Iraqi Journal of Statistical Science (11) 2007
P.P. [27-46]

Modifying Of Barzilai and Borwein Method for Solving Large-Scale Unconstrained Optimization Problems.

Khalil K. Abbo*

ABSTRACT:

In this paper we present a technique for computing the minimum value of an objective function in the frame of gradient descent methods based on combination of Barzilai and Borwein approximation of Hessian matrix of objective function and Lipchetz constant in the gradient flow algorithm which is derived from a system of ordinary differential equations associated to unconstrained optimization problem. This algorithm suitable for large- scale unconstrained optimization problems, computational results for this algorithm is given and compared with BB method showing a considerable improvement.

* **College of Computers Sciences and Mathematics /Department of Mathematics/ University of Mosul**

Received: 29/ 8 /2006 _____Accepted: 30 / 10 / 2006

طريقة برزيليا وبورين المعدلة لحل مسائل الامثلية غير المقيدة ذات القياس الواسع.

المستخلص

تم في هذا البحث اقتراح خوارزمية لحساب النهاية الصغرى لدالة الهدف f(x) في اطار طرائق التدرج المنحدر استندت الى استخدام التقريب لمصفوفة هيسيان الذي استخدمه برزيليا وبروين وثابت ليبشز في خوارزمية الجريان المندرج التي تم اشتقاقها من نظام معادلات تفاضلية اعتيادية مرتبطة مع مسألة الأمثلية غير المقيدة. هذه الخوارزمية ملائمة لدوال تحتوي على عدد كبير من المتغيرات والنتائج العددية (لبعض دوال الاختبار) لهذه الخوارزمية مقارنة مع طريقة BB تشير الى كفاءة الخوارزمية.

1- Introduction

A well-known algorithm for the unconstrained optimization of function f(x) in n variables

$$f: R^n \to R \; ; \quad x \in R^n \qquad \ldots\ldots\ldots(1)$$

having Lipchetz continuous first partial derivatives whose gradient $\nabla f(x) = g(x)$ is available, is the steepest descent method first proposed by Cauchy in 1874. The iterations are made according to the following equation

$$x_{k+1} = x_k + \alpha_k d_k \qquad \ldots\ldots\ldots(2)$$

Where $d_k = -g_k$ and α_k is a step size. It's well known that the negative gradient direction has the following optimal property see (Dai et. al. 1998)

$$-g_k = \underset{d \in R^n}{Min}\ \underset{\alpha \to +0}{Lim}\left[f(x_k) - f\left(x_k + \frac{\alpha d}{\|d\|^2} \right) \right] \times \frac{1}{\alpha} \quad \ldots\ldots\ldots(3)$$

Where $\|.\|$ is Euclidean norm. In the classical steepest descent method, the step size is obtained by carrying out an exact line search namely

$$\alpha_k = \arg\min_{\alpha} f(x_k + \alpha_k d_k) \quad \ldots\ldots\ldots(4)$$

However, despite the simplicity of the method and the optimal properties (3) and (4), the steepest descent method convergence slowly and is badly affected by ill-conditioning (Fletcher 1987), therefore not recommended for practical use. Different modifications are made to this method corresponding to different ways of choosing step size or modifying search directions.

The paper is organized as follows. In section 2 we review Barzilai and Borwein method. In section 3 steepest descent method with adaptive step size (SDAS). In section 4 problem reformulation and gradient flow algorithm are introduced. In section 5 our algorithm are derived and finally in section 6 numerical results are presented in order to compare the performance of the new algorithm.

2- Barzilai and Borwein (BB) method

Barzilai and Borwein in 1988 proposed two point step size gradient (BB) method (Barzilai and Borwein, 1988) by regarding $H_k = \lambda_k I$ as an approximation to the Hessian of f at x_k and imposing some–quasi–Newton property on H_k.

Denote $v_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$ by minimizing $\|v_{k-1} - H_k y_{k-1}\|$ they obtained

$$\lambda_k^{BB} = \frac{v_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}} \qquad \ldots\ldots(5)$$

With this, the method of Barzilai and Borwein is given by the following iteration scheme

$$x_{k+1} = x_k - \frac{1}{\lambda_k^{BB}} g_k \qquad \ldots\ldots\ldots\ldots\ldots\ldots(6)$$

The quantity given in (5) is frequently referred as a Rayleigh quotient. Indeed, if f is twice continuously differentiable, we have (Andrei 2005)

$$y_k = \left[\int_0^1 \nabla^2 f(x_k + tv_k)dt\right] v_k \ldots\ldots\ldots\ldots\ldots(7)$$

Therefore

$$\lambda_k^{BB} = v_k^T \left[\int_0^1 \nabla^2 f(x_k + tv_k)dt\right] v_k \Big/ v_k^T v_k \ldots\ldots\ldots(8)$$

Which lies between the largest and the smallest eigenvalue of the Hessian Average.

$$\int_0^1 \nabla^2 f(x_k + tv_k)dt$$

The scalar λ_k^{BB} has been already used as scaling factor in the context of limited memory quasi Newton algorithms see for example (Liu and Nocedal 1989) or conjugate gradient algorithms (Shanno and Phua 1980) and (Andrei 2005).

The BB method received a great deal of attention for its simplicity and numerical efficiency for well-conditioned problems, and analyzed by Raydan (Raydan, 1993), have a number of interesting feature that make them attractive for the numerical solution of (1). The most important features of this method is that only gradient directions are used, that the memory requirements are minimal and that they do not involve a decrease in the objective function, which allows fast local convergence. They have been applied successfully to find local minimizers of large scale real problems (Luengo and Raydan, 2003).

Raydan proved that for strictly convex function with any variable the (BB) method is globally convergence, despite of these advances of (BB) method on quadratic functions, still many open questions about this method on non-quadratic functions although

Fletcher (Fletcher, 2001) show that the method may very slow on some problems.

3- Steepest Descent with Adaptive Stepsize (SDAS)

In the scheme (2) the search direction d_k satisfied descent condition i.e

$$d_k^T g_k < 0 \qquad \text{.....(9)}$$

Which guarantees that d_k is a descent direction of $f(x)$ at x_k. In order to guarantee the global convergence it's usually required to satisfy the condition

$$g_k^T d_k \le -c\|g_k\| \qquad \text{.....(10)}$$

Where $c>0$ is a constant

Many procedures for step size computation α_k have been proposed for example minimizer rule, Armijo rule, limited minimization rule, and strong wolf rule which states that at the k-th iteration, α_k satisfies simultaneously

$$f(x_k + \alpha_k d_k) \le f(x_k) + \sigma_1 \alpha_k g_k^T d_k \text{ .....(11)}$$

And

$$g(x_k + \alpha_k d_k)^T d_k \ge \sigma_2\, g_k^T d_k \text{ .....(12)}$$

Where $\sigma \in (0, \frac{1}{2})$ and $\rho \in (\sigma, 1)$, and there are many other line search procedures for example see (Vrahatis et al 2000).

In the recent paper of zhen, (Zhen and Jie 2005)

An estimation of stepsize is introduced by means of lipschitz constant. As follows

For $k=0$ select $L_0 > 0$ and for $k \geq 1$

$$L_k = Max\left[L_{k-1}, \frac{\|y_{k-1}\|}{\|v_{k-1}\|} \right] \ldots\ldots\ldots(13)$$

And they proved that their algorithm is globally convergent and rate of convergence is linear (Zhen and Jie 2005), her we review briefly their algorithm.

Algorithm (A):

Step (1): $k = 0$ *choose* $x_0 \in R^n, \delta \in (0,2)$ *and* $L_0 > 0, \varepsilon = 10^{-6}$

Step (2): *if* $\|g_k\| < \varepsilon$ *then stop.else goto step* (3)

Step (3): *Estimate* L_k *from*(13)

Step (4): $x_{k+1} = x_k + \alpha_k d_k$ *where* $\alpha_k = \frac{\delta}{L_k}$

Step (5): $k = k + 1$ *goto step* (2)

In the above algorithm line search procedure is avoided at each iteration, which may reduce the cost of computation. However, we must estimate L_k at each iteration. Main draw back of algorithm (A) is, if L_k is very large then α_k will be very small and will slow the convergence rate of the method, on the other hand if L_k is very small

then α_k will be large and hence the method may fail to guarantee the global convergence.

4-Problem Reformulation and Gradient Flow Algorithm

For the unconstrained optimization problem given in equation (1), as we know a necessary condition for point x^* be an optimal solution is

$$g(x^*) = 0 \ldots\ldots\ldots(14)$$

This is a system in n Non-linear equations which must be solved to get the optimal solution x^*. In order to fulfill this optimality condition the following continuous gradient flow reformulation of the problems is suggested (khalaf and Al-Wagih, 2001). Solve the following system of ordinary differential equation

$$\frac{dx(t)}{dt} = -g(x(t)) \ldots\ldots\ldots(15)$$

With initial condition

$$x(0) = x_0 \ldots\ldots\ldots\ldots\ldots(16)$$

The solution of the system (15) with initial condition (16) is convergence to optimal solution which is minimum of the function given in (1) according to the following theorems, see (Andrei, 2003)..

Theorem (1):

Consider that x^* is a point satisfying (14) suppose that $G = \nabla^2 f(x^*)$ is positive definite, if x_0 is close enough to x^*, then $x(t)$ solution of (15) tends to x^* as $t \to \infty$.

Theorem (2):

Let $x(t)$ be the solution of (15), for fixed $t_0 \geq 0$ if $g(x(t)) \neq 0$ for all $t > t_o$, then $f(x(t))$ is strictly decreasing with respect to t for all $t > t_o$.

As we have seen solving the unconstrained optimization problem (1) has been reduced to that of integration of the ordinary differential equation (15) with initial condition (16). Andrei proposed an algorithm for solving the system (15) as follows

Let $0 = t_0 < t_1 < \ldots < t_k < \ldots$ be sequence of time points and consider

$h_k = t_{k+1} - t_k$ the sequence of time distance between two successive time points, consider the following (Andrei, 2003), discretization of (15)

$$\frac{x_{k+1} - x_k}{h_k} = -\left[(1-\theta)g_k + \theta g_{k+1}\right] \ldots\ldots\ldots (17)$$

where $\theta \in [0,1]$ is a parameter, then

$$x_{k+1} = x_k - h_k\left[(1-\theta)g_k + \theta g_{k+1}\right] \ldots\ldots\ldots (18)$$

It's clear that when $\theta = 0$ the above discretization is the explicit forward Euler's scheme. On the other hand, when $\theta = 1$ we have the implicit backward Euler's scheme. But

$$g_{k+1} = g_k + G_k v_k + o(\|v\|^2) \ldots\ldots\ldots(19)$$

Omitting the last term equation (19) and substituting in equation (18) we obtain

$$x_{k+1} = x_k - h_k[I + h_k \theta G_k]^{-1} g_k \ldots\ldots\ldots(20)$$

In fact the algorithm given in (20) introduced by Zghier but he derived it by using generalized trapezoidal rule with $\theta = 0.5$ see (Zghier 1981) or (Brown and Biggs 1989).

Many authors (for example Botsaris(1978), Brown(1986), and others solved systems(15) and (20) with initial condition (16) by some well known integration methods. Andrei (2003) showed if x_0 as initial guss close enough to x^* and if G_k is positive then the algorithm given in equation (20) is convergence for $\theta \in [0,1]$ and rate of convergence is linear when $\theta = 0$, super linear for $0 < \theta < 1$ and order of convergence is two if $\theta = 1$

5- Proposed Algorithm (Gradient Flow Steepest Descent GFSD say)

The method based on the equation (20) has quite good performance if G_k is positive definite and have desirable features, but not recommended for practical use, the major drawback of the algorithm is computing $(I + h\theta G_k)^{-1}$ at each iteration and

also there is no specified value of h. However one can deduce a simple implementation of the algorithm given (20) with preserving useful theoretical features as follows:

Let $G_k = \lambda_k^{BB} I$, where I_{nxn} Identity matrix as an approximation of G_k and put $h_k=L_k$ then substitute in (20)

$$x_{k+1} - x_k = -L_k \left[I + L_k \theta \lambda_k^{BB} I \right]^{-1} g_k$$

$$\frac{1}{L_k}\left(x_{k+1} - x_k\right) = -\left[I + \theta \frac{y_k^T y_k}{v_k^T v_k} \frac{v_k^T y_k}{y_k^T y_k} \right]^{-1} g_k$$

$$\frac{1}{L_k} v_k = -\left[1 + \theta \frac{v_k^T y_k}{v_k^T v_k} \right]^{-1} g_k$$

$$d_k = -\left[\frac{v_k^T v_k + \theta v_k^T y_k}{v_k^T v_k} \right]^{-1} g_k$$

$$d_k = -\left[\frac{v_k^T v_k}{v_k^T v_k + \theta y_k^T v_k} \right] g_k \qquad \text{.....(21)}$$

Therefore one can compute the new point from the following algorithm

$$x_{k+1} = x_k - \alpha_k g_k \quad \text{where } \alpha_k = \frac{v_k^T v_k}{v_k^T v_k + \theta y_k^T v_k} \qquad (22)$$

which is generalization of the BB method It's clear that if $\theta = 0$ the algorithm restarts with steepest descent direction. The convergence properties of the method given in (22) can be studied providing that $d_k = -g_k$ is descent direction and using the following proposition .

Proposition(1)

Assume that the step size α_k satisfies wolf conditions (11) and (12) and that d_k is descent direction then $y_k^T v_k > 0$.

for proof see (Barzilae and Borwein 1988) .

Theorem (3):

Suppose that f is bounded below in R^n and that f is continuously differentiable in neighborhood of the level set $L = \{x : f(x) \le f(x_0)\}$ - Assume also that the gradient g_k is lipchitz continuous i.e there exists a constant $c > 0$ s.t $\|g(x) - g(y)\| \le c\|x - y\| \ \forall x, y \in$

Consider any iteration of the form

$x_{k+1} = x_k + \alpha_k d_k$ where $\alpha_k \dfrac{v_k^T v_k}{v_k^T v_k + \theta y_k^T v_k}$ and $d_k = -g_k$ and α_k satisfies wolf conditions then $\lim_{k\to\infty} \|g_k\| = 0$.

proof :

from equation (12) we have

$(g_{k+1} - g_k)^T d_k \ge (\sigma_2 - 1) g_k^T d_k$ .....(23)

on the other hand ,the lipchitz condition

$(g_{k+1} - g_k)^T d_k \le \alpha_k c \|d_k\|^2$ .....(24)

from (23) and (24) we get

$$\alpha_k \ge \left(\frac{\sigma_2 - 1}{c}\right) \frac{g_k^T d_k}{\|d_k\|^2} \quad \text{.....(25)}$$

using equations (11) and (25) we have

$$f_{k+1} \le f_k + \sigma_1 (\frac{\sigma_2 - 1}{c}) \frac{(g_k^T d_k)^2}{\|d_k\|^2} \qquad (26)$$

now using the relation

$\|g_k\| \|d_k\| \cos\gamma_k = -g_k^T d_k$ where γ_k is the angle between g_k and d_k

then equation(26) can be written as

$$f_{k+1} \le f_k + t\|g_k\| \cos^2\gamma_k \qquad (27)$$

$$\text{where } t = \frac{\sigma_1(\sigma_2 - 1)}{c}$$

summing the expression in equation (27) and recalling f bounded below, we obtain $\sum \cos^2\gamma_k \|g_k\|^2 < \infty \qquad (28)$

assuming that $\cos^2\gamma_k > \delta > 0$ for all k , then we conclude that (Nocedal 1992)

$$\lim_{k\to\infty} \|g_k\| = 0$$

Out line of the algorithm (GFSD)

Step (1): $k = 0$ *choose* $x_0 \in R^n, \varepsilon > 0,\ \theta \in [0,1]; d_0 = -g_0$

Step (2): *if* $\|g_k\| < \varepsilon$ *stop.else goto step* (3)

Step 3):

Compute α_k from equation (22) and test for Wolfe conditions *if satisfied* accept α_k *as* stepsize else set $\alpha_k = 1$ and use backtraking to satisfy Wolf conditions

Step (4): $x_{k+1} = x_k - \alpha_k g_k$

Step (5): *compute* $\mathrm{g_k}, \mathrm{v_k}, \mathrm{y_k}$

Step (6): $k = k + 1$ *goto step* (2)

6-Numerical Results

We present the numerical results for the Barzilai-Borwein method and proposed (GFSD) method for some well known test functions (Bongartz and el at,1995), these algorithms are coded in double precision FORTRAN 90 language. The criteria for stopping the iterations are

$\|g_{k+1}\| < 10^{-6}$ or $\alpha_{k+1}\left|g_{k+1}^T g_{k+1}\right| < 10^{-20}|f_{k+1}|$

For both methods initial step size are computed by using backtracking procedure with $\sigma = 0.0001$ and $\rho = 0.8$. Also Wolfe conditions are used for accepting step size, the complete set of results are given (a) with $1000 \le \mathrm{n} \le 5000$ and table (b) with $6000 \le \mathrm{n} \le 10000$. In Tables (a) and (b) we present the comparison results of BB and GFSD methods for different dimensions with $1000 \le \mathrm{n} \le 5000$ consisting number of iteration (NOI), function gradient F&G ev evaluations (they are equal in these algorithms) and the execution time in nanosecond are compared it shown that the proposed algorithm is better than BB method in most cases. In Tables (a1) and (b1) we see that there is an improvement about % 5 in NOI , % 15.14 in F&G ev and %21 in execution time for

dimensions $1000 \le n \le 5000$. And %4 in NOI, %11 in F&G ev and %13 in execution time for dimensions with $6000 \le n \le 10000$. These algorithms are not compared with algorithm (A) since it shown by (Andrei,2005) BB method is better than algorithm (A).

Table(a) Comparison of BB method and GFSD with $1000 \le n \le 5000$

Test Fun.	N	Barzilai-Bowein Algorithm			GFSD Algorithm		
		NOI	F&Gev	Time	NOI	F&G ev	Time
Freudenstein & Roth	1000	218	1615	58	130	941	45
Extended Trigonometric	1000	44	148	25	48	147	26
Extended Rosenbrock	1000	248	1981	57	310	2435	65
Beal Fun.	2000	153	614	31	82	612	27
Penalty Fun.	1000	54	232	26	45	187	22
Raydan 1	1000	1134	5631	350	1130	4570	290
Raydan 2	2000	21	44	11	13	30	11
Ge. Tridigonal 1	1000	29	90	13	29	88	13
Extended Wood F	1000	1043	7083	201	1004	6251	181
DIXMAAN (cute)	2000	425	2335	297	381	2066	187
Freudenstein & Roth	3000	190	1410	70	128	932	45
Extended Trigonometric	4000	30	124	55	33	130	59
Extended Rosenbrock	5000	703	5544	491	473	3261	223
Beal Fun.	3000	152	614	79	83	612	39
Penalty Fun.	3000	46	219	12	46	207	10
Raydan 1	2000	1832	9419	1082	1913	8424	980
Raydan 2	5000	21	44	11	12	31	11
Ge. Tridigonal 1	3000	31	94	17	32	97	17
Extended Wood F	3000	1126	7545	511	994	6258	298
Dixmaan (cute)	4000	641	3499	701	642	3456	675
Total		8109	48285	68.3 sec	7719	40735	53.73 sec

Table(a1) Percentage of improving the GFSD with $1000 \le n \le 5000$

Tools	BB method	GFSD method
NOI	%100	%95.19
F&G EV	%100	%84.36
Time	%100	%78.62

Table(b) Comparison of BB method and GFSD with $6000 \le n \le 10000$

Test Fun.	N	Barzilai-Bowein Algorithm			GFSD Algorithm		
		NOI	FGEV	Time	NOI	F&G EV	Time
Freudenstein & Roth	6000	270	1995	197	130	935	92
Extended Trigonometric	7000	32	133	115	34	137	106
Extended Rosenbrock	9000	408	3183	391	450	3134	380
Beal Fun.	7000	46	608	137	108	301	91
Penalty Fun.	8000	55	241	30	56	237	29
Raydan 1	6000	4345	23262	8016	4561	21591	7478
Raydan 2	7000	20	38	07	21	44	06
Ge. Tridigonal 1	8000	37	102	22	30	85	16
Extended Wood F	9000	1094	7388	1238	932	5694	839
DIXMAAN (cute)	6000	697	3594	1772	614	3201	1650
Freudenstein & Roth	8000	295	2173	203	125	901	120
Extended Trigonometric	10000	32	137	148	33	138	150
Extended Rosenbrock	10000	527	4193	620	549	3742	507
Beal Fun.	10000	152	612	142	102	597	128
Penalty Fun.	9000	58	255	39	55	248	36
Raydan 1	10000	6301	33159	1732	6148	31082	17761
Raydan 2	10000	22	44	29	23	45	31
Ge. Tridigonal 1	10000	39	115	33	29	84	22
Extended Wood F	10000	1052	6978	1172	983	6052	983
Dixmaan (cute)	7000	624	3609	1796	512	3466	1730
Total		16206	91821	297.316 sec	15555	81714	257.58 sec

Table(b1) Percentage of improving the GFSD with $6000 \le n \le 10000$

Tools	BB method	GFSD method
NOI	%100	%95.8
F&G EV	%100	%88.9
Time	%100	%86.6

Conclusion

These types of algorithms are suitable for large-scale unconstrained optimization problems. Our numerical results indicates that there are an improvements of proposed algorithm especially on F&G EV I think which means that step size given by BB method is ether typically large or small and hence it require more functions and gradient evolutions to accept the step size to reduce in function value.

References:

1- Andrei. N "Gradient flow algorithm for unconstrained optimization"
Research Institute for informatics center for advanced Modeling
Optimization 8-10 Averescu Avenue, Bucharest, 2003
E-mail: nandrei@u3.ici.ro.

2- Andrei. N "Scaled conjugate Gradient algorithms for unconstrained Optimization"
ICI. Technical Report, Bucharest, 2005.

3- Barzilai. J and Borwein. M "Two Points stepsize Gradient Methods"
IMA J. Number Anal. , 8, 1988.

4- Brown. A and Batholomew-Biggs. C

"Some Effective Methods For unconstrained optimization Based on the Solution of systems of differential equations"

Journal of optimization theory and applications, Vol.62, no. 2, 1989.

5- Brown. A "Optimization Methods Involving The solution Of Ordinary

Differential Equations"

PhD Thesis, Hatfield Polytechnic. 1986

7- Botsaris. C "Aclass Of Methods For Unconstrained Minimization Based On Stable Numerical Integration Techniques"

Journal of Mathematical Analysis and Application 63. 1978

8- Bongartz, I., Conn. A, Gould. N and Toint. P

"CUTE: Constrained and unconstrained testing environments"

ACM Trans. Math Software, 1995.

9- Dai. Y., Yuan. J and Yuan. X

"Modified two point stepsize Gradient Methods for unconstrained Optimization"

Report No. ICM.98-044 July 1998

E-mail: dyh@Isec.cc.ac.cn

10- Fletcher. R

"Practical Methods of Optimization"

(2nd Edition). John Wiley Chichester 1987

11- Fletcher. R

"On the Barzilai-Borwein Method"

Numerical Analysis Report NA/207 October 2001

12- Khalaf. B and Al-Wagih. K

"Parallel shooting Method for unconstrained Numerical Optimization"

Raf. Jour. Sci. Vol.12, No.2, 2001.

13- Liu. D and Nocedal. J

"On the limited Memory BFGS Method for Large Scale Optimization"

Math. Programming, 45, 1989.

14- Luengo. F and Raydan. M

"Gradient Method With Dynamical Retards For Large-Scale Optimization Problems" Electronic Transaetions on Numerical Analysis vol(16), 2003 Kent state University

15- Raydan. M

"On the Barzilai-Borwein Choice of the step length for the gradient Method"

IMA J.Number Anal. 13, 1993.

15- Nocedal. J "Theory Of Algorithms For Unconstrained Optimization"

Acta Numerica pp 199-242, 1992

16- Shanno. F and Phua.H

"Remark on algorithm 500: minimization of unconstrained multivariate Functions"

ACM. Trans. Math. Software. 6, 1980.

17- Vrahatis. N. Androulakis. S and Lambrinos. N

"A class of gradient unconstrained Minimization Algorithms with Adaptive stepsize"

J. Computational and applied Mathematics, 114, 2000.

18- Zhen. J and Jie. S

"Stepsize Estimation for unconstrained Optimization Methods"

J. Computational and applied Mathematics, Vol.24,No.3, 2005.

19- Zghier. A

"The use of Differential Equations in Optimization"

phD thesis. Loughborough University, 1981.