A New effected Three-Term Hestenes-Stiefel Conjugate-Gradient Method for Solving Unconstrained Optimization Problems

Abbas Y. Al-Bayati[1] , Abbas H. Taqi[2] , Yoksal A. Sadiq[3]
[1] Basic Education College, University of Telafer , Telafer , Iraq
[2] Science College, University of Kirkuk , Kirkuk , Iraq
[3] Computer Sciences and Mathematic College, University of Mosul , Mosul , Iraq

Abstract

In this paper a new three–term Conjugate Gradient (CG) method is suggested, the derivation of the method based on the descent property and conjugacy condition, the global convergence property is analyzed; numerical results indicate that the new proposed CG-method is well compared against other similar CG-methods in this field.

1. Introduction

Consider the unconstrained optimization problem :

$$\min \{ f(x) \mid x \in R^n \} \quad (1)$$

where f is a continuously differentiable function of n variables. In order to introduce our new modified CG-method which is a generalization of three-term (Hestenes and Stiefel, 1952) (HS)-CG method. Let us simply recall the well-known BFCG quasi-Newton (QN) direction (Dennis and More et al.,1977). QN-methods for solving (1) often needed the new search direction d_k at each iteration by :

$$d_k = -H_k g_k \quad (2)$$

where $g_k = \nabla f(x_k)$ is the gradient of f evaluated at the current iterate x_k . One then computes the next iterate by

$$x_{k+1} = x_k + \alpha_k d_k \quad (3)$$

where the step size α_k satisfies the Wolfe–conditions

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta_1 \alpha_k d_k^T g_k \quad (4)$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \delta_2 d_k^T g_k \quad (5)$$

where $0 \prec \delta_1 \prec 1/2$ and $\delta_1 \prec \delta_2 \prec 1$, and H_{k+1} is an approximation to $\{\nabla^2 f(x_k)\}^{-1}$. The matrix H_{k+1} satisfies the actual quasi-Newton condition

$$H_{k+1} y_k = \rho_k v_k \quad (6)$$

where $y_k = g_{k+1} - g_k$, $v_k = x_{k+1} - x_k$, ρ_k is a scalar, for exact QN-condition $\rho_k = 1$.

For BFGS-update, see (Al-Bayati and Hassan, 2006), where H_{k+1} is obtained by the following BFGS formula:

$$H_{k+1} = H_K + \left(1 + \frac{y_k^T H_K y_k}{s_k^T y_k}\right) \frac{s_k s_K^T}{s_k^T y_k} - \frac{s_k y_k^T H_k + H_K y_k s_k^T}{s_k^T y_k} \quad (7)$$

If $H_k = I$ (where I is the identity matrix). Then the above BFGS method becomes the memoryless BFGS method introduced by Shanno (Shanno, 1978). In this case the search direction d_{k+1} can be defined as:

$$d_{k+1} = -g_{k+1} + \left(\frac{y_k^T g_{k+1}}{s_k^T y_k} - \left(1 + \frac{y_k^T y_k}{s_k^T y_k}\right)\frac{s_k^T g_{k+1}}{s_k^T y_k}\right) s_k + \frac{s_k^T g_{k+1}}{s_k^T y_k} y_k \quad (8)$$

which shows that d_{k+1} possesses the following form:

$$d_{k+1} = -g_{K+1} + \beta_k s_k - \delta_k y_k \quad (9)$$

which is called the three-term CG-algorithm.

(Nazareth, 1977) proposed another CG- algorithm using a three -term recurrence formula:

$$d_{k+1} = -y_k + \frac{y_k^T y_k}{y_k^T d_k} d_k + \frac{y_{k-1}^T y_k}{y_{k-1}^T d_k} d_{k-1} \quad (10)$$

with $d_{-1} = 0$, $d_0 = 0$.

If f is quadratic convex function, then for any step length α_k the search direction generated by (10) are conjugate subject to the Hessian of the nonlinear function f, even without exact line search. In the same context, (Zhang et al., 2007) proposed another descent modified HSCG method with three–term, say, ZTCG where its search direction was defined as:

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k}{s_k^T y_k} s_k - \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k \quad (11)$$

Where $d_o = -g_0$. A remarkable property of this method is that produce descent direction i.e.

$$d_k^T g_k = -\|g_{k+1}\|^2 \quad (12)$$

The convergent properties of (11) for a convex optimization are given in (Zhang et al., 2009). Zhang in the same paper introduced another three–term CG-method based on the Dia-Lia method, say, ZDTCG whose search direction was defined by:

$$d_{k+1} = -g_{K+1} + \frac{g_{k+1}^T (y_k - t s_k)}{s_k^T y_k} s_k - \frac{g_{k+1}^T s_k}{s_k^T y_k} (y_k - t s_k) \quad (13)$$

where $d_o = -g_0$ *and* $t \ge 0$.

There are many possibilities in choosing search directions in this type of methods and it must be said that there is no single choice that is superior to others in most situations. Below we will introduce a new formulated three term CG-method which its idea is based on two important properties, i .e. the descent property and conjugacy condition.

2. A New Three-Term CG-Method (New).

Consider the search direction which is suitable for any three-term CG-type methods is defined by the following formula:

$$d_{k+1} = -g_{K+1} + \frac{a}{s_k^T y_k} s_k - \frac{b}{s_k^T y_k} y_k \quad (14)$$

where $d_1 = -g_1$ and a, b are any unknown parameters.

The main advantages for the this type of search direction are employing the descent and conjugacy properties, therefore we use these two different types of theoretical properties to find the two unknown parameters a and b.

2.1. Assume that the search direction defined by equation (14) satisfies the sufficient descent property i.e.

$$d_{k+1}^T g_{k+1} = -g_{k+1}^T g_{k+1}$$

multiply both sides of (14) by g_{k+1} to get:

$$d_{k+1}^T g_{k+1} = -g_{k+1}^T g_{k+1} + \frac{s_k^T g_{k+1}}{s_k^T y_k} a - \frac{y_k^T g_{k+1}}{s_k^T y_k} b = -g_{K+1}^T g_{k+1}$$

or

$$\frac{s_k^T g_{k+1}}{s_k^T y_k} a - \frac{y_k^T g_{k+1}}{s_k^T y_k} b = 0$$

Again, multiply both sides of the above equation by the scalar parameter $s_k^T y_k > 0$ to get:

$s_K^T g_{k+1} a - y_{k+1}^T g_{k+1} b = 0$; or

$$b = \frac{s_k^T g_{k+1}}{y_k^T g_{k+1}} a \quad (15)$$

Again, trying to use the conjugacy property i.e., $d_{k+1}^T y_k = 0$, in (14) and multiplying both sides of (14) by y_k^T, yields:

$$d_{k+1}^T y_k = -g_{k+1}^T y_k + \frac{y_k^T s_k}{s_k^T y_k} a - \frac{y_k^T y_k}{s_k^T y_k} b = 0$$

$$-g_{k+1}^T y_k + a - \frac{y_k^T y_k}{s_k^T y_k} b = 0$$

$$\therefore\ a = \frac{y_k^T y_k}{s_k^T y_k} b + g_{k+1}^T y_k \quad (16)$$

Now, form equations (14) and (15) we have to get:

$$a = \frac{(g_{k+1}^T y_k)^2\, s_k^T y_k}{s_k^T y_k\, g_{k+1}^T y_k - y_k^T y_k\, s_k^T g_{k+1}} \quad (17)$$

$$b = \frac{s_k^T g_{k+1}\, s_k^T y_k\, g_{k+1}^T y_k}{s_k^T y_k\, g_{k+1}^T y_k - y_k^T y_k\, s_k^T g_{k+1}} \quad (18)$$

substituting a and b in equation (14), we will get the new formulated three-them CG-method i.e.

$$d_{k+1} = -g_{k+1} + \frac{(g_{k+1}^T y_k)^2}{s_k^T y_k\, g_{k+1}^T y_k - y_k^T y_k\, s_k^T g_{k+1}} s_k - \frac{s^T g_{k+1}\, y_k^T g_{k+1}}{s_k^T y_k\, g_{k+1}^T y_k - y_k^T y_k\, s_k^T g_{k+1}} y_k \quad (19)$$

2-2. Some Remarks on The New Method.

1- If the line search is exact i.e. $g_{K+1}^T s_k = 0$, then the search direction in (19) reduces to the classical HSCG search direction .

2- If the objective function is quadratic convex and line search is exact, then $g_{k+1}^T g_k = 0$ and $s_k^T g_{k+1} = 0$, hence, the search direction defined in (19) will reduce to the classical Conjugate-Descent method since, $g_{k+1}^T s_k = 0$, and $(g_{k+1}^T y_k)^2 = (g_{k+1}^T g_{k+1} - g_{k+1}^T g_k)^2 = (g_{k+1}^T g_{k+1})^2$.

3. Outlines of The New Algorithm (New).

Step1. Given an initial point and. Set

Step2. Set k=k+1 and calculate.

Step3. Check if , then stop.

Step4. Calculate step length using Wolfe line searches (4) and (5).

Step5. Set .

Step6. Calculate and.

Step7. Calculate The search direction defined in (19).

Step8. Go to Step2.

To show that the search directions of (19) are descent directions:

3.1. Proposition.

Suppose that the line search satisfies the Wolfe condition (4) and (5) then d_{k+1} given by (19) is a descent direction.

<u>Proof</u>.

$$d_{k+1} = -g_{k+1} + \frac{(g_{k+1}^T y_k)^2}{s_k^T y_k\, g_{k+1}^T y_k - y_k^T y_k\, s_k^T g_{k+1}} s_k - \frac{s_k^T g_{k+1}\, y_k^T g_{k+1}}{s_k^T y_k\, g_{k+1}^T y_k - y_k^T y_k\, s_k^T g_{k+1}} y_k$$

$$if\ k = 0 \ \ then\ \ d_1 = -g_1\ and\ d_1^T = -g_1^T g_1 = -\|g_1\|^2 < 0$$

$$d_{k+1}^T g_{k+1} = -g_{k+1}^T g_{k+1} + \frac{(g_{k+1}^T y_k)^2 s_k^T g_{k+1}}{s_k^T y_k g_{k+1}^T y_k - y_k^T y_k s_k^T g_{k+1}} - \frac{s_k^T g_{k+1} g_{k+1}^T y_k g_{k+1}^T y_k}{s_k^T y_k g_{k+1}^T y_k - y_k^T y_k s_k^T g_{k+1}}$$

$$d_{k+1}^T g_{k+1} = -g_{k+1}^T g_{k+1} + \frac{(g_{k+1}^T y_k)^2 s_k^T g_{k+1}}{s_k^T y_k g_{k+1}^T y_k - y_k^T y_k s_k^T g_{k+1}} - \frac{(g_{k+1}^T y_k)^2 s_k^T g_{k+1}}{s_k^T y_k g_{k+1}^T y_k - y_k^T y_k s_k^T g_{k+1}}$$

$$d_{k+1}^T g_{k+1} = -g_{k+1}^T g_{k+1}$$

To complete the theoretical required proofs for the new formulated three-term CG-method we have to prove the following main property.

4. Convergence Analysis Property.

In this section, we have to prove the basic global convergence property of the (New) proposed algorithm under the following assumptions:

4.1. The level set $S = \{x \in R^n : f(x) \le f(x_1)\}$ is bounded, i.e. there exists a positive constant $B > 0$ such that, for all:

$\|x\| \le B \qquad , \quad \forall\ x \in s$

$\|s_k\| \le B_1 \qquad , \quad \forall\ x \in s$

4.2. In a neighborhood N of S the function f is continuously differentiable and its gradient is Lipschitz continuous , i.e. there exists a constant $L > 0$ such that:

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|$$

Under these assumptions on f, there exists a constant $c \ge 0$ such that $\|\nabla f(x)\| \le c$, for all $x \in s$:

$$\|y_k\| \le c_1 \quad (20)$$

Observe that in the above assumption, the function f is bounded below is weaker than the usual assumption that the level set is bounded. Although the search directions generated by (19) are always descent directions, to ensure convergence of the algorithm we need to constrain the choice of the step length α_k. Now, the following proposition shows that the Wolfe line search always gives a lower bound for the step length α_k.

4.3. Proposition.

Suppose that d_K is a descent direction and that the gradient ∇f satisfies the Lipschitz condition $\|\nabla f(x) - \nabla f(x_k)\| \le L\|x - x_k\|$ for all x on the line segment connecting x_k and x_{k+1}, where L is a positive constant. If the line search satisfies the Wolfe conditions (4) and (5), then:

$$\alpha_k \ge \frac{(1-\sigma)|g_k^T d_k|}{L\|d_k\|^2} \quad (21)$$

Proof : See (Andrei et al ,2013)

To prove the global convergence we need the following lemma (Zoutendijk, 1970).

4.4. Lemma.

Suppose that x_1 is a starting point for which assumptions (4.1) and (4.2) hold. Let x_k be generated by the descent algorithm (New) with α_k satisfies the Wolfe line search conditions (4) and (5) then we have:

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \quad (22)$$

It easy to get from **Propositions (3.1)** that (22) is equivalent to the following equation:

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty \quad (23)$$

4.5. Theorem.

Suppose that assumptions (4.1) and (4.2) holds, and consider the new algorithm (New), where α_k is computed by the Wolfe line search conditions (4) and (5) then:

$$\lim_{k\to\infty} \inf \|g_k\| = 0 \quad (24)$$

Proof.

The prove is by contradiction we suppose that the conclusion is not true. Then there exist a constant $r > 0$ such that:

$$\|g_k\| > r \qquad \forall\ k > 0 \quad (25)$$

since $\|g_k\| \ne 0$ and with **Proposition (3. 1)** it follows that $d_k \ne 0$. Consider the search direction defined by the equation (19):

$$d_{k+1} = -g_{k+1} + \frac{(g_{k+1}^T y_k)^2}{s_k^T y_k\ g_{k+1}^T y_k - y_k^T y_k\ s_k^T y_k} s_k - \frac{s_k^T g_{k+1}\ y_k^T g_{k+1}}{s_k^T y_k\ g_{k+1}^T y_k - y_k^T y_k\ s_k^T y_k} y_k \quad (26)$$

Note that the from Lipschitz condition with $0 < L$ we have:

$$a: s_k^T y_k\ |y_k^T g_{k+1}| \ge \frac{1}{L} y_k^T y_k\ |y_k^T g_{k+1}|$$

$$\ge \frac{1}{L}\|y_k\|^2\ y_k^T g_{k+1}$$

$$b: y_k^T y_k\ s_k^T g_{k+1} \le L\ y_k^T y_k\ y_k^T g_{k+1}$$

Hence

$$-y_k^T y_k\ s_k^T g_{k+1} \ge -L\ y_k^T y_k\ y_k^T g_{k+1}$$

Therefore

$$s_k^T y_k\ y_k^T g_{k+1} - y_k^T y_k\ s_k^T g_{k+1} \ge \frac{1}{L} y_k^T y_k\ y_k^T g_{k+1} - L\ y_k^T y_k\ y_k^T g_{k+1}$$

$$= (\frac{1}{L} - L)\|y_k\|^2\ y_k^T g_{k+1}$$

$$\therefore\ s_k^T y_k\ y_k^T g_{k+1} - y_k^T y_k\ s_k^T g_{k+1} \ge (\frac{1-L^2}{L})\|y_k\|^2\ y_k^T g_{k+1}$$

Hence

$$\|d_{k+1}\| \le \left\| -g_{k+1} + \frac{L(y_k^T g_{k+1})^2}{(1-L^2)\|y_k\|^2\ y_k^T g_{k+1}} s_k - \frac{L\ s_k^T g_{k+1}\ y_k^T g_{k+1}}{(1-L^2)\|y_k\|^2\ y_k^T g_{k+1}} \right\|$$

$$\le \|g_{k+1}\| + \frac{L\|g_{k+1}\|\|s_k\|}{(1-L^2)\|y_k\|} + \frac{L\|s_k\|\|g_{k+1}\|}{(1-L^2)\|y_k\|}$$

$$= (1 + \frac{L\|s_k\|}{(1-L^2)\|y_k\|})\|g_{k+1}\| \le (1 + \frac{1}{(1-L^2)})\|g_{k+1}\|$$

Again for Lipschitz condition

$$\|y_k\| \le L\|s_k\|$$

Therefore

$$1 \le \frac{L\|s_k\|}{\|y_k\|}$$

Hence

$$\|d_{k+1}\| \le (\frac{L\|s_k\|}{\|y_k\|} + \frac{L\|s_k\|}{(1-L^2)\|y_k\|})\|g_{k+1}\|$$

$$= \frac{L\|s_k\|}{\|y_k\|}(1 + \frac{1}{(1-L^2)})\|g_{k+1}\|$$

$$= \frac{L\|s_k\|}{\|y_k\|}(\frac{2-L^2}{1-L^2})\|g_{k+1}\|$$

$$\therefore \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} \ge \sum_{k=0}^{\infty} c^2 \frac{1}{\|g_{k+1}\|^2} \quad (27)$$

Where

$$C=\frac{\|y_k\|}{L\|s_k\|}\left(\frac{1-L^2}{2-L^2}\right) \qquad (28)$$

multiply both side of (27) by $\|g_{k+1}\|^4$ to get

$$\sum_{k=0}^{\infty}\frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2}\geq\sum_{k=0}^{\infty}c^2\|g_{k+1}\|^2>\sum_{k=0}^{\infty}c^2\gamma^2=\infty$$

which is contradiction with lemma (22) , then

$$\lim_{k\to\infty}\inf\|g_{k+1}\|=0$$

5. Numerical Results .

The main work of this section is to report the performance of the new three-terms algorithm (**New**) on a set of test problems. The codes are written in Fortran and compiled with F77 (default Compiler settings). All the tests were performed on a PC. We selected (30) large-scale unconstrained optimization test functions in generalized or extended form (Bongartz et al., 1995) library. For each test function, we have taken numerical experiments with the number of variables n=100 and 1000 and their details are given in the Appendix. In order to assess the reliability of our new proposed method, we have tested it against (Fletcher, 1978); (Shanno, 1978) and (Zhang, et al., 2007), using the same test problems. The algorithm implements the acceleration Wolfe line search conditions with $\delta=10^{-4}$, $\sigma=0.9$ and the same stopping criterion $\|g_k\|_\infty\leq10^{-6}$, where $\|\cdot\|_\infty$ is the maximum absolute component of a vector.

Tables (1) and (2) show numerical results for employing **(30)** test functions with **four** different algorithms, namely; **(FRCG; SHANNO; ZTCG,** and **New)** using **n=100** and **1000** only; tools **(NOI** and **NOFG).** All of these tables indicate:

n = Dimension of the problem.

NOI = Number of iterations.

NOFG = Number of function and gradient evaluations.

In **Tables ((3); (4) and (5))** we have compared the percentage performance of the **New** algorithm against the **FRCG**; **ZTCG** and **SHANNO** algorithms respectively for **n=100** and with respect to **NOI** and **NOFG** taking over all the tools as 100%.

In **Tables ((6); (7) and (8))** we have compared the percentage performance of the **New** algorithm against the **FRCG**; **ZTCG** and **SHANNO** algorithms respectively for **n=1000** and with respect to **NOI** and **NOFG** taking over all the tools as 100%.

Table (1) Comparison between (FRCG; ZTCG & SHANNO against New) methods for the total of (30) Problems with n= 100

Prob.	New		FRCG		ZTCG		SHANNO	
	NOI	NOFG	NOI	NOFG	NOI	NOFG	NOI	NOFG
1	31	67	34	76	34	72	42	88
2	13	26	37	71	16	29	18	32
3	8	15	16	28	8	15	10	18
4	90	136	168	201	87	134	108	170
5	62	99	124	155	71	111	57	94
6	28	44	48	71	27	44	30	47
7	21	42	60	86	21	42	21	42
8	38	59	182	219	41	61	33	54
9	4	8	6	12	4	8	4	8
10	10	19	14	28	10	19	10	19
11	171	302	365	677	121	210	254	442
12	8	17	123	222	66	122	74	138
13	44	75	601	794	71	151	73	170
14	9	22	92	173	41	73	43	76
15	27	52	140	227	31	59	28	53
16	79	174	100	191	78	181	79	187
17	84	136	194	229	92	146	87	141
18	23	54	125	199	24	52	24	52
19	92	146	338	376	120	181	118	189
20	32	51	53	75	33	53	31	52
21	13	26	31	61	17	34	15	30
22	82	124	137	167	84	131	79	125
23	28	49	71	95	27	46	74	47
24	80	123	176	211	99	150	108	169
25	437	689	1047	1180	556	880	535	864
26	24	46	41	67	27	47	26	46
27	12	23	254	402	15	25	15	26
28	20	36	207	891	21	37	18	33
29	53	130	477	796	85	184	81	179
30	30	50	46	70	30	51	40	61
Total	**1653**	**2840**	**5307**	**8050**	**1957**	**3348**	**2135**	**3652**

Table (2) Comparison between (FRCG; ZTCG & SHANNO against New) methods for the total of (30) Problems with n= 1000

Prob.	New		FRCG		ZTCG		SHANNO	
	NOI	NOFG	NOI	NOFG	NOI	NOFG	NOI	NOFG
1	26	53	62	115	34	70	40	87
2	13	25	16	30	13	25	17	31
3	10	20	15	31	10	20	9	18
4	315	494	804	859	342	539	324	521
5	182	303	300	362	185	305	187	316
6	342	9380	741	21790	318	8659	507	15041
7	27	50	69	102	30	239	38	540
8	63	99	207	244	45	71	52	84
9	4	8	6	12	4	8	4	8
10	10	19	16	30	19	31	18	29
11	419	734	481	941	263	467	278	475
12	67	124	142	250	67	122	73	140
13	67	153	611	818	68	166	73	175
14	138	240	339	622	152	268	156	277
15	27	52	84	157	24	48	21	42
16	79	171	111	207	78	179	75	181
17	337	525	744	799	342	529	335	537
18	35	85	2469	3258	38	90	38	88
19	360	566	1621	1725	372	587	477	765
20	36	55	59	78	31	48	37	58
21	14	28	49	95	12	24	11	21
22	212	332	378	407	236	372	247	404
23	40	65	2162	2610	35	58	37	60
24	331	517	804	859	333	532	321	516
25	321	494	938	998	308	474	344	544
26	79	2062	178	4558	110	2907	95	2058
27	10	20	260	384	12	24	14	27
28	19	35	1182	20065	22	39	17	30
29	130	296	279	499	114	253	286	631
30	31	55	47	72	40	62	34	57
Total	**3744**	**17060**	**15174**	**62977**	**3657**	**17216**	**14165**	**23761**

Table (3); N=100

Tools	**FRCG**	**New**
NOI	100	31.14
NOFG	100	35.27

Table (4); N=100

Tools	**ZTCG**	**New**
NOI	100	84.46
NOFG	100	84.82

Table (5); N=100

Tools	**SHANNO**	**New**
NOI	100	77.42
NOFG	100	77.76

Table (6); N=1000

Tools	**FRCG**	**New**
NOI	100	24.67
NOFG	100	27.09

Table (7); N=1000

Tools	**ZTCG**	**New**
NOI	100	101.02
NOFG	100	99.09

Table (8); N=1000

Tools	**SHANNO**	**New**
NOI	100	26.43
NOFG	100	71.79

6. Discussions.

It is clear that from **Table (3) and for n=100** that taking, over all, the tools as a 100% for the **FRCG** method, the new method **(New)** has an improvements about; **(68)% NOI; (64)% NOFG**.

It is clear that from **Table (4) and for n=100** that taking, over all, the tools as a 100% for the **ZTCG** method, the new **method (New)** has an improvements about; **(15)% NOI; (15)% NOFG.**

It is clear that from **Table (5) and for n=100** that taking, over all, the tools as a 100%, for **SHANNO** method, the new method **(New)** has an improvements about; **(22)% NOI; (22)% NOFG.**

It is clear that from **Table (6) and for n=1000** that taking, over all, the tools as a 100% for the **FRCG**

method, the new method **(New)** has an improvements about; **(75)% NOI; (72)% NOFG**.
It is clear that from **Table (7) and for n=1000** that taking, over all, the tools as a 100% for the **ZTCG** method, the new **method (New)** has an improvements about; **(1)% NOI; (1)% NOFG.**

It is clear that from **Table (8) and for n=1000** that taking, over all, the tools as a 100%, for **SHANNO** method, the new method **(New)** has an improvements about; **(73)% NOI; (28)% NOFG.**
In general, These results indicate that all the four new three-term CG-methods are, in general, the best.

Appendix.

1- Extended White & Holst Function

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

$$x_0 = [-1.2,1,...,-1.2,1], \qquad c = 100$$

2- Extended Beale function.

$$f(x) = \sum_{i=1}^{n/2} \left(1.5 - x_{2i-1}(1 - x_{2i})\right)^2 + \left(2.25 - x_{2i-1}(1 - x_{2i}^2)\right)^2$$

$$+ \left(2.625 - x_{2i-1}(1 - x_{2i}^3)\right)^2, \quad x_0 = [1,0.8,\ldots,1,0.8]^T.$$

3- Extended Penalty Function

$$f(x) = \sum_{i=1}^{n-1} (x_{i-1})^2 + \left(\sum_{j=1}^{n} x_j^2 - 0.25\right)^2,$$

$$x_0 = [1, 2, \ldots, n]^T.$$

4- Perturbed Quadratic

$$f(x) = \sum_{i=1}^{n} i x_i^2 + \frac{1}{100}\left(\sum_{i=1}^{n} xi\right)^2$$

$$x_0 = [0,5,0,5,...,05]$$

5- Raydan (1) Function

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

$$x_1 = [1,1,...,1]$$

6- Hager Function

$$f(x) = \sum_{i=1}^{n} (\exp(x_i) - \sqrt{i x_i}),$$

$$x_0 = [1,1,...,1]^T.$$

7- Generalized Tridiagonal-1 Function

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

$$x_0 = [2,2,...,2]^T$$

8- Generalized Tridiagonal 2 Function

$$f(x) = ((5 - 3x_1 - x_1^2)x_1 - 3x_2 + 1)^2 +$$

$$\sum_{i=1}^{n-1} ((5 - 3x_i - x_i^2)x_i - x_{i-1} - 3x_{i+1} + 1)^2 +$$

$$((5 - 3x_n - x_n^2)x_n - x_{n-1} + 1)^2,$$

$$x_1 = [-1,-1,...,-1]$$

9-Diagonai3

$$f(x) = \sum_{i=1}^{n} \left(\exp(x_i) - i\sin(x_i)\right),$$

$$x_0 = [1, 1, \ldots, 1]^T.$$

10- Extended Himmelblau Function

$$f(x) = \sum_{i=1}^{n/2} \left(x_{2i-1}^2 + x_{2i} - 11\right)^2 + \left(x_{2i-1} + x_{2i}^2 - 7\right)^2,$$

$$x_0 = [1, 1, \ldots, 1]^T.$$

11- Extending Powell Function

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

$$+ (x_{4i-2} - 2x_{4i-1})^2 + 10(x_{4i-3} - x_{4i})^4,$$

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

12- Extended PSC1 Function

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

$$x_0 = [3,-1,0,1,...,3,-1,0,1].$$

13. Extended Maratos Function (c=100)

$$f(x) = \sum_{i=1}^{n/2} x_{2i-1} + c(x_{2i-2}^2 + x_{2i}^2 - 1)^2$$

$$x_1 = [1.1,0.1,......,1.1,0.1]$$

14- Extended Cliff Function.

$$f(x) = \sum_{i=1}^{n/2} \left(\frac{x_{2i-1} - 3}{100}\right)^2 - (x_{2i-1} - x_{2i}) + \exp(20(x_{2i-1} - x_{2i}),$$

$$x_0 = [0,-1,...,0,-1].$$

15- Quadratic Diagonal Perturbed

$$f(x) = \left(\sum_{i=1}^{n} xi\right)^2 + \sum_{i}^{n} \frac{i}{100} x_i^2,$$

$$x_0 = [0,5,0,5,...,0,5]$$

16- Full Hessian FH2 Function.

$$f(x) = (x_i - 5)^2 + \sum_{i=1}^{n} (x_1 + x_2 + \ldots + x_{i-1})^2,$$

$$x_0 = [0.01,0.01,...,0.01].$$

17-Full Hessian FH3 Function.

$$f(x) = \left(\sum_{i=1}^{n} x_i^2\right)^2 + \sum_{i}^{n} \frac{i}{1000}(\sin x_i + \cos x_i),$$

$$x_0 = [1,1,...,1]$$

18-Tridiagonal White & Holst (c=4):

$$f(x) = \sum_{i=1}^{n-1} c(x_{i+1} + x_i^3)^2, + (1 + x_i)^2, c = 4,$$

$$x_0 = [-1.2,1,...,-1.2,1],$$

19-Diagonal Double Boarded Arrow Up:

$$f(x) = \sum_{i=1}^{n} 4(x_i^2 - x_i)^2, + (x_i - 1)^2$$

$$x_0 = [4,0,...,4,0],$$

20- ARWHEAD

$$f(x) = \sum_{i=1}^{n-1} \left(-4x_i + 3\right) + \sum_{i=1}^{n-1} \left(x_i^2 + x_n^2\right)^2,$$

$$x_0 = [1, 1, \ldots, 1]^T.$$

21-DQDRTIC Function

$$f(x) = (x_i - 5)^2 + \sum_{i=1}^{n} (x_1 + x_2 + \ldots + x_{i-1})^2,$$

$$x_0 = [0.01, 0.01, \ldots, 0.01].$$

22- Fletcher (CUTE)

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

$$x_0 = [0., 0., \ldots, 0.]^T. \quad c = 100.$$

23- DENSCHNA Function.

$$f(x) = \sum_{i=1}^{nl2} x_{2i-1}^4 + \sum_{i=1}^{n} (x_{2i-1} + x_{2i})^2 + (-1 + \exp(x_{2i}))^2,$$

$$x_0 = [8, 8, \ldots, 8].$$

24- DENSCHNB Function.

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

$$x_0 = [1, 1, \ldots, 1]^T.$$

25-DENSCHNC Function.

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

$$x_0 = [8, 8, \ldots, 8].$$

26- Perturbed Quadratic Function

$$f(x) = \sum_{i=1}^{n} (i x_i^2 + \frac{1}{100} (\sum_{i=1}^{n} x_i)^2),$$

$$x_0 = [0.5, 0.5, \ldots, 0.5].$$

27-Extended Quadratic Penalty QP1 Function

$$f(x) = \sum_{i=1}^{n-1} (x_i^2 - 2)^2 + \left(\sum_{i=1}^{n} x_i^2 - 0.5 \right)^2,$$

28-Tridiagonal TS1 Function.

$$f(x) = \sum_{i=2}^{n-1} (x_i + x_{i+1} - i)^2,$$

$$x_0 = [1, 1, \ldots, 1].$$

29-Tridiagonal TS1 Function.

$$f(x) = \sum_{i=2}^{n} (x_{i-1} + x_i - i)^2,$$

$$x_0 = [1, 1, \ldots, 1].$$

30 -PRODCOS Function.

$$f(x) = (x_1^2 + x_2^2 + \ldots + x_m^2)(\cos(x_1) + \cos(x_2) + \ldots + \cos(x_n)),$$

$$m = n - 1$$

$$x_0 = [5, 5, \ldots, 5].$$

References

[1] Al-Bayati, A. and Hassan, B. (2006), Some Theoretical Results for Oren Variable Metric Method. J. of Kirkuk University, Iraq, 1, 26-31.

[2] Andrei, N. (2013), A simple three-term conjugate gradient algorithm for unconstrained optimization , Journal of Computational and Applied Mathematics, 241, 19–29.

[3] Dennis, J. and More, J. (1977), Quasi-Newton methods, motivation and theory, SIAM Review, 19, 46-89.

[4] Fletcher, R. (1987), Practical Methods of Optimization (second edition)", John Wiley and Sons, New York.

[5] Hestenes, M. and Stiefel, E. (1952) , Method of conjugate gradients for solving linear systems", J. Research Nat. Standards , 49, 409-436.

[6] Nazareth, L. (1977), A conjugate direction algorithm without line search. Journal of Optimization Theory and Applications, 23, 373-387.

[7] Shanno, D. (1978), Conjugate gradient methods with inexact searches, Math. of operation Research, 3, 244-256.

[8] Zhang, L. Zhou, Y. (2012), A note on the convergence properties of the original three-term Hestenes–Stiefel method, AMO-Advanced Modeling and Optimization, 14, 159–163.

[9] Zhang, L. Zhou, W. and Li, D. (2007), Some descent three-term conjugate gradient methods and their global convergence. Optimization Methods and Software, 22, 697- 711.

[10] Zhang, J.; Xiao, Y. and Wei, Z. (2009), Nonlinear conjugate gradient methods with sufficient descent condition for large-scale unconstrained optimization. Math. Prog. Eng., Article ID 243290, 16. DOI: 10.1155/2009/243290.

[11] Bongartz, I.; Conn, A.; Gold, N. and Toint, P. (1995), CUTE: constrained and unconstrained testing environment, ACM Trans., Math. Software, 21.

طريقة جديدة كفؤة للتدرج المترافق ذات الحدود الثلاثة لحل مسائل الامثلية اللاخطية

عباس يونس البياتي [1] ، عباس حسن تقي [2] ، يوكسل عبدالستار صادق [3]

[1] كلية التربية الأساسية ، جامعة تلعفر ، تلعفر ، العراق

[2] كلية العلوم ، جامعة كركوك ، كركوك ، العراق

[3] كلية علوم الحاسوب والرياضيات ، جامعة الموصل ، الموصل ، العراق

الملخص

تم في هذا البحث اقتراح خوارزمية جديدة كفؤة للتدرج المترافق ذات الحدود الثلاثة لحل مسائل الامثلية غير الخطية وتم اشتقاق هذه الطريقة بالاعتماد على شروط خاصية الانحدار وشروط الترافق مع دراسة نظرية للاستقرارية وتحليل التقارب المطلق لهذه الطريقة. النتائج العددية تشير إلى كفاءة هذه الخوارزمية مقارنة مع عدد كبير من الطرائق المتشابهه لها في هذا المجال