Global convergence of new modified CG method with inexact line search

Latif S. Ivan * and Mohammed J. Lajan**

* Dept. of Mathematics, College of Education, University of Salahaddin, Erbil, Iraq, e-mail: ivansubhi2001@yahoo.com
** Dept. of Mathematics, Faculty of Science and Education, School of Science, University of Sulaimani, Sulaimani, Iraq, e-mail: lajan_jalil@yahoo.co.uk

Received: 26 Feb. 2014, Revised: 7 Apr. 2014, Accepted: 24 April 2014
Published online: 7 June 2014

Abstract:

The conjugate gradient (CG) method has played a special role in solving non-linear unconstrained optimization problems due to the simplicity of their-iterations and their very low memory requirements. In this paper we take a modified to the Dai-Yuan (DY) conjugate gradient methods such that the direction generated by the modified method provides a descent direction for the optimization function and establish some global convergence of the proposed method. Numerical results effective and promising by comparing with CG method.

Key words: Unconstrained optimization; general line search method; conjugate gradient method (CG); inexact line search; global convergence

1. Introduction

This Consider an unconstrained minimization problem

$$\min f(x) \quad , \ x \in R^n \qquad (1)$$

Where R^n denotes an n-dimensional Euclidean space where $f(x): R^n \to R^1$ is a continuously differentiable function. There are many iterative schemes for solving (1), the iterative formula given by

$$x_{k+1} = x_k + \alpha_k d_k \ , \quad k = 1,2,3, \ldots \quad (2)$$

For convenience if x_k is the current iterate and x^* is a minimal solution or stationary point of (1) then we denote $f(x_k)$ by f_k, $\nabla f(x_k)$ by g_k, and $d_k = -g_k$ results in the steepest descent method [R.Fletcher and C.Reeves(1964), M.R.Hestenes, E.Stiefel (1952), E.Polak,G.Ribiere(1969), M.A.Wolfe (1978)]. Where d_k is a search direction of $f(x_k)$ at the current iterate x_k and α_k is a step size. The search direction $d_k$is generally required to satisfy

$$g_k^T d_k < 0 \qquad (3)$$

Which guarantees that d_k is a descent direction of $f(x)$ at x_k [Y.Yuan (1993)]. In order to guarantee the global convergence, we sometimes require d_k satisfying the sufficient descent condition

$$g_k^T d_k \le -c\|g_k\|^2 \qquad (4)$$

Where $c > 0$ constant and $\|.\|$ stands for the Euclidean norm of vectors. Moreover, the angle property is often used in proving the global convergence of related line search methods, that is

$$\cos(-g_k, d_k) = -\frac{g_k^T d_k}{\|g_k\|.\|d_k\|} \ge r \qquad (5)$$

Where $0 > r \ge 1$

After the descent direction d_k is determined at the k-th iteration, we should seek a step size along the descent direction and complete on iteration. The conjugate gradient method is an important line search method that has the form (2) with

$$d_k = \begin{cases} -g_k & \text{if} \quad k = 0 \\ -g_k + \beta_k d_{k-1} & \text{if} \quad k \geq 1 \end{cases} \qquad (6)$$

Where β_k is a parameter well- known conjugate gradient methods include. The best-known formulas for β_k can be defined by

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \qquad (7)$$

$$\beta_k^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{\|g_{k-1}\|^2} \qquad (8)$$

$$\beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})} \qquad (9)$$

$$\beta_k^{LS} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T g_{k-1}} \qquad (10)$$

$$\beta_k^{CD} = -\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}} \qquad (11)$$

$$\beta_k^{DY} = \frac{g_k^T g_k}{(g_k - g_{k-1})^T d_{k-1}} \qquad (12)$$

The corresponding method is respectively called FR (Fletcher-Reeves) , PRP(Polak-Ribiére-Polyak) , HS(Hestenes-Stiefel) , LS(Liu-Storey) , CD(Conjugate Descent) and DY(Dai-Yuan) conjugate gradient method. The conjugate gradient method is an approach for solving large scale minimization problems due to its decreased storage requirements and simple computation.

In the implementation of any CG method the step size α_k is often determined by certain line search conditions such as the Wolfe condition namely

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta\, \alpha_k\, g_k^T d_k \qquad (13)$$

And

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma\, g_k^T d_k \qquad (14)$$

Or the strong Wolfe conditions namely

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma\, g_k^T d_k \qquad (15)$$

Where $0 < \delta < \sigma \leq 1$

The types of line search involve extensive computation of function values and gradients, which often becomes a significant burden for optimization problem. In this paper we used a simple modified inexact step size see [8] formula with new modified Dai-Yuan and damped CG and study the convergence property of the new modification method is promising.

2. Algorithm of Dai-Yuan Conjugate Gradient Method

Through this section, we assume that every search direction d_{k+1} satisfies the descent condition (3). The following step show Dai-Yuan Conjugate Gradient Method (V.H.Dai,Y.Yuan 1999)which are used to find minimum point of unconstrained optimization problems..

2.1 The outline of Dai-Yuan Conjugate Gradient Method

Step 1: Given a starting point x_0 , a tolerance $\varepsilon = 1x10^{-6}$, n

Step 2: set $k = 1 , d_0 = -g_0$

Step 3: Terminate if $\|g_k\| \leq \varepsilon$, stop otherwise go to **step 4**

Step 4: Compute x_{k+1} used formula (2) and α_k is obtain from line search procedure satisfy formula (13)

Step 5: Find the new direction d_{k+1} used formula (6), (12)

Step 6: Go to **step 3.**

3. Result for New Modified Conjugate Gradient Methods

3.1 New propose method

In this section we used new inexact modified step size rules full described used (Grippo-Lucidi)line search rule to find the best step size parameter along the new modified search direction iterative α_k where

$$\alpha_k = \max\left\{\rho_1^j \frac{|g_k^T d_k|}{\|d_k\|^2} , \frac{|g_k^T d_k|}{\rho_2 \|d_k\|^2} , 0 ; j = 1,2,3, \dots\right\} \quad (16)$$

Satisfies (13), (14) where

$$-c_2\|g_{k+1}\|^2 \le g_{k+1}^T d_{k+1} \le -c_1 \|g_{k+1}\|^2 \quad (17)$$

With $\delta > 0$, $0 < c_1 < 1 < c_2$, $\rho_1 \in (0,1)$, and $\rho_2 = \min\{\delta , c_1\}$. Also we propose a modified line search method is proposed in which the proposed on the basic of β_k

$$\beta_k^{MDY} = \frac{t_k\, g_k^T\, g_k}{t\,(g_k - g_{k-1})^T d_{k-1}} \quad (18)$$

Where $t_k = \frac{\|g_k\|}{\|g_{k+1}\|}$, $t > 0$. If $t_k = t$ we have special case of Dai-Yuan Conjugate Gradient β_k (Dai-Yuan) scalar.

In order to prove the global convergence of New Modification Algorithm with inexact line search. We make the following basic assumptions on the objective function

(**H1**)The objective function $f(x)$ is continuously differentiable and has lower bound on R^n,

(**H2**) The gradient $g(x) = \nabla f(x)$ of $f(x)$ is Lipschitz continuous,

(**H3**) We assume that ever d_k satisfies the descent condition (3).

3.2 The outline of New modified Dai-Yuan CG with Inexact line search

Step 1: Given a starting point x_0, a tolerance $\varepsilon = 1x10^{-6}$, n

Step 2: Set $k = 1$, $d_0 = -g_0$,

Step 3: Terminate if $\|g_k\| \le \varepsilon$ stop otherwise go to **step 4**

Step 4: Compute x_{k+1} used formula (2) and α_k is obtain from inexact line search procedure (16) satisfy formula (13), (17)

Step 5: Find the new direction d_{k+1} used formula (6) (18)

Step 6: Set $k = k + 1$, go to **step 3**.

3.3 Convergent Analysis

In convergence properties which we present in this section prove the global convergence for the general CG methods. We assume that every d_k atisfies the descent condition as formulas (3).

Theorem: suppose that the assumptions (**H1**), (**H2**) holds, consider any iteration methods of the form (2), (6) where d_k satisfied (3) and α_k is obtained by (16) satisfies (13), (14) then

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \quad (19)$$

The following theorem is a general and positive result for conjugate gradient methods.

Theorem: suppose that the assumptions (**H1**), (**H2**) holds, consider the method of form (2), (6) with d_k satisfied (3) and where α_k satisfied (13), (14) then either

$$\lim_{k\to\infty} \inf\|g_k\| = 0 \tag{20}$$

Or

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < +\infty \tag{21}$$

Proof: formula (6) indicates that for all $k \geq 2$

$$d_k + g_k = \beta_k d_{k-1} \tag{22}$$

Squaring both sides of formula (22) we obtain

$$\|d_k\|^2 = -\|g_k\|^2 - 2\, g_k^T d_k + \beta_k^2 (\|d_{k-1}\|)^2 \tag{23}$$

It follows form this relation and (5) that

$$\|d_k\|^2 \geq \beta_k^2 \|d_{k-1}\|^2 - \|g_k\|^2 \tag{24}$$

Use (6) implies the following relation

$$g_k^T d_k - \beta_k g_k^T d_{k-1} = -\|g_k\|^2 \tag{25}$$

To above inquilinity used Assumption (**H2**) and (8) Cauchy-Schwartz inequality yield

$$\left(g_k^T d_k\right)^2 + \beta_k^2 \left(g_{k-1}^T d_{k-1}\right)^2 \geq c_1 \|g_k\|^4 \tag{26}$$

Where c_1 defined (17) is positive constant .therefore, it follows from (24), (26) that

$$\frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} + \frac{\left(g_{k-1}^T d_{k-1}\right)^2}{\|d_{k-1}\|^2}$$
$$= \frac{1}{\|d_k\|^2}\left[\left(g_k^T d_k\right)^2 + \frac{\|d_k\|^2}{\|d_{k-1}\|^2}\left(g_{k-1}^T d_{k-1}\right)^2\right]$$
$$\geq \frac{1}{\|d_k\|^2}\left[\left(g_k^T d_k\right)^2 + \beta_k^T\left(g_{k-1}^T d_{k-1}\right)^2 - \frac{\left(g_{k-1}^T d_{k-1}\right)^2}{\|d_{k-1}\|^2}\|g_k\|^2\right]$$
$$\geq \frac{1}{\|d_k\|^2}\left[c_1\|g_k\|^4 - \frac{\left(g_{k-1}^T d_{k-1}\right)^2}{\|d_{k-1}\|^2}\|g_k\|^2\right] \tag{27}$$

If (20) is not true, relation (27) and (19) imply that the following inequality

$$\frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} + \frac{\left(g_{k-1}^T d_{k-1}\right)^2}{\|d_{k-1}\|^2} \geq \frac{c_1}{2}\frac{\|g_k\|^4}{\|d_k\|^2} \tag{28}$$

Holds for all sufficiently large k. Now inequality (21) follows from (2) and (19). ■

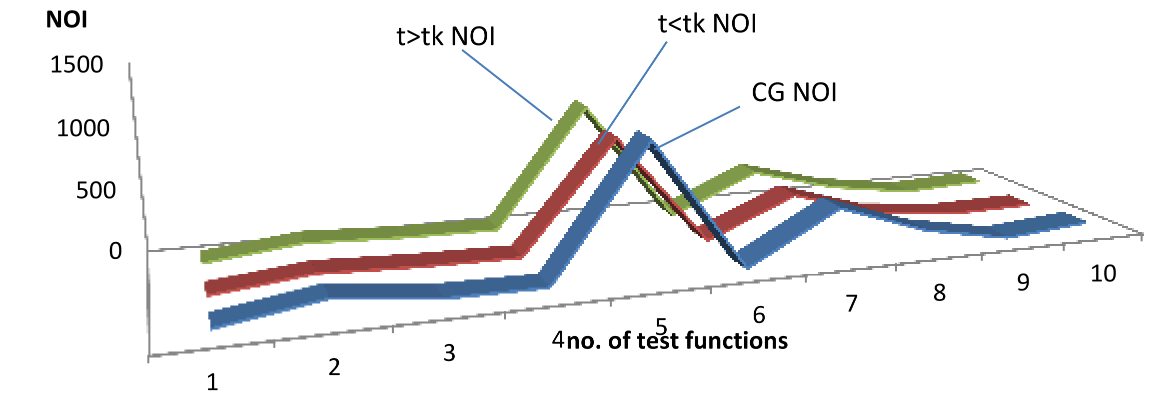
4. Numerical Result

In this section, we report some preliminary numerical experiments. The test problems are unconstrained problems in [J.Nocedal & J.S.Wright 2006] test problems used Math Lab (MATLAB 7.14) to library. We stop the iteration if the inequality $\|g_k\| \leq 10^{-6}$ is satisfied. Table 1 lists the number of iteration NOI and the number of function evaluation NOF for four different dimensions testing of each problem used algorithm (2.1) compared with new algorithm $t > t_k$ & $t < t_k$. where figure (1), (2),…, (8) shows the performance of the new algorithm .that is, we plot the ten well-known test problems used at index for four different dim $n = 5,10,100,1000$ .the left side of the figure gives the percentage of the test problem for which algorithm is faster used number of iteration & number of evaluation function all test function when take scalar $t < t_k$ is some efficient and stable.

Table 1: Comparison between the CG algorithm (2.1) and new algorithm (3.2) using different value of scalar t , t_k and four different value of n dimension the tools for each test functions

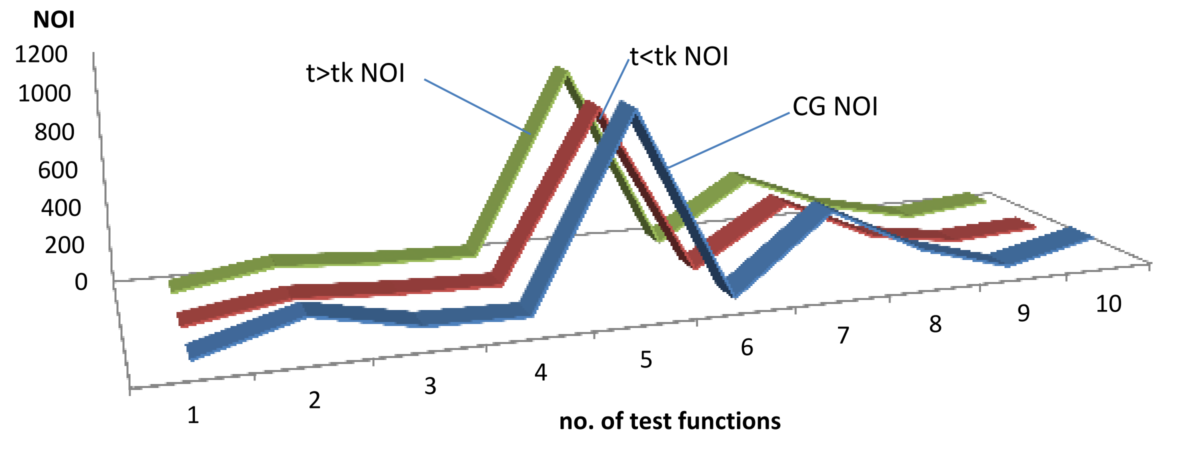
no. of test	Test functions	n-dim	CG algorithm(2.1)		New algorithm(3.2)			
					with $t < t_k$		with $t > t_k$	
			NOI	NOF	NOI	NOF	NOI	NOF
1	Generalized Rosen Brock	5	48	117	17	89	22	94
		10	37	130	17	89	22	94
		100	39	140	17	89	26	98
		1000	39	140	17	89	26	98
2	Extended Powell	5	165	331	96	212	102	224
		10	181	363	96	212	104	226
		100	189	379	98	226	116	242
		1000	199	399	103	318	118	263
3	Extended Tridiagonal -1	5	68	140	63	138	66	151
		10	68	140	63	138	66	151
		100	70	156	63	138	66	151
		1000	70	156	63	138	66	151
4	Extended Tridiagonal -2	5	57	142	48	124	50	128
		10	54	142	48	124	50	128
		100	58	120	50	132	52	132
		1000	58	120	50	132	52	132
5	Dquadratic	5	1080	2051	928	1820	1020	2041
		10	1025	2051	928	1821	1020	2041
		100	1305	2611	1021	1861	1020	2051
		1000	1305	2611	1024	1872	1305	2610
6	Extended Fred	5	26	59	20	48	22	54
		10	26	59	20	48	22	54
		100	27	61	22	51	24	56
		1000	27	61	22	51	24	56
7	Extended Freudenstein and Roth	5	422	763	326	642	330	652
		10	422	763	326	642	330	652
		100	449	817	328	644	396	702
		1000	506	931	328	644	396	702
8	Biggsb1	5	166	207	94	104	129	198
		10	168	210	94	104	129	198
		100	170	220	116	118	123	182
		1000	178	234	116	118	123	182
9	Mill and Cornwell	5	23	25	17	20	18	22
		10	23	25	17	20	18	22
		100	24	26	19	21	18	22
		1000	24	26	19	21	18	22

10	General non diagonal	5 10 100 1000	73 129 343 385	222 302 803 826	23 44 46 48	63 138 142 146	44 63 68 129	138 123 129 302



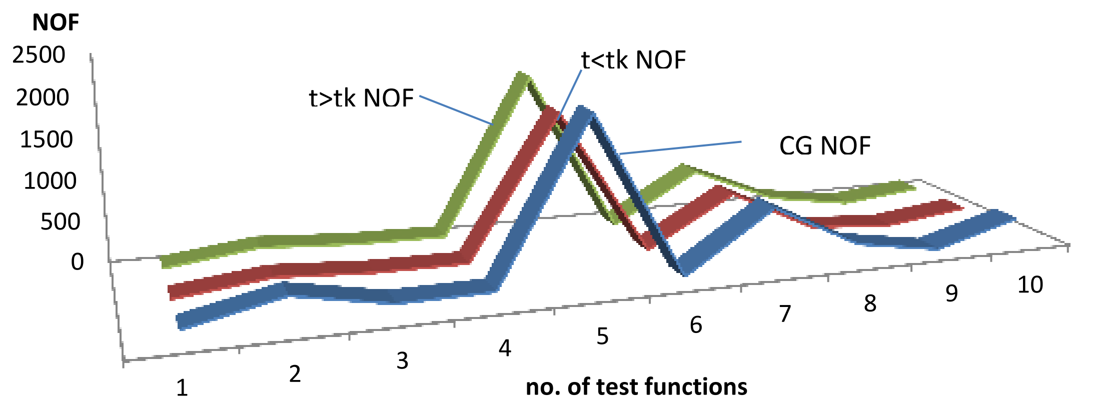
	1	2	3	4	5	6	7	8	9	10
CG NOI	48	165	68	57	1080	26	422	166	23	73
t< tk NOI	17	96	63	48	928	20	326	94	17	23
t>tk NOI	22	102	66	50	1020	22	330	129	18	44

Fig. 1 : comparison between the algorithm (2.1) and New algorithm (3.2) using different value of scalar using NOI number of iteration function of dimension 5 for each test functions



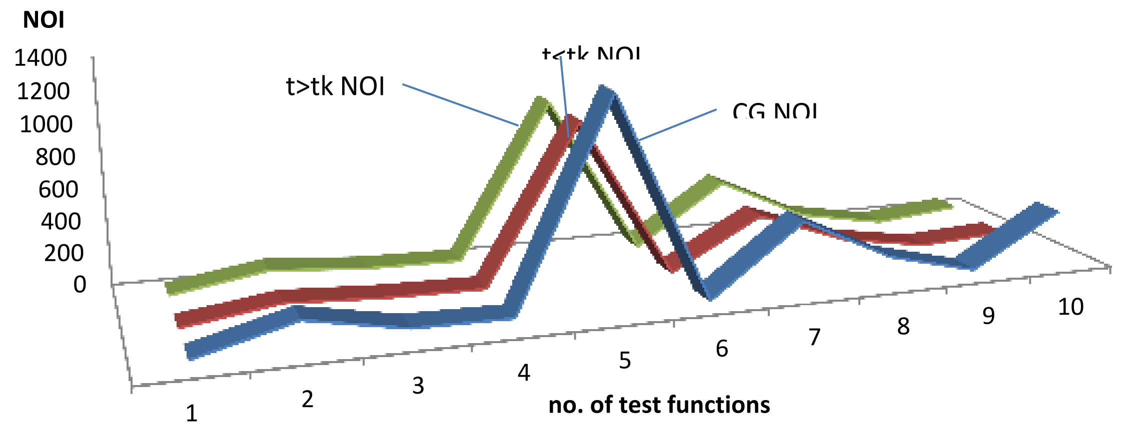
	1	2	3	4	5	6	7	8	9	10
CG NOI	37	181	68	54	1025	26	422	168	23	129
t<tk NOI	17	96	63	48	928	20	326	94	17	44
t>tk NOI	22	104	66	50	1020	22	330	129	18	63

Fig. 2: comparison between the algorithm (2.1) and New algorithm (3.2) using different value of scalar using NOI number of iteration function of dimension 10 for each test functions



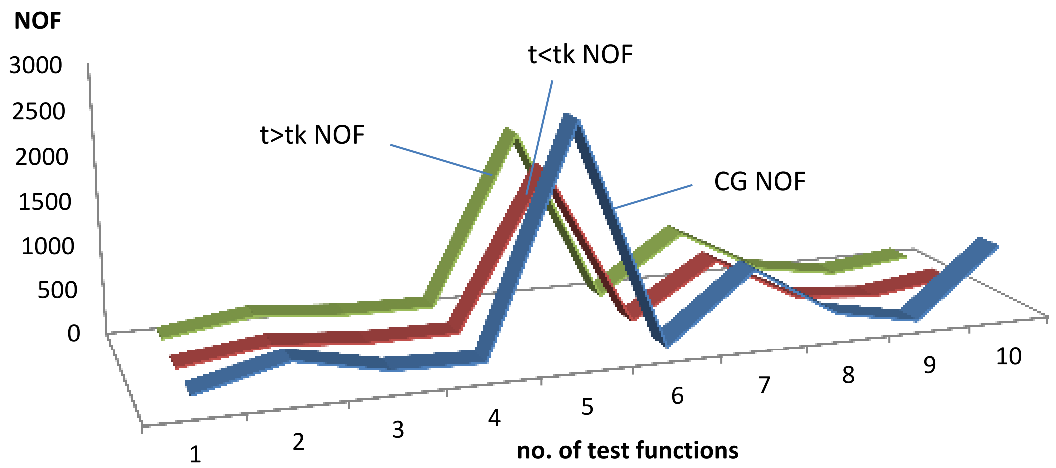
	1	2	3	4	5	6	7	8	9	10
CG NOF	130	363	140	142	2051	59	763	210	25	302
t<tk NOF	89	212	138	124	1821	48	642	104	20	138
t>tk NOF	94	226	151	128	2041	54	652	198	22	123

Fig. 3: comparison between the algorithm (2.1) and New algorithm (3.2) using different value of scalar using NOF number of iteration function of dimension 10 for each test functions



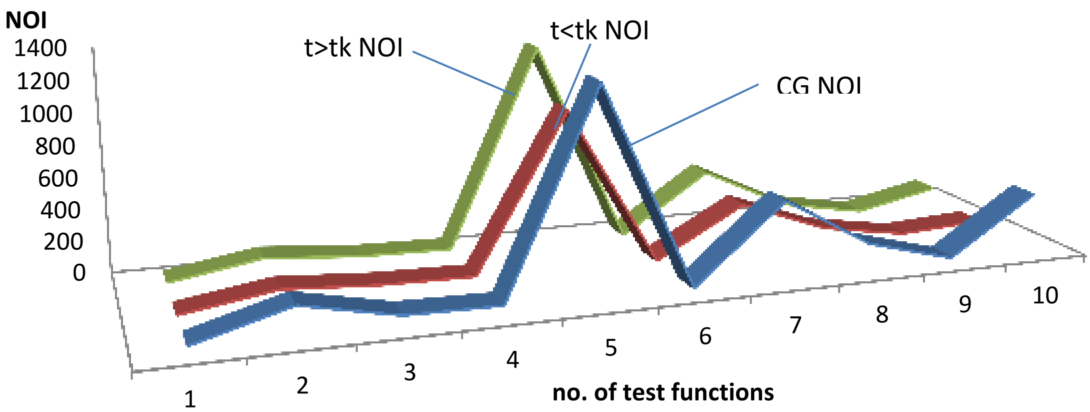
	1	2	3	4	5	6	7	8	9	10
CG NOI	39	189	70	58	1305	27	449	170	24	343
t<tk NOI	17	98	63	50	1021	22	328	116	19	46
t>tk NOI	26	116	66	52	1020	24	396	123	18	68

Fig. 4: comparison between the algorithm (2.1) and New algorithm (3.2) using different value of scalar using NOI number of iteration function of dimension 100 for each test functions



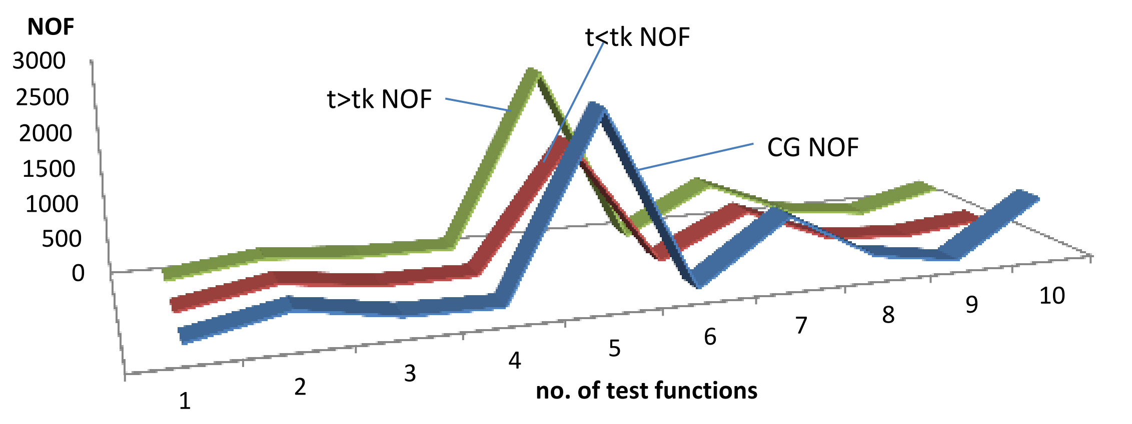
	1	2	3	4	5	6	7	8	9	10
CG NOF	140	379	156	120	2611	61	817	220	26	803
t<tk NOF	89	226	138	132	1861	51	644	118	21	142
t>tk NOF	98	242	151	132	2051	56	702	182	22	129

Fig. 5: comparison between the algorithm (2.1) and New algorithm (3.2) using different value of scalar using NOF number of iteration function of dimension 100 for each test functions



	1	2	3	4	5	6	7	8	9	10
CG NOI	39	199	70	58	1305	27	506	178	24	385
t<tk NOI	17	103	63	50	1024	22	328	116	19	48
t>tk NOI	26	118	66	52	1305	24	396	123	18	129

Fig. 6: comparison between the algorithm (2.1) and New algorithm (3.2) using different value of scalar using NOI number of iteration function of dimension 1000 for each test functions



	1	2	3	4	5	6	7	8	9	10
CG NOF	140	399	156	120	2611	61	931	234	26	826
t<tk NOF	89	318	138	132	1872	51	644	118	21	146
t>tk NOF	98	263	151	132	2610	56	702	182	22	302

Fig. 7: comparison between the algorithm (2.1) and New algorithm (3.2) using different value of scalar using NOF number of iteration function of dimension 1000 for each test functions

5. Discussions

In this paper we have presented global convergence result for nonlinear conjugate gradient methods. Using modified for (Dai-Yuan) scalar and where the step length is computed that all the search directions are descent directions, we have established convergence results for the new modification of conjugate gradient

Finally, some of the numerical results have been reported, which show the effectiveness of the new formula.

Appendix

All the test functions used in this paper are from general literature Nocedal (1980-2006).

1. **Generalized Rosen Brock Banana function:**

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

2. **Extended Powell function:**

$$f(x) = \sum_{i=1}^{n/4}(x_{4i-3} + 10\, x_{4i-2})^2 + 5\,(x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4 \ ,$$
$$x_0 = (3,-1,0,1, \dots ,3,-1,0,1)^T$$

3. **Extended Tridiagonal-1 function:**

$$f(x) = \sum_{i=1}^{\frac{n}{2}}(x_{2i-1} + x_{2i} - 3)^2 + (x_{2i} - x_{2i} + 1)^4,$$
$$x_0 = (2,2, \dots ,2)^T$$

4. **Extended Tridiagonal -2 function:**

$$f(x) = \sum_{i=1}^{n-1}(x_i x_{i+1} - 1)^2 + c\,(x_i + 1)(x_{i+1} + 1),$$
$$x_0 = (2,2, \dots ,2)^T; c = 0.1$$

5. **Dquadratic function:**

$$f(x) = \sum_{i=1}^{n-2}(x_i^2 + c\, x_{i+1}^2 + d\, x_{i+2}^2\ )^2,$$
$$x_0 = (3,3, \dots ,3)^T \ ; \ c=100,\ d=100$$

6. **Extended Fred function:**

$f(x)=\sum_{i=1}^{n/2}(-13+x_{2i-1}+(5-x_{2i})+(x_{2i}-2)(x_{2i})\ )^2+\sum_{j=1}^{n/2}(-29+x_{2j-1}+(1-x_{2j})+(x_{2j}-14)(x_{2j})\ )^2$,
$x_0=(1.,2.,\dots,n)^T$

7. Extended Freudenstein and Roth function:

$f(x)=\sum_{i=1}^{n/2}(-13+x_{2i-1}+((5-x_{2i})x_{2i}-2)x_{2i}\ )^2+(-29+x_{2i-1}+((x_{2i}+1)x_{2i}-14)x_{2i}\ )^2$,
$x_0=(0.5,-2.,0.5,-2.\dots,0.5,-2.)^T$

8. Biggsb1 function:

$f(x)=(x_1-1)^2+\sum_{i=1}^{n-1}(x_{i+1}-x_i)^2+(1-x_n)^2$, $x_0=(1.,1.,\dots,1.,1.)^T$

9. Mill and Cornwell function:

$f(x)=\sum_{i=1}^{\frac{n}{4}}[\exp(x_{4i-3}+10\,x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2(x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4]$,
$x_0=(1.,2.,2.\dots,1.,2.,2.)^T$

10. General non diagonal function:

$f(x)=\sum_{i=2}^{n}[100(x_1-x_i^2)^2+(1-x_i)^2$,
$x_0=(-1.,\dots,-1.)^T$

References

[1] Beale, E., "Introduction to optimization"; Wiley Inter Science series in Discrete Mathematics and Optimization, (1988).

[2] Bazarad, M., Sherali, H. and Sheety, C.M.,"nonlinear Programing Theory and Application", Johan Wiley and Sons, New York,(1993).

[3] Bunday, B.D., "Basic Optimization Methods" Edward Arnold,Bedford square, London, (1984).

[4] Dai Y.H., Yuan Y., "A nonlinear conjugate gradient method with a strong global convergence property ", SIAM J.Optim, No. 10, pp. 177-182, (1999).

[5] Edwin K.P. and Stanislaw H.Zak., "An Introduction to optimization", Wiley inter Science series in Discrete Mathematics and Optimization 2nd Edition, (2001).

[6] Fletcher R., Reeves C., "Function minimization by conjugate gradients, Compute ", No. J.7, pp. 149-154, (1964).

[7] Fletcher R., "practical Method of optimization", vol (1): Unconstrained Optimization, 2nd Edition, John Wiley and Sons Ltd., New York, (1987).

[8] Yn G., Zhao Y., Wei Z., "A descent nonlinear conjugate gradient method for large scale unconstrained optimization", Applied Mathematics and computation, No. 187, pp. 636-643, (2007).

[9] Hestenes, M.R., "Conjugate Direction Method in Optimization", Spring-Verlage, New York, Heidderg, Berlin,(1980).

[10] Hestenses M.R., Stiefel I., "Method of conjugate gradient for solving linear equations", J.Res.Nati. Bur.Standerds No. 49, pp. 409-436, (1952).

[11] Nocedal J. and Wright S.J., "Numerical optimization" Springer-Verlag New York, (2006).

[12] Polak E., Ribere G., " Note sur , la convergence de directions conjugates", Rev.Francainse informate Recherché Optimization ,No. 3,pp. 35-43, (1969).

[13] Wolfe M.A., "numerical methods for Unconstrained Optimization", Van Nostrand Reihold Company Ltd,New York,(1978).

[14] Yaun Y., "Numerical Methods for Nonlinear Programming", Shanghai Scientific and Technical Publishers, Chinese, (1993).