AMO - Advanced Modeling and Optimization, Volume 18, Number 1, 2016

A variant feasible sequential quadratic programming method and its superlinear convergence[1]

Lirong Wang and Quan Huang

The Department of Information science and Engineering,

Hunan University of Humanities, Science and Technology, Loudi, 417000,P.R.China

Abstract. In this paper, a variant feasible sequential quadratic programming method is proposed to solve the nonlinear programming problem. The method has the following advantages: starts from any initial point, automatically adjusts penalty parameter, the subproblem is feasible at each iterate point, and the global and superlinear convergence are proved under some suitable conditions. Furthermore, numerical results reported show that the algorithm in this paper is effective.

Key words. Nonlinear programming, SQP method, Global convergence, superlinear convergence

1. Introduction

Consider the following nonlinear programming :

$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_j(x) \le 0, j \in L = \{1, 2, \dots, m\}, \end{array} \tag{1.1}$$

where function $f_0(x) : R^n \to R$ and $f_j(x) : R^n \to R$ are assumed to be twice continuously differentiable.

SQP(Sequential Quadratic Programming) method is one of the most effective methods for solving nonlinear programming. Many papers contributed to this method, such as [1] - [7]. SQP algorithms generate iteratively the main search direction d_0 by solving the following quadratic programming(QP) subproblem:

$$\begin{array}{ll} \min & \nabla f_0(x)^T d + \frac{1}{2} d^T B d \\ \text{s.t.} & f_j(x) + \nabla f_j(x)^T d \le 0, j = 1, 2, \dots, m, \end{array} \tag{1.2}$$

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

$$x^{k+1} = x^k + \lambda_k d_k,$$

where d_k is the desired solution for solving the subproblem (1.2) and λ_k is the step-size chosen to yield the greatest decrease in the function $f_0(x)$ by some line search.

However, such type SQP algorithms have two serious shortcomings: (1) SQP algorithms require that the relate QP subproblem (1.2) must be consistency. (2) There exists Matatos effect [1].

To overcome these faults, [3]-[5] etc. modified the quadratic subproblem to make sure the associated constraint region of QP subproblem is nonempty and a robust convergence theory can be established. However, Han and Burke's method is only implemented in theory and can not be implementable practically.Zhou's method can be implemented in practice with the help

[1]This work was supported in part by Scientific Research Fund of Hunan Provincial Education Department (NO.14C0609 and 12C0743).

AMO -Advanced Modeling and Optimization. ISSN: 1841-4311

of exact line search, but it can not overcome Maratos effect. Zhang's method is obtained by combing Armijo line search and nonmonotone technique to overcome the Maratos effect, consequently global convergence and local superlinear convergence are reached, however, its prove process is too.

In this paper, a modified FSQP method is proposed on the basic of [5]. However, we simple the method by not using the nonmonotone technique. The method has the following advantages: starts from any initial point, automatically adjusts penalty parameter, the subproblem is feasible at each iterate point, under some reasonable conditions, the global convergence and superlinear convergence are shown.

This paper is organized as follows. Some definitions and lemmas are given in section 2. In section 3, our algorithm is presented. The well-defined of our approach is also discussed, the accountability of which allows us to present global convergence guarantees under common conditions in Section 4, while in Section 5 we deal with superlinear convergence. Finally, in Section 6, some numerical experiments are implemented.

2. Signs and Lemmas

In this section, we adopt the penalty function associated with (1.1) as a merit function for obtaining the global and local superlinear convergence properties.

$$F(x) = \max_{j\in I}\{f_j(x),\ 0\}. \tag{2.1}$$

Then the direction derivative of $F(x)$ along any direction $d \in R^n$ is

$$F'(x;d) = \max_{j\in I_0(x)}\{\nabla f_j(x)^T d\}, \tag{2.2}$$

where $I_0(x) = \{j : f_j(x) = F(x), j \in L\}$.

Generally speaking, $F'(x;d)$ is not continuous. In[4], gave a continuous approximation to $F'(x;d)$ which is called pseudo directional derivative along any direction d at x:

$$F^*(x;d) = \max_{j\in I_0(x)}\{f_j(x) + \nabla f_j(x)^T d\} - F(x) \tag{2.3}$$

Lemma 2.1. *[4]*
1) For any $x, d \in R^n$, we have

$$F^*(x;d) \geq F'(x;d) \tag{2.4}$$

and there exists $\delta > 0$ such that

$$F^*(x;td) = F'(x;td), \forall t \in [0,\delta]. \tag{2.5}$$

2)For any $x \in R^n, F^*(x;\cdot) : R^n \to R$ is a convex function.

Let

$$\psi(x) = \max\{f_j(x), j \in L\}. \tag{2.6}$$

For any $x, d \in R^n$, let $\psi^*(x;d)$ be the first-order approximation to $\psi(x+d)$:

$$\psi^*(x;d) = \max\{f_j(x) + \nabla f_j(x)^T d, j \in L\}. \tag{2.7}$$

For any $\sigma > 0$, function $\psi(x,\sigma), \psi^0(x,\sigma) : R^n \times R^+ \to R$ are defined as follows:

$$\psi(x,\sigma) = \min\{\psi^*(x;d) : ||d|| \le \sigma\}. \tag{2.8}$$

$$\psi^0(x,\sigma) = \max\{\psi(x,\sigma), 0\}. \tag{2.9}$$

REMARK: (2.8) is equivalent to the following linear programming ($LP(x,\sigma)$):

$$\min\{z : f_j(x) + \nabla f_j(x)^T d \le z, j \in L, ||d||_\infty \le \sigma\}$$

Let

$$\theta(x,\sigma) = \psi(x,\sigma) - \psi(x), \tag{2.10}$$

$$\theta^0(x,\sigma) = \psi^0(x,\sigma) - \psi(x), \tag{2.11}$$

$$G = \{x : f_j(x) \le 0, j \in L\} \Leftrightarrow \{x : \psi(x) \le 0\}, \tag{2.12}$$

$$G^c = \{x : \psi(x) > 0\}. \tag{2.13}$$

Definition 2.1. *The Mangasarian-Fromowitz constraint qualification (MFCQ) is said to be satisfied by $f_j(x) \le 0$ at x if there exists a $z \in R^n$ such that*

$$\nabla f_j(x)^T z < 0, \forall j \in \{j : f_j(x) \ge 0\}, j \in L\}$$

Lemma 2.2. *[4]*
1) If MFCQ is satisfied for any $x \in G^c$, then $\theta(x,\sigma) < 0$.
2) $\psi(x,\sigma), \psi^0(x,\sigma), \theta(x,\sigma), \theta^0(x,\sigma)$ are continuous on $R^n \times R^+$.
3)If $\theta(x,\sigma) < 0$ for any $x \in G^c$, then $\theta^0(x,\sigma) < 0$.

We define the following set

$$E(x,\sigma,\beta) = \{d \in R^n : f_j(x) + \nabla f_j(x)^T d \le \psi^0(x,\sigma), j \in L, ||d||_\infty \le \beta\},$$

where $\beta > \sigma$. It can be seen if $d^* \in R^n$ is the solution of $LP(x,\sigma)$, then $d^* \in E(x,\sigma,\beta)$. So $E(x,\sigma,\beta)$ is nonempty. Now, we describe the modification to the subproblem (1.2)with the following convex programming problem $Q(x^k, B_k, \sigma_k, \beta_k)$:

$$\begin{array}{ll} \min & \nabla f_0(x)^T d + \frac{1}{2} d^T B_k d \\ \text{s.t.} & f_j(x^k) + \nabla f_j(x^k)^T d \le \psi^0(x^k, \sigma_k), j \in L, \\ & ||d||_\infty \le \beta \end{array} \tag{2.14}$$

It is obvious the convex programming $Q(x^k, B_k, \sigma_k, \beta_k)$ is feasible when $\sigma_k \le \beta_k$. And when B_k is positive definite then the solution of $Q(x^k, B_k, \sigma_k, \beta_k)$ is unique and bounded. The convex programming has the following properties:

Theorem 2.3. *[5] Let $x^k \in R^n, 0 < \sigma_k < \beta_k$, and $B_k \in R^{n\times n}$ be a symmetric and positive definite matrix. If MFCQ is satisfied at x^k, then*
(1) the convex programming problem $Q(x^k, B_k, \sigma_k, \beta_k)$ has a unique solution d_k where d_k satisfies the following KKT conditions: there exists vectors $\widetilde{U}^k = (\widetilde{u}_1^k, \widetilde{u}_2^k, \ldots, \widetilde{u}_m^k)^T, V^k = (v_1^k, v_2^k, \ldots, v_n^k)^T$ and $L^k = (l_1^k, l_2^k, \ldots, l_n^k)^T, e = (1, 1, \ldots, 1)^T$ that
a) $f_j(x^k) + \nabla f_j(x^k)^T d_k \le \psi^0(x^k, \sigma_k), j \in L, ||d_k||_\infty \le \beta,$
b) $\widetilde{U}^k \ge 0, V^k \ge 0, L^k \ge 0,$
c) $\nabla f_0(x^k) + B_k d_k + \nabla f_j(x^k)^T \widetilde{U}^k + V^k - L^k = 0,$
d) $\sum_{j=1}^{m} \widetilde{u}_j^k (f_j(x^k) + \nabla f_j(x^k)^T d_k - \psi^0(x^k, \sigma_k)) = 0, V^{kT}(d_k - \beta_k e) = 0, L^{kT}(d_k + \beta_k e) = 0;$
(2) if $d_k = 0$ is the solution of $Q(x^k, B_k, \sigma_k, \beta_k)$, then x^k is a KKT point of the problem (1.1).

Lemma 2.4. *[5] 1) For any $x \in G^c, 0 < \sigma \le \beta$, if MFCQ is satisfied at x and $d \in E(x, \sigma, \beta)$, then $F^*(x; d) \le \theta^0(x, \sigma) < 0$.*
2) For any $x \in G, 0 < \sigma \le \beta, d \in E(x, \sigma, \beta)$, we have $\psi^*(x; d) = 0$.

3. Description of Algorithm

Now, we describe the algorithm as follows.

Algorithm:

Step 0 : Given $x^0 \in R^n, c_0 > 0, \delta > 0, 0 < \sigma_l < \sigma_r < \bar{\beta}, 0 < \alpha < \frac{1}{2}, 0 < \gamma < 1, 2 < \tau < 3$, $\sum$ is a compact set which consists of symmetric positive definite matrices, $B_0 \in \sum, k = 0$.

Step 1 : Compute $\psi(x^k, \sigma_k), \psi^0(x^k, \sigma_k)$.

Step 2 : Let d_0^k be the solution of the convex programming problem $Q(x^k, B_k, \sigma_k, \beta_k)$. If $d_0^k = 0$, then x^k is a KKT point of (1.1).STOP.

Step 3 : If $\nabla f_0(x^k)^T d_0^k + c_k F^*(x^k, d_0^k) \le -d_0^k B_k d_0^k$, then $c_{k+1} = c_k$. Otherwise, let

$$c_{k+1} = \max\{\frac{\nabla f_0(x^k)^T d_0^k + d_0^{kT} B_k d_0^k}{-F^*(x^k; d_0^k)}, 2c_k\}.$$

Step 4 : Let d_1^k be the least norm solution of the following linear equation system:

$$f_j(x^k + d_0^k) + \nabla f_j(x^k)^T d = -||d_0^k||^\tau e, j \in E(x^k) = \{j : j \in L, \widetilde{u}_j^k > 0\},$$

and if the above linear equation system is inconsistent or $||d_1^k|| > ||d_0^k||$, then let $d_1^k = 0$.

Step 5 : Let $x^{k+1} = x^k + \lambda_k d_0^k + \lambda_k^2 d_1^k$, where λ_k is the largest value of the sequence $\{1, \gamma, \ldots\}$ satisfying

$$F_{c_{k+1}}(x^k + \lambda_k d_0^k + \lambda_k^2 d_1^k) \le F_{c_{k+1}}(x^k) + \alpha\lambda_k(\nabla f_0(x^k)^T d_0^k + c_{k+1} F^*(x^k; d_0^k)).$$

Step 6 : Choose $B_{k+1} \in \sum, \sigma_{k+1} \in [\sigma_l, \sigma_r], \beta_{k+1} \in (\sigma_{k+1}, \bar{\beta}]$. Let $k = k + 1$, go to step 1.

4. Global Convergence of Algorithm

Throughout this paper, the following general assumptions are true.

H 4.1. *The functions $f_0, f_j, j \in L$ are two-times continuously differentiable.;*

H 4.2. *The sequences* $\{x^k\}$ *is bounded;*

H 4.3. *There exists* $0 \le a \le b$ *such that* $a||y||^2 \le y^T B_k y \le b||y||^2, \forall y \in R^n, k = 1, 2, \ldots$

Lemma 4.1. *[5] If MFCQ holds, suppose* $x^k \to x^*, B_k \to B^*, \sigma_k \to \sigma^*, \beta_k \to \beta^*$, *then* $d_0^k \to d^*$ *where* d_0^k *is the solution of* $Q(x^k, B_k, \sigma_k, \beta_k)$ *and* d^* *of* $Q(x^*, B^*, \sigma^*, \beta^*)$.

Now we prove that the subproblem (2.14) is feasible at each iterate point whether it is a feasible point or not.

Lemma 4.2. *Suppose that* $\{x^k\}$ *is an infinite sequence generated by algorithm. Then, when* k *is large enough, the penalty parameter* c_k *tends to infty, so any cluster point* x^* *of* $\{x^k\}$ *satisfies the constraint conditions of (1.1).*

Proof. To the contrary, x^* does not satisfy the constraint conditions of (1.1), i.e., $f_j(x^*) > 0$, then, $x^* \in G^c$. According to lemma 2.2, we know $\theta^0(x^*, \sigma) < 0$. Furehmore, because x^* is a cluster point of $\{x^k\}$, then there exists a subsequence $\{x^{k_i}\}$ such that $x^{k_i} \to x^*, i \to \infty$.

By using of lemma 4.1 we have that d^* is the solution of $E(x^*, B_*, \sigma_*, \beta_*)$ where $B_{k_i} \to B_*, \sigma_{k_i} \to \sigma_*, d_0^{k_i} \to d^*, \beta_{k_i} \to \beta_*$. From lemma 2.4 we know $F^*(x^*; d^*) \le \theta^0(x^*, \sigma_*) < 0$.

By H4.1 we obtained $F^*(x; d)$ is continuous in $R^n \times R^n$, then $F^*(x^{k_i}; d_0^{k_i}) \to F^*(x^*; d^*), i \to \infty$.

H4.1 and the way of computing $d_0^{k_i}$ imply that that $\nabla f_0(x^{k_i})^T d_0^{k_i} + d_0^{k_i T} B_{k_i} d_0^{k_i}$ is bounded. On the other hand, from updating rule of the penalty parameter we know $c_k \to \infty$, i.e., $\dfrac{\nabla f_0(x^{k_i})^T d_0^{k_i} + d_0^{k_i T} B_{k_i} d_0^{k_i}}{-F^*(x^{k_i}; d_0^{k_i})} \to \infty$. Then, $F^*(x^{k_i}; d_0^{k_i}) \to 0$.

We obtain a contradiction. The proof is finished. ∎

Form the above statement and lammas, it can be seen if MFCQ holds at any point c_k is a constant when k is sufficiently large. Without loss of generality, we assume $c_k = c > 0, \forall k$ in the sequel analysis.

Then, we prove that the algorithm is well defined. That is to say there exists a step-size t_k satisfied the inequality of the Armijo linear search.

Lemma 4.3. *If* x^k *is not a* $K - T$ *point of (1.1), then there exists a* $\lambda_k > 0$ *satisfying the following inequality:*

$$F_c(x^k + \lambda_k d_0^k + \lambda_k^2 d_1^k) \le F_c(x^k) + \alpha\lambda_k(\nabla f_0(x^k) d_0^k + cF^*(x^k; d_0^k))$$

Lemma 4.4. $\lim\limits_{k\to\infty} ||x^{k+1} - x^k|| = 0; \lim\limits_{k\to\infty} ||\lambda_k d_0^k|| = 0.$

Proof. In view of $F_c'(x^k, d_k) < 0$, sequence $\{F_c(x^k)\}$ is a non-increasing sequence. Therefore, from step 3 and step 6, we have

$$F_c(x^{k+1}) \le F_c(x^k) + \alpha\lambda_k(\nabla f_0(x^k) d_0^k + cF^*(x^k; d_0^k)) \le F_c(x^k) - \alpha\lambda_k d_0^{kT} B_k d_0^k. \tag{4.1}$$

From (4.1) and H4.1, we get

$$\lim_{k\to\infty} ||\lambda_k d_0^k|| = 0.$$

Since $||d_1^k|| < ||d_0^k||$, then

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

∎

Lemma 4.5. $\lim_{k\to\infty} d_0^k = 0$.

Proof. To the contrary, there must exist a subsequence $\{d_0^{k_i}\}$ of $\{d_0^k\}$ and an $\epsilon > 0$ such that

$$||d_0^{k_i}|| \geq \epsilon, i \to \infty. \tag{4.2}$$

From lemma 4.4 we know that $\lambda_k d_0^k \to 0, k \to \infty$. Now, we prove that there exists $\lambda^{'} > 0$ such that

$$\lambda_i \geq \lambda^{'}, \forall i. \tag{4.3}$$

Similarly, we prove (4.3) holds to the contrary. That is to say there exists a subsequence of $\{\lambda_i\}$ (without loss of generality, we can suppose that the subsequence is $\{\lambda_i\}$ itself) such that $\lambda_i \to 0, i \to \infty$.

Lemma 4.3 implies that $F_c(x^i+\lambda_i d_0^{k_i}+\lambda_i^2 d_1^{k_i}) \leq F_c(x^i)+\lambda_i(\nabla f_0(x^i)d_0^{k_i}+cF^*(x^i;d_0^{k_i}))+o(\lambda_i)$.
From step 5 of algorithm, we have that

$$F_c(x^i + \frac{\lambda_i}{\gamma}d_0^{k_i} + \frac{{\lambda_i}^2}{\gamma^2}d_1^{k_i}) \geq F_c(x^i) + \alpha\frac{\lambda_i}{\gamma}(\nabla f_0(x^i)^T d_0^{k_i} + cF^*(x^i;d_0^{k_i})).$$

We know that for i sufficiently that

$$\begin{aligned}
&\frac{\lambda_i}{\gamma}(\nabla f_0(x^i)d_0^{k_i} + cF^*(x^i;d_0^{k_i})) + o(\frac{\lambda_i}{\gamma}) \\
&\quad \geq F_c(x^i + \tfrac{\lambda_i}{\gamma}d_0^{k_i} + \tfrac{{\lambda_i}^2}{\gamma^2}d_1^{k_i}) - F_c(x^i) \\
&\quad \geq \alpha\tfrac{\lambda_i}{\gamma}(\nabla f_0(x^i)^T d_0^{k_i} + cF^*(x^i;d_0^{k_i})).
\end{aligned}$$

This shows that $(1-\alpha)\frac{\lambda_i}{\gamma}(\nabla f_0(x^i)d_0^{k_i} + cF^*(x^i;d_0^{k_i})) + o(\frac{\lambda_i}{\gamma}) \geq 0$.
According to the choice of c, (4.2) and H4.3 we get that

$$-(1-\alpha)b_1\epsilon^2 + \frac{o(\frac{\lambda_i}{\gamma})}{\frac{\lambda_i}{\gamma}} \geq 0.$$

Note that $\lambda_i \to 0, i \to \infty$, then $-(1-\alpha)b_1\epsilon^2 \geq 0$. This contradicts $0 < \alpha < \frac{1}{2}$. So (4.3) holds. However (4.2) and (4.3) imply that $\lambda_i d_0^{k_i} \not\to 0$. We obtain a contradiction. The proof is finished. ∎

Combining the above lemmas, we obtain the global convergence of the algorithm.

Theorem 4.6. *If MFCQ holds at any $x \in R^n$, the algorithm either stops at the KKT point of the problem(1.1) in finite iteration, or generates an infinite sequence $\{x^k\}$ any accumulation point x^* of which is a KKT point of the problem(1.1).*

5. The rate of Convergence

Now we discuss the convergent rate of the algorithm, for this purpose, we must make further assumption.

H 5.1. *The second-order sufficiency conditions with strict complementary are satisfied at the $K-T$ point pair (x^*, μ^*) of the problem (1.1).*

H 5.2. *$\forall x \in R^n$, the vectors $\{\nabla f_j(x) : j \in I(x)\}$ are linearly independent, where $I(x) = \{j \in L | f_j(x) = 0\}$.*

H 5.3. *Let $\{B_k\}$ satisfy*

$$||P_k(B_k - \nabla^2_{xx}\widetilde{L}(x^k, \widetilde{u}^k))d_0^k|| = o|||d_0^k|| \Longleftrightarrow ||P_k(B_k - \nabla^2_{xx}\widetilde{L}(x^*, \mu^*))d_0^k|| = o(||d_0^k||)$$

where

$$P_k = E_n - A_k(A_k^T A_k)^{-1}A_k^T, \qquad A_k = (\nabla f_j(x^k), j \in I^* \triangleq I(x^*)),$$

$$\nabla^2_{xx}\widetilde{L}(x^k, \widetilde{u}^k) = \nabla^2 f_0(x^k) + \sum_{j \in I^*} \widetilde{u}_j^k \nabla^2 f_j(x^k), \nabla^2_{xx}L(x^*, \mu^*) = \nabla^2 f_0(x^*) + \sum_{j \in L} \mu_j^* \nabla^2 f_j(x^*).$$

From the all above assumptions and lemma 4.5, it is easy to see that the entire sequence $\{x^k\}$ converges to x^*, i.e., $x^k \to x^*$, $k \to \infty$. In addition, for k large enough, $\psi^0(x^k, \sigma_k) = 0$ and $||d_0^k|| \to 0$, i.e., the constraint condition $||d||_\infty \le \beta_k$ in $Q(x^k, B_k, \sigma_k, \beta_k)$ is redundant. Hence the subproblem $Q(x^k, B_k, \sigma_k, \beta_k)$ is equivalent to the following quadratic programming subproblem

$$\begin{array}{ll} \min & \nabla f_0(x)^T d + \frac{1}{2} d^T B_k d \\ \text{s.t.} & f_j(x^k) + \nabla f_j(x^k)^T d \le 0, j \in L. \end{array} \tag{5.1}$$

For k large enough, then $\lambda_k \to \mu^*$, $\widetilde{U}^k \to \mu^*, k \to \infty$, where μ^* is a $K-T$ multiplier of problem (1.1) and λ_k is of problem (5.1).

Lemma 5.1. *For k is large enough, $||d_1^k|| = O(||d_0^k||^2)$.*

Proof. It is clear that

$$f_j(x^k + d_0^k) = f_j(x^k) + \nabla f_j(x^k)^T d_0^k + O(||d_0^k||^2) = O(||d_0^k||^2).$$

According to the equation system of step 4 in the algorithm, we get

$$f_j(x^k + d_0^k) = O(||d_0^k||^2) = -||d_0^k||^\tau e - \nabla f_j(x^k)^T d_1^k.$$

In view of $\tau \in (2, 3)$, then

$$O(||d_0^k||^2) + ||d_0^k||^\tau e = -\nabla f_j(x^k)^T d_1^k$$

$$O(||d_0^k||^2) = -\nabla f_j(x^k)^T d_1^k$$

$$||d_1^k|| = O(||d_0^k||^2)$$

∎

Now, we prove the superlinear convergence of the algorithm.

Lemma 5.2. *For k large enough, $\lambda_k \equiv 1$.*

Proof. Now we need to prove the following inequality holds for $\lambda_k = 1$.

$$b_k \triangleq F_c(x^k + d_0^k + d_1^k) - F_c(x^k) - \alpha(\nabla f_0(x^k)^T d_0^k + cF^*(x^k; d_0^k)) \le 0.$$

According to the definition of $F_c(x)$, $K-T$ condition and the compute ways of d_1^k, then the above inequality can be expanded to the following:

$$\begin{aligned} b_k &\triangleq \nabla f_0(x^k)^T d_0^k + cF^*(x^k; d_0^k) + \nabla f_j(x^k)^T d_1^k + \tfrac{1}{2}(d_0^k)^T \nabla^2 f_0(x^k) d_0^k + o(||d_0^k||^2) \\ &\leq (\alpha - \tfrac{1}{2}) b ||d_0^k||^2 + \tfrac{1}{2}(d_0^k)^T(\nabla_{xx}^2 \widetilde{L}(x^k, \widetilde{U}^k) - B_k) d_0^k + o(||d_0^k||^2). \end{aligned}$$

Denote $P_* = E_n - A_*(A_*^T A_*)^{-1} A_*^T, A_* = (f_j(x^*), j \in I_*)$, then $P_k \to P_*$. Let

$$d_0^k = P_* d_0^k + y^k, y^k = A_*(A_*^T A_*)^{-1} A_*^T d_0^k = o(||d_0^k||) - A_*(A_*^T A_*)^{-1} f_j(x^k).$$

It is easy to know that

$$||y^k|| = o(||d_0^k||) + O(z_k),$$

where $z_k = (\sum\limits_{j \in I_*} g_j^2(x^k))^{\frac{1}{2}}$. So

$$\begin{aligned} b_k &\leq b(\alpha - 1)||d_0^k||^2 + \tfrac{1}{2}((d_0^k)^T P_* + (y^k)^T)(\nabla_{xx}^2 \widetilde{L}(x^k, \widetilde{U}^k) - B_k) d_0^k + o(||d_0^k||^2) \\ &= b(\tfrac{1}{2} - \alpha)||d_0^k||^2 + o(||d_0^k||^2) + o(z^k) \\ &\leq 0. \end{aligned}$$

i.e., for k large enough, (5.2)holds. ∎

Moreover, in view of lemma 5.2 and the way of Theorem 5.2 in [**?**], we may obtain the following theorem.

Theorem 5.3. *Under all above-mentioned assumptions, the algorithm is superlinearly convergent, i.e., the sequence* $\{x^k\}$ *generated by the algorithm satisfies* $||x^{k+1} - x^*|| = o(||x^k - x^*||)$.

6. Numerical experiments

In this section, we carry out numerical experiments based on the Algorithm in section 3. The code of the proposed algorithm is written by using MATLAB 7.0 and utilized the optimization toolbox. The results show that the algorithm is very effective. During the numerical experiments, it is chosen at random some parameters as follows: $\delta = 1, \theta_l = 1, \theta_r = 2, \bar{\beta} = 3, \alpha = 0.25, \gamma = 0.5, \beta_k = 10, \sigma_k = 10, c_k = 100, \tau = 2.5, B_0 = I \in R^{n \times n}$. B_k is updated by the BFGS formula [8]. In the implementation, the stopping criterion of Step 2 is changed to $If\ \|d_0^k\| \leq 10^{-8} \quad STOP.$

This algorithm has been tested on some problems from Ref.[9], a feasible initial point is either provided or obtained easily for each problem. The results are summarized in Table 1. The columns of this table has the following meanings:
No.: the number of the test problem in [9];
NT: the number of iterations;

Table 1 Numerical results of Algorithm A

NO.	x_0	NT	x^*	$f(x^*)$
HS05	(2, 2)	4	(0.811262176188244, 1.22099772824924)	1.46193760965203
HS32	(0.1, 0.7, 0.2)	20	(0.00000000000829, 0.00000000000016, 0.99999999999999)	1.00000005665777
HS43	(0.5, 0.2, 0.5, 0.3)	10	(-0.97012709077022, 2.27467923493263, -3.25275103455275, 4.08518466555261)	-7.18253130771858
HS113	(2, 3, 5, 5, 1, 2, 7, 3, 6, 10)	50	(2.00000000001807, 0.999999999983118, 8.999999999961392, 5.000000000007303, 1.00000000012861, 0.99999999997318, 0.99999999999496, 11.00000000003116, 9.00000000002467, 8.00000000000379)	24.309999993831

References

[1] N.Maratos. Exact penalty functions for finite dimensional and control optimization problems[D], Ph.D. thesis, Univ. of Science and Technology, London (1978).

[2] ,P. Boggs, J. Tolle, P. Wang,P *On the Local Convergence of Quasinewton Methods for Constrained Optimization*. SIAM J.Optim,1982 54: 161-171.

[3] J.V Burke, S. Han. *A Robust SQP Method*. Mathematical Programming, 1989,43: 277-303.

[4] G. Zhou. *A Modified SQP Method and Its Global Convergence*. Journal of Optimization,1997, 11:193-205.

[5] J. Zhang, X. Zhang. *A Modified SQP Method with Nonmonotone Linesearch Technique*. Journal of Optimization,2001, 21:201-218.

[6] Z.Zhu, S. Wang. *A superlinearly convergent numerical algorithm for nonlinear programming*. Nonlinear Analysis Real World Applications, 2012, 13(5):2391-2402.

[7] Z. Luo, Z. Zhu, G. Chen et al.*A superlinearly convergent SQP algorithm for constrained optimization problems*. Journal of Computational Information Systems, 2013, 9(11):4443-4450.

[8] M. J. D. Powell. A Fast Algorithm for Nonlinearly Constrained Optimization Calculations. In:Waston, G.A.(ed). Numerical Analysis. Springer,Berlin 1978:144-157.

[9] W. Hock, K. Schittkowski. Test examples for nonlinear programming codes, Lecture Notes in Economics and Mathematical Systems, vol. 187,Springer, Berlin, 1981.