Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 8 Issue 1, January - 2021

A Modified Algorithm of Successive Quadratic Programming Type

Zilong Zang
School of Mathematics, Lanzhou City University, Lanzhou, 730070, P.R. China
1786915586@qq.com

***Abstract*—This paper presents an algorithm of successive quadratic programming type for programming problems with nonlinear equality and inequality constraints. Under some suitable conditions, we prove the global and superlinear convergence properties.**

Keywords—General nonlinear programming; successive quadratic programming; exact penalty functions; rate of convergence.

I. INTRODUCTION

Successive quadratic programming is currently considered to be one of the most promising approaches for solving constrained nonlinear optimization problems [1,9,10,12-14]. An attractive feature of the method is that it possesses a fast local convergence property, provided that the data of quadratic programming subproblems are suitably chosen. In order to obtain global convergence, the method is often designed by making use of an exact penalty function . It has been observed, however, that the use of a nonsmooth exact penalty function may deteriorate the desirable local convergence property. This unfavorable phenomenon, comm0nly called the Maratos effect, has recently drawn much attention, and some remedies have been proposed to retain a rapid rate of convergence.

In the next section we describe how the modified quadratic programming subproblems are obtained from the second-order approximations to the problem. We formally state the algorithm in Section 3 and establish some convergence results in Section 4.

II. MOTIVATION

Consider the nonlinear programming problem

Minimize $f(x)$

subject to

$$\begin{aligned} &c_i(x)\le 0,\ i=1,2,\cdots,m',\\ &c_i(x)=0,\ i=m'+1,\cdots,m, \end{aligned} \qquad (2.1)$$

where the functions f and c_i, $i=1,\cdots,m$, are twice continuously differentiable. We define the exact penalty function F_r by

$$F_r(x)=f(x)+r\left[\sum_{i=1}^{m'}\max\{0,c_i(x)\}+\sum_{i=m'+1}^{m}|c_i(x)|\right] \qquad (2.2)$$

where $r>0$. It well known [6] that ,if r is sufficiently large, then local minima of (2.2) normally coincide with those of (2.1).

In the following, we let x be the current estimate to an opimal solution of (2.1) .The quadratic programming subproblem solved to determine a search direction in the methods of Han [10] and [12] takes the form

minimize $g^Td+\frac{1}{2}d^TBd$

$$\text{subject to } \begin{aligned} &c_i+g_i^Td\le 0,\ i=1,2,\cdots,m',\\ &c_i+g_i^Td=0,\ i=m'+1,\cdots,m, \end{aligned} \qquad (2.3)$$

where $g=\nabla f(x), c_i=c_i(x)$ and $g_i=\nabla c_i(x)$,and the matrix B is anapproximation to the Hessian of the Lagrangian

$$L(x,u)=f(x)+\sum_{i=1}^{m}u_ic_i(x) \qquad (2.4)$$

With respect to x .The solution $\bar{d}$ of (2.3) is then used to obtain the next iterate x^+ as

$$x^+=x+\alpha\bar{d}, \qquad (2.5)$$

where the step size α is selected in such a way that the exact penlty function (2.2) is decreased. For example, we may choose α if it satisfies the condition

$$F_r(x+\alpha\bar{d})\le F_r(x)+\sigma\alpha\left[\overline{F_r}(x,\bar{d})-F_r(x)\right] \qquad (2.6)$$

where σ is a constant such that $0<\sigma<1$ and $\overline{F_r}(x,d)$ is defined by

$$\overline{F_r}(x,d)=f(x)+g^Td+\frac{1}{2}d^TBd$$

$$+r\left[\sum_{i=1}^{m'}\max\{0,c_i+g_i^Td\}+\sum_{i=m'+1}^{m}|c_i+g_i^Td|\right]. \qquad (2.7)$$

Note that, if B is positive definite and r is sufficiently large, then the quantity $\overline{F_r}(x,\overline{d})-F_r(x)$ is negative and (2.6) is achieved by choosing σ to be small enough [14]. Global convergence of the iterates thus generated may be established under appropriate conditions [10,14]. In order to ensure a fast ultimate convergence, however, the step size $\alpha=1$ needs to be accepted by the line search criterion (2.6) eventually.

Unfortunately, the last property does not necessarily hold for the original Han-Powell methods, especially when the current point x is very close to the constraint surface. This is due to the fact that the exact penalty function F_r has a discontinuity of the first derivatives across that surface. To cope with this difficulty, let us consider the following second-order approximation to (2.1):

$$\text{minimize} \quad g^T d+\frac{1}{2}d^T Bd$$

$$\text{subject to} \quad \begin{aligned} &c_i+g_i^T d+\frac{1}{2}G_i d^T\le 0, i=1,2,\cdots,m', \\ &c_i+g_i^T d+\frac{1}{2}G_i d^T=0, i=m'+1,\cdots,m, \end{aligned} \qquad (2.8)$$

where G and G_i denote the Hessian matrices $\nabla^2 f(x)$ and $\nabla^2 c_i(x)$, respectively. Problem (2.8) itself is not second derivatives. Therefore, try to modify (2.8) into an ordinary quadratic programming problem.

To this end, let us consider the Kuhn-Tucker conditions for problem (2.8)

$$Gd+\sum_{i=1}^{m}u_i(G_i d+g_i)=-g \quad g_i^T d+\frac{1}{2}d^T G_i d\le -c_i, u_i\ge 0$$

$$u_i\left[c_i+g_i^T d+\frac{1}{2}d^T G_i d\right]=0, i=1,2,\cdots,m',$$

$$g_i^T d+\frac{1}{2}d^T G_i d=-c_i, i=m'+1,\cdots,m.$$

Rearranging terms, we may rewrite these conditions as

$$\left(G+\sum_{i=1}^{m}u_i G_i\right)d+\sum_{i=1}^{m}u_i\left(g_i+\frac{1}{2}G_i d\right)=-g+\frac{1}{2}\sum_{i=1}^{m}u_i G_1 d$$

$$\left(g_i+\frac{1}{2}G_i d\right)^T d\le -c_i, u_i\ge 0$$

$$\begin{aligned} &u_i\left[c_i+\left(g_i+\frac{1}{2}G_i d\right)^T d\right]=0, i=1,2,\cdots,m', \\ &\left(g_i+\frac{1}{2}G_i d\right)^T d=-c_i, i=m'+1,\cdots,m. \end{aligned} \qquad (2.9)$$

Now we replace some of the unknown quantities in (2.9) by those that appear in (2.3) or that are obtained by solving (2.3). Specifically, we put

$$p=g-\frac{1}{2}\sum_{i=1}^{m}\overline{u}_i(\overline{g}_i-g_i) \qquad (2.10)$$

and

$$a_i=\frac{1}{2}(\overline{g}_i+g_i), i=1,2,\cdots,m, \qquad (2.11)$$

where $\overline{u}_i$ are optimal Lagrange multipliers of (2.3) and $\overline{g}_i$ denote $\nabla c_i(x+\overline{d})$ for $1\,i=1,2,\cdots,m$ .Note that

$$p=g-\frac{1}{2}\sum_{i=1}^{m}\overline{u}_i G_i\overline{d}+\sum_{i=1}^{m}\overline{u}_i o(\|\overline{d}\|) \qquad (2.12)$$

And

$$a_i=g_i+\frac{1}{2}G_i\overline{d}+o(\|\overline{d}\|), i=1,2,\cdots,m \qquad (2.13)$$

Thus it seems reasonable to substitute p and $a_i, i=1,2,\cdots m$ for $g-\frac{1}{2}\sum_{i=1}^{m}u_i G_i d$ and

$$g_i+\frac{1}{2}G_i d, i=1,2,\cdots,m$$

respectively. Also, we may naturally replace $G+\sum_{i=1}^{m}u_i G_i$ by the matrix B, which appears in (2.3) as an approximation to the Hessian of the Lagrangian. As a result, we derive the following conditions from (2.9):

$$Bd+\sum_{i=1}^{m}u_i a_i=-p, a_i^T\le -c_i, u_i\ge 0$$

$$\begin{aligned} &u_i\left[c_i+a_i^T d\right]=0, i=1,2,\cdots,m', \\ &a_i^T d=-c_i, i=m'+1,\cdots,m. \end{aligned} \qquad (2.14)$$

It is then straightforward to get a quadratic programming problem whose Kuhn-Tucker conditions are (2.14),that is ,

$$\text{minimize} \quad p^T d+\frac{1}{2}d^T Bd$$

$$\text{subject to} \quad \begin{aligned} &c_i+a_i^T d\le 0, i=1,2,\cdots,m', \\ &c_i+a_i^T d=0, i=m'+1,\cdots,m. \end{aligned} \qquad (2.15)$$

where p and $a_i, i=1,2,\cdots m$,are given by (2.10) and (2.11), respectively. Let d^* and u^* denote a solution of (2.15) and corresponding Lagrange multipliers, respectively.

If the multipliers $\overline{u}_i$ are bouned, then, by (2.12) and (2.13), we may expect that the solution d^* of (2.15) is almost equal to the solution d^* of (2.3), when $\|\overline{d}\|$ is sufficiently small. Thus the iteration

$$x^+ = x + d^* \qquad (2.16)$$

is supposed to exhibit the same local convergence property as that of (2.5) with $\alpha = 1$. In addition, (2.12) suggest that d^* is a good approximation to the solution of (2.8), so that $x + d^*$ will satisfy the constraints of (2.1) to second order in case x lies on the constraint boundary. This implies x^+ given by (2.16) is eventually accepted by the line search criterion and hence that the Marators effect does not occur. These matters will be closely examined later on.

We should note that d^* is not necessarily a descent direction of the exact penalty function F_r defined by (2.2). So we perform a one-dimensional search along the arc

$$x(\alpha) = x + \alpha\bar{d} + \alpha^2(d^* - \bar{d}) \qquad (2.17)$$

with the search criterion

$$F_r(x(\alpha)) \le F_r(x) + \sigma\alpha\left[\overline{F_r}(x,\bar{d}) - F_r(x)\right] \quad (2.18)$$

where $0 < \sigma < 1$ and $\overline{F_r}(x,\bar{d})$ is given by (2.7). Since $x(\alpha) = x + \alpha\bar{d}$ for α small enough, search along arc $x(\alpha)$ by a procedure of Armijo type will yield a sufficient decrease of F_r, which is essential to ensure global convergence of an algorithm. Obviously, if the step size α is one, then (2.17) reduces to (2.16). Note that search arcs similar to (2.17) are also employed by Mayne and Polak [11] and Gabay [8].

III. ALGORITHM

Suppose that the parameters r and σ in (2.2) and (2.18) are appropriately selected. Also,let β be a real number such that $0 < \beta < 1$. Then a prototype of the algorithm may be sated as follows:

Step1. Choose a starting point x and a matrix B.

Step2. Solve (2.3) to obtain $\bar{d}$. If $\bar{d} = 0$, then stop.

Step3. Solve (2.15) to obtain d^*.

Step4. Find $\alpha = \beta^j$,where j is the smallest nonnegative integer such that (2.18) is satisfied.

Step5. Set $x = x + \alpha\bar{d} + \alpha^2(d^* - \bar{d})$,update the matrix B and return to Step 2.

Some comments on this algorithm are in order. Because the purpose of introducing d^* is to avoid the Maratos effect, it is clearly superfluous to solve (2.15) on every iteration. To be more practical, the algorithm should therefore be modified in such a way that (2.15) is solved only when it is needed. A possible modification of the algorithm is to calculate d^* when the point $x + \bar{d}$ does not satisfy (2.6) and, at the same time, $c_i + a_i^T\bar{d} = 0, i = 1,2,\cdots,m$.are much better than $c_i + g_i^T\bar{d} = 0, i = 1,2,\cdots,m$,as approximations to $c_i(x + \bar{d}), i = 1,2,\cdots,m$.

More specifically, let $\Delta_1 = \max\left\{\left|c_i + a_i^T\bar{d} - c_i(x+\bar{d})\right|\right\}$ and $\Delta_2 = \max\left\{\left|c_i + g_i^T\bar{d} - c_i(x+\bar{d})\right|\right\}$, where the maximum is taken over active constraints. Then we solve (2.15) if the inequality $\Delta_1 \le \mu\Delta_2$ is satisfied for some prescribed constant $\mu \in (0,1)$. This strategy requires the evaluation of constraint gradients at $x + \bar{d}$ in order to decide whether (2.15) is to be solved. Nevertheless, we may expected that the overall efficiency of the algorithm is improved compared with those using only $\bar{d}$,because the calculation of d^* can be viewed as an alternative to reducing the step length and using the search direction $\bar{d}$.

The algorithm assumes that problems (2.3) and (2.15) are always solvable. However,this assumption is rather restrictive, because the constraints of these subproblems may be inconsistent, even if the given problem is feasible, To cope with such difficulties, a practical implementation of the algorithm should include a suitable technique of resolving constraint inconsistency, such as those suggested by Powell [12],Bartholomew-Biggs [1] and Tone [15].

In order that the algorithm works efficiently, we have to specify a formula of updating the matrices B in such a way that B approximates the Hessian of the Lagrangian (2.4). Although there are many possibilities of doing this, the modified BFGS formula presented by Powell [12,13] seems most suitable, because it preserves the positive definiteness of B and it yields superlinear convergence of the iteration (2.5) with $\alpha = 1$ under certain conditions.

IV. CONVERGENCE

In this section we discuss convergence properties of the algorithm presented at the beginning of the previous section. for simplicity, notation such as $\{x\}$ will be used to represent a sequence generated by the algorithm.

We first consider the global behavior of the algorithm. To establish a global convergence theorem, we need the following assumptions:

(a) Problems (2.3) and (2.15) are always feasible.

(b) the matrices $\{B\}$ are symmetric, positive

semidefinite, and uniformly bounded.

(c) The sequences $\{x\}, \{\overline{d}\}$ and $\{\overline{d}\}$ produced by the algorithm are bounded.

(d) The penalty parameter r is large enough to satisfy

$$r > \sup\left\{\max_{1\le i\le m} \overline{u}_i\right\}.$$

Where $\overline{u}_i$ are optimal Lagrange multipliers of (2.3) and 'sup' is taken with respect to the sequence generated by the algorithm. Note that this condition assumes the boundedness of $\{\overline{u}\}$.

Theorem1. Let the above assumptions be satisfied. Then the algorithm either terminates at a Kuhn-Tucker point of (2.1) or generates an infinite sequence $\{x\}$ whose limitpoints are Kuhn-Tucker points of (2.1).

Proof. Because the algorithm terminates only if $\overline{d} = 0$, it is easily seen that , when the generated sequence is finite, the last iterate x is a Kuhn-Tucker point. Suppose that $\overline{d} \neq 0$ and let $x(\alpha)$ be defined by (2.17). Then it can shown that

$$F_r(x(\alpha)) - F_r(x) \le \alpha\left[\overline{F_r}(x,\overline{d}) - F_r(x)\right] + o(\alpha)$$

This implies that Step 4 of the algorithm can always find an $\alpha > 0$ satisfying (2.18). Moreover, by assumption (c) , there exists for each $\eta > 0$ a positive constant $\beta(\eta)$ such that (2.18) holds for any $\alpha \in [0, \beta(\eta)]$, whenever $F_r(x) - \overline{F_r}(x,\overline{d}) > \eta$. This fact enables us to follow the proof of Theorem 1 in [14] and hence the rest of the proof is omitted.

REFERENCES

[1] M.c.Bartholomew-Biggs,"Recursive quadratic

Programming methods for nonlinear constraints", in:M.J.D.Powell, ed., Nonlinear Optimization 1981 (Academic Press, New York,1982) pp.213-221.

[2] P.T.Boggs,J.W.Tolle and P.Wang," On the local convergence of quasi-Newton methods for constrain

-ed optinization",SIAM Journal on Control and Optim

-ization 20(1982) 161-171.

[3] R.M.Chamberlain, M.J.D. Powell, C. Lemarechal and H.C. Pedersen, "The watchdog technique for forcing convergence in algorithms for constrained optimization", Mathematical Programming Study 16

(1982) 1-17.

[4]T.F.Coleman and A.R. Connn,"Nonlinear programm

-ing via an exact penalty function: Asympotic analysis

", Mathematical Programming 24 (1982) 123-136.

[5]T.F.Coleman and A.R. Connn,"Nonlinear programm

-ing via an exact penalty function: Global analysis ", Mathematical Programming 24 (1982) 137-161.

[6] R.Fletcher, Practical methods of optimization,Vol. 2:Constrained optimization (John Wiley, Chichester, 1981).

[7] R.Fletcher, "Second order corrections for nondiff

-erentiable optimization", in:G.A.Watson,ed.,Numeri

-cal analysis, Dundee 1981 (Springer-Verlag, Be rlin, 1982) pp.85-114.

[8] D.Gabay, "Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization", Mathematical Programming Study 16

(1982) 18-44.

[9] S.P.Han, "Superlimearly convergent variable metric algorithms for general nonlinear programming problems", Mathematical Programming 11 (1976) 263-282.

[10] S.P.Han, "A globally convergent method for nonlinear programming", Journal of Optimization Theory and Applications 22 (1977) 297-309.

[11] D.Q.Mayne and E.Polak, "A superlinearly convergent algorithm for constrained optimization problems", Mathematical Programming Study 16(1982) 45-61.

[12] M.J.D.Powell, "A fast algorithm for nonlinearly constrained optimization calculations", in: G.A.Watson ed., Numerical Analysis, Dundee 1977(Springer-Verlag, Berlin,1978)pp.144-157.

[13] M.J.D.Powell, "The convergence of variable metric methods for nonlinearly constrained optimization calculations", in: O.L.Mangasarian , R.R.

Meyer and S.M.Robinson,des.,Nunlinear programm

-ing 3 (Academic Press, New York,1978) 27-63.

[14] M.J.D.Powell, "Variable metric methods for constrained optimization", in: A.Bachem, M.Grotschel and B.Korte , eds., Mathematical Programming :The state of the art,Bonn 1982 (Springer-Verlag,Berlin, 1983)pp.288-311.

[15] K.Tone,"Revisions of constraint approximations in the successive QP method for nonlinear programming problems ", Mathematical Programming 26 (1983) 144-152.