AMO - Advanced Modeling and Optimization, Volume 17, Number 2, 2015

A modified sequential quadratic programming method for nonlinear programming[1]

Zhibo Luo, Zhijun Luo and Lifang Deng

The Department of Mathematics and Econometrics,

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

Abstract. In this paper, based on the idea of generalized gradient projection method, a modified sequential quadratic programming method is presented to solve nonlinear programming. Under some suitable assumptions, the algorithm is proven to be globally and superlinearly convergent. The numerical results show that the method in this paper is effective.

Key words. Nonlinear programming, Sequential quadratic programming algorithm, Quadratic programming, Global convergence, Superlinear convergence rate

1. Introduction

In this paper, consider the following nonlinear optimization problem.

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_j(x) \leq 0, j \in L = \{1, 2, \cdots, m\}. \end{array} \tag{1.1}$$

Where, $f(x)$, $g_j(x)(j \in L)$ are continuously differentiable functions. Denote $\chi = \{g_j(x) \leq 0,\ j \in L\}$, $I(x) = \{j \in I \mid g_j(x) = 0\}$.

Since sequence quadratic programming(SQP) algorithm birth on 1970s, it has always been one of very active research topics in the society of optimization and has been one of the most efficient algorithms for solving constrained optimization problems [1]-[5]. However, the traditional SQP algorithms make it necessary to solve relatively complex and highly computational cost QP problems per single iteration, or let the Hessian matrix of the quadratic programming subproblem be positive definite or uniformly positive definite [6]-[8]. In order to simplify the structure of the algorithm, weaken hypothesis conditions, reduce the computational cost, and quicken the convergence rate, a lot of authors present many different types algorithms. For example [9] proposed a new SQP method for solving inequality constrained optimization, which is not necessary that the Hessian matrix is positive definite.

In this paper, combining the generalized gradient projection technique [10] with the ideas of the sequential quadratic programming method, we proposed a modified SQP method for the solution of problem (1.1). At single iteration, it is only necessary to solve one QP subproblem with equality constraints. Thus, the computational cost is reduced. Under some suitable assumptions, we prove that the algorithm is global convergence as well as superlinear convergence. Finally, some numerical results are reported to show the effectiveness of the proposed algorithm.

[1]Corresponding author:Zhijun Luo. E-mail addresses: ldlzj123@163.com. This research was supported by Scientific Research Fund of Hunan Provincial Education Department (NO. 14C0609)and the fund of the educational reform research fund of Hunan University of Humanities, Science and Technology (NO. RKJGY1526 and RKJGY1436).

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

The plan of the paper is as following : In section 2, the algorithm is proposed. In section 3, we show that the algorithm is globally convergent. While the superlinear convergence rate is analyzed in section 4. Finally, in Section 5, some numerical experiments are implemented.

2. Description of Algorithm

It is well known that standard SQP method for (1.1) generates a decent direction at the point $x \in \chi$ by solving the quadratic programming subproblem

$$\begin{array}{ll} \min & \nabla f(x)^T d + \frac{1}{2} d^T H d \\ \text{s.t.} & g_j(x) + \nabla g_j(x)^T d \leq 0, j \in L \end{array} \tag{2.1}$$

where H is the approximate Hessian matrix of Lagrangian function associated with (1.1). However, the direction d is not a feasible direction, and can not avoid the Maratos effect. So, combining with generalized gradient projection method,we define some variants as follows:

$$\begin{array}{c} N(x) = \left(\nabla g_j(x), j \in L\right), D(x) = \text{diag}(D_j(x), j \in I) = \text{diag}(g_j^2(x), j \in L), \\ B(x) = \left(N(x)^T N(x) + D(x)\right)^{-1} N(x)^T, \ \nu(x) = -B(x)\nabla f(x). \end{array} \tag{2.2}$$

For an appropriate small parameter $\sigma > 0$, define

$$L(x) = \left\{ j \in L \mid -\sigma|\nu_j(x)| \leq g_j(x) \leq 0 \right\}. \tag{2.3}$$

For (1.1), we consider the following equality constrained quadratic programming

$$\begin{array}{ll} \min & \nabla f(x)^T d + \frac{1}{2} d^T H d \\ \text{s.t.} & g_j(x) + \nabla g_j(x)^T d = -\min\{0, \nu_j(x)\}, j \in L(x). \end{array} \tag{2.4}$$

Now, the algorithm for the solution of the problem (1.1) can be stated as follows.

Algorithm: Given a starting point $x^0 \in \chi$, and an initial symmetric matrix $H^0 \in^{n\times n}$. Choose parameters $\xi, \upsilon \in (0,1), \alpha \in (0, \frac{1}{2}), \tau \in (2,3), \delta > 2$. Set $k = 0$;

Step 1 : From (2.2) and (2.3), compute $N_k = N(x^k),\ D_k = D(x^k),\ B_k = B(x^k)$, $nu_k = \nu(x^k),\ L_k = L(x^k)$;

Step 2 : Obtain $\left(d_0^k, \widetilde{u}^k\right)$ by solving the problem (2.4) at x^k, if $d_0^k = 0$, STOP;

Step 3 : Compute

$$d_1^k = -B_k^T \left(||d_0^k||^\tau e + G(x^k + d_0^k)\right), d^k = d_0^k + d_1^k, \tag{2.5}$$

where $e = (1, \ldots, 1)^T \in R^{|L|}$,

$$G(x^k + d_0^k) = \left(G_j(x^k + d_0^k), j \in L\right), G_j(x^k + d_0^k) = \begin{cases} g_j(x^k + d_0^k), & j \in L_k; \\ 0, & j \in L \setminus L_k. \end{cases}$$

If

$$\nabla f(x^k)^T d_0^k \leq \min\left\{-\xi||d_0^k||^\delta, -\xi||d^k||^\delta\right\}, \tag{2.6}$$

$$||H_k d_0^k|| \leq \xi||d_0^k||^{\frac{1}{2}}, \min\left\{\widetilde{u}_j^k, j \in L_k\right\} \geq -\xi||d_0^k||, \tag{2.7}$$

$$f(x^k + d^k) \leq f(x^k) + \alpha\nabla f(x^k)^T d_0^k, \tag{2.8}$$

$$g_j(x^k + d^k) \leq 0, \ \ j \in L. \tag{2.9}$$

Let $\lambda_k = 1$;

Step 4 : Let H_{k+1} be a new symmetric approximation of the Hessian matrix. Set $x^{k+1} = x^k + \lambda_k d^k$ and $k = k + 1$; Go back to Step 2.

3. Global Convergence of Algorithm

In this section, we firstly prove the global convergence of algorithm. Under mild conditions, we will discuss the strong convergence of the proposed Algorithm.

Throughout this paper, the following assumption holds.

H 3.1. *The feasible sets are nonempty, i.e.,* $\chi \neq \phi$ *; the functions* f *and* $g_j(x)(j \in L)$ *are two-times continuously differentiable;* $\forall x \in \chi$*, the vectors* $\{\nabla g_j(x), j \in I(x)\}$ *are linearly independent.*

According to Lemma 2.2 in [4], we can obtain the following result.

Lemma 3.1. *Suppose that H3.1 holds, then* $\forall x \in \chi$*, the matrix* $(N(x)^T N(x) + D(x))$ *is positive definite.*

Lemma 3.2. *Suppose that* $(d_0(x), \widetilde{u}(x))$ *is a KKT point pair of (2.4), if* $d_0(x) = 0$*, then, we obtain that* x *is the KKT point of the problem (1.1).*

Proof. Let

$$u(x) = (u_j(x), j \in L), u_j(x) = \begin{cases} \widetilde{u}_j(x), & j \in L(x), \\ 0, & j \in L \setminus L(x). \end{cases}$$

If $d_0(x) = 0$, then, from (2.4), we have

$$\begin{aligned} &\nabla f(x) + \nabla g_{L(x)}(x)\widetilde{u}(x) = 0, \\ &g_j(x) + \min\{0, \nu_j(x)\} = 0, j \in L(x). \end{aligned} \tag{3.1}$$

Where $\nabla g_{L(x)}(x) = \left(\nabla g_j(x),\ j \in L(x)\right)$. Thus $g_j(x) = 0,\ \nu_j(x) \geq 0,\ j \in L(x)$, $D_j(x)\hat{u}_j(x) = 0$, $j \in L$, and we have

$$\left(N(x)^T N(x) + D(x)\right) \hat{u}(x) = N(x)^T \nabla g_{L(x)}(x)\widetilde{u}(x) = -N(x)^T \nabla f(x),$$

i.e.

$$\hat{u}(x) = -\left(N(x)^T N(x) + D(x)\right)^{-1} N(x)^T \nabla f(x) = \nu(x).$$

So, from (3.1), we obtain

$$\begin{aligned} &\nabla f(x) + \nabla g_{L(x)}(x)\widetilde{u}(x) = 0, \\ &g_j(x) = 0, \widetilde{u}_j(x) \geq 0, j \in L(x). \end{aligned}$$

Which shows that x is a KKT point of the problem (1.1). ∎

Theorem 3.3. *The algorithm either stops at a KKT point* x^k *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).*

Proof. The first statement is obvious. Thus, assume that the algorithm generates an infinite sequence $\{x^k\}$, x^* is an accumulation point. Because $J_k \subseteq L$ is a finite set, there exists an infinite index set K, such that

$$x^k \to x^*, J_k \equiv J,\ k \in K.$$

If there exists $K_1 \subseteq K$, such that for all $k \in K_1$, in view of (2.6), (2.8), we obtain that $\{f(x^k)\}$ is monotone decreasing. Thus, from $\{x^k\}_{k\in K} \to x^*$, together with H 3.1 we can get

$$f(x^k) \to f(x^*), k \to \infty. \tag{3.2}$$

From (2.6) and (2.8), it is easy to see that

$$0 = \lim_{k\in K}(f(x^{k+1}) - f(x^k)) \leq \lim_{k\in K} \alpha\nabla f(x^k)^T d_0^k \leq \lim_{k\in K}(-\alpha\xi||d_0^k||^\delta) \leq 0, \tag{3.3}$$

Then, we have $d_0^k \to 0,\ k \in K$ and $H_k d_0^k \to 0,\ k \in K$. From (2.4), we get

$$\begin{aligned}&\nabla f(x^k) + H_k d_0^k + \nabla g_J(x^k)\widetilde{u}^k = 0,\\ &g_j(x^k) + \min\{0, \nu_j^k\} + \nabla g_j(x^k)^T d_0^k = 0, j \in J.\end{aligned} \tag{3.4}$$

So, $g_j(x^*) = 0,\ j \in J$, and $J \subseteq I(x^*)$hold. Thus, from H3.1, we obtain that the matrix $\big(\nabla g_J(x^*)^T\nabla g_J(x^*)\big)$ is nonsingular at x^*, so, for $k \in K$ large enough, the matrix $\big(\nabla g_J(x^k)^T\nabla g_J(x^k)\big)$ is nonsingular, and we have

$$\big(\nabla g_J(x^k)^T\nabla g_J(x^k)\big)^{-1} \to \big(\nabla g_J(x^*)^T\nabla g_J(x^*)\big)^{-1}, k \in K, k \to \infty.$$

Then from (3.4) it can be seen that

$$\begin{aligned}\widetilde{u}^k &= -\big(\nabla g_J(x^k)^T\nabla g_J(x^k)\big)^{-1}\nabla g_J(x^k)^T(\nabla f(x^k) + H_k d_0^k)\\ &\to -\big(\nabla g_J(x^*)^T\nabla g_J(x^*)\big)^{-1}\nabla g_J(x^*)^T\nabla f(x^*) \triangleq \lambda^*, k \in K,\end{aligned} \tag{3.5}$$

So, from (2.7) and (3.4), it is easy to get

$$\begin{aligned}&\nabla f(x^*) + \nabla g_J(x^*)\lambda^* = 0,\\ &g_j(x^*) = 0, \lambda_j^* \geq 0, j \in J.\end{aligned} \tag{3.6}$$

Which implies that x^* is a KKT point of (1.1). ∎

To obtain the strong convergence of Algorithm, the following assumption is necessary.

H 3.2. *Suppose that the sequence $\{x^k\}$ of points generated by Algorithm is bounded, and has a limit point x^*. The second-order sufficiency conditions with strict complementary slackness are satisfied at the KKT point x^* and corresponding multipliers u^* of the problem (1.1).*

According to H3.2 and Proposition 4.1 in article [3], we can obtain the following conclusion.

Lemma 3.4. *Let H3.1∼ H3.2 holds, $\lim_{k\to\infty}||x^{k+1}-x^k|| = 0$. Thereby, the entire sequence$\{x^k\}$ converges to x^*, i.e. $x^k \to x^*, k \to \infty$.*

4. Superlinear Convergence of Algorithm

In order to obtain superlinear convergence, we make another assumption.

H 4.1. *$H_k \to H_*, k \to \infty$, and H_* is positive definite on the subspace $Y(x^*)$. Where $Y(x^*) = \big\{d \in R^n \mid \nabla g_j(x^*)^T d = 0, j \in I(x^*)\big\}$.*

Lemma 4.1. *Suppose that H3.1 $\sim$ H3.2 holds, for k large enough, then the solution of (2.4) is unique, and*

$$J_k \equiv I(x^*) \overset{\triangle}{=} I_*, \lim_{k\to\infty} d_0^k = 0, \lim_{k\to\infty} \pi^k = u^*, \lim_{k\to\infty} \widetilde{u}^k = (u_j^*, j \in I_*).$$

Lemma 4.2. *For k large enough,*

1) d_1^k and d^k obtained in step 3 satisfy:

$$||d^k|| \sim ||d_0^k||, \; ||d_1^k|| = O\left(||d_0^k||^2\right). \tag{4.1}$$

2) there exists constants $b, \eta > 0$, such that

$$\sum_{j\in I_*} \widetilde{u}_j^k g_j(x^k) \le -\eta z_k, z_k = \left(\sum_{j\in I_*} g_j^2(x^k)\right)^{\frac{1}{2}}, \tag{4.2}$$

$$-(d_0^k)^T H_k d_0^k \le -b||d_0^k||^2 + o(z_k), \tag{4.3}$$

$$\nabla f(x^*)^T d_0^k \le -b||d_0^k||^2. \tag{4.4}$$

proof. 1). In view of Lemma 4.1 and H3.2, we know that $\pi_j^k > 0, \widetilde{u}_j^k > 0, j \in I_*$. Then from (2.4) we have

$$g_j(x^k) + \nabla g_j(x^k)^T d_0^k = 0, \; j \in I_*. \tag{4.5}$$

So

$$g_j(x^k + d_0^k) = g_j(x^k) + \nabla g_j(x^k)^T d_0^k + O\left(||d_0^k||^2\right) = O\left(||d_0^k||^2\right), \; j \in I_*,$$

i.e.

$$||G(x^k + d_0^k)|| = O\left(||d_0^k||^2\right).$$

Considering $B_k \to B_*$, $\tau \in (2,3)$, we obtain $||d_1^k|| = O\left(||d_0^k||^2\right), \; ||d^k|| \sim ||d_0^k||$.

2). According to $\widetilde{u}_j^k > 0$, $j \in I_*$, there exists a constant $\eta > 0$, such that

$$\sum_{j\in I_*} \widetilde{u}_j^k g_j(x^k) = -\sum_{j\in I_*} \widetilde{u}_j^k \left|g_j(x^k)\right| \le -\sum_{j\in I_*} \eta \left|g_j(x^k)\right| \le -\eta z_k.$$

Denote

$$P_* = I_n - A_*(A_*^T A_*)^{-1} A_*^T, P_k = I_n - A_k(A_k^T A_k)^{-1} A_k^T, \tag{4.6}$$

where $A_* = \nabla g_{I_*}(x^*), A_k = \nabla g_{I_*}(x^k)$. Let

$$d_0^k = P_* d_0^k + y_k, y_k = A_*(A_*^T A_*)^{-1} A_*^T d_0^k, \tag{4.7}$$

and $G(x^k) = \left(g_j(x^k), j \in I_*\right)$, then by (4.5), it can be seen that

$$y_k = A_*(A_*^T A_*)^{-1}(A_* - A_k)^T d_0^k - A_*(A_*^T A_*)^{-1} G(x^k),$$

so,

$$||y_k|| = O\left(||d_0^k||\right) = o\left(||d_0^k||\right) + O(z_k).$$

Hence,

$$\begin{aligned} -(d_0^k)^T H_k d_0^k &= -\left(P_* d_0^k + y_k\right)^T H_k \left(P_* d_0^k + y_k\right) \\ &= -\left(P_* d_0^k\right)^T H_* \left(P_* d_0^k\right) + \left(P_* d_0^k\right)\left(H_* - H_k\right)\left(P_* d_0^k\right) + O\left(||d_0^k|| \cdot ||y_k||\right) \\ &= -\left(P_* d_0^k\right)^T H_* \left(P_* d_0^k\right) + o\left(||d_0^k||^2\right) + o(z_k). \end{aligned}$$

We know that $d_0^k \to 0, P_* d_0^k \in Y(x^*)$, so, there exists a constant $b_1 > 0$, such that

$$\begin{aligned} -(d_0^k)^T H_k d_0^k &\leq -b_1 ||P_* d_0^k||^2 + o\left(||d_0^k||^2\right) + o(z_k) = -b_1 ||d_0^k - y_k||^2 + o\left(||d_0^k||^2\right) + o(z_k) \\ &= -b_1 ||d_0^k||^2 + o\left(||d_0^k||^2\right) + o(z_k) \leq -b||d_0^k||^2 + o(z_k). \end{aligned}$$

In addition, from (3.4) and (4.5), we have

$$\begin{aligned} \nabla f(x^k)^T d_0^k &= -(d_0^k)^T H_k d_0^k - (A_k^T d_0^k)^T \widetilde{u}^k = -(d_0^k)^T H_k d_0^k + \sum_{j \in I_*} \widetilde{u}_j^k g_j(x^k) \\ &\leq -b||d_0^k||^2 + o(z_k) - \eta z_k \leq -b||d_0^k||^2. \end{aligned} \tag{4.8}$$

∎

To ensure the step size unit can be accepted, the following assumption about the symmetric matrix satisfied:

H 4.2. *Let*

$$\left\| P_k \left(H_k - \nabla_{xx}^2 \widetilde{L}(x^k, \widetilde{u}^k) \right) d_0^k \right\| = o\left(||d_0^k||\right),$$

which, obviously is equivalent to

$$\left\| P_k \left(H_k - \nabla_{xx}^2 L(x^*, u^*) \right) d_0^k \right\| = o\left(||d_0^k||\right),$$

where

$$\nabla_{xx}^2 \widetilde{L}(x^k, \widetilde{u}^k) = \nabla^2 f(x^k) + \sum_{j \in I_*} \widetilde{u}_j^k \nabla^2 g_j(x^k), \nabla_{xx}^2 L(x^*, u^*) = \nabla^2 f(x^*) + \sum_{j \in I} u_j^* \nabla^2 g_j(x^*).$$

Lemma 4.3. *For k large enough, the inequalities of Step 3 are satisfied. i.e.* $x^{k+1} = x^k + d^k, \lambda_k \equiv 1$.

proof. Firstly, in view of lemma 4.1, lemma 4.2 and H3.2, it follows that (2.7) holds. Considering $\delta > 2$, (4.1) and (4.4), so (2.6) holds. Thus, in order to finish the proof of this lemma, we only need to prove that (2.8) and (2.9) are true. According to Lemma 4.1 and H3.2, for k large enough, the perturbation term of the right side of quadratic programming (2.4) is disappeared. Then, (2.4) becomes the following quadratic programming.

$$QP_k \quad \begin{array}{ll} \min & \nabla F_c(x^k)^T d + \frac{1}{2} d^T H_k d \\ \text{s.t.} & g_j(x^k) + \nabla g_j(x^k)^T d = 0, j \in E_*. \end{array}$$

So, from Lemma 4.2, imitating Theorem 4.2 in [4], it is easy to prove the conclusion holds. ∎

Moreover, in view of Lemma 4.3 and the way of Theorem 5.2 in [11], we obtain the following theorem:

Theorem 4.4. *Under all above-mentioned assumptions, the algorithm is superlinearly convergent, i.e., the sequence* $\{x^k\}$ *generated by the algorithm satisfied* $||x^{k+1} - x^*|| = o(||x^k - x^*||)$.

5. 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: $\xi = 0.001, \upsilon = 0.01, \delta = 2.5, \alpha = 0.25, \tau = 2.25, H_1 = I.H_k$ is updated by the BFGS formula [12]. In the implementation, the stopping criterion of Step 1 is changed to $If\ \|d_0^k\| \le 10^{-8} \quad STOP$.

The results are summarized in Table 1. The columns of this table has the following meanings:
Prob: the number of the test problem in [8];
NT: the number of iterations;
FV: the final value of the objective function.

Table 1: The Information of Numerical Experiments

Prob.	X_0,	NT,	X_k,	FV
EX2 [8]	$(5,5,5,5)^T$,	19,	$(1.225163018, 1.225163018, 1.225163018, 1.225163018)^T$,	6.00409788
	$(7,8,9,10)^T$,	22,	$(1.000341406, 1.143247322, 1.286153237, 1.429059152)^T$,	6.004097581
EX4 [8]	$(5,5,1)^T$,	15,	$(0.00045033, 0.00045033, 1.99574882)^T$,	-1.99708530
	$(10,20,0)^T$,	17,	$(0.000469241, 0.000469241, 1.997377941)^T$,	-1.996949359
	$(3,18,2)^T$,	20,	$(0.000464581, 0.000464581, 1.997405157)^T$,	-1.996982945

References

[1] R. B. Wilson, A simplical algorithm for concave programming Ph D Thesis, Graduate School of Business Administration , Harvard University ,(1963).

[2] S. P. Han, A globally convergent method for nonlinear programming, *Journal of Optimization Theory and Applications*, (1977), 297 - 309.

[3] E. R. Panier, A. L. Tits, A superlinearly convergent feasible method for the solution of inequality constrained optimization problems, *SIAM J. Control Optimization*, **4**(1987), 934-950.

[4] J. B. Jian, C. M. Tang, Q. J. Hu, H. Y. Zheng, A feasible descent SQP algorithm for general constrained optimization without strict complementarity,*Journal of Computational and Applied Mathematics*, **180** (2005),391-412.

[5] Z.B.Zhu, S. Wang. A superlinearly convergent numerical algorithm for nonlinear programming, *Nonlinear Analysis*,**13**(2012),239-2402.

[6] M. Fukushima, A Successive Quadratic Programming Algorithm with Global and Superlinear Convergence Properties, *Mathematical Programming*, **35** (1986),253-264.

[7] P. Spellucci, An SQP Method for General Nonlinear Programs Using Only Equality Constrained Subproblems, *Mathematical Programming*, **82** (1998),413-448.

[8] G.L.Zhou. A modified SQP method and its global convergence. *Jouunal of Global optimization*, **11**(1997), 193-205.

[9] Z. B. Zhu, A new SQP algorithm for inequality constained optimization, *Mathmematica numerica*, **11** (2004), 413-426.

[10] J. B. Jan, Algorithm of sequential systems of linear equations with superlinear and quadratical convergence for general constrained optimization, *Journal of Engineering Mathematics*, **8** (2003), 24-30.

[11] F. Facchinei, S. Lucidi, Quadraticly and Superlinearly Convergent for the Solution of Inequality Constrained Optimization Problem,*JOTA.*, **85** (1995), 265-289. tg

[12] Powell M J D. A fast algorithm for nonlinearly constrained optimization calculations[M]. Numerical analysis. Springer Berlin Heidelberg, 1978, 144-157.