A New Family of Spectral CG-Algorithm

Abbas Y. Al Bayati **Runak M. Abdullah**
profabbasalbayati@yahoo.com
College of Computer Sciences and Mathematics *College of Sciences*
University of Mosul, Iraq *University of Suleimani*
Received on: 07/01/2007 **Accepted on: 23/01/2007**

ABSTRACT

A new family of CG –algorithms for large-scale unconstrained optimization is introduced in this paper using the spectral scaling for the search directions, which is a generalization of the spectral gradient method proposed by Raydan [14].

Two modifications of the method are presented, one using Barzilai line search, and the others take $\alpha = 1$ at each iteration (where α is step-size). In both cases tested for the Wolfe conditions, eleven test problems with different dimensions are used to compare these algorithms against the well-known Fletcher –Revees CG-method, with obtaining a robust numerical results.

Keywords: Unconstrained optimization, spectral conjugate gradient method, inexact line search.

عائلة جديدة لخوارزميات التدرج المترافق الطيفي

عباس يونس البياتي **روناك محمد عبدالله**
جامعة الموصل/كلية علوم الحاسوب والرياضيات *جامعة السليمانية/كلية العلوم*
تاريخ استلام البحث: 2007/01/07 **تاريخ قبول البحث: 2007/01/23**

الملخص

تم اقتراح عائلة جديدة من خوارزميات التدرج المترافق في الامثلية غير المقيدة ذات القياس العالي التي تستخدم القياس الطيفي لخطوط البحث والتي هي توسيع للخوارزمية الطيفية المقترحة من قبل Raydan [14] .

تم تطوير العائلة بوسيلتين أحداهما باستخدام خط بحث Barzilai والآخرى باستخدام $\alpha = 1$ في كل خطوة تكرارية مع استخدام شرط Wolfe في الحالتين.

تم مقارنة أحدى عشرة دالة لاخطية بإبعاد مختلفة باستخدام العائلة الجديدة مقارنة مع الخوارزمية القياسية لـ FR مع الحصول على نتائج عددية ذات كفاءة عالية.

الكلمات المفتاحية: الامثلية غير المقيدة، طريقة التدرج المترافق الطيفي، خط بحث غير تام.

1. Introduction

Unconstrained optimization is one of the fundamental problems of numerical analysis with numerous applications.

The problem is the following:

For a function $f : R^n \to R$ and an initial point x_0 , find a point x^* (the minimizer of f) which minimizes the function $f(x)$, i.e.

$$\min_{x \in R^n} f(x) \qquad \ldots(1)$$

Usually x^* exists and is locally unique. It is a assumed that f is continuously differentiable for all k where k is the number of iterations. Methods for unconstrained optimization are generally iterative methods in which the user typically provides an initial estimate x_0 of x^* with possibly some additional information. A sequence of iterates $\{x_k\}$ is then generated according to some algorithm. Usually function values $\{f_k\}$ is monotonically decreasing (f_k denotes $f(x_k)$).

A well-known algorithm for solving problem given in equation(1) is the Steepest Descent method first proposed by Cauchy in 1874. The iterations are made according to the following equation:

$$x_{k+1} = x_k + \alpha_k d_k \quad , \quad k = 0,1,\ldots \qquad \ldots(2)$$

where $d_k = -g_k$ and α_k is a step-size, which is obtained by carrying out an exact line search. It's well-known that the negative gradient direction has the following optimal property (see [7]).

$$-g_k = \underset{d \in R^n}{Min} \underset{a \to 0^+}{Lim} [f_k - f[x_k + \frac{\alpha d}{\|d\|_2}]] \frac{1}{\alpha} \qquad \ldots (3)$$

Despite the simplicity of the method and the optimal property (3), the Steepest Descent method converges slowly and is badly affected by ill-conditioning (see [9] or [15]).

In 1988, a paper by Barzilai and Borwein [5] proposed a Steepest Descent method (the BB method) that uses a different strategy for choosing the step-size α_k along the negative gradient direction which is obtained from two point approximation to the secant equation underlying Quasi-Newton methods,

Considering $H_k = \gamma_k I_{nxn}$ as an approximation to the Hessian of f at x_k, they choose γ_k such that

$$H_k = \arg\min \|Hs_k - y_k\|_2 ,$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$, yielding (see[2] or [5]),

$$\gamma_k^{BB} = \frac{s_k^T y_k}{s_k^T s_k} \qquad \ldots(4)$$

with these, the method of Barzilai and Borwein is given by the following iterative scheme:

$$x_{k+1} = x_k - \alpha_k g_k \qquad \dots(5)$$

where $\alpha_k = \frac{1}{\gamma^{BB}}$

The scalar γ^{BB} has been already used as scaling factor in the Quasi-Newton algorithms or Conjugate Gradient algorithms (see[4] and [11]).

The BB method has been shown to converge [14] and it's convergence is linear [13], despite at these advances of BB method on quadratic functions, still there are many open questions about this method on non-quadratic functions although Fletcher [9] shows that the method be very low on some test functions.

In recent paper Abbo [1] proposed a modification of BB by the following way [1].

Let $G_k = \gamma_k^{BB} I_{nxn}$

where I is the identity matrix as an approximation of Hessian matrix G_k, from convex combination of forward and backward Euler's scheme

$$x_{k+1} = x_k - h_k[(1-\varepsilon)g_k + \varepsilon g_k],\ 0 \le \varepsilon \le 1,\ h \text{ is a step-size} \qquad \dots(6)$$

and using Taylor's series for g(x) about x_{k+1}, i.e.

$$g_{k+1} = g_k + G_k s_k + o(\|s\|^2) \qquad \dots(7)$$

2. Conjugate Gradient Method (CG-Methods)

Conjugate Gradient Methods depend on the fact that for quadratic function, if we search along a set of n mutually conjugate directions d_k, $k = 1,2,...,n$, then we will find the minimum in at most n steps if line searches are exact. Moreover, if we generate this set of directions by known gradients, then each direction can be simply expressed as

$$d_0 = -g_0 \qquad \dots(8)$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k \qquad \dots(9)$$

where β_k can be calculated by

$$\beta_{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \qquad \dots(10)$$

$$\beta_{perry} = \frac{(y_k - s_k)^T g_{k+1}}{s_k^T y_k} \qquad \ldots(11)$$

All these β_k's are equivalent on quadratic function with exact line searches and starting with steepest descent direction, but when extended to general non-linear functions, the conjugate gradient algorithm with different β are quite different in efficiency. Formula (11) gives better algorithms than (10) in practice , a reason for this is given by Powell [13]. One of the reasons for the inefficiency of CG-method is that none of the β in (10) and (11) takes into consideration the effect of inexact line searches [10]. To overcome this drawback some authors proposed the so called spectral conjugate gradient methods (see for example [3],[6]).

Birgin and Martinez in [6] introduced an spectral conjugate gradient (SCG), in which the search directions are generated by

$$d_k = -\theta_k g_k \quad , \ k = 0$$
$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k \qquad \ldots(12)$$

where $$\theta_{k+1} = \frac{s_k^T s_k}{s_k^T y_k} \qquad \ldots(13)$$

and $$\beta_k = \frac{(\theta_k y_k - s_k)^T g_{k+1}}{s_k^T y_k} \qquad \ldots(14)$$

For if $\theta_k = 1$ this formula was introduced by Perry in [12], if we assume that

$s_j^T g_{j+1} = 0 \quad , \quad j = 0,1,\ldots,k$ then

$$\beta_k = \frac{\theta_k y_k^T g_{k+1}}{\alpha_k \theta_k g_k^T g_k} \ . \qquad \ldots(15)$$

Finally, assuming that the successive gradients are orthogonal, we obtain the generalization of FR formula:

$$\beta_k = \frac{\theta_k g_{k+1}^T g_{k+1}}{\alpha_k \theta_k g_k^T g_k} \qquad \ldots(16)$$

In fact, SCG algorithm is a generalization of the Raydan [14] spectral gradient algorithm defined by

$$d_k = -\theta_k g_k \qquad \ldots(17)$$

where θ as in (13).

3. Outlines of the spectral CG-algorithm algorithm

Let $x_0 \in R^n$, , $d_0 = -g_0$, $k = 0$, $\alpha_0 = 1$

Step(1) : if $g_k = 0$ stop, otherwise go to step(2)

Step(2) : compute

$$\alpha_k = \frac{\alpha_{k-1}\|d_{k-1}\|}{\|d_k\|} \qquad \ldots(18)$$

such that Wolfe-condition is satisfied and hence a new x_{k+1} is computed

Step(3) : compute θ_{k+1} by (13) and β_k by (15) or (16) and define

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k$$

Step(4) : If $d_k^T g_{k+1} \le -10^{-3}\|d_k\|_2 \|g_{k+1}\|$ …(19)

then set $d_{k+1} = d_k$ else $d_{k+1} = -\theta g_{k+1}$

Step(5) : k=k+1 go to step(1)

4. New family of SCG methods (NSCG say)

In [10] Birgin gives a nice comparison by asking the following questions:

1- Is the choice (13) better than $\theta = 1$?
2- Which is the best choice of β_k among (15) and (16)?
3- Which is the best choice of α_k?

According to these inquires let us consider the following:
From the last term in (7) and substituting in (6) we obtain

$$x_{k+1} - x_k = -h_k[(1-\varepsilon)g_k + \varepsilon(g_k + G_k s_k)]$$

$$s_k = -h_k[g_k + \varepsilon G_k s_k]$$

$$s_k + \varepsilon h_k G_k s_k = -h_k g_k$$

$$(I + \varepsilon h_k G_k)s_k = -h_k g_k$$

$$\frac{x_{k+1} - x_k}{h_k} = -(I + \varepsilon h_k G_k)^{-1} g_k \qquad \ldots(20)$$

Let $L_k = \dfrac{\|g_{k+1} - g_k\|^2}{\|x_{k+1} - x_k\|^2}$, Lipschitz constant , let $G_k = \lambda_k I$ where I is $n \times n$ identity matrix and put $h_k = L_k$ in (20)

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

$$\frac{1}{L_k}(x_{k+1}-x_k)=-[I+\varepsilon\frac{y^Ty}{s^Ts}.\frac{s^Ty}{y^Ty}]^{-1}g_k$$

$$\frac{1}{L_k}s_k=-[\frac{s_k^Ts_k}{s_k^Ts_k+\varepsilon s_k^Ty}]g_k$$

$$\because\quad d_k=-\frac{s_k^Ts_k}{s_k^Ts_k+\varepsilon s_k^Ty}g_k$$

$$x_{k+1}=x_k+d_k$$

$$\text{where}\,\theta=\frac{s^Ts}{s^Ts+\varepsilon s^Ty} \qquad \ldots(21)$$

From (21) it is clear that setting $\varepsilon=0$ this gives $\theta=\frac{s^Ts}{s^Ts}=1$, this will answer one of the inquiries of Birgin. Also taking $\varepsilon=1$ will give $\theta=\frac{s^Ts}{s^Ts+s^Ty}$. To answer the 2nd inquiry, it is clear that β_k in (14) is very effective since the line search which is used in this paper is not exact. To answer the 3rd inquiry we suggest a new hybrid computations for the scalar α as shown in step(2) from the new algorithm.
We are going to list outlines of the new proposed algorithm (NSCG).

4.1 Outline of the algorithm (NSCG)

Let $x_0\in R^n$, $0<\sigma<\gamma<1$, $d_0=-g_0$, $k=0$
Step(1) : if $g_k=0$ stop, else go to step(2)
Step(2):First compute $\alpha_k=1$ and second compute

$$\alpha_k=\begin{cases}1 & k=0\\ \frac{\alpha_{k-1}\|d_{k-1}\|}{\|d_k\|} & k>0\end{cases}$$

Such that $f(x_k+\alpha_kd_k)\le f(x_k)+\sigma\alpha_kg_k^Td_k$ …(22)

And $g_{k+1}^Td_k\ge\gamma g_k^Td_k$ …(23)

$$x_{k+1}=x_k+\alpha_kd_k$$

Step(3) : compute θ by (21) and β_k by (16) and define

$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k$

Step(4) : If $d_k^T g_{k+1} \le -10^{-3} \|d\|_2 \|g_{k+1}\|$

then $d_{k+1} = d_k$ else $d_{k+1} = -\theta_{k+1} g_{k+1}$

Step(5) : k=k+1 go to step(1)

4.2 Some theoretical results

4.2.1 Theorem:

If α_k satisfies Wolf condition defined by (22) and (23) then the search direction will be descent , i.e. $y_k^T s_k > 0$.
For proof see [5].

4.2.2 Theorem:

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. $\|g(x) - g(y)\| \le c\|x - y\| \quad \forall x, y \in R^n$.

Consider any iteration of the form

$x_{k+1} = x_k + \alpha_k d_k$ where $\alpha = 1$ and if $d_k = -g_k$ and α_k satisfies Wolfe conditions defined in (22) and (23) then $\lim_{k \to \infty} \|g_k\| = 0$.

Proof : From equation (22) we have $(g_{k+1} - g_k)^T d_k \ge (\sigma_2 - 1) g_k^T d_k$ …(24)
on the other hand, the lipchitz condition $(g_{k+1} - g_k)^T d_k \le \alpha_k c \|d_k\|^2$ …(25)
from (24) and (25) we get $\alpha_k \ge \left(\frac{\sigma_2 - 1}{c}\right) \frac{(g_k^T d_k)^2}{\|d_k\|^2}$ …(26)

using equations (22) and (26) we have $f_{k+1} \le f_k + \sigma_1 (\frac{\sigma_2 - 1}{c}) \frac{(g_k^T d_k)^2}{\|d_k\|^2}$ …(27)

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 the equation (27) can be written as $f_{k+1} \le f_k + t\|g_k\| \cos^2 \gamma_k$ …(28)
where $t = \frac{\sigma_1(\sigma_2 - 1)}{c}$ and $\sigma_1, \sigma_2 \in (0, \frac{1}{2})$

summing the expression in equation (28) and since f is bounded below, we obtain

$$\sum \cos^2 \gamma_k \|g_k\|^2 < \infty \qquad \ldots(29)$$

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

$$\lim_{k\to\infty} \|g_k\| = 0 \qquad \ldots(30)$$

5. Numerical results

The comparative test involves eleven well-known standard test functions(given in the appendix) with different dimensions. The results are given in the Table(1) is specifically quoting the number of function evaluations (NOF) . All programs are written in FORTRAN 90 language and for all cases the stopping criterion is taken to be $\|g_{k+1}\| < 1x10^{-5}$. The results are given in table (1):

Table (1)
Comparison results between the new (NSCG)
and Birgin spectral standard SCG for β_{FR}

		New (SCG)		Standard SCG)	
Test Function	**N**	$\alpha_k = \frac{\alpha_{k-1}\|d_{k-1}\|}{\|d_k\|}$	$\alpha_k = 1$	$\alpha_k = \frac{\alpha_{k-1}\|d_{k-1}\|}{\|d_k\|}$	$\alpha_k = 1$
		f & g Eva	f & g Eva	f & g Eva	f & g Eva
Extended Trigonometric	1000	44	35	44	32
	5000	101	40	99	161
	10000	86	40	86	152
Extended Rosenbrock	1000	59	92	64	121
	5000	60	92	64	106
	10000	64	99	64	105
Perturbed Quadratic	1000	662	431	513	364
	5000	1239	833	1351	938
	10000	1504	1198	1703	2001
Raydan 1	1000	1619	622	636	591
	5000	#	873	#	2327
	10000	#	#	#	#
Diagonal 2	1000	307	388	337	344
	5000	486	826	747	830
	10000	522	1189	33[illegible]7	1200
Generalized Tridiagonal-1	2000	54	42	320	469
	5000	429	185	732	537
	10000	1324	325	1666	321
Extended Three Exponential Terms	3000	1769	116	5663	195
	4000	1911	181	1524	425

	10000	2634	438	4364	768
Generalized PSC1	5000	#	#	#	#
Extended Powell	1000	172	152	147	590
	3000	146	179	183	2062
	5000	158	155	178	712
Extended Maratos	1000	37	141	43	46
	6000	37	141	39	307
	10000	98	141	422	310
Extended Wood	1000	184	71	184	81
	5000	192	71	202	79
	10000	178	74	188	83
Total		16076	9170	24970	16257

From Table (1) taking the standard Birgin (SCG) as %100 NOF we can get the following values.

Table(2)

	NOF+NOG	
	$\alpha_k = \frac{\alpha_{k-1}\lVert d_{k-1}\rVert}{\lVert d_k\rVert}$	$\alpha_k = 1$
Standard SCG	100%	100%
New SCG	64%	56%

From table (2) it is clear that the new proposed algorithm with it's both versions has an improvements of about (33-36)% NOF according to our selected number of test functions.

6. Appendix :

All the test functions used in this paper are from general literature:

1. Extended Trigonometric Function

   f(x) = $\sum_{i=1}^{n}((n-\sum_{j=1}^{n}\cos x_j)+i(1-\cos x_i)^2$, $x_0=[0.2,0.2,\ldots,0.2]^T$

2. Extended Rosenbrock Function

   f(x) = $\sum_{i=1}^{n/2}c(x_{2i}-x_{2i-1}^2)^2+(1-x_{2i-1})^2$, $x_0=[-1.2,1,..,-1.2,1]^T$

3. Perturbed Quadratic Function

   f(x) = $\sum_{i=1}^{n}ix_i^2+\frac{1}{100}(\sum_{i=1}^{n}x_i)^2$, $x_0=[0.5,0.5,\ldots,0.5]^T$

4. Raydan1 Function

   f(x) = $\sum_{i=1}^{n}\frac{i}{10}(\exp(x_i)-x_i)$, $x_0=[1,1,\ldots,1]^T$

5. Diagonal2 Function

   f(x) = $\sum_{i=1}^{n}(\exp(x_i)-\frac{x_i}{i})$, $x_0=[1/1,1/2,\ldots,1/n]^T$

6. Generalized Tridigonal-1 Function

   f(x) = $\sum_{i=1}^{n-1}(x_i+x_{i+1}-3)^2+(x_i-x_{i+1}+1)^4$, $x_0=[2,2,\ldots,2]^T$

7. Extended Three Exponential Terms

   f(x) = $\sum_{i=1}^{n/2}(\exp(x_{2i-1}+3x_{2i}-0.1)+\exp(x_{2i-1}-3x_{2i}-0.1)+\exp(-x_{2i-1}-0.1)$,

   $x_0=[0.5,0.5,\ldots,0.5]^T$

8. Generalized PSC1 Function

   f(x) = $\sum_{i=1}^{n-1}(x_i^2+x_{i+1}^2x_ix_{i+1})^2+\sin^2(x_i)+\cos^2(x_i)$, $x_0=[3,0.1,\ldots,3,0.1]^T$

9. Extended Powell Function

   f(x) =

   $\sum_{i=1}^{n/4}(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+(x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4$,

   $x_0=[3,-1,0,1,\ldots,3,-1,0,1]^T$

10. Extended Maratos Function

$$f(x) = \sum_{i=1}^{n/2} x_{2i-1} + c(x_{2i-1}^2 + x_{2i}^2 - 1)^2 \quad , \quad x_0 = [1.1, 0.1, \ldots, 1.1, 0.1]^T$$

11. Extended Wood Function

$$f(x)= \sum_{i=1}^{n/4} 100(x_{4i-3}^2 - x_{4i-2})^2 + (x_{4i-3} - 1)^2 + 90(x_{4i-1}^2 - x_{4i})^2 + (1 - x_{4i-1})^2 +$$

$$10.1\{(x_{4i-2} - 1)^2 + (x_{4i} - 1)\} + 19.8(x_{4i-2} - 1)(x_{4i} - 1) \quad ,$$

$$x_0 = [-3, -1, -3, -1, \ldots, -3, -1, -3, -1]^T$$

REFERENCES

[1] Abbo, K. " Modifying of Barzilai and Borwein Method for solving Large scale Unconstrained Optimization", accepted for publication in **Iraqi journal of Statistical Sciences**, 2006

[2] Andrei, N. "A New Gradient Descent Method for Unconstrained Optimization" **Research Institute Informatics** AMS 65F30. Bucharest 2002

[3] Andrei, N. "A New Gradient Descent Method for Unconstrained Optimization" **ICI. Technical Report.** Bucharest 2004.

[4] Andrei, N. "Scaled Conjugate Gradient Algorithm for Unconstrained Optimization" **ICI. Technical Report.** Bucharest 2005.

[5] Barzilai, J. and Borwein.M " Two Points Step-size Gradient Methods" **IMA J. Number Anal.** ,9, 1988.

[6] Bergin, E. and Martinez. M " a Spectral Conjugate Gradient Method for Unconstrained Optimization Applied Math. And Optimization, 43, 2001.

[7] Dai, Y. and Ynan. X. " Modified Two Point Step-size Gradient Methods for Unconstraine Optimization" **Report No. ICM.** 98-044 July-1988.

[8] Dai, Y. and Liao. L " R- linear Convergence of Barzilai and Borwein Gradient Method" Research Report, 1999 (Accepted by **IMA J.** Number Anal.).

[9] Fletcher, R. "Practical Methods of Optimization" (2nd Edition). John Wiley Chichester 1987.

[10] Hu, Y. and Storeg. C " On Unconstrained Conjugate Gradient Optimization Methods and Their Interrelationship Mathematics Report A 129, July 1990.

[11] Lin, D. and Nocedal. J " On the Limited Memory BFGS Method for Large Scale- Optimization" **Math. Programming** , 95, 1989.

[12] Perry, A. " A Modified Conjugate Gradient Algorithm" operations Research 26, 1978.

[13] Powell. M. " Restart Procedures for the Conjugate Gradient Method" **Math. Programming** , Vol (12), 1977.

[14] Raydan, M. "On the Barzilai and Borwein Choice of steplength for the Gradient Method", Research Report, 1999 (accepted by **IMA J**. Number Anal.).

[15] Zhen, J. and Jie. S " Step-size Estimation for Unconstrained Optimization Methods" Computational and Applied Mathematics Vol. (24) No.(3), 2005.