Noname manuscript No.
(will be inserted by the editor)

Step lengths in BFGS method for monotone gradients

Yunda Dong

Received: date / Accepted: date

Abstract In this paper, we consider how to directly apply the BFGS method to finding a zero point of any given monotone gradient and thus suggest new conditions to locate the corresponding step lengths. The suggested conditions involve curvature condition and merely use gradients' computations. Furthermore, they can guarantee convergence without any other restrictions. Finally, preliminary numerical experiments indicate their practical effectiveness in solving some systems of nonlinear equations arising in boundary value problems and others.

Keywords Monotone gradient · Quasi-Newton method · BFGS method · Convergence

1 Introduction

In this paper, we consider the following problem of finding an $x \in R^n$ such that

$$g(x) = 0, \tag{1}$$

where $g : R^n \to R^n$ is gradient whose function f is a continuously differentiable convex function. Special cases include some systems of nonlinear equations arising in boundary value problems and others (see Sect. 5 below).

A class of effective iterative procedures for solving the problem above are quasi-Newton methods (cf. [3, 14, 17]): For an initial point $x_0 \in R^n$ and an initial symmetric positive definite matrix W_0, quasi-Newton methods produce the new iterate from the

Y.D. Dong (✉)
School of Mathematics and Statistics, Zhengzhou University, Zhengzhou, 450001, PR China
E-mail: ydong@zzu.edu.cn

current iterate x_k by the recursion formula

$$x_{k+1} = x_k + \alpha_k d_k, \qquad \text{with} \quad d_k := -W_k g(x_k), \tag{2}$$

where $\alpha_k > 0$ is a step length and W_k is a symmetric positive definite matrix.

In this paper, we mainly consider the quasi-Newton method with BFGS formula of updating W_k (see the formula (4) below), i.e., the BFGS method.

Our focus is on how to design conditions on step lengths. In fact, the answer is not obvious at all. If the BFGS method is with the Wolfe conditions, then possibly unavailable function's evaluations become undesirable. If one turns to make use of approximate Wolfe conditions [2] (see two inequalities (10) below), then the method's convergence can not be guaranteed.

To resolve these issues, we suggest the following way of locating step lengths in the BFGS method: Choose α_k such that

$$c_2 g(x_k)^T d_k \le g(x_k + \alpha_k d_k)^T d_k \le c_1 g(x_k)^T d_k, \tag{3}$$

where $0 < c_1 < c_2 < 1$. Moreover, we can prove that, if the BFGS method is with the suggested conditions, then Powell's convergence theory can be guaranteed.

Note that this pair of inequalities (3) first appeared in [4, Lemma 3] and was used for analyzing the Fletcher-Reeves method [5] there. As shown below, the inequality on the right-hand side plays a role similar to Armijo condition in the Wolfe conditions and the other is the same as curvature condition in the Wolfe conditions.

To the best of knowledge, in the setting of unmodified BFGS method, the suggested conditions appear to be the first of three nice features at the same time. They keep the curvature condition so as to take advantage of second-order information, and they only need gradient's computations, and the corresponding Powell's convergence theory remains valid under standard assumptions (without any other restrictions).

The rest of paper is organized as follows. In Sect. 2, we will give a detailed description of the BFGS method with the Wolfe conditions and the corresponding Powell's convergence theory. In Sect. 3, we will discuss some basic properties of the suggested conditions. In Sect. 4, we will prove that, for the BFGS method with the suggested conditions, Powell's convergence remains valid. And in Sect. 5, we will confirm that its limited memory and scaled version effectively solved systems of nonlinear equations arising in boundary value problems and others. Finally, we will close this paper by some concluding remarks.

Notation. Throughout this paper, $x^T y$ stands for the usual inner product for $x, y \in R^n$, and $\|x\|$ for the induced norm by $\|x\| = \sqrt{x^T x}$, where T stands for the transpose of an element in R^n. The capital letter I denotes the identity matrix. g_k and $g(x_k)$ have the same meaning.

2 Preliminary knowledge

In this section, we review the BFGS method and also give some useful preliminary results.

Let f be a real continuously differentiable function in R^n. Its gradient is given by

$$g(x) := (\partial f/\partial x_1, ..., \partial f/\partial x_n)^T.$$

Recall that the gradient g is called monotone if the following relation

$$(x-y)^T(g(x)-g(y)) \geq 0$$

holds for all $x,\ y \in R^n$. Of course, if f is a real continuously differentiable convex function in R^n, then its gradient g must be monotone.

Below we state the BFGS method for solving the unconstrained minimization problem $\min_{x\in R^n} f(x)$, where its gradient $g(x)$ at any given $x \in R^n$ can be computed.

Algorithm 1 – BFGS Method

Step 0. Choose $x_0 \in R^n$ and a symmetric positive definite initial matrix W_0. Set $k := 0$.
Step 1. Compute $d_k = -W_k g_k$ and locate the step length $\alpha_k > 0$. Then the new iterate is given by

$$x_{k+1} = x_k + \alpha_k d_k.$$

Step 2. Compute $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. Update W_k by

$$W_{k+1} = (I - \gamma_k y_k s_k^T)^T W_k (I - \gamma_k y_k s_k^T) + \gamma_k s_k s_k^T, \text{ with } \gamma_k = 1/(s_k^T y_k). \quad (4)$$

Set $k := k+1$, and go to Step 1.

To analyze convergence properties of the BFGS method, we need to make the following assumptions.

Assumption 1
a) f is a real convex function in R^n.
b) The level set $\{x \in R^n : f(x) \leq f(x_0)\}$ is bounded.
c) f is twice continuously differentiable on this level set.

Based on these assumptions, Powell proposed the following convergence result in the year 1976.

Proposition 1 *Assume that the function f satisfies Assumption 1 and the step length $\alpha_k > 0$ satisfies the Wolfe conditions*

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k g_k^T d_k, \tag{5}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq c_2 g_k^T d_k, \tag{6}$$

where $0 < c_1 < c_2 < 1$ and d_k is the search direction given by the BFGS method. Then the sequence $\{x_k\}$ generated by the BFGS method converges to a minimizer of the function f.

Note that if the function f is not convex then the BFGS method may fail to converge, see [1, 12] for further discussions. In non-convex case, [10] analyzed how to modify the BFGS method to ensure its global convergence.

3 The suggested conditions

In this section, we first give the suggested conditions and then discuss some relevant issues.

As shown before, the suggested conditions are composed of the following two inequalities:

$$g(x_k + \alpha_k d_k)^T d_k \leq c_1 g_k^T d_k, \tag{7}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq c_2 g_k^T d_k, \tag{8}$$

where $0 < c_1 < c_2 < 1$. Note that the first inequality originally takes the following form:

$$g(x_k + \alpha_k d_k)^T d_k + 0.5\nu_k \|d_k\|^2 \leq c_1 g_k^T d_k, \tag{9}$$

where ν_k is a real number approximately computed by

$$\nu_k \approx \max\left\{0, -\frac{d_k^T(g(x_k + \alpha_{k-1} d_k) - g(x_k))}{\alpha_{k-1}\|d_k\|^2}\right\}, \quad \alpha_{-1} := 1,$$

in practical implementations. Yet, since the gradient has been assumed to be monotone, we may set $\nu_k \equiv 0$ throughout this paper. The proposal of the inequality (9) was enlightened by [8]; see [4, Appendex] for more details.

At first sight, the suggested conditions are much like the following approximate Wolfe conditions [2] used for analyzing the Fletcher-Reeves method

$$c_2 g_k^T d_k \leq g(x_k + \alpha_k d_k)^T d_k \leq -(1 - 2\delta) g_k^T d_k, \tag{10}$$

where $0 < \delta < \min\{0.5, c_2\}$, because both conditions merely involve gradients' computations. Yet, the coefficient $-(1 - 2\delta)$ here is always negative and thus is widely different from the parameter c_1 in (7), which is positive.

Such difference has two consequences. One is that the suggested conditions rule out optimal step length α_k^* satisfying

$$g(x_k + \alpha_k^* d_k)^T d_k = 0.$$

The other is that they shorten the interval of acceptable values, when compared to approximate Wolfe conditions. These two aspects appear to be restrictive. However, we would like to stress that, for quasi-Newton methods, whether or not the conditions are practical mainly depends on the acceptance frequency of trial value one. As done in Sect. 5, we can introduce a practical form (see (13) below) of the suggested conditions and it desirably accepts trial value one in most iterations.

Next, we give a simple way of locating such a step length at k-th iteration, which is well known (e.g., [9]) in optimization community.

Algorithm A

Step 0. Set $\mu = 0$ and $\nu = +\infty$. Choose $\alpha = 1$. Set $j = 0$.

Step 1. If α does not satisfy (7), then set $j = j + 1$, and turn to Step 2. If α does not satisfy (8), then set $j = j + 1$, and turn to Step 3. Otherwise, set $\alpha_k = \alpha$.

Step 2. Set $\nu = \alpha$, $\alpha = 0.5(\mu + \nu)$. Then turn to Step 1.

Step 3. Set $\mu = \alpha$, $\alpha = 2\mu$. Then turn to Step 1.

Since in the context of quasi-Newton methods, a trial value of α in the first line of Algorithm A is chosen to be one. This is reminiscent of one as a constant step length in Newton method.

Lemma 1 *Assume that $x_k \in R^n$, $d_k \in R^n$ satisfy $g_k^T d_k < 0$ and there exists the corresponding positive number α_k^* such that*

$$g(x_k + \alpha_k^* d_k)^T d_k = 0. \tag{11}$$

If the gradient $g(x)$ is continuous, then there exists some $\hat{\alpha}$ satisfying (7)-(8).

Proof Let $c = 0.5(c_1 + c_2)$. Then, $0 < c < 1$. Denote

$$\varphi_k(\alpha) = g(x_k + \alpha d_k)^T d_k - c g_k^T d_k.$$

It follows from the gradient's continuity and the assumption $g_k^T d_k < 0$ that

$$\lim_{\alpha \downarrow 0} \varphi_k(\alpha) = (1 - c) g_k^T d_k < 0.$$

On the other hand, from the assumption, we can get

$$\varphi_k(\alpha_k^*) = g(x_k + \alpha_k^* d_k)^T d_k - c g_k^T d_k = -c g_k^T d_k > 0.$$

Hence, the gradient's continuity implies that there exists some $\hat{\alpha} \in (0, \alpha_k^*)$ such that

$$\varphi_k(\hat{\alpha}) = 0.$$

So, the desired results follow. □

In the above Lemma 1, the first assumption $g_k^T d_k < 0$ holds automatically in the context of the BFGS method (cf.[3, 14, 17]). Here we shall be aware of that if $g_k^T d_k < 0$ then the curvature condition (8) yields

$$[g(x_k + \alpha_k d_k) - g_k]^T d_k \geq (c_2 - 1) g_k^T d_k > 0 \quad \Rightarrow \quad y_k^T s_k > 0.$$

As to the second assumption (11), it is not restrictive because it is implied by that the function f is bounded below along the ray $\{x_k + \alpha d_k : \alpha > 0\}$.

Lemma 2 *If the assumptions in Lemma 1 hold, then Algorithm A terminates within finite steps.*

Proof We prove it by contradiction. It follows from the assumption (11) and Lemma 1 that there must exist some α such that both (7) and (8) hold. In particular,

$$\varphi_k(\alpha_k^*) = g(x_k + \alpha_k^* d_k)^T d_k - c_1 g_k^T d_k = -c_1 g_k^T d_k > 0.$$

This implies that there exists a positive number $\bar{\nu}$ such that

$$g(x_k + \bar{\nu} d_k)^T d_k > c_1 g_k^T d_k.$$

Now we assume that the algorithm does not terminate within finite steps. Then, it eventually generates a nested sequence of finite intervals, halving in length at each iteration, i.e., for each $j = 0, 1, ...,$ we always have that $[\mu_{j+1}, \nu_{j+1}]$ is nested in the finite interval $[\mu_j, \nu_j]$ and

$$0 \leq \nu_{j+1} - \mu_{j+1} = \frac{1}{2}(\nu_j - \mu_j) \to 0, \text{ as } j \to +\infty.$$

The facts above imply

$$\lim_{j \to +\infty} \nu_j = \lim_{j \to +\infty} \mu_j = \alpha.$$

On the other hand, according to this algorithm, for each $j = 0, 1, ...,$ we always have

$$g(x_k + \nu_j d_k)^T d_k > c_1 g_k^T d_k, \quad g(x_k + \mu_j d_k)^T d_k < c_2 g_k^T d_k.$$

So, taking the limits along j yields

$$g(x_k + \alpha d_k)^T d_k \geq c_1 g_k^T d_k, \quad g(x_k + \alpha d_k)^T d_k \leq c_2 g_k^T d_k.$$

This contradicts $0 < c_1 < c_2 < 1$ and $g_k^T d_k < 0$. The proof is complete. □

4 Convergence

In this section, we mainly confirm that, under standard assumptions, Powell's convergence theory remains valid for the BFGS method with the suggested conditions.

Theorem 1 *For any given $x, d \in R^n$ and for all $\alpha \geq 0$, if the function f is continuously differentiable convex in R^n, then the following relation holds*

$$f(x + \alpha d) \leq f(x) + \alpha g(x + \alpha d)^T d.$$

Proof This is immediate and can be viewed an instance of [4, Lemma 1].

Theorem 2 *Assume that the function f of the gradient $g : R^n \to R^n$ satisfies Assumption 1. If the step length α_k is located by (7)-(8) then it must satisfy the following Wolfe conditions:*

$$\begin{aligned} &f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha g_k^T d_k, \\ &g(x_k + \alpha d_k)^T d_k \geq c_2 g_k^T d_k. \end{aligned}$$

Hence the sequence $\{\|g_k\|\}$ generated by the BFGS method converges to zero.

Proof Assume that the step length α_k is located by (7)-(8). Then we have

$$c_2 g_k^T d_k \leq g(x_k + \alpha_k d_k)^T d_k \leq c_1 g_k^T d_k, \tag{12}$$

where $0 < c_1 < c_2 < 1$. Meanwhile, the function f is continuously differentiable convex in R^n. Thus, it follows from Theorem 1 and the inequality on the right-hand side of (12) that

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k g(x_k + \alpha_k d_k)^T d_k \leq f(x_k) + c_1 \alpha_k g_k^T d_k.$$

This shows that the step length satisfies the first inequality in the Wolfe conditions. From this fact and the inequality on the left-hand side of (12) that, we know that the located step length satisfies the Wolfe conditions.

On the other hand, we also know that Assumption 1 holds. So, the convergence easily follows from Proposition 1 because f is a continuously differentiable convex function. □

From the proof of Theorem 2, we can see that the condition (7) plays a role similar to Armijo condition. It is used for ensuring sufficient reduction of the function value.

5 Preliminary numerical experiments

In this section, we checked numerical performances of the BFGS method with a practical form of the suggested conditions by solving systems of nonlinear equations arising in some boundary value problems and others. In our writing style, rather than striving for maximal test problems, we have tried to make the basic ideas and techniques as clear as possible.

All numerical experiments were run in MATLAB (R2014a) on a desktop computer with an Intel(R) Core(TM) i3-2120 CPU 3.30 GHz and 2 GB of RAM. The operating system is Windows XP Professional.

In practical implementations, we terminated the method whenever the stopping criterion $\|g_k\|_\infty \leq 10^{-12}$ was satisfied. At each iteration of the method, we set 20 to be the maximum value of the times of trials.

Note that, when the number of variables is large, it can be expensive to store and manipulate W_k used in (4) because this matrix is generally dense. So, we shall keep storing the ℓ most recent vector pairs $\{s_i, y_i\}$ for $i = k - \ell, ..., k - 1$ to update W_k (cf. [14, Chapter 7]). This is the basic idea of limited memory updates for the associated matrices in the BFGS method [11]; see Hager and Zhang [7] for recent developments. We set $\ell = 15$ and initial matrix

$$W_k := \frac{s_k^T y_k}{\|y_k\|^2} I, \quad \text{if } \operatorname{mod}(k, \ell) = 1,$$

instead of the unit matrix (as is well known, the choice of ℓ is problem-dependent in general.). This is a simple and effective way of introducing a scale in the implementations, suggested by Shanno and Phua [16].

From now on, "Algorithm NEW" stands for limited memory ($\ell = 15$) and scaled BFGS method with the following practical form of the suggested conditions:

$$0.9 g_k^T d_k \leq g(x_k + \alpha_k d_k)^T d_k \leq c_{1,k} g_k^T d_k, \quad \text{with } c_{1,k} = 0.0001 \times (1 - 0.9^k) - 0.9^k, \quad (13)$$

where the limit of the sequence $\{c_{1,k}\}$ is the constant 0.0001, which corresponds to that in Armijo condition in practical implementations. And "Algorithm OLD" stands for limited memory ($\ell = 15$) and scaled BFGS method with approximate Wolfe conditions with $\delta = 0.4$ and $c_2 = 0.9$.

The first test problem is one-dimensional and is given by

$$g(x) = 0.5 - \ln(1 + |x|) = 0.$$

The starting point is -3.69.

Table 1 Numerical results of Algorithm NEW for the test problems

Problem	starting point	n	k	Ngrad	Time	$\|g_k\|_\infty$	Times(1)	Ratio
1	-3.69	1	6	28	0.005	1.22e-015	4	0.67
2	$(-1.2, 1)^T$	2	28	49	0.032	0	23	0.82
3	$(h, ..., nh)^T$	8^2	298	420	0.258	9.92e-013	206	0.69
4	$(0, ..., 0)^T$	32^2	5	7	0.474	5.55e-016	5	1.00
4	$(1, ..., 1)^T$	32^2	8	10	0.920	3.94e-015	8	1.00

The second test problem is related to the gradient of Rosenbrock function $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, and it is given by

$$g_1(x) = 400x_1(x_1^2 - x_2) + 2(x_1 - 1) = 0, \quad g_2(x) = 200(x_2 - x_1^2) = 0.$$

The starting point is $(-1.2, 1)^T$.

The third test problem is to find a root of systems of nonlinear equations arising in boundary value problems [15, Chapter 1]. Below we set $h := 1/(n + 1)$ be the discretization mesh size:

$$\begin{aligned} &2x_1 - x_2 + h^2 c \sin(x_1) - p = 0, \\ &-x_{i-1} + 2x_i - x_{i+1} + h^2 c \sin(x_i) = 0, \quad 1 < i < n, \\ &-x_{n-1} + 2x_n + h^2 c \sin(x_n) - q = 0, \end{aligned}$$

where c is a constant number, and p, q are two real numbers. The i-th entry of the starting point is jh, $j = 1, ..., n$. In our implementations, we set $p = 0$ and $c = q = 1$.

The fourth test problem is to find a root of system of nonlinear equations

$$g_j(x) = x_j - \frac{j}{n} + \frac{1}{2}\frac{j}{n^2}\sum_{i=1}^{n} \cos(x_i) = 0, \quad j = 1, ..., n. \tag{14}$$

This problem corresponds to the discretization of the following integral equation problem (see, e.g., [6, Example 2]) of finding a continuous function x defined on the interval $[0, 1]$ such that

$$x(s) - s + \frac{1}{2}\int_0^1 s \cos(x(t))dt = 0, \quad \forall s \in [0, 1].$$

In practical implementations, we chose 0 and $(1, ..., 1)^T$ as starting points, respectively.

For all the test problems above, we ran both Algorithm NEW and Algorithm OLD. The former's numerical results were reported in the following table, where "Ngrad" stands for the number of gradient evaluations, "Time" for solution time in seconds, "Times(1)" for times of appearance of unit step lengths, and "Ratio" for appearance's frequency of unit step lengths during the whole iteration process.

From Table 1, we can see that

- Firstly, Algorithm NEW successfully solved all these test problems. In particular, it solved the test problems 1,2,4 up to at least 10^{-14} efficiently.
- Secondly, Algorithm NEW accepted the trial value 1 itself in most iterations. The acceptance frequency was nearly 70 percent or even higher. This implies that the practical form (13) of the suggested conditions appear to be of a desirable property of making trial value one generally acceptable in practical implementations.
- Thirdly, Algorithm NEW solved the test problem 4 very effectively, and this is surprising. In fact, this problem corresponds to systems of nonlinear equations arising in integral equation problems and its Jacobian matrices is thus not symmetric. In such case, it is more general than (1) because the latter's Jacobian matrix (if such matrix exists) must be symmetric. This interesting phenomenon indicates general practical applications of Algorithm NEW.

Note that, as mentioned above, we also ran Algorithm OLD for all these test problems. The returned numerical results confirmed that it can be as effective as Algorithm New. Yet, we would like to specially stress that the latter can guarantee theoretical convergence in the case of the test problem 3 whereas Algorithm OLD failed to do so. In addition, for all these test problems, the corresponding dimensions are not large because the BFGS method is for small-scale and medium-scale problems.

6 Conclusions

It is well-known that the BFGS method is one of the few fundamental iterative procedures for solving unconstrained minimization in Euclidean spaces. Unfortunately, for a long time, one has known little about its convergence properties when directly applied to finding a zero point of monotone gradient, in which case the function of gradient has been assumed to be unavailable.

In this paper, we have settled such suspense successfully by suggesting new conditions for the BFGS method. The suggested conditions merely involve gradients' computations and also keep the curvature condition so as to take advantage of second-order information. One the one hand, the suggested conditions can guarantee that Powell's convergence theory remains valid without any other restrictions. On the other hand, the practical form of the suggested conditions can make unit step length acceptable in most iterations, as approximate Wolfe conditions do.

In this research direction, there are several issues to be solved. We only specify two of them. We have already confirmed convergence of the BFGS method with the suggested conditions. Then, whether or not it is possible to analyze its theoretical convergence when applied to solving problem like (14), in which the corresponding Jacobian matrix is not symmetric. Finally, we would like to give a second issue: What is the relation between $f(x_k) - \min f(x)$ and $1/k$ if Assumption 1 holds? These issues deserve further investigation.

Acknowledgments Special thanks go to Dr. Y.Y. Huang and Dr. X. Zhang for writing Matlab codes, and Dr. C.Z. Zhang for a critical modification. Also, to X.X. Bai for his help.

References

1. Dai Y.H.: Convergence properties of the BFGS algorithm. SIAM J. Optim. **13**(3), 693-701 (2003)
2. Dai Y.H., Yuan Y.X.: Convergence properties of the Fletcher-Reeves method. IMA J. Numer. Anal. **16**, 155-164 (1996)
3. Dennis J.E., Moré J.J.: Quasi-Newton method, motivation and theory, SIAM Rev. **19**, 46-89 (1977)
4. Dong Y.D.: New step lengths in conjugate gradient methods. Comput. Math. Appl. **60**(3), 563-571 (2010)
5. Fletcher R., Reeves C.M.: Function minimization by conjugate gradients. J. Comput. **7**, 149-154 (1964)
6. Gutiérrez J.M., Hernández M.A.: An acceleration of Newton's method: Super-Halley method. Appl. Math. Comput. **117**(2-3), 223-239 (2001)
7. Hager W.W., Zhang H.C.: The limited memory conjugate gradient method. SIAM J. Optim. **23**, 2150-2168 (2013)
8. Konnov I.V.: Combined relaxation method for finding equilibrium points and solving related problems. Russ. Math. (Iz. VUZ). **37**(2), 44-51 (1993)
9. Lewis A.S., Overton M.L.: Nonsmooth optimization via quasi-Newton methods. Math. Program. **141**, (2013) 135-163.
10. Li D.H., Fukushima M.: On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. SIAM J. Optim. **11**, 1054-1064 (2001)
11. Liu D.C., Nocedal J.: On the limited-memory BFGS method for large scale optimization. Math. Program. **45**, 503-528 (1989)
12. Mascarenhas W.F.: The BFGS method with exact line searches fails for non-convex objective functions. Math. Program. **99**, 49-61 (2004)
13. Moré J.J., Garbow B.S., Hillstrom K.E.: Testing unconstrained optimization. ACM Trans. Math. Software. **7**, 17-41 (1981)
14. Nocedal J., Wright J.S.: *Numerical Optimization, Second Edition.* Springer, New York (2006)
15. Ortega J.M., Rheinboldt W.C.: *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York (1970)
16. Shanno D.F., Phua K.H.: Matrix conditioning and nonlinear optimisation. Math. Program. **14**, 149-160 (1978)
17. Sun W.Y., Yuan Y.X.: *Optimization Theory and Methods: Nonlinear Programming.* Springer, New York (2006)