JOURNAL OF THE
CHUNGCHEONG MATHEMATICAL SOCIETY
Volume **21**, No. 1, March 2008

THE PERFORMANCE OF A MODIFIED ARMIJO LINE SEARCH RULE IN BFGS OPTIMIZATION METHOD

MinSu Kim*, SunJoo Kwon**, and SeYoung Oh***

Abstract. The performance of a modified Armijo line search rule related to BFGS gradient type method with the results from other well-known line search rules are compared as well as analyzed. Although the modified Armijo rule does require as much computational cost as the other rules, it shows more efficient in finding local minima of unconstrained optimization problems. The sensitivity of the parameters used in the line search rules is also analyzed. The results obtained by implementing algorithms in Matlab for the test problems in [3] are presented.

1. Introduction

The unconstrained optimization problem requires only the objective function as

$$\min\ f(x),\ x \in R^n \tag{1.1}$$

where R^n is an n-dimensional Euclidean space and $f : R^n \to R^1$ is a continuously differentiable function. There are no functional constraints although the side constraints are necessary to keep the solution finite.

While unconstrained optimization does not occur often in engineering design, its numerical techniques provide the means to solve constrained problems by transforming it into an unconstrained one.

The typical numerical techniques for solving (1.1) are iterative and also refers to as line search methods. The iterations choose a search direction by moving along the direction while taking an appropriate step size.

Each iteration of the line search methods for (1.1) is given by

$$x_{k+1} = x_k + \alpha_k d_k,\ k = 1, 2, 3, \dots, \tag{1.2}$$

Received February 12, 2008.
2000 Mathematics Subject Classification: Primary 65M60 76E06.
Key words and phrases: unconstrained optimization, line search method, BFGS method.

where $x_1 \in R^n$ is an initial point, d_k is a descent direction of $f(x)$ at x_k, and the positive scalar α_k is the step size.

Line search methods differ in how the search direction and the step size are chosen. The different choices of d_k and α_k yield different convergence properties. Generally the first-order conditions(FOC), $\nabla f(x^*) = 0$, is used to check for local convergence to a stationary point x^*.

The search directions can be constructed in many ways. According to the information regarding the order of the derivative of the objective function, the techniques for the search directions can be classified in the following: gradient methods, conjugate gradient methods, Newton's methods, quasi-Newton's methods, and derivative-free methods.

After the descent direction d_k is taken, line search methods investigate a step size α_k along the search direction. The different choices of the step size α_k ensure that the sequence of the iterates x_k defined by (1.2) is globally convergent with the some rate of convergence.

The exact value of the step size α_k is $\alpha_k = \underset{\alpha>0}{argmin}\ f(x_k+\alpha d_k)$ that is the minimizer of the function f in terms of α along the descent direction d_k. However, the exact step size is difficult and often impossible to find in practical computation. To find even a local minimizer of $f(x_k + \alpha d_k)$ generally requires too many evaluations of the objective function f and possibly the gradient ∇f. More practical strategies in performing an inexact line search to identify a step length may help achieve adequate reductions in f at minimal cost.

Typical line search algorithms are tried by applying a sequence of candidate values for α_k while certain conditions are satisfied. Although sophisticated line search algorithms can be quite complicated, the line search is done in two following stages: A bracketing phase finds an interval containing desirable step lengths, and a bisection or interpolation phase computes a good step length within this interval.

Many inexact line search methods have been proposed: Armijo[8], Wolfe[9], Porta and Shi[6], Goldstein[2], and others. Recently Shi and Shen[7] proposed a new inexact line search rule similar to the Armijo rule analyzed the global convergence.

We present the performance of a modified Armijo line search rule associated with BFGS(Broyden, Fletcher, Goldfarb, Shanno[10], [4]) gradient type method to compare with other well-known line search rules. The modified Armijo rule proposed by Shi in [7] has very similar behaviors with the trust region method. Although it requires as much computational cost as the other rules, it shows more efficient in finding local minima of unconstrained optimization problems. The sensitivity of the parameters used in the line search rules is also analyzed. The algorithms implemented in Matlab were run with many test problems shown in [3] on 2.4MHz Intel desktop. We denote the function value $\nabla f(x_k)$ by g_k.

Section 2 contains a brief discussion of BFGS for a descent direction. Here, the concepts about the Wolfe and Goldstein rules in [5] are presented. The new inexact line search called modified Armijo rule is also summarized. Some numerical results and efficiency of the modified Armijo step size rule with BFGS algorithm are shown in Section 3. Conclusions are given in Section 4.

2. Line search methods

We assume that $f(x)$ is bounded below and its gradient $g(x)$ is Lipschitz continuous in an open convex set. The search direction d_k is generally required to satisfy

$$\nabla f(x_k)^T d_k < 0 \quad \text{or} \quad \nabla f(x_k)^T d_k \leq -c \left\| \nabla f(x_k) \right\|^2, \tag{2.1}$$

which guarantees that d_k is a sufficient and descent direction of $f(x_k)$ at x_k and $c > 0$ is a constant.

The search directions can be determined in many ways. The techniques for finding the search direction primarily differ in how the search direction is established. BFGS method is a gradient-based method which is one of quasi-Newton method as it behaves like Newton's method in approaching the solution. It has the property of quadratic convergence and robustness while carrying the information forward from the previous iterations. It currently is the most popular one in the variable metric methods.

Algorithm 1. BFGS(Broyden-Fletcher-Goldfarb-Shanno) method.

Step 0. Choose x_0, B_0, $k := 1$.
Step 1. If $\|\nabla f(x_k)\| \leq eps$, then stop; else, go to Step 2.
Step 2. The search direction is obtained as a solution to

$$B_k d_k = -\nabla f(x_k)$$

and set $x_{k+1} = x_k + \alpha_k d_k$.
Step 3. Updated by

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k},$$

where $y_k = \nabla f(x_{k+1}) - \nabla f(x_k),\ s_k = x_{k+1} - x_k$.
Step 4. $k := k + 1$; Go to Step 1.

Most line search algorithms require d_k to be a descent direction because this property guarantees that the function f can be reduced along this direction. Moreover, the search direction often has the form

$$B_k d_k = -\nabla f(x_k), \tag{2.2}$$

where B_k is a symmetric and nonsingular matrix. In the steepest descent method B_k is simply the identity matrix I, while in Newton's method B_k is the exact Hessian $\nabla^2 f(x_k)$. In quasi-Newton methods, such as BFGS and DFP method, B_k is an approximation of the Hessian that is updated at every iteration by means of a low-rank formula. However, the difference between the DFP and BFGS is the way the metric is updated: the former converges to the inverse of the Hessian, while the latter converges to the Hessian itself. In a sense the BFGS is more direct as the BFGS replaces the conjugate gradient techniques as a workhorse in solving nonlinear equations.

When d_k is defined by (2.2) and B_k is positive definite, we have

$$d_k^T \nabla f(x_k) = -\nabla f(x_k)^T B_k^{-1} \nabla f(x_k) < 0, \tag{2.3}$$

and therefore d_k is a descent direction.

An initial choice of positive definite matrix for B_k is usually the identity matrix as a default choice.

More practical strategies are needed in order to perform an inexact line search to identify a step length that achieves adequate reductions in f at minimal cost.

One of the well-known and useful inexact line search rules is proposed by Armijo, called Armijo line search rules. Given an initial step size $s > 0$ and $\beta \in (0, 1)$, choose α_k to be the largest one in $\{s, s\beta, s\beta^2, \ldots\}$ such that

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f(x_k)^T d_k. \tag{2.4}$$

The α_k should give sufficient decrease in the objective function f, as measured by (2.4).

The sufficient decrease condition may not be enough by itself to ensure that the algorithm makes reasonable progress, since it is satisfied for all sufficiently small values of α. To rule out unacceptably short steps α_k is required to satisfy

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f(x_k)^T d_k, \tag{2.5}$$

for some constant $c_2 \in (c_1, 1)$, where c_1 is the constant from (2.4). For this curvature condition, typical value of c_2 is 0.9 when the search direction d_k is chosen by a Newton, quasi-Newton, as well as BFGS methods.

The sufficient decrease (2.4) and curvature conditions (2.5) are known collectively as the Wolfe conditions. The strong Wolfe conditions require α_k to satisfy

$$f(x_k + \alpha_k d_k) \le f(x_k) + c_1 \alpha_k \nabla f(x_k)^T d_k \tag{2.6}$$

and

$$\left|\nabla f(x_k + \alpha_k d_k)^T d_k\right| \le c_2 \left|\nabla f(x_k)^T d_k\right|, \tag{2.7}$$

with $0 < c_1 < c_2 < 1$. The condition (2.7) does not allow the derivative to be too positive. Hence, we exclude points that are far from stationary points of $\phi(\alpha) = f(x_k + \alpha d_k),\ \alpha > 0$.

It is not difficult to prove that there exist step lengths which satisfy the Wolfe conditions for every function f that is smooth and bounded below.

THEOREM 2.1. *([6]) Suppose that $f : R \to R$ is continuously differentiable. Let d_k be a descent direction at x_k, and assume that f is bounded below along the line $\{x_k + \alpha d_k | \alpha > 0\}$. Then if $0 < c_1 < c_2 < 1$, there exist intervals of step lengths satisfying the Wolfe conditions and the strong Wolfe conditions.*

The Wolfe conditions are scale-invariant in a broad sense; multiplying the objective function by a constant or making an affine change of variables does not alter them. The convergence results are proved in [9] which may be used in most line search methods, particularly important in the implementation of quasi-Newton methods.

How to choose the parameters (such as s, c_1, c_2, β) is very important for practical problems. Several valid techniques have been appearing in many literatures.

Similar to the Wolfe conditions, the Goldstein conditions also ensure that the step length achieves sufficient decrease while preventing from being too small such as

$$\begin{aligned} f(x_k) + (1 - c_1)\alpha_k \nabla f(x_k)^T d_k &\le f(x_k + \alpha_k d_k) \\ &\le f(x_k) + c_1 \alpha_k \nabla f(x_k)^T d_k, \end{aligned} \tag{2.8}$$

with $0 < c_1 < \frac{1}{2}$. Thus, the Goldstein and Wolfe conditions have much in common, and their convergence theories are quite similar. The Goldstein conditions are often used in Newton-type methods, however, are not well suited for quasi-Newton methods with a positive definite Hessian approximation.

Algorithm 2. Modified Armijo line search rule.

Step 0. Choose $x_1 \in R^n$, a symmetric positive definite matrix B_1, $\beta \in (0,1)$, and $c_1 \in (0, \frac{1}{2})$; $k := 1$.
Step 1. $g_k := \nabla f(x_k)$. If $\|g_k\| \leq eps$ then stop else go to Step 2.
Step 2. $d_k = -B_k^{-1} g_k$.
Step 3. $s_k = -\frac{g_k^T d_k}{d_k^T B_k d_k}$ and choose α_k to be the largest one in $\{s_k, s_k\beta, s_k\beta^2, \ldots\}$ such that $f(x_k + \alpha d_k) - f(x_k) \leq c_1 \alpha \left[\nabla f(x_k)^T d_k + \frac{1}{2}\alpha d_k^T B_k d_k\right]$.
Step 4. Set $x_{k+1} = x_k + \alpha_k d_k$.
Step 5. Set $s_k = x_{k+1} - x_k$, $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. Modify B_k as B_{k+1} by using BFGS formula.
Step 6. $k := k + 1$; Go to Step 1.

THEOREM 2.2. *([9]) If $f(x)$ is bounded below and its gradient $g(x)$ is Lipschitz continuous in an open convex set, d_k satisfies* (2.1) *and α_k is defined by the new Armijo search. Algorithm 2 generates an infinite sequence x_k with a bounded sequence B_k, that is, there is a β such that $\|B_k\| \leq \beta$, $\forall$ k. Then $\lim_{k\to\infty}(-\frac{g_k^T d_k}{\|d_k\|}) = 0$ and $\lim_{k\to\infty}\|g_k^T\| = 0$.*

3. Numerical results

In this section, the performances of BFGS algorithm with Wolfe, as well as Goldstein conditions, and the modified Armijo step size rule are discussed for the comparison purpose. The descent directions for the line search methods are determined by BFGS algorithm, and three different step size rules presented in the previous section are used for appropriate reductions of the objective function value. The implemented algorithm for the unconstrained optimization problem is stated as follows.

Algorithm 3.

Step 0. Given some parameters $\beta \in (0,1)$, $c_1 \in (0, \frac{1}{2})$, $c_2 \in (c_1, 1)$, $B_1 = I_n$, $eps = 10^{-2}$, let $x_1 \in R^n$ and set k:=1.
Step 1. If $\|\nabla f(x_k)\| \leq eps$, then stop; else, go to Step 2.
Step 2. Choose d_k to satisfy (2.2).
Step 3. Set $x_{k+1} = x_k + \alpha_k d_k$, where α_k is defined by the new rule, Wolfe's rule and Goldstein rule.
Step 4. B_{k+1} is determined by BFGS formula.
Step 5. $k := k + 1$; Go to Step 1.

The codes were written in Matlab. Five test problems in [3] and one problem in the Navier-Stocks equations from fluid mechanics are chosen. The problems were tested on a Intel PC with 2.4MHz. For each test problem, we chose a few initial points x^0 at each distance r ($= 1, 2, 3$ or 10) from the known optimal points x^* and investigated the average number of iterations and execution cpu times for three step size rule algorithms. The stopping condition was $\|\nabla f(x_k)\| \leq 10^{-2}$. The solution is dependent on the starting point, and the tolerance for one-dimensional step size search as well as the derivatives are computed numerically using the finite difference method. Note that each component in Tables is an average number of iterations(execution cpu time(sec)).

PROBLEM 3.1. (Rosenbrock function)
(i) $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

TABLE 1. $\beta = 0.99,\ c_1 = 10^{-1},\ c_2 = 0.9,\ \|x_0 - x^*\| = r$

	New rule	Wolfe's rule	Goldstein's rule
$r = 1$	40(0.1)	41(0.1)	98(0.8)
$r = 2$	65(0.2)	60(0.2)	87(0.9)
$r = 3$	81(0.2)	76(0.1)	226(5.1)

(ii) $f(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right],\ n = 100$

TABLE 2. $\beta = 0.99,\ c_1 = 10^{-6},\ c_2 = 0.1,\ \|x_0 - x^*\| = r$

	New rule	Wolfe's rule	Goldstein's rule
$r = 1$	539(3.5)	421(11.2)	539(3.7)
$r = 2$	853(5.9)	154(9.7)	853(5.8)
$r = 3$	1013(7.2)	176(10.4)	1013(7.1)

PROBLEM 3.2. (McCormic curved valley)
$f(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1$

TABLE 3. $\beta = 0.99,\ c_1 = 0.1,\ c_2 = 0.9,\ \|x_0 - x^*\| = r$

	New rule	Wolfe's rule	Goldstein's rule
$r = 1$	6.75(0.01)	6.5 (0.01)	6.5 (0.03)
$r = 2$	6.5 (0.01)	6.25(0.02)	6.25(0.03)
$r = 3$	9 (0.01)	9 (0.05)	9 (0.04)

PROBLEM 3.3. (Powell's quartic function)
$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$

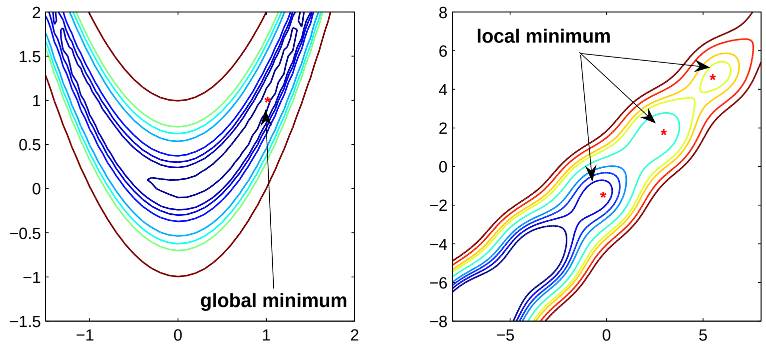

FIGURE 1. Contour plots of Rosenbrock function(left) and McCormic curved valley(right).

TABLE 4. $\beta = 0.9,\ c_1 = 10^{-2},\ c_2 = 0.1,\ \|x_0 - x^*\| = r$

	New rule	Wolfe's rule	Goldstein's rule
$r = 1$	19(0.03)	16(0.03)	19(0.03)
$r = 2$	25(0.03)	19(0.03)	25(0.03)
$r = 3$	26(0.03)	19(0.03)	25(0.03)
$r = 10$	69(0.18)	362(9.2)	69(0.18)

PROBLEM 3.4. (Wood's function)

$$f(x) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$$

TABLE 5. $\beta = 0.85,\ c_1 = 10^{-3},\ c_2 = 0.5,\ \|x_0 - x^*\| = r$

	New rule	Wolfe's rule	Goldstein's rule
$r = 1$	62.6 (0.04)	33.3(0.03)	62.6 (0.04)
$r = 2$	59.75(0.04)	60 (0.04)	59.75(0.04)
$r = 3$	60 (0.04)	61 (0.04)	60 (0.04)
$r = 10$	82 (0.31)	312 (7.64)	82 (0.31)

PROBLEM 3.5. (Watson function)

$f(x) = \sum_{i=1}^{31} f_i(x)^2$, where

$$f_i(x) = \sum_{j=2}^{30}(j-1)x_j t_i^{j-2} - \left(\sum_{j=1}^{n} x_j t_j^{j-1}\right)^2 - 1,\ t_i = \tfrac{i}{29},\ 1 \le i \le 29,$$
$$f_{30}(x) = x_1,\ f_{31}(x) = x_2 - x_1^2 - 1.$$

TABLE 6. $\beta = 0.9,\ c_1 = 10^{-3},\ c_2 = 0.4$

Start point	New rule	Wolfe's rule	Goldstein's rule
$(0,0,\ldots)$	290(9.32)	311(12.79)	290(9.40)
$(1,1,\ldots)$	390(22.11)	223(32.14)	failed
$(0,1,0,1\ldots)$	270(11.15)	207(28.23)	failed

PROBLEM 3.6. (Three-dimensional flow near a rotating disk[8]) This example represents another exact solution to the Navier-Stokes equations from fluid mechanics. The nonlinear two-point boundary value problem describes a viscous flow around a flat disk which rotates about an axis perpendicular to its plane with a constant angular velocity. Once the problem is redefined mathematically, the new set of coupled nonlinear equations with boundary conditions at two points is as follows:

$$\begin{aligned} 2F + H' &= 0 \\ F^2 + F'H - G^2 - F'' &= 0 \\ 2FG + HG' - G'' &= 0 \\ F(0) = 0,\ G(0) = 1,\ H(0) &= 0 \\ F(\infty) = 0,\ G(\infty) &= 0 \end{aligned} \tag{3.1}$$

Equations are converted to state space form. The design variables are the missing boundary conditions on $F'(0)$, $G'(0)$ and the final value of the independent variable. The objective function is the squared error in the integrated value at the final condition.

The optimization problem

$$min\ f(x_1,x_2,x_3):\ y_1(x_3)^2 + y_3(x_3)^2$$

where the state variables are obtained from the solution of the following initial value problem:

$$\begin{bmatrix} y_1' \\ y_2' \\ y_3' \\ y_4' \\ y_5' \end{bmatrix} = \begin{bmatrix} y_2 \\ y_1^2 + y_2y_5 - y_3^2 \\ y_4 \\ 2y_1y_3 + y_4y_5 \\ -2y_1 \end{bmatrix};\ \begin{bmatrix} y_1(0) \\ y_2(0) \\ y_3(0) \\ y_4(0) \\ y_5(0) \end{bmatrix} = \begin{bmatrix} 0 \\ x_1 \\ 1 \\ x_2 \\ 0 \end{bmatrix}$$

The state equations are integrated using Matlab function *ode45*.

TABLE 7. $\beta = 0.8,\ c_1 = 10^{-3},\ c_2 = 0.5$

Start point	New rule	Wolfe's rule	Goldstein's rule
$(1, -1, 8)$	25(6.9)	20(28.5)	25(7.2)
$(3, -2, 10)$	89(184.3)	failed	failed

The numerical results are shown in Tables 1-7. The (extended) Rosenbrock's problem is notorious since it requires a large number of iterations for convergence. As the result indicates, the new rule and Goldstein's rule have performed better than Wolfe's rule. For small problems such as McCormic curved valley, Powell's quadratic function, and Wood's function, all three rules have performed similary as shown in Tables 3-5. However, the new rule needed less number of iterations and execution time when the dimension of variables are increased as shown in Tables 6-7. Especially the Goldstein's rule that had very similar performances to the new rule for the smaller problems has often failed in finding the optimal solutions.

4. Conclusions

An application of Shi's line search rule which is combined with BFGS descent method is presented and its efficiency is investigated. In such algorithm, we can use an approximation of the Hessian as the objective function to get a large step size in each iteration and still maintain the global convergence of BFGS method. As shown in the computational results, when the dimension n of the problems is increased, the new step size rule has performed better than other rules such as Wolfe's and Goldstein's rule.

References

[1] L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific Journal of mathematics **6** (1966), 1-3.

[2] A. A. Goldstein, *On steepest descent*, SIAM Journal on Control **3** (1965), 147-151.

[3] J. J. Morè, B. S. Garbow and K. E. Hillstron, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software, **7** (1981), 17-41.

[4] J. Nocedal, *Theory of algorithms for unconstrained optimization*, Acta Numerica **1** (1992), 199-242.

[5] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer-Verlag, New York, 2006.
[6] J. M. Ortega, *Introduction to paralled and vector solution of linear systems*, New York: Plenum press, 1988.
[7] F. A. Potra and Y. Shi, *Efficient line search algorithm for unconstrained optimization*, Journal of Optimization Theory And Applicaions **85** (1995), 677-704.
[8] Z. J. Shi and J. Shen, *New inexact line search method for unconstrained optimization*, Journal of Optimization Theory And Applicaions **127** (2005), 425-446.
[9] Z. J. Shi and X. S. Zhang, *From line search method to trust region method*, International Symposium on OR and Its Applications 2005, 156-170.
[10] P. Venkataraman, *Applied optimization with MATLAB programming*, Wiley, 2001.
[11] P. Wolfe, *Convergence conditions for descent methods*, SIAM Review, **11** (1968), 226-235.
[12] Y. Yuan, *A modified BFGS algorithm for unconstrained optimization*, IMA J. Numerical Analysis **11** (1991), 325-332.

*
Department of Mathematics
Chungnam National University
Daejeon 305-764, Republic of Korea
E-mail: mskim@math.cnu.ac.kr

**
Department of Mathematics
Chungnam National University
Daejeon 305-764, Republic of Korea
E-mail: sjkw@cnu.ac.kr

***
Department of Mathematics
Chungnam National University
Daejeon 305-764, Republic of Korea
E-mail: soh@cnu.ac.kr