A Combined Cubic and Novel Line Search CG-Algorithm

Abbas Y. Al-Bayati **Hamsa TH. Chilmeran**
College of Computer Sciences and Mathematics
University of Mosul

Received on: 10/12/2006 **Accepted on: 15/08/2007**

الملخص

في هذا البحث تم استخدام خوارزمية جديدة من خوارزميات التدرج المترافق في الأمثلية غير المقيدة والمتمثل بربط تقنيتين من تقنيات خط البحث (Cubic and Novel). أن التقنية الجديدة تم مقارنتها مع خوارزمية التدرج المترافق التي تستخدم Cubic Interpolation وبصورة عامة الخوارزميات المقترحة في هذا البحث ذات فعالية عالية عند مقارنتها مع مثيلاتها من الخوارزميات السابقة في مسائل الأمثلية المقيدة.

ABSTRACT

In this paper a new line search technique is investigated. It uses (cubic and novel) line searches in the standard CG-algorithm for unconstrained optimization. Applying our new modified version on CG-method shows that, it is too effective when compared with other established algorithms, in this paper, to solve standard unconstrained optimization problems.

1. Introduction.

The Conjugate Gradient (CG) method is particularly useful for minimizing function of many variables because it does not require the storage of any matrices. However, the rate of convergence of the algorithm is only linear unless the iterative procedure is "restarted" occasionally. At present it is usual to restart the CG- algorithm every n or (n+1) iterations, where n is number of variables, but it is known that frequency of restarts should depend on the objective function. CG-algorithm is useful to find minimum of a function $f : R^n \to R$, In general, the method has the following form

$$d_{k+1} = \begin{bmatrix} -g_k & for \quad k=1 \\ -g_{k+1} + \beta_k d_k & for \quad k>1, \end{bmatrix}$$

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

where g_k denotes the gradient $\nabla f(x_k)$, d_k is the search direction, λ_k is a step-length obtained by a line search, and β_k is chosen so that it becomes the k-th conjugate direction when the function is quadratic and the line search is exact (Sun, et al, 2003), a Well Known formula for β_k is given by:

$$\beta_k = \frac{y_k^T g_{k+1}}{d_k^T y_k} \text{ (Hestenes \& Stiefel, 1952)} \qquad \ldots(1.a)$$

$$\beta_k = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \text{ (Fletcher \& Reeves, 1964)} \qquad \ldots(1.b)$$

$$\beta_k = \frac{g_{k+1}^T\left(g_{k+1} - g_k\right)}{\|g_k\|^2} \text{ (Polak \& Ribiere, 1969)} \qquad \text{…(1.c)}$$

$$\beta_k = \frac{-\|g_{k+1}\|^2}{d_k^T g_k} \text{ (Dixon, 1975)} \qquad \text{…(1.d)}$$

$$\beta_k = \frac{-g_{k+1}^T y_k}{d_k^T g_k} \text{ (Al-Bayati \& Al-Assady, 1986)} \qquad \text{…(1.e)}$$

where $y_k = g_{k+1} - g_k$ …(2)

The successive direction are conjugate vectors for successive gradients obtained as the method progresses. At step k one evaluates the current negative gradient and adds it to a linear combination of the pervious direction vectors to obtain a new conjugate direction along which to move (Cristain, 2003).

2. Conjugate Gradient Algorithm with a Novel Line Search:

Let $f = f(x_k)$ be continuously differentiable of $x \in R^n$ with gradient $g(x) = \nabla f(x)$. The problem of finding a minimizer of f is often investigated by a CG-algorithm $x_{k+1} = x_k + \lambda_k d_k$, where λ_k is the optimal step size, computed by a novel line search technique (Sun, et al, 2003).

2.1 The One – Dimensional Novel Line Search Procedure:

The following steps describe novel line search technique:

Step 1: Let $f(x)$ be any unimodal function. Set $x_1 = a, f_1 = f(x_1)$ and $k = 1$, ε is any small parameter. Set $x_2 = T_0, f_2 = f(x_2)$, where T_0 is the initial step length $0 < T_0 < 1$ and a is assumed to be available; $a = 0.0$ is normally taken.

Step 2: If $f_2 \geq f_1$ then, set $T_0 = T_0 / 3$ and go to step (1) else go to the next step (3).

Step 3: a) Set $k = k + 1$

b) $f_k = f(x_k)$, where $x_k = 2T_0$

c) if $f_{k+1} \geq f_k$, then go to step 4 else set $T_0 = 2T_0$ and go to position a.

Step 4: Steps (1) to (3) give the initial interval of uncertainty. Let:

$y_1 = x_{k-2}$ $\qquad f_{y_1} = f_{k-2}$

$y_3 = x_{k-1}$ $\qquad f_{y_3} = f_{k-1}$

$y_5 = x_k$ $\qquad f_{y_5} = f_k$,

also let $f_s = f_{y_3}$

Step 5: Obtain two more points in this interval of uncertainty as follows:

$$y_2 = y_3 - \left|(f_{y_5} - f_{y_3}) / f_{y_5}\right| [(y_5 - y_3 / y_5]; \quad f_{y_2} = f(y_2)$$

and

$$y_4 = y_3 - \left|(f_{y_1} - f_{y_3}) / f_{y_1}\right| [(y_3 - y_1 / y_3]$$

Now there are five points y_1, y_2, y_3, y_4 and y_5 . if the value of points are not in the increasing order, order them such that $y_1 < y_2 < y_3 < y_4 < y_5$ and $f_{y_1}, f_{y_2}, f_{y_3}, f_{y_4}$, and f_{y_5} are their corresponding function values. choose a point , which has the lowest function value among the five points and set it as $\widetilde{B}$. Let $f_{\widetilde{B}}$ be its corresponding function value. Take adjacent point of $\widetilde{B}$ and set them as $\widetilde{A}$ and $\widetilde{C}$, such that: $\widetilde{A} < \widetilde{B} < \widetilde{C} \rightarrow f_{\widetilde{A}} \geq f_{\widetilde{B}} \leq f_{\widetilde{C}}$; $X_0 = [(\widetilde{A}^2 + \widetilde{B}^2 + \widetilde{C}^2)/3]^{1/2}$; $f = f(X_0)$. Now we have four points $X_0, \widetilde{A}, \widetilde{B}, \widetilde{C}$ and their respective function values $f_0, f_{\widetilde{B}}, f_{\widetilde{A}}, f_{\widetilde{C}}$. Again choose three appropriate points and set them as y_1, y_3 and y_5 such that $y_1 < y_3 < y_5 \Rightarrow f_{y_1} \geq f_{y_3} \leq f_{y_5}$; $f_{y_1}, f_{y_3}, f_{y_5}$ their respective function values.

Step 6: If the inequality $|(f_s - f_0)/f_0| \leq \varepsilon$, is satisfied , then the search is terminated with $x^* = y_3$ else $f_0 = \min(f_s, f_0)$ and go to (5). The ε parameter control the accuracy of the solution – desired. it is value may be recommend a value of $\varepsilon = 10^{-5}$ for most the usually encountered problems (Rao and Chandra, 1983).

3. Cubic Interpolation Technique:

In the proposed cubic polynomial method , a gradient and a function evaluation are made at every iteration at k .At each iteration an update is performed when a new point x_{k+1} that satisfies the condition $f(x_{k+1}) \leq f(x_k)$ in many of the n-dimensional routines, and the values of the first derivative are calculated. It may be desirable to use this information. An efficient algorithm based on fitting a cubic equation through the data at two points is often using this algorithm.

To start the procedure, point (a) is selected. The derivative is evaluated and a step h taken in the direction of decreasing f The derivative $\tilde{f}(a+h)$ is then evaluated. If $f'(a+h)$ has the same sign $f'(a)$ then the point $(a+h)$ is chosen as the new origin. The step size h is doubled and the step repeated. If $f'(a+h)$ is zero or has changed sign, then a cubic is fitted to cubic and may be written as

$$f(x) = Ax^3 + Bx^2 + Cx + D \qquad \ldots(3)$$

They may be found by solving a set of four equations, for convenience, the origin will be taken at the point a

$$f(a) = D \qquad \ldots(4)$$

$$f(b) = Ax^3 + Bx^2 + Cx + D \qquad \ldots(5)$$

$$v = f'(a) = C \qquad \ldots(6)$$

$$U = f'(b) = 3Ax^2 + 2Bx + C \qquad \ldots(7)$$

At the required solution:

$$f'(x) = 3Ax^2 + 2Bx + C = 0 \qquad \ldots(8)$$

And

$$f''(x) = 6Ax + 2B\phi 0 \qquad \ldots(9)$$

The solution of above equation are

$$x = (-B\mu\sqrt{B^2 - 3AC})/3A \qquad \text{if } A \neq 0 \qquad \ldots(10)$$

$$x = -C/2B \qquad \text{if } A = 0 \qquad \ldots(11)$$

And , of the former of these , only the positive singe satisfies equation (9)
Let us define:

$$s = -3(f(a) - f(b)0/b = 3(Ab^2 + Bb + C) \quad \ldots(12)$$

$$\hat{z} = s - u - v = Bb + C \quad \ldots(13)$$

$$w^2 = \hat{z}^2 - uv = b^2(B^2 - 3AC) \quad \ldots(14)$$

Then the most readily obtainable solution is obtained from:

$$v - \hat{z} + w = b(-B\mu\sqrt{B^2 - 3AC}) \quad \ldots(15)$$

And

$$u + v - \hat{z} = 3Ab^2 \quad \ldots(16)$$

And the minimum is given by

$$x = b(v - \hat{z} + w)/(u + v - \hat{z}) \quad \ldots(17)$$

This solution is not acceptable, however, as when $A = 0$, it becomes indeterminate. Davidon (1959) therefore replaces it by:

$$x = b\left(1 - \frac{u + w + \hat{z}}{u - v + 2w}\right) \quad \ldots(18)$$

In an n-dimensional problem. It is usual to accept the point x predicted by equation (18) as the minimum provided $f(x) < f(a)$ and $f(x) < f(b)$. To fit the cubic we will use the function and its gradient at two points as a, $f(a)$, $f'(a)$, b, $f(b)$, $f'(b)$ (Sun, J., etc, 2003, 2005).

3.1 Outline of the Cubic Technique (Bundy, 1984):

Step 1: Set a_1, h, ε.

Step 2: Find $v = f'(a)$ and if $f'(a) > 0$ then $h = -h$ and go to step (3); else set $h = +h$.

Step 3: Find $b_1 = a_1 + h$; compute $u = f'(b)$.

Step 4: If $uv < 0$, then go to step (5); else set $h = 2h,\ v = u,\ a_1 = b_1$ and go to step (3).

Step 5: For $k = 1,2,........,$ repeat.

Step 6: Set $z = 3/(b_k - a_k)(f_a - f_b) + u + v$

$w = (z^2 - uv)^{1/2}$

$x^* = a_k + (b_k - a_k)\{1 - (u - +w - z/u - v + 2w)\}$

and evaluate $f_x = f(x^*),\ g_x = g(x^*)$.

Step 7: If $g_x < 0$ *or* $f_x < f(a_x)$, then set $a_{k+1} = x^*$, $b_{k+1} = b_k$ and $f_a = f_x, f'_a = g_x$ go to step (9), else set $a_{k+1} = a_k$, $b_{k+1} = x^*$ and $f_b = f_x$, $f'_b = g_x$.

Step 8: If $b_{k+1} - a_{k+1} \le \varepsilon$, then x^* is the root, else go to step (5).

4. A new combined Novel and Cubic Line Searches:

This new technique combines between the cubic interpolation & Novel line searches to find the minimum of multi-dimension function subject to necessary condition $f'(x) = 0$ and sufficient condition $f(x_{k+1}) < f(x_k)$.

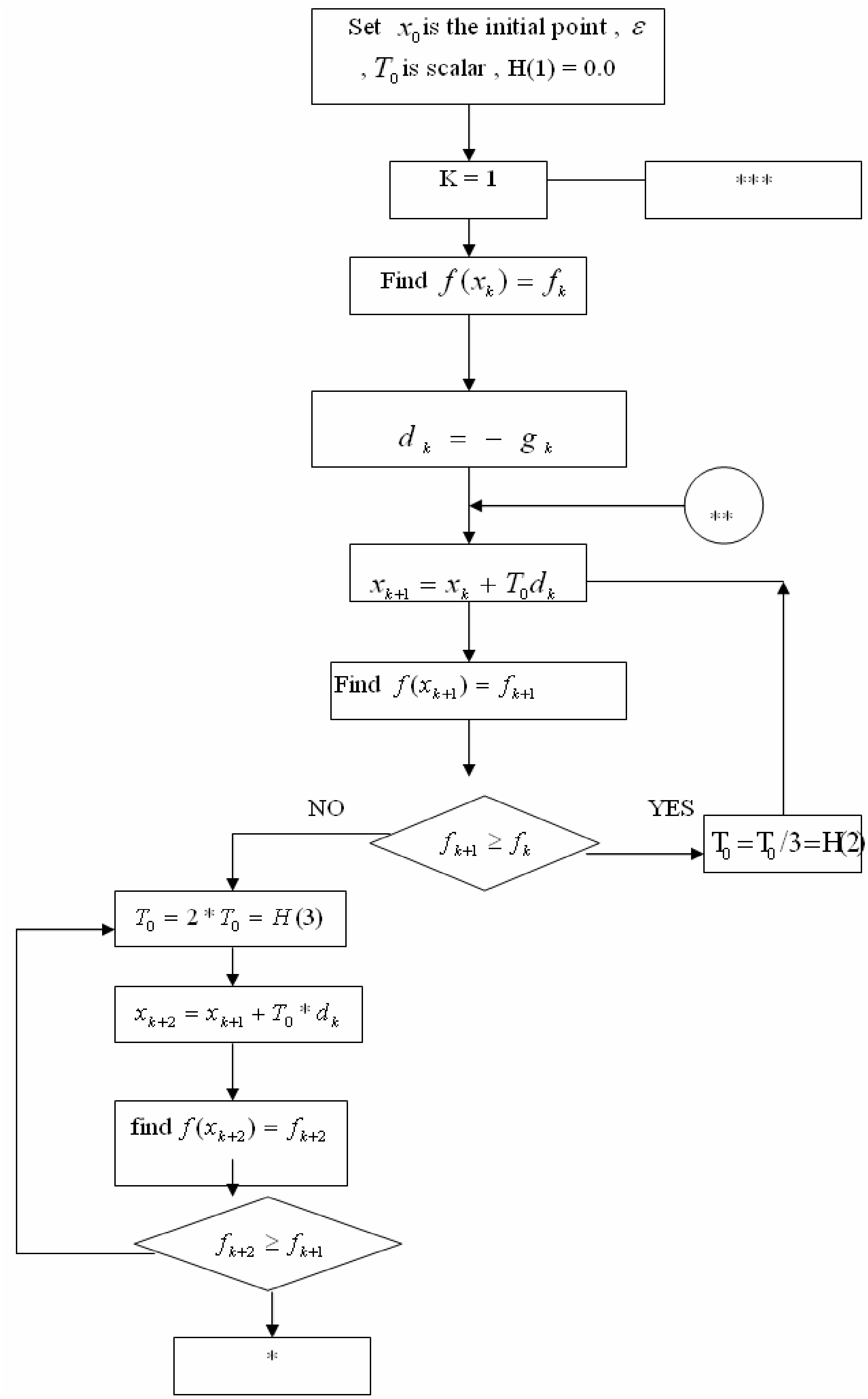
4. 1 Outline of the Combined Technique are as follows:

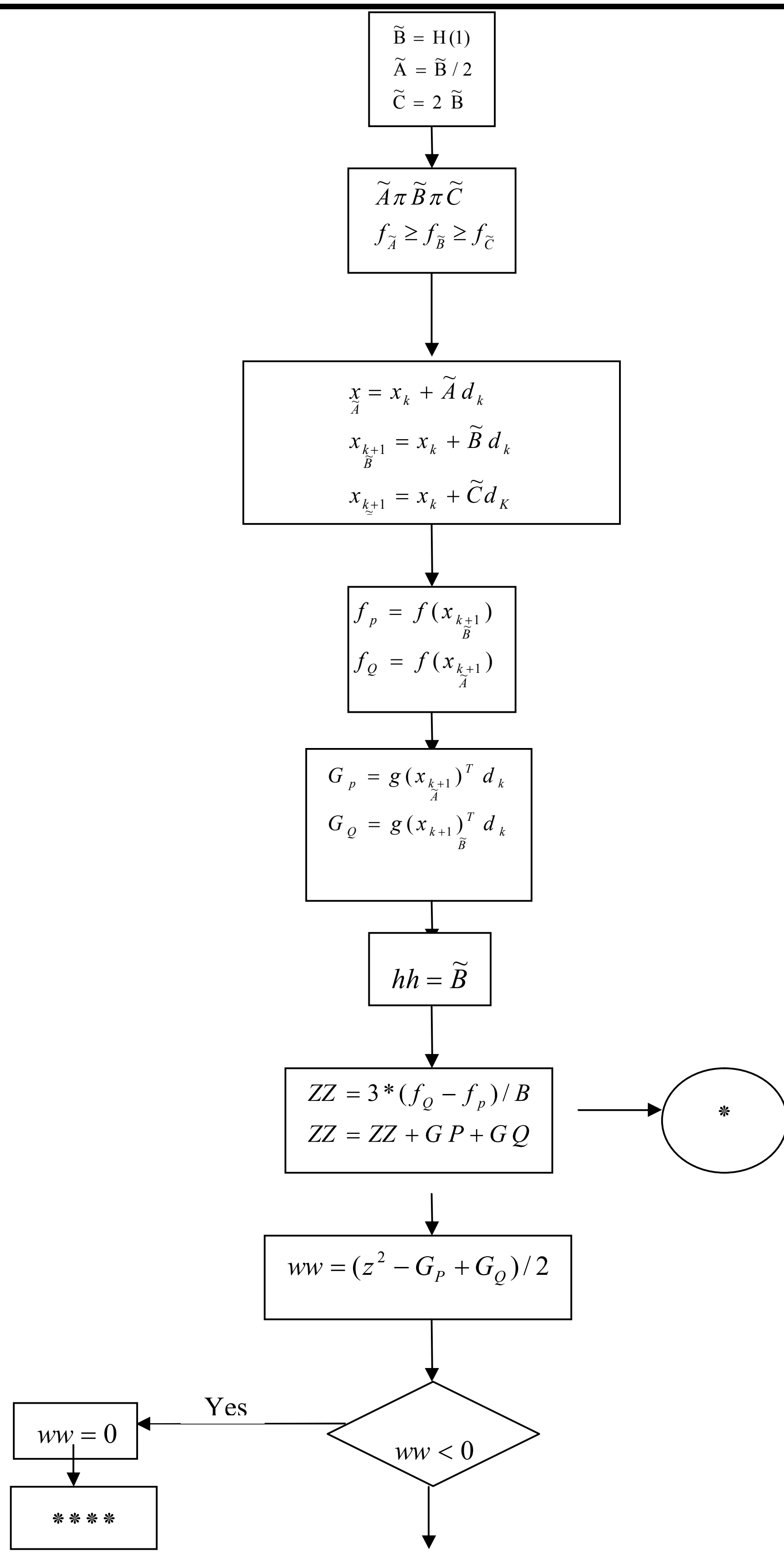
Step 1: Let x_0 be the initial point; set ε, $H(1)=0$, $T_0=1$.
Step 2: Find the function of x_k, $f(x_k)=f_k$.
Step 3 : Set $k=1$.
Step 4 : Set $d_k=-g_k$.
Step 5 : Set $x_{k+1}=x_k+T_0 d_k$.
Step 6 : Find the function of x_{k+1}, $f(x_{k+1})=f_{k+1}$.
Step 7: Check if $f_{k+1}\geq f_k$ is satisfied then set $T_0=T_0/3=H(2)$ and go to step (5); otherwise go to step (8).
Step 8: Set $T_0=2T_0=H(3)$; $x_{k+2}=x_{k+1}+T_0 d_k$.
Step 9: Check if $f_{k+2}\geq f_{k+1}$. Set

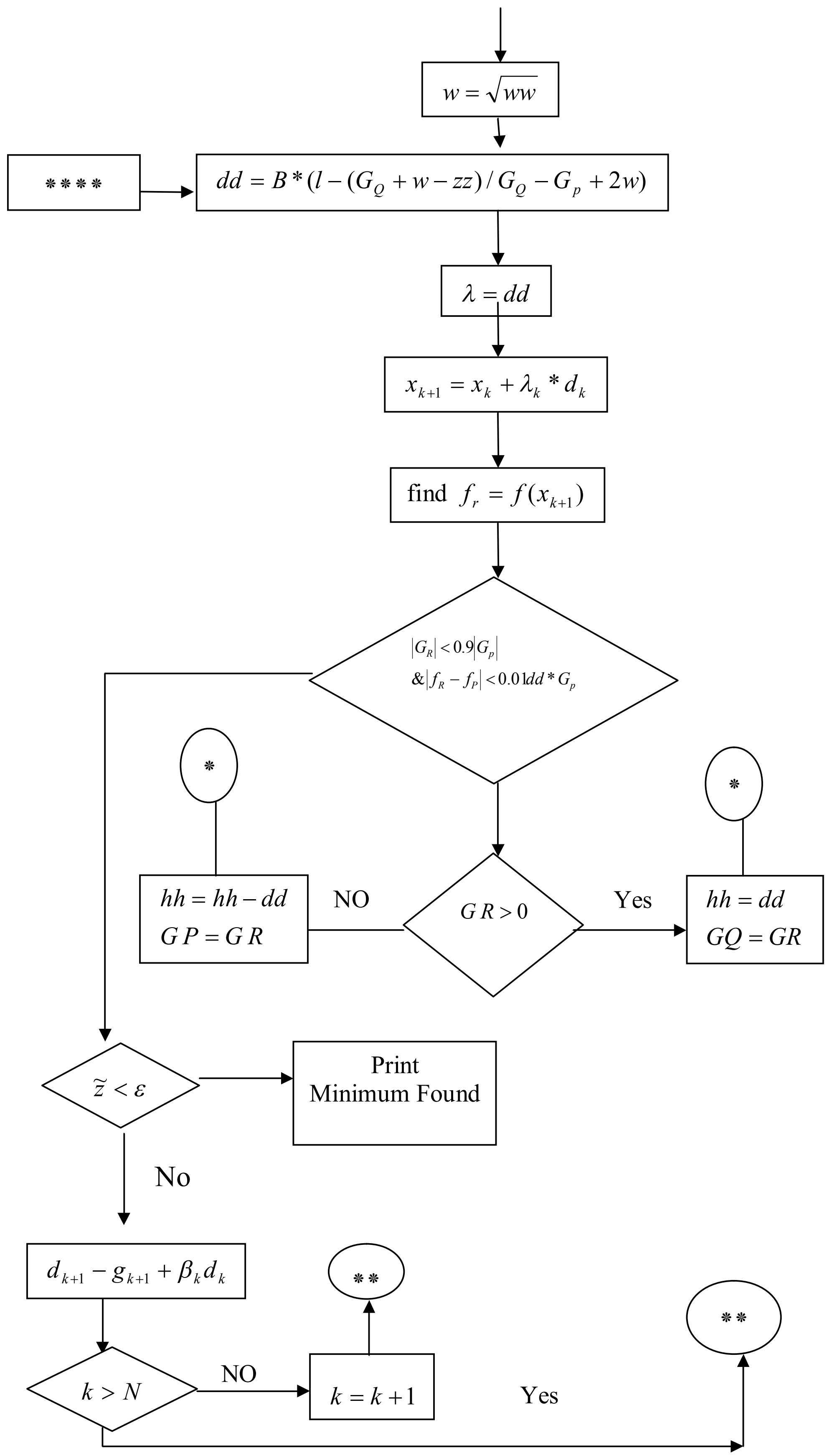
$$\widetilde{B}=H(1)$$
$$\widetilde{A}=\widetilde{B}/2$$
$$\widetilde{C}=2\widetilde{B}$$
$$\widetilde{A}\leq\widetilde{B}\leq\widetilde{C}$$

else go to step (8)
Step 10: Set $x_{\underset{\widetilde{C}}{k+1}}=x_k+\widetilde{C}d_k$.
Step 11: Set $x_{\underset{\widetilde{B}}{k+1}}=x_k+\widetilde{B}d_k$ and $x_{\underset{\widetilde{A}}{k+1}}=x_k+\widetilde{A}d_k$.
Step 12: Compute $f_p=f(x_{\underset{\widetilde{C}}{k+1}})$ and $f_Q=f(x_{\underset{\widetilde{A}}{k+1}})$.
Step 13: Compute $G_p=g_{k+1}^T d_{k+1}$ and $G_Q=g_{k+1}^T d_{k+1}$.
Step 14: Put $B=hh$
Step 15: Set $zz=3*(F_p-F_Q)/hh$ and $zz=zz+G_p+G_Q$.
Step 16: Compute $ww=zz^2-G_p*G_Q$.
Step 17: Check if $ww<0$, then set $ww=0$ and go to step (19) else put $w=(ww)^{1/2}$.
Step 18: Compute $dd=hh*(1.-(G_Q+w-zz)/(G_Q-G_p+2*w)$ and set $\lambda_k=dd$.
Step 19: Find $x_{k+1}=x_k+\lambda_k d_k$.
Step 20: Find $f_R=f(x_{k+1})$.
Step 21: Find $G_R=g(x_{k+1})^T d(x_{k+1})$.
Step 22: Check if $|G_R|<0.9|G_p|$ *and* $|f_R-f_p|\leq 0.01dd*G_p$ is satisfied then go to step(26); otherwise
Step 23: Check if $G_R>0$.
Step 24: Set $hh=dd$; $G_Q=G_R$ Go to step (16); else set $hh=hh-dd$ and $G_P=G_R$ then go to step (16).
Step 25: Check if $\left|\overset{\sigma}{Z}\right|\leq\varepsilon$. Print minimum found; else
Step 26: Set $d_{k+1}=-g_{k+1}+\beta_k d_k$.
Step 27: Check if $k>n$ go to step (3); else set $k=k+1$ and go to step (4).

2. Flowchart of CG algorithm with New Combined Novel and Cubic Line Searches:




$\tilde{B} = H(1)$
$\tilde{A} = \tilde{B} / 2$
$\tilde{C} = 2\tilde{B}$
$\tilde{A} \pi \tilde{B} \pi \tilde{C}$
$f_{\tilde{A}} \geq f_{\tilde{B}} \geq f_{\tilde{C}}$
$x_{\tilde{A}} = x_k + \tilde{A} d_k$
$x_{k+1, \tilde{B}} = x_k + \tilde{B} d_k$
$x_{k+1} = x_k + \tilde{C} d_K$
$f_p = f(x_{k+1, \tilde{B}})$
$f_Q = f(x_{k+1, \tilde{A}})$
$G_p = g(x_{k+1, \tilde{A}})^T d_k$
$G_Q = g(x_{k+1, \tilde{B}})^T d_k$
$hh = \tilde{B}$
$ZZ = 3 * (f_Q - f_p) / B$
$ZZ = ZZ + GP + GQ$
*
$ww = (z^2 - G_P + G_Q) / 2$
$ww < 0$
Yes
$ww = 0$
****


$w = \sqrt{ww}$
∗∗∗∗
$dd = B * (l - (G_Q + w - zz) / G_Q - G_p + 2w)$
$\lambda = dd$
$x_{k+1} = x_k + \lambda_k * d_k$
find $f_r = f(x_{k+1})$
$|G_R| < 0.9|G_p|$
$\& |f_R - f_P| < 0.01dd * G_p$
∗
∗
$hh = hh - dd$
$GP = GR$
NO
$GR > 0$
Yes
$hh = dd$
$GQ = GR$
$\tilde{z} < \varepsilon$
Print
Minimum Found
No
$d_{k+1} - g_{k+1} + \beta_k d_k$
∗∗
∗∗
$k > N$
NO
$k = k + 1$
Yes

5. Numerical Results:

In this section, we discuss computational results of H/S-CG method using cubic and novel line searches techniques for finding the minimum of nonlinear objective functions. The results are obtained using Pentium IV Computer. The programs were written in FORTRAN Language. The comparative performance of the algorithms is evaluated by considering NOF , NOI , and TW is the best measure of actual work done. Table (1) includes the numerical results of (H/S) CG-algorithm by using cubic line search and Novel line search for $4 \le n \le 1000$. Table (2) includes the numerical results of (F/R)-CG method by using cubic line search and employ new combined cubic-Novel line research for test function of $4 \le n \le 1000$.

Table (1). Comparisons of CG-algorithm (Novel line search) with CG-algorithm (cubic line search)

Test fun.	Dim	H/S- algorithm (cubic and novel line searches)		H/S- algorithm (cubic line searches)	
		TW=NOF	NOI	TW=NOF(N+1)	NOI
Central	4	733	50	2760	51
Mmiele	4	1857	126	2090	107
Powell	4	2311	163	560	49
Sum	4	153	8	265	6
Osp	4	138	8	7474	5
Central	100	453	27	25957	27
Powell3	100	331	29	4747	22
Powell	100	2152	120	46662	203
Herical	100	488	41	21311	103
Osp	1000	1600	159	340340	154
Wolfe	1000	456	46	127127	63
Total		**10672**	**777**	**579293**	**790**

Table (2). Comparisons of our new algorithm with standard CG- algorithm

Test fun.	Dim	F/R- algorithm (cubic and novel line searches)		F/R- algorithm (cubic line searches)	
		TW=NOF(novel)+(n+1)NOF (cubic)	NOI	TW=NOF(N+1)	NOI
Rosen	4	1593	62	1320	87
Miele	4	425	16	395	38
Cubic	4	469	18	290	25

Powell3	4	269	10	235	23
Herical	4	1318	52	2170	215
Shallow	4	164	6	120	10
Central	4	404	16	435	20
Rosen	1000	288680	71	310301	110
Miele	1000	269605	67	345345	163
Powell3	1000	57147	14	71071	35
Herical	1000	264610	66	684684	340
Shallow	1000	28076	7	30030	13
Sum	1000	56148	14	109109	26
Cubic	1000	9229	23	72072	32
Central	1000	109212	27	192192	30
Total		**1170344**	**469**	**1819769**	**1167**

From the above two tables we can find the following:

Table (3). Performance percentage of the new algorithm compared with standard CG-algorithm (cubic line search)

Tools	F/R- algorithm **(cubic line search)**	F/R- algorithm **(cubic and novel line searches)**
NOI	٪100	٪40
TW	٪100	٪64

Appendix

1. Generalized Osp (Oren and Spedicato Function:

$$f(x)=\left[\sum_{i=1}^{n} i x_i^2\right]^2, \qquad x_0=(1,\ldots)^T.$$

2. Wolfe Function:

$$f(x)=[-x_1(3-x_1/2)+2x_2-1]^2+\sum_{i=1}^{n-1}[x_{i-1}-x_i(3-x_i/2+2x_{i+1}-1)]^2+[x_{n-1}-x_n(3-x_n/2)-1]^2, \qquad x_0=(-1,\ \ldots)^T$$

3. Generalized Cantrell Function:

$$f(x)=\sum_{i=1}^{n/4}\left[\begin{array}{l}(\exp(x_{4i-3})-x_{4i-2})^4+100(x_{4i-2}-x_{4i-1})^6+\\ \arctan(x_{4i-1}-x_{4i}))^4+x_{4i-3}\end{array}\right], \quad x_0=(1,2,2,2;\ldots)^T.$$

4. Generalized Miele Function:

$$f(x)=\sum_{i=1}^{n/4}\begin{bmatrix} exp(x_{4i-3}-x_{4i-1})^2+100(x_{4i-2}-x_{4i-1})^6+ \\ (tan(x_{4i-1}-x_{4i}))^4+x_{4i-3}^8+(x_{4i}-1)^2 \end{bmatrix},\quad x_0=(1,2,2,2;\ldots)^T.$$

5.Generalized Powell Function:

$$f(x)=\sum_{i=1}^{n/4}\begin{bmatrix} (x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+ \\ (x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4 \end{bmatrix},\quad x_0=(3,-1,0,3;\ldots)^T.$$

6.Generalized Powell 3 Function:

$$f(x)=\sum_{i=1}^{n/3}\left\{3-\left[\frac{1}{1+(x_i-x_{2i})^2}\right]-sin(\frac{\pi\, x_{2i}x_{3i}}{2})-exp\left[-(\frac{x_i+x_{3i}}{x_{2i}}-2)^2\right]\right\},x_0=(0,1,2;\ldots)^T$$

7.Rosenborck Function:

$$f(x)=\sum_{i=1}^{n/2}100(x_{2i}-x_{2i-1}^2)^2+(1-x_{2i-1})^2\quad,\quad x_0=(-1.2,1,\ldots)^T.$$

8. Generalized Cubic Function:

$$f(x)=\sum_{i=1}^{n/2}\left[100(x_{2i}-x_{2i-1}^3)^2+(1-x_{2i-1})^2\right],\quad x_0=(-1.2,1,\ldots)^T.$$

9. Generalized Shallow Function:

$$f(x)=\sum_{i=1}^{n/2}\left[x_{2i-1}^2-x_{2i}\right]^2+(1-x_{2i-1})^2\ ,\quad x_0=(-2,\ -2;\ldots)^T.$$

10. Sum of Quatrics (SUM) function:

$$f(x)=\sum_{i=1}^{n}(x_i-i)^4,\quad x_0=(1\ ,\ldots)^T.$$

11. Herical function:

$$f(x)=100(x_{3i})-10(\frac{1}{2\pi}ATAN(\frac{x_{3i-1}}{x_{3i-2}}))^2+100(\sqrt{x_{3i-2}^2+x_{3i-1}^2}-1)^2+x_{3i}^2$$

$$x_0=(-1\ ,0,0,\ldots)^T.$$

REFERENCES

[1] Al-Bayati, A.Y. and Al-Assady, N.H (1986), "**Conjugate Gradient Method**", Technical Report, no. (1/86), School of Computer Studies, Leeds University. U.K

[2] Bundy, B.(1984), "**Basic optimization methods"**, Edward Arnold Bedford square, London, U.K.

[3] Davidon, W.C. (1959), "**Variable Metric Methods for Minimization**", A.E.C. Research and Development, Report ANL-5990, Argonne National Laboratory, Argonne, I 11 nois.

[4] Dixon, L.G.W.(1975), "**Conjugate Direction Without Linear Search",** journal of Institute mathematics and it is applications 11.pp: 317- 328.

[4] Fletcher, R. and Reeves, C.M., (1964), "***Function Minimization by Conjugate Gradient***", Computer Journal, Vol. 7, PP.149-154.

[5] Hestenes, M. R. and Stiefel E.(1952)" ***Method of Conjugate Gradient for Solving Linear System*** "Journal of Research of the National Bureau of Standards, 49, pp.409-436

[6] Cristian, C. (2003), "**Optimization Techniques (Tehnici de Optimizare) Faculty of Automatic Control and Computers**", University Polytechnic Bucharest.

[7] Polak, E. (1969), "**Computational Method in Optimization**", A Unified Approach, Academic Press, New York.

[8] Rao , M. and Chandra K., (1983), "**A new line search for optimization algorithms**", proc. Conf. On systems, M, AM.

[9] Sun, J., Yang, x. AND Yang, X. (2003), "**Quadratic Cost Flow and the Conjugate Gradient Method**", Grants of National, University of Singapore.