



A modification of Dai and Yuan Method for Solving Unconstrained Optimization Problems

Hussein Kurdosh Aljarjary^{1,*}, Khalil K. Abbo², Edrees M Nori Mahmood³

^{1,2}Department of Mathematics, College of Computer Sciences and Mathematics, University of Mosul, Iraq

³College of Basic Education, University of Telafer, Iraq

Emails: hussein.21csp18@student.uomosul.edu.iq, kh_196538@yahoo.com, edreesnori@uomosul.edu.iq

Article information

Article history:

Received:12/2/2024

Accepted:31/5/2024

Available online:15/12/2024

Abstract

The conjugate gradient (CG) method is the most widely used and has an important role in solving unconstrained optimization problems over a large scale due to its simplicity and lack of need for memory. Whereas a new parameter was proposed that enhances the Dai-Yuan formula so that it achieves better numerical results, in addition to the overall regression and convergence properties of the proposed method. The proposed method was compared with classical methods.

Keywords:

Conjugate gradient method, global convergence, sufficient descent condition.

Correspondence:

Author: **Hussein Kurdosh Aljarjary**

Email: hussein.21csp18@student.uomosul.edu.iq

I. Introduction

Conjugate gradient (CG) method is a typical method to solve large scale unconstrained optimization problems in the form:

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

Where $f: \mathbb{R} \rightarrow \mathbb{R}^n$ is differentiable and continuously. Using the iterative form:

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

Where x_k is the k_{th} iterative point and d_k is the search direction, $\alpha_k > 0$ represented the step length and d_k is calculated by:

$$d_{k+1} = \begin{cases} -g_{k+1} & \text{if } k = 0 \\ -g_{k+1} + \beta_k d_k & \text{if } k > 0 \end{cases} \quad (3)$$

$\beta_k \in \mathbb{R}$ Defined as a coefficient for the CG method [1,2], Note that they are of various formats and have undergone many and various modifications. Here we mention the most famous of them: Fletcher-Reeves (FR)[3], Polark-Ribi`ere-Polyak (PRP)[4], Hestenes-Stiefel (HS)[5], conjugate-Descent (CD)[6], Liu-Storrey (LS)[7], and Dai-Yuan

(DY)[8].

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k},$$

Where g_k is the derivative of $f(x)$ at x_k and

$$y_k = g_{k+1} - g_k.$$

Let us now turn our attention to the line search. An efficient strategy. studied by wolf [9], consists in accepting a positive step length α is it satisfies:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g_k^T d_k \quad (4)$$

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k \quad (5)$$

Where $0 < \rho < \sigma < 1$. Often the strong – wolfe line search is used in the implementation of conjugate gradient methods. These are given by (5) and:

$$|g_{k+1}^T d_k| \leq -\sigma g_k^T d_k \quad (6)$$

The global convergence of the DY method has been proven. Noting that it was used many different lines search as exact line search and standred-wolfe line search (SWP) line search.

In our method, we used robust strong –Wolfe line search [11,

12]. The numerical experiments presented demonstrated good efficiency and competitiveness of the new method. The rest is divided as follows: In Section 2 we write the new formulation and algorithm based on modifications of the DY method. In Section 3 we prove the regression and global convergence properties of the proposed method. The numerical results and conclusions will be in Sections 4 and 5, respectively.

II. The new algorithm

Dai and Yuan [6] proved the global convergence of the conjugate gradient algorithm:

$$|\beta_k| = \theta \beta^{DY} \quad (7)$$

Where $\theta \in [-\frac{1-\sigma}{1+\sigma}, 1]$. on the other hand Dai and Liu in [DL] modified the conjugate condition to the

$y_k^T d_{k+1} = -t s_k^T g_{k+1}$ where $t > 0$ and $s_k = x_{k-1} - x_k$. Based on the eq. (7) and conjugated condition by Dai-Liu with suitable t we get:

$$\theta = \left(\frac{y_k^T g_{k+1}}{\|g_{k+1}\|^2} - \rho \frac{|y_k^T g_k| s_k^T g_{k+1}}{\|g_{k+1}\|^2 s_k^T y_k} \right) \text{ where } \rho > 0 \text{ and } \beta^{HDY} = \theta \beta^{DY}$$

Hence, we can write d_k

$$d_{k+1} = \begin{cases} -g_{k+1} + (\theta \beta^{DY} s_k) & \text{if } \theta \in (0,1) \\ -g_{k+1} + \beta^{DY} s_k & \text{O.w} \end{cases} \quad (8)$$

In this section we suggest a new algorithm by modified (DY) method as follows:

Consider $\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{y_k^T s_k}$ the proposed β_k^{HDY} will be as follows:

$$\beta_k^{HDY} = \left[\frac{g_{k+1}^T y_k}{y_k^T s_k} - \frac{\rho |y_k^T g_{k+1}| s_k^T g_{k+1}}{(y_k^T s_k)^2} \right] \quad (9)$$

Where $\rho > 0$, and $d_{k+1} = -g_{k+1} + \beta^{HDY} s_k$

The proposed algorithm will be as follows:

2.1 Algorithm:

Step (1): Choose an initial point x_0 and ε sufficiently small. Set $d_0 = -g_0$ and $k = 0$.

Step (2): Test a criterion for stopping the iterations. If this test is satisfied, then stop; otherwise, continue with step (3).

Step (3): Using the Wolfe line search conditions, determine the step size α_k and Compute

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

Step (4): Compute the conjugate gradient parameter β_k^{HDY} by eq. (9)

Step (5): Compute the search direction

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

Step (6): Restart criterion. $f |g_{k+1}^T g_k| >$

$0.2 \|g_{k+1}\|^2$ or $k = n$ then set

$d_{k+1} = -g_{k+1}$ and go to step (2)

Step (7): Set $k = k + 1$ and go to step(3)

III. Global Convergence Properties

To prove the global convergence of the suggested conjugate gradient method, we will assume the following.

Assumption 3.1:(see [10])

the set $\Omega = \{x \in \mathbb{R}^n, f(x) \leq f(x_0)\}$ is bounded if x is the initial point. For some neighborhood N of the set Ω , assume g be Lipchitz continuous on Ω , i.e., $\exists$ a positive constant $L > 0$, such that:

$$\|g(x) - g(y)\| \leq L \|x - y\|, \forall x, y \in \Omega \quad (10)$$

In addition to this assumption, we may deduce that a positive constant r exists, such that:

$$\|g(x)\| \leq r, \forall x \in \Omega \text{ and } \beta > 0, |x| \leq \beta \quad (11)$$

Moreover, when the function is uniformly convex, the following relationship will hold:

$$y_k^T s_k \geq \mu \|s_k\|^2 \quad (12)$$

Lemma 1. If we presume that α_k satisfies Strong Wolfe-Powell (SWP), then the

direction of the search, d_{k+1} , which is produced by:

$$d_{k+1} = -g_{k+1} + \beta_k^{HDY} s_k \quad (13)$$

Satisfy the condition:

$$d_{k+1}^T g_{k+1} \leq -c_1 \|g_{k+1}\|^2 \quad (14)$$

Proof: from (13), we get:

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k s_k}{s_k^T y_k} - \frac{\rho |g_{k+1}^T y_k| s_k^T g_{k+1}}{(s_k^T y_k)^2} s_k \quad (15)$$

by multiplying through (15) by (g_{k+1}^T) gives:

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

Since $s_k^T g_{k+1} \leq s_k^T y_k, g_{k+1}^T y_k \leq \|g_{k+1}\| \cdot \|y_k\|$, we get :

$$\begin{aligned} &\leq -\|g_{k+1}\|^2 + \frac{(g_{k+1}^T y_k)(s_k^T y_k)}{s_k^T y_k} - \frac{\rho |g_{k+1}^T y_k| (s_k^T y_k)^2}{(s_k^T y_k)^2} \\ &\leq -\|g_{k+1}\|^2 + \|g_{k+1}\| \|y_k\| - \rho \|g_{k+1}\| \|y_k\| \\ &\leq -\left(1 - \frac{\|y_k\|}{\|g_{k+1}\|} + \frac{\rho \|y_k\|}{\|g_{k+1}\|}\right) \|g_{k+1}\|^2 \end{aligned}$$

Since $0 < \frac{\|y_k\|}{\|g_{k+1}\|} < 1$, let $c_1 = \left(1 - \frac{\|y_k\|}{\|g_{k+1}\|} + \frac{\rho \|y_k\|}{\|g_{k+1}\|}\right)$

$$\therefore d_{k+1}^T g_{k+1} \leq -c_1 \|g_{k+1}\|^2$$

Theorem 1: Assuming assumption (3.1) is valid, α_k

satisfies Strong Wolfe-Powell (SWP) conditions, and

Lemma 1 holds, Then,

$$\lim_{k \rightarrow \infty} \|g_k\| = 0, \forall k \geq 0 \quad (16)$$

Proof: Reformulating the equation for determining the search direction (13) by

$$d_{k+1} = -g_{k+1} + \beta_k^{HDY} s_k \quad (17)$$

Where β_k^{HDY} is defined by eq. (9), we get

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k s_k}{s_k^T y_k} - \frac{\rho |g_{k+1}^T y_k| s_k^T g_{k+1} s_k}{s_k^T y_k s_k^T y_k}$$

Since $0 < \beta < 1 \rightarrow \beta^{HS} = c_2$, we get

$$\|d_{k+1}\| \leq \|g_{k+1}\| + c_2 \|s_k\| - \rho c_2 \frac{s_k^T y_k}{s_k^T y_k} s_k$$

Table 1. Comparison concerning (NOI and NOF) for dimensions n=1000, 5000, and 10000.

Function	N	Problem	Alg. DY		Alg. CD		Alg. PRP		Alg. HDY	
			Iter.	Fg.	Iter.	Fg.	Iter.	Fg.	Iter.	Fg.
1	1000	Freudenstein & Roth	15	33	23	39	14	30	35	584
	5000		13	27	1003	27820	21	43	35	485
	10000		13	28	525	15034	20	42	1003	28183
2	1000	Extended Rosenbrock SROSENBR (CUTE)	41	82	40	75	44	89	35	73
	5000		41	90	47	91	41	87	36	79
	10000		40	86	43	97	41	87	36	78
3	1000	Extended White & Holst	35	69	41	87	35	72	32	68
	5000		34	72	34	69	41	91	36	80
	10000		34	72	38	81	38	81	27	55
4	1000	Extended Beale BEALE (CUTE)	15	28	15	29	15	28	14	27
	5000		13	24	17	31	16	30	15	28
	10000		14	27	14	26	16	31	15	28
5	1000	Hager	29	51	32	53	28	46	27	44
	5000		121	2379	198	5105	139	3139	301	8555
	10000		499	14538	595	17952	451	13310	289	7748
6	1000	Generalized Tridiagonal 1	23	45	23	45	23	45	28	187
	5000		24	49	29	140	25	49	26	48
	10000		30	130	64	1236	33	125	29	52
7	1000	Extended Three Expo Terms	20	122	11	17	9	14	15	24
	5000		26	408	13	21	8	14	21	32
	10000		13	75	105	2927	38	515	18	29
8	1000	Extended Himmelblau HIMMELBC (CUTE)	11	20	11	20	11	20	11	20
	5000		23	36	12	22	11	20	11	20
	10000		23	36	23	36	12	22	12	22

$$\leq \omega + c_2 \|s_k\| + \rho c_2 \|s_k\| = D_2 \quad (18)$$

$$\sum_{k \geq 1} \frac{1}{\|d_{k+1}\|^2} \geq \frac{1}{D_2^2} \sum_{k \geq 1} 1 = \infty$$

IV. Numerical results

In this section, we presented our numerical results that explain the new algorithm specified by equation (9) is compared to the classical algorithm CD [6] as well as algorithm PRP [4] and algorithm DY [8]. The numerical results showed clear superiority over other algorithms with

regard to function evaluations and number of iterations. Many researchers have adopted the principle of modification as well as hybridization to obtain more accurate results. The specific parameters are as follows: $\rho = 0.5$, $\varepsilon = 10^{-5}$, and the stopping condition is $\|g_{k+1}\| \leq 10^{-6}$. The results are shown in **Table 1**.

Function	N	Problem	Alg. DY		Alg. CD		Alg. PRP		Alg. HDY	
9	1000	Extended Powell	Iter.	Fg.	Iter.	Fg.	Iter.	Fg.	Iter	Fg.
	5000		403	715	433	739	1003	1489	309	539
	10000		1003	1323	410	749	1003	1166	528	911
10	1000	Extended PSC1	357	639	502	826	1003	1096	651	1275
	5000		81	153	84	160	65	124	77	143
	10000		77	137	78	146	72	132	92	173
11	1000	Extended Maratos	99	182	70	129	72	145	89	165
	5000		72	158	78	174	67	157	68	162
	10000		64	139	106	251	79	182	113	349
12	1000	Full Hessian FH1	98	228	120	402	74	165	65	150
	5000		12	30	11	29	1003	1036	12	31
	10000		13	27	13	26	17	33	10	23
13	1000	Full Hessian FH2	17	50	14	34	16	39	16	49
	5000		81	177	86	186	86	186	80	175
	10000		81	176	87	187	86	185	78	170
14	1000	Full Hessian FH3	81	174	91	193	86	183	81	174
	5000		94	195	94	145	96	148	100	152
	10000		258	399	254	396	245	391	224	346
15	1000	Tridiagonal White & Holst (c=4)	366	578	34	535	341	540	30	471
	5000		25	52	29	61	27	58	24	55
	10000		72	163	76	182	75	197	72	219
16	1000	BDQRTIC (CUTE)	38	91	37	88	37	88	36	86
	5000		16	31	12	22	12	23	12	23
	10000		13	25	12	23	13	25	13	25
17	1000	FLETCHCR (CUTE)	16	32	13	26	17	33	15	29
	5000		86	134	80	123	78	123	83	129
	10000		184	284	220	346	163	259	17	286
18	1000	ENGVAL1 (CUTE)	225	352	250	394	239	373	253	393
	5000		88	137	91	136	8	139	81	119
	10000		188	292	199	316	197	317	13	345
19	1000	DENSCHNA (CUTE)	263	426	72	454	227	376	33	394
	5000		31	48	33	51	30	52	29	53
	10000		41	65	39	65	42	68	40	64
20	1000	DENSCHNA (CUTE)	38	63	34	64	40	67	35	61
	5000									
	10000									

Function	N	Problem	Alg. DY		Alg. CD		Alg. PRP		Alg. HDY	
			Iter.	Fg.	Iter.	Fg.	Iter.	Fg.	Iter.	Fg.
20	1000	DENSCHNB (CUTE)	105	162	99	151	104	163	105	159
	5000		246	384	222	349	229	373	209	322
	10000		336	533	375	595	324	519	298	469
21	1000	DENSCHNC (CUTE)	107	167	103	161	102	164	99	157
	5000		244	387	258	404	288	456	232	363
	10000		356	547	344	546	330	525	299	474
22	1000	QP1 Extended Quadratic Penalty	16	27	15	27	17	30	14	26
	5000		10	20	10	20	10	20	10	20
	10000		14	27	14	25	13	25	13	25
23	1000	PRODCos	36	61	34	59	32	55	35	56
	5000		36	60	32	57	36	61	36	58
	10000		36	56	34	57	36	59	39	64
24	1000	Extended Trigonometric ET1	35	97	30	62	25	84	20	36
	5000		76	121	73	113	71	114	64	106
	10000		39	66	36	65	39	66	30	46
25	1000	Perturbed Quadratic	124	187	113	172	110	152	99	148
	5000		253	404	267	395	246	387	230	377
	10000		420	486	437	597	445	701	346	547
26	1000	Raydan 1	76	117	85	131	80	126	70	110
	5000		238	407	268	440	675	991	207	330
	10000		506	952	452	880	638	981	407	755
27	1000	TR- SUMMBEAL	315	728	470	1126	305	6678	287	437
	5000		213	324	217	331	216	330	115	218
	10000		164	327	184	426	178	331	137	328
28	1000	Diagonal 1	65	107	69	111	59	106	55	104
	5000		140	233	157	259	152	250	122	206
	10000		214	350	219	352	220	374	197	353
29	1000	Extended Tridiagonal 1	11	22	11	22	14	27	11	22
	5000		12	23	12	23	12	23	12	23
	10000		16	30	14	27	14	27	13	26
30	1000	Tridiagonal Double Bordered Arrow Up	15	29	14	27	11	20	9	18
	5000		11	22	11	22	10	20	8	16
	10000		19	38	17	34	15	30	11	22

Function	N	Problem	Alg. DY		Alg. CD		Alg. PRP		Alg. HDY	
31	1000	Extended Cliff	Iter.	Fg.	Iter.	Fg.	Iter.	Fg.	Iter.	Fg.
	5000		50 110	89 202	54 135	92 236	55 110	96 192	46 103	80 202
	10000		156	273	181	317	158	278	191	342
32	1000	Quadratic Diagonal Perturbed	30 25	60 49	36 28	59 54	30 25	56 49	28 21	56 46
	5000		27	52	28	50	34	64	26	51
	10000									
33	1000	Diagonal Double Bordered Arrow Up	112 282	174 435	119 278	179 429	116 1003	181 1170	103 238	161 372
	5000		382	601	408	649	395	625	341	539
	10000									
34	1000	TRIDIA (CUTE)	3 3	6 6	3 3	6 6	3 3	6 6	3 3	6 6
	5000		3	6	3	6	3	6	3	6
	10000									
35	1000	ARWHEAD (CUTE)	41 35	61 53	40 48	67 71	39 48	65 74	38 58	61 547
	5000		39	56	78	1313	63	519	41	69
	10000									
36	1000	Extended Hiebert	9 10	17 76	10 9	37 54	9 8	17 14	20 10	58 76
	5000		18	178	9	57	26	260	7	14
	10000									
37	1000	Almost Perturbed Quartic	17 35	37 70	15 31	31 65	13 22	26 46	11 15	22 26
	5000		36	78	34	69	41	82	30	68
	10000									
38	1000	DENSCHNF (CUTE)	27 40	48 335	26 85	49 1987	26 109	46 2651	25 26	46 68
	5000		41	280	34	233	40	211	62	127
	10000									
39	1000	HIMMELBG (CUTE)	421 411	665 630	442 387	703 590	470 361	748 560	403 323	723 525
	5000		390	467	456	826	395	733	423	759
	10000									
40	1000	HIMMELBH (CUTE)	18 20	34 44	18 21	34 46	15 24	31 53	22 20	41 44
	5000		32	67	19	41	20	45	21	46
	10000									
Total	NOI		15518		13463		17567		11788	
	NOF		81677		52343		79940		36731	

V. Conclusion

In his paper, the (Dai-Yuan, 1999) algorithm was modified to solve unconstrained optimization problems. This modified proved its effectiveness through numerical results, and the regression and global convergence of the proposed method was proven.

Acknowledgement

The authors would express they're thanks to the College of Computers Sciences and Mathematics, University of Mosul to support this report.

References

- [1] G. H. Yu, Nonlinear self-scaling conjugate gradient methods for large-scale optimization problems, Doctorial thesis, Sun Yat-Sen University, Guangzhou, China, 2007.
- [2] G. Yuan and Z. Wei, "New line search methods for unconstrained optimization," *Journal of the Korean Statistical Society*, vol. 38, no. 1, pp. 29–39, 2009.
- [3] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal* vol. 7, pp. 149–154, 1964.
- [4] E. Polak and G. Ribiere, "Note sur la convergence de m `ethodes de directions conjugues," *Revue Francaise Informatique de Recherche Operiionelle*, vol. 3, no. 16, pp. 35–43, 1969.
- [5] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.
- [6] R. Fletcher, *Practical Methods of Optimization. Vol. 1: Unconstrained Optimization*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1997.
- [7] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms. I. Theory," *Journal of Optimization Theory and Applications*, vol. 69, no. 1, pp. 129–137, 1991.
- [8] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.
- [9] P. Wolf : Convergence Conditions for ascent methods. *SIAM Rer.*,11 (1969).
- [10] H. Yabe and N. Sakaiwa, *Journal of the Operations Research Society of Japan* 48, 284–296 (2005).
- [11] Mahdi, M. M. and Shiker, M. A. K., (2020), Three- Term of New Conjugate Gradient Projection Approach under Wolfe Condition to Solve Unconstrained Optimization Problems, *Journal of Advanced Research in Dynamical and Control Systems*, 12: 7, pp. 788- 795.
- [12] Wasi, H. A. and Shiker, M. A. K., (2020), A new hybrid CGM for unconstrained optimization problems, "in press", accepted paper for publication in *IOP Science*, 1st International Virtual Conference on Pure Sciences (IVCPS)- Iraq.