introduce the nonmonotone line search technique.

THEOREM 5.2. *When k is sufficient large, Step 4 in Algorithm A is always satisfied, hence Step 5 and Step 6 will not be executed.*

Proof. We need only to prove that for sufficiently large k we have

$$P_\alpha(x_k + d_k) - \mu d_k^T H_k d_k \leqslant P_\alpha(x_{k-1}). \tag{37}$$

Now we assume that k is so large that d_k is calculated by (33) and $t_k = 1$, so

$$\begin{aligned} g_j(x_k + d_k) &= g_j(x_k) + \nabla g_j(x_k)^T d_k + O(\|d_k\|^2) \\ &\leqslant O(\|d_k\|^2). \end{aligned}$$

Since

$$\lim_{k\to\infty} \frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} = 0,$$

then

$$\begin{aligned} P_\alpha(x_k + d_k) &= f(x_k + d_k) + \alpha \max_{j\in M\cup\{0\}} \{g_j(x_k + d_k)\} \\ &\leqslant f(x^* + x_k + d_k - x^*) + O(\|d_k\|^2) \\ &= f(x^*) + \nabla f(x^*)^T (x_k + d_k - x^*) + O(\|x_k + d_k - x^*\|^2) \\ &\quad + O(\|d_k\|^2) \\ &= P_\alpha(x^*) - \textstyle\sum_{j\in \hat{I}(x^*)} \lambda_j^* \nabla g_j(x^*)^T (x_k + d_k - x^*) \\ &\quad + o(\|x_k - x^*\|^2) + O(\|d_k\|^2). \end{aligned}$$

Since $U^k \to \lambda^*$, then $\hat{I}(x^*) \subseteq I(x_k)$. So for $j \in \hat{I}(x^*)$

$$\begin{aligned} O(\|d_k\|^2) &= g_j(x_k) + \nabla g_j(x_k)^T d_k + O(\|d_k\|^2) \\ &= g_j(x_k + d_k) \\ &= g_j(x^* + x_k + d_k - x^*) \\ &= g_j(x^*) + \nabla g_j(x^*)^T (x_k + d_k - x^*) + O(\|x_k + d_k - x^*\|^2) \\ &= \nabla g_j(x^*)^T (x_k + d_k - x^*) + o(\|x_k - x^*\|^2). \end{aligned}$$

By Theorem 5.1 and note that

$$\lim_{k\to\infty} \frac{\|d_k\|}{\|x_k - x^*\|} = 1,$$

we obtain

$$\begin{aligned} P_\alpha(x_k + d_k) - \mu d_k^T H_k d_k &= P_\alpha(x^*) + o(\|x_k - x^*\|^2) + O(\|d_k\|^2) \\ &= P_\alpha(x^*) + O(\|x_k - x^*\|^2) \\ &= P_\alpha(x^*) + o(\|x_{k-1} - x^*\|^2) \\ &\leqslant P_\alpha(x_{k-1}), \end{aligned}$$

where the last inequality follows from Lemma 1 in Chamberlin et al. (1982). □

6. Some Discussions and Numerical Examples

In this section, we give some numerical examples to show the success of the proposed method. Updating of H_k is done by BFGS formula, i.e.,

$$H_{k+1} = \begin{cases} H_k, & \text{if } s_k^T y_k \leqslant 0; \\ H_k + \dfrac{y_k y_k^T}{y_k^T s_k} - \dfrac{H_k s_k s_k^T H_k}{s_k^T H_k s_k}, & \text{if } s_k^T y_k > 0, \end{cases}$$

where $s_k = x_{k+1} - x_k$, $y_k = (\nabla f(x_{k+1}) - \nabla g(x_{k+1}) U^{k+1}) - (\nabla f(x_k) - \nabla g(x_k) U^k)$, and U^k is defined as in Theorem 3.1. The stop criteria is $\|d_k\| \leqslant 10^{-6}$. And the algorithm parameters were set as follows: $\alpha_0 = 100$, $\delta = 1$, $\sigma_l = 1$, $\sigma_r = 2$, $\bar{\beta} = 3$, $\mu = 0.25$, $\gamma = 0.5$ and $H_0 = I \in R^{n\times n}$. The program is written in MATLAB and call for the inner function QP in matlab to solve the quadratic subproblems.

EXAMPLE 1.

$$\begin{aligned} \min \quad & f(x) = x - \tfrac{1}{2} + \tfrac{1}{2}\cos^2 x, \\ \text{s.t.} \quad & x \geqslant 0. \\ & x_0 = 2,\ x^* = 0,\ f(x^*) = 0, \text{ iterate} = 2. \end{aligned}$$

EXAMPLE 2.

$$\begin{aligned} \min\ & f(x) = \textstyle\sum_{i=1}^{4} x_i^2, \\ \text{s.t.}\ & 6 - \textstyle\sum_{i=1}^{4} x_i^2 \leqslant 0. \\ & x_0 = (2, 2, 2, 2)^T,\ x^* = (1.224745, 1.224745, 1.224745, 1.224745)^T, \\ & f(x^*) = 9, \text{ iterate} = 7. \end{aligned}$$

EXAMPLE 3.

$$\begin{aligned} \min\ & f(x) = \textstyle\sum_{i=1}^{3} x_i^2 x_{i+1}^2 + x_1 x_4, \\ \text{s.t.}\ & 4 - \textstyle\sum_{i=1}^{4} x_i \leqslant 0, \\ & 1 - \textstyle\sum_{i=1}^{4} (-1)^{i+1} x_i \leqslant 0. \\ & x_0 = (2.5, 1.5, 0, 0)^T, \\ & x^* = (1.240023, 0.753253, 1.259977, 0.746746)^T, \\ & f(x^*) = 3.515915, \text{ iterate} = 6. \end{aligned}$$

EXAMPLE 4.

$$\begin{aligned} \min\ & f(x) = \tfrac{4}{3}(x_1^2 - x_1 x_2 + x_2^2)^{\frac{3}{4}} - x_3, \\ \text{s.t.}\ & x \geqslant 0,\ x_3 \leqslant 2. \\ & x_0 = (0, 0.25, 0)^T,\ x^* = (0, 0, 2)^T,\ f(x^*) = -2, \text{ iterate} = 8. \end{aligned}$$

From the above, we know that the algorithm can solve these problems. Comparing with the results in Zhou (1997), the computation in each iteration in this paper is less than that in Zhou's method since they use exact line search to obtain step-size. Because we use nonmonotone line search technique in our method, the iterate number for some problems is less than that in Zhou (1997).

Although the method in this paper is proposed for inequality constrained problem, we can apply this method to solve general optimization problem. If an equality constraint $h(x) = 0$ exists in the original problem, it can be handled as two corresponding inequality $h(x) \leqslant 0$ and $h(x) \geqslant 0$, and we can apply the above algorithm.

The method proposed in this paper has advantage over traditional SQP method. The following example demonstrates situations in which the algorithm proposed in this paper succeeds while the SQP method developed by Wilson, Han and Powell can fail if the initial value of x is set to 3.

$$\begin{aligned} \min\ & x \\ \text{s.t.}\ & x \leqslant 1, \\ & x^2 \geqslant 0. \end{aligned}$$

Acknowledgements

The authors thank the referees who gave us valuable suggestions and comments for improving the paper. This work is supported in part by the National Natural Science Foundation of China (Grant No. 39830070).

References

Bazaraa, M.S. and Goode, J.J. (1982), An algorithm for solving linearly constraint minimax problem, *European Journal of Operational Research* 11: 158–166.

Burke, J.V. and Han, S.P. (1989), A robust SQP methods, *Mathematical Programming* 43: 277–303.

Boggs, P.T., Tolle, J.W. and Wang, P. (1982), On the local convergence of quasinewton methods for constrained optimization, *SIAM J. on Control and Optimization* 20: 161–171.

Bonnans, J.F. and Launay, G. (1995), Sequential quadratic programming with penalization of displacement, *SIAM J. Optimization* 54: 792–812.

Bonnons, J.F., Panier, E.R., Titts, A.L. and Zhou.J.L. (1992), Avoiding the Maratos effect by means of nonmonotone linesearch, Inequality constrained problems – feasible iterates, *SIAM J. Numerical Analysis* 29: 1187–1202.

Chamberlin, R.M., Powell, M.J.D., Lemarechal, C. and Pedersen, H.C. (1982), The watchdog technique for forcing convergence in algorithms for constrained optimization, *Mathematical Programming Study* 16: 1–7.

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

Grippo, L., Lampariellp, F. and Lucidi, S. (1986), A nonmonotone line search technique for Newton's method, *SIAM J. Numerical Analysis* 23: 707–716.

Han, S.P. (1976), Superlinearly convergent variable metric algorithm for general nonlinear programming problems, *Mathematical Programming* 11: 263–282.

Han, S.P. (1977), A global convergence method for nonlinear programming, *Journal of Optimization Theory and Application* 22: 217–309.

Han, S.P. and Mangasarian, O.L. (1976), Exact penalty function in nonlinear programming, *Mathematical Programming* 17: 251–269.

Maratos, N.(1978), *Exact penalty function algorithm for finite dimensional and control optimization problems*, Ph.D. Thesis, University of London, UK.

Mayne, D.Q. and Polack, E. (1982), A superlinealy convergent algorithm for constrained optimization problems, *Mathematical Programming Study* 16: 45–61.

Panier, E.R. and Titts, A.L. (1991), Avoiding the Maratos effect by means of a nonmonotone line search I. General constrained problems, *SIAM J. Numerical Analysis* 28: 1183–1195.

Powell, M.J.D. (1978), A fast algorithms for nonlinearly constrained optimization calculations, in Watson, G.A. (ed.), *Numerical Analysis*, Springer-Verlag, Berlin, 144–157.

Powell, M.J.D. (1982), Variable metric methods for constrained optimization, in Bächen et al. (eds.), *Mathematical programming – The state of art*, Springer-Verlag, Berlin.

Robison, S.M. (1982), Generalized equations and their applications, Part II: application to nonlinear programming, *Mathematical Programming Study* 19: 200–221.

Xue, G.L. (1991), Algorithms and their convergent properties for solving a class linearly constraint minimax problems, *Comn. on Appl. Math. and Comput.* 5: 11–18.

Zhou, G.L. (1997), A modified SQP method and its global convergence, *J.Global Optimization* 11: 193–205.