Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM)

www.csmj.mosuljournals.com

New investigation Development of the Dai-Yuan method for solving Unconstrained Optimization Problems

Hussein Ali AL-Moussawi[1,*], Khalil K. Abbo[2]

[1,2]*Department of Mathematics, College of Computer Science and Mathematics, University of Mosul, Iraq*
Emails: *hussein.21csp25@student.uomosul.edu.iq* , *kh_196538@yahoo.com*

Article information

Article history:
Received :12/2/2024
Accepted :25/5/2024
Available online :15/12/2024

Abstract

This paper proposes a novel parameter modification to the classical Dai-Yuan conjugate gradient (CG) method to enhance its efficiency in solving large-scale unconstrained optimization problems. The new approach maintains simplicity, low memory usage, and theoretical robustness, making it suitable for high-dimensional applications. A detailed theoretical analysis proves global convergence and descent properties under standard assumptions. Numerical experiments on benchmark problems demonstrate that the proposed method outperforms classical CG algorithms by delivering more accurate solutions with fewer iterations. These findings confirm the method's potential as an effective and reliable alternative, offering improved numerical stability and faster convergence for large-scale optimization tasks.

Keywords:
Conjugate gradient method, global convergence, sufficient descent condition.

Correspondence:
Author: Hussein Ali Al-Moussawi
Email: *hussein.21csp25@student.uomosul.edu.iq*

II. Introduction

The conjugate gradient (CG) approach may be useful for large-scale unconstrained optimization problems. The second derivative or an approximation of it is not required, which is an advantage over Newton's technique. It is very simple and straightforward to use the conjugate gradient approach. Let us have a look at it:

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

When $f(x): R^n \to R$ is continuously differentiable and confined below; the iteration approach is used, which is expressed as:

$$x_{k+1} = x_k + \alpha_k d_k, k = 0,1,2, \ldots \tag{2}$$

The step size α_k it may be calculated using the following equations, which represent a Strong Wolfe-Powell (SWP) line search:

$$f(x_k + \alpha_k d_k) \le c_1 \alpha_k \nabla f(x_k)^T d_k \tag{3}$$

$$|g_{k+1}^T d_k| \le -\sigma g_k^T d_k \tag{4}$$

Where $g_k = g(x_k) = \nabla f(x_k), and\ 0 < c_1 < \sigma < 1$ (5)

is a search direction defined using the CG method:

$$d_{k+1} = \begin{cases} -g_{k+1}, & k = 0 \\ -g_{k+1} + \beta_k d_k, & k \ge 1 \end{cases} \tag{6}$$

Where $\beta_k \in R$ stands for the conjugate gradient update parameter. There are several classes of CG techniques, for example, in [1] and [2]. The most well-known CG techniques are as follows: Hestenes-Steifel (HS) [3], Fletcher Reeves (FR) [4], Polak-Ribeire (PR) [5][6] , Dai Yuan (DY) [7], and Dai and Liao [8], A nonnegative parameter t was included into Perry's condition [9]:

$$d_{k+1}^T y_k = -t g_{k+1}^T s_k, \tag{7}$$

Where $s_k = x_{k+1} - x_k, and\ t > 0$.

Later, researchers were inspired by the **DL** approach and suggested versions of the **DL** method by different selections of t_k [10], [11], [12]. This follows the rest of the article: In **Section 2**, we derived a new parameter called β_k that represents an improvement to the **DL-CG** procedure.

Section 3 examines a new formula that uses the robust Wolfe line search and analyses its descent and global convergence. **Section 4** presents numerical results, and Section **5** describes Conjugate gradient method implementation. Finally, the conclusion is in **section 6.**

II. Derivation of New Parameter

Solving nonlinear unconstrained optimization problems, the proposed method uses a parameter essential in finding optimal problem optimization with less computational effort based on theoretical analysis and numerical results. The most significant distinction between the spectral gradient approach and the conjugate gradient method is calculating the search path. The spectral gradient method's search path is as follows:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k^{DY} s_k \quad (8)$$

Where $s_k = \alpha_k d_k$ *and* θ_{k+1} *interpolation is the spectral parameter.We multiply the above eq. by* y_k *we get:*

$$d_{k+1}^T y_k = -\theta_{k+1} g_{k+1}^T y_k + \frac{\|g_{k+1}\|^2}{s_k^T y_k} s_k^T y_k \quad (9)$$

Using eq.(7), we get:

$$-t s_k^T g_{k+1} = -\theta_{k+1} g_{k+1}^T y_k + \frac{\|g_{k+1}\|^2}{s_k^T y_k} s_k^T y_k \quad (10)$$

Take $t = \frac{y_k^T s_k}{\|s_k\|^2} + \frac{\|y_k\|}{\|s_k\|}$ *defined from Andrei (Andrei, 2018) and subs. In eq.(10) we get:*

$$-\frac{y_k^T s_k s_k^T g_{k+1}}{\|s_k\|^2} - \frac{\|y_k\| s_k^T g_{k+1}}{\|s_k\|} = -\theta_{k+1} g_{k+1}^T y_k + \frac{\|g_{k+1}\|^2}{s_k^T y_k} s_k^T y_k$$

Then the new spectral is known as the following shape:

$$\theta_{k+1} = \frac{\|g_{k+1}\|^2}{y_k^T g_{k+1}} + \frac{y_k^T s_k s_k^T g_{k+1}}{\|s_k\|^2 y_k^T g_{k+1}} + \frac{\|y_k\| s_k^T g_{k+1}}{\|s_k\| y_k^T g_{k+1}} \quad (11)$$

If we use ELS (exact line search) the spectral θ_{k+1} is equal to one and the direction is become classical CG direction.

III. Global Convergence Properties

An efficient strategy will have to fulfill both the descent requirement and the convergence condition. it's necessary to make the following assumptions:

Assumptions 3.1: [13] The level set $\Omega=\{x|f(x) \le f(x_1)|\}$ is bounded ; meaning that there exists a positive constant p such that:

$$\|x\| \le p \,, \forall x \in \Omega \quad (12)$$

In some neighbourhood $Q \; of \; \Omega$, f is continuously differentiable, and its gradient is Lipschitz continuous; that is ; for all $x, y \in Q$, there exists a constant $L > 0$ such that:

$$\|g(x) - g(y)\| \le L|x - y| \quad (13)$$

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

$$\|g(x)\| \le B \,, \forall x \in \Omega \quad (14)$$

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

$$y_k^T s_k \ge \mu \|s_k\|^2 \quad (15)$$

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:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k^{DY} d_k \quad (16)$$

And θ_{k+1} is defined by (11) Satisfy the condition:

$$d_{k+1}^T g_{k+1} \le 0 \quad (17)$$

Proof: From (16), we get:

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

Multiply the above eq. by $\frac{g_{k+1}^T}{\|g_{k+1}\|^2}$ we get:

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

Since $y_k^T g_{k+1} \le \|y_k\| . \|g_{k+1}\|$, $s_k^T g_{k+1} \le -\sigma s_k^T g_k$

$$\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} \le \frac{-\|g_{k+1}\|^2}{\|y_k\| \|g_{k+1}\|} + \frac{\sigma y_k^T s_k s_k^T g_k}{\|s_k\|^2 \|y_k\| \|g_{k+1}\|} + \frac{\sigma \|y_k\| s_k^T g_k}{\|s_k\| \|y_k\| \|g_{k+1}\|} - \frac{\sigma s_k^T g_k}{s_k^T y_k}$$

Since $s_k^T g_k \le \frac{-s_k^T y_k}{(\sigma+1)}$ we get:

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

$$\le \frac{\sigma}{(\sigma+1)} - \frac{\sigma}{\|s_k\| \|g_{k+1}\| (\sigma+1)}$$

$$\le \frac{-\sigma}{(\sigma+1)\|s_k\| \|g_{k+1}\|} \left(1 - \|s_k\| \|g_{k+1}\|\right)$$

$$\therefore d_{k+1}^T g_{k+1} \le 0 \quad (18)$$

Theorem 1: Assuming assumption (3.1) is valid, α_k satisfies Strong Wolfe Powell (SWP) conditions, and Lemma 1 holds, Then

$$\lim_{k\to\infty} inf\|g(x)\| = 0 \quad (19)$$

Proof: The search direction is determined by:

$$d_{k+1} = -\theta_{k+1}g_{k+1} + \beta_k^{DY}d_k \quad (20)$$

Where θ_{k+1} is defined by eq.(11)

$$|\theta_{k+1}| = \left|\frac{-\|g_{k+1}\|^2}{y_k^T g_{k+1}} - \frac{y_k^T s_k s_k^T g_{k+1}}{\|s_k\|^2 y_k^T g_{k+1}} - \frac{\|y_k\| s_k^T g_{k+1}}{\|s_k\| y_k^T g_{k+1}}\right| \quad (21)$$

$$|\theta_{k+1}| \le \left|\frac{-\|g_{k+1}\|^2}{y_k^T g_{k+1}}\right| + \left|\frac{-y_k^T s_k s_k^T g_{k+1}}{\|s_k\|^2 y_k^T g_{k+1}}\right| + \left|\frac{-\|y_k\| s_k^T g_{k+1}}{\|s_k\| y_k^T g_{k+1}}\right|$$

Since $y_k^T g_{k+1} \le \|y_k\|\|g_{k+1}\|$, $s_k^T g_{k+1} \le s_k^T y_k$

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

Since $s_k^T y_k \le L\|s_k\|^2$, we get

$$\|\theta_{k+1}\| \le \frac{\|g_{k+1}\|}{\|y_k\|} + \frac{L^2\|s_k\|^4}{\|s_k\|^2\|y_k\|\|g_{k+1}\|} + \frac{L\|y_k\|\|s_k\|^2}{\|s_k\|\|y_k\|\|g_{k+1}\|}$$

$$\|\theta_{k+1}\| \le \frac{\|g_{k+1}\|}{\|y_k\|} + \frac{L^2\|s_k\|^2}{\|y_k\|\|g_{k+1}\|} + \frac{L\|s_k\|}{\|g_{k+1}\|} = \delta$$

Since $|\beta_k| = \frac{\|g_{k+1}\|^2}{s_k^T y_k} \le \varphi$

$$\therefore \|d_{k+1}\| \le |\theta_{k+1}|\|g_{k+1}\| + |\beta_k|\|s_k\| \quad (22)$$

$$\|d_{k+1}\| \le \delta\|g_{k+1}\| + \varphi\|s_k\| = C_3$$

$$\sum_{k\ge1}\frac{1}{\|d_{k+1}\|^2} \ge \frac{1}{C_3^2}\sum_{k\ge1} 1 = \infty$$

∴ It is global

IV. Numerical β^{KDY}:

In this research we will find results for each of four methods: the new algorithm specified by equation (11) is compared to the classical DY algorithm (Dai and Yuan ,1999) as well as AMDYN algorithm (Andrei,2010), which define θ_{k+1} as follows:

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

And HDY algorithm[14] , which define θ_{k+1} as follows :

$$\theta_{k+1} = \frac{\|g_{k+1}\|^2}{y_k^T g_{k+1}} + \left(\frac{y_k^T y_k}{s_k^T y_k}\right)\frac{s_k^T g_{k+1}}{y_k^T g_{k+1}}$$

We chose 31 unconstrained optimizations in the range [n = 1000,5000,10000],broadly and based on generalized (Andrei,2008a) .All algorithms used Wolfe condition c_1 = 0. 9, δ = 0.0001. The codes are adopted with double precision and using the Fortran language. We applied the performance profile of Dolan and More (Dolan and More ,2002) to demonstrate the algorithm's efficiency, this proposed method has been shown to be better than other methods in terms of the number of functions, iterations, and time.

V. Conclusion

In his research paper, a method (Dai-Yuan, 1999) was developed to solve unconstrained optimization problems. This development proved its effectiveness through the numerical results in Table (1-1), and the regression and global convergence of the proposed method were also 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] M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Standards, 49, no. 6, (1952), 409–435.

[2] R. Fletcher, M. Colin, Function minimization by conjugate gradients. The computer Journal, Oxford University Press, 7, no. 2, (1964), 14-154

[3] Elijah Polak, Gerard Ribiere, Note sur la convergence de methodes de directions conjuguees, The computer Journal, 3, no. R1, (1969), 35–43.

[4] Boris Teodorovich Polyak,, The conjugate gradient method in extremal problems, USSR Computational Mathematics and Mathematical Physics, Elsevier, 9, no. 4, (1969), 94–112.

[5] Yuhong Dai, Jiye Han, Guanghui Liu, Defeng Sun, Hongxia Yin, Yaxiang Yuan, Convergence properties of nonlinear conjugate gradient methods, SIAM Journal on Optimization, 10, no. 2 ,(2000), 345–358.

[6] B. T. Polyak, "The conjugate gradient method in extremal problems," USSR Comput . Math. Math. Phys., Vol. 9, No. 4, pp. 94–112 (1969),doi:10.1016/0041-5553(69)90035-4.

[7] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," SIAM J. Optim., Vol. 10, No. 1, pp. 177–182 (1999), doi: 10.1137/S1052623497318992.

[8] Y. H. Dai and L. Z. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," Appl. Math. Optim., Vol. 43, No. 1, pp. 87–101 (2001), doi: 10.1007/s002450010019.

[9] A. Perry, "Technical Note—A Modified Conjugate Gradient Algorithm," Oper. Res., Vol. 26, No. 6, pp. 1073–1078 (1978), doi:10.1287/ opre.26.6.1073.

[10] W. W. Hager and H. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search," SIAM J. Optim., Vol.16, No. 1, pp. 170–192 (2006), doi: 10.1137/030601880.

[11] N. Andrei, "A Dai-Liao conjugate gradient algorithm with clustering of eigenvalues," Numer. Algorithms, Vol. 77, No. 4, pp. 1273–1282(2018), doi: 10.1007/s11075-017-0362-5.

[12] S. Yao, L. Ning, H. Tu, and J. Xu, "A one-parameter class of three-term conjugate gradient methods with an adaptive parameter choice," Optim. Methods Softw., Vol. 35, No. 6, pp. 1051–1064 (2020), doi:10.1080/10556788.2018.1510926.

[13] J. C. Gilbert and J. Nocedal, "Global Convergence Properties of Conjugate Gradient Methods for Optimization," SIAM J. Optim., Vol.2, No. 1, pp. 21–42 (1992), doi: 10.1137/0802003.

[14] Najm Huda, Y., & Ahmed Huda, I. (2022). A modification of Dai-Yuan's conjugate gradient algorithm for solving unconstrained optimization. Bulletin of the South Ural State University. Ser. Mathematical Modelling , Programming & Computer Software (Bulletin SUSU MMCS), 15(3), 127-133.

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

Function	N	Problem	Alg. KDY		Alg. DY		Alg. HDY		Alg. AMDYN	
			Iter.	Fg.	Iter.	Fg.	Iter.	Fg.	Iter	Fg.
1	1000	Freudenstein & Roth	412	433	1080	20870	432	474	858	1159
	5000		159	2086	925	21381	352	6357	443	6956
	10000		67	611	314	3659	985	24147	1856	22210
2	1000	Generalized Tridiagonal 1	54	629	44	73	285	428	44	73
	5000		68	1143	71	1084	293	1235	57	489
	10000		57	748	93	1924	275	1590	139	3251
3	1000	Extended Tridiagonal 1	15	27	39	72	22	40	26	50
	5000		119	232	59	115	28	53	85	147
	10000		65	109	27	52	31	59	25	48
4	1000	Extended Beale BEALE (CUTE)	47	65	44	79	138	219	23	49
	5000		68	112	37	65	95	174	54	91
	10000		31	58	52	80	77	149	24	40
5	1000	Extended Three Expo Terms	26	51	16	27	13	24	16	27
	5000		26	51	16	29	35	285	15	26
	10000		27	53	12	22	38	705	10	18
6	1000	Extended Himmelblau HIMMELBC (CUTE)	26	52	23	46	844	1120	23	46
	5000		26	52	23	45	824	1096	19	37
	10000		26	52	21	41	869	1165	13	25
7	1000	ARWHEAD (CUTE)	35	54	55	688	39	55	44	67
	5000		88	1605	137	3004	47	97	155	3587
	10000		191	5098	115	2430	252	7088	94	1999
8	1000	Extended Powell	1417	3681	1558	3070	1468	3855	2002	3195
	5000		986	2260	1904	2934	1454	2904	1863	5822
	10000		952	2060	1420	4557	1088	11631	1575	2874
9	1000	Extended Hiebert	157	855	105	296	491	3977	68	124
	5000		71	622	111	347	2002	7274	7	14
	10000		5	11	36	264	1322	3168	5	11
10	1000	DENSCHNF (CUTE)	52	209	52	354	197	794	57	685
	5000		40	70	103	1769	262	2745	72	948
	10000		75	1012	60	850	249	1666	46	359
11	1000	SINQUAD (CUTE)	8	22	8	22	8	22	8	22
	5000		5	15	5	15	5	15	5	15
	10000		7	20	7	20	7	20	7	20

Function	N	Problem	Alg. KDY Iter.	Alg. KDY Fg.	Alg. DY Iter.	Alg. DY Fg.	Alg. HDY Iter.	Alg. HDY Fg.	Alg. AMDYN Iter	Alg. AMDYN Fg.
12	1000	Perturbed Quadratic	620	1003	589	959	498	849	691	1122
	5000		1338	2121	1379	2795	1189	1963	1400	2294
	10000		2002	3493	2002	3338	2002	3254	2002	3333
13	1000	GENROSNB (CUTE)	695	1356	714	1123	748	1273	734	1165
	5000		1895	3648	2002	3198	2002	3453	2002	3154
	10000		2002	3371	2002	3270	2002	3656	2002	3178
14	1000	QP2 Extended Quadratic Penalty	593	960	559	876	536	898	463	729
	5000		1890	3050	1712	3564	1956	3221	1626	2573
	10000		2002	3197	1928	3566	2002	3280	2002	3177
15	1000	Extended Trigonometric ET2	71	1207	101	2009	334	3707	103	1984
	5000		220	2786	151	3670	227	2740	240	6667
	10000		302	8904	215	5994	276	5246	206	5562
16	1000	PRODsin	32	189	21	36	30	57	20	37
	5000		26	48	23	39	37	725	20	36
	10000		29	88	24	39	28	60	22	39
17	1000	Extended Trigonometric ET1	42	66	31	57	30	52	31	56
	5000		26	45	20	39	23	45	24	43
	10000		19	36	18	35	24	47	20	37
18	1000	Extended Rosenbrock SROSENBR (CUTE)	186	282	127	231	1317	1806	116	189
	5000		432	570	137	209	447	692	326	428
	10000		113	211	139	209	445	607	282	383
19	1000	Extended Tridiagonal 1	15	27	39	72	22	40	26	50
	5000		119	232	59	115	28	53	85	147
	10000		65	109	27	52	31	59	25	42
20	1000	Extended Three Expo Terms	26	51	16	27	13	24	16	27
	5000		26	51	16	29	35	285	15	26
	10000		27	53	12	22	38	705	10	18
21	1000	Diagonal 1	330	557	407	674	806	1192	385	709
	5000		818	1345	852	1207	561	862	257	647
	10000		336	533	375	595	324	519	298	469
22	1000	Hager	404	768	443	950	637	1164	342	709
	5000		464	885	358	704	484	859	332	563
	10000		456	934	434	726	389	625	499	878
23	1000	Generalized Tridiagonal 2	216	359	150	327	878	1330	82	135
	5000		23	58	22	56	17	39	5	9
	10000		20	35	24	41	42	73	13	24

Function	N	Problem	Alg. KDY		Alg. DY		Alg. HDY		Alg. AMDYN	
			Iter.	Fg.	Iter.	Fg.	Iter.	Fg.	Iter	Fg.
24	1000	Diagonal Full Borded	5	10	5	10	5	10	5	10
	5000		5	10	5	10	5	10	5	10
	10000		5	10	5	10	5	10	5	10
25	1000	Tridiagonal Double Borded Arrow Up	9	18	15	25	46	81	10	20
	5000		15	30	13	26	41	73	12	24
	10000		18	36	16	32	45	77	13	25
26	1000	Extended Block-Diagonal BD1	63	97	66	98	63	117	56	88
	5000		52	63	39	66	44	74	22	39
	10000		65	98	51	101	56	88	25	46
27	1000	Extended Maratos	417	565	610	761	432	830	337	529
	5000		518	752	612	810	1087	1593	352	567
	10000		523	852	872	1055	538	981	439	803
28	1000	Full Hessian FH1	31	62	47	96	35	76	19	20
	5000		21	42	17	34	16	32	11	13
	10000		16	32	18	36	15	30	12	24
29	1000	Quadratic Diagonal Perturbed	588	1032	183	268	981	1520	199	323
	5000		140	233	157	259	152	250	122	209
	10000		667	1240	297	479	584	890	231	364
30	1000	Tridiagonal White & Holst (c=4)	588	1032	183	268	981	1520	199	323
	5000		140	233	157	259	152	250	122	209
	10000		667	1240	297	479	584	890	231	364
31	1000	TRIDIA (CUTE)	3	6	3	6	3	6	3	6
	5000		4	7	4	7	4	7	4	7
	10000		33	66	30	60	48	96	28	56
Total	**NOI**	**28815**	**29129**		**41826**		**27185**		**28815**	
	NOF	**115282**	**101769**		**138567**		**72940**		**115282**	