Received: xxx **Accepted:** xxx **Published:** xxx.
DOI. xxxxxxx
xxx Volume xxx, Issue xxx, (1-14)
Research Article

Open Access

Control and Optimization in Applied Mathematics - COAM

Some Hybrid Conjugate Gradient Methods Based on Barzilai-Borwein Approach for Solving Two-Dimensional Unconstrained Optimization Problems

Farzad Rahpeymaii ✉, **Majid Rostami**

Department of Basic Sciences, Technical and Vocational University (TVU), Tehran, Iran.

✉ **Correspondence:**
Farzad Rahpeymaii
E-mail:
frahpeymaei@tuv.ac.ir

How to Cite
Rahpeymaii, F., Rostami, M. (2026). "Some hybrid conjugate gradient methods based on Barzilai-Borwein approach for solving two-dimensional unconstrained optimization problems", Control and Optimization in Applied Mathematics, 11(): 1-14, doi: 10.30473/coam.2025.74997.1317.

Abstract. The conjugate gradient (CG) method is one of the simplest and most widely used approaches for unconstrained optimization, and our focus is on two-dimensional problems with numerous practical applications. We devise three hybrid CG methods in which the hybrid parameter is constructed from the Barzilai–Borwein process, and in these hybrids, the weaknesses of each constituent method are mitigated by the strengths of the others. The conjugate gradient parameter is formed as a linear combination of two well-known CG parameters, blended by a scalar, enabling our new methods to solve the targeted problems efficiently. Under mild assumptions, we establish the descent property of the generated directions and prove the global convergence of the hybrid schemes. Numerical experiments on ten practical examples indicate that the proposed hybrid CG methods outperform standard CG methods for two-dimensional unconstrained optimization.

Keywords. Unconstrained optimization, Hybrid conjugate gradient methods, Wolfe conditions, Global convergence, Barzilai-Borwein process.

MSC. 90C34; 90C40.

https://mathco.journals.pnu.ac.ir

1 Introduction

Consider the nonlinear unconstrained optimization problem

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

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a continuously differentiable function and bounded below. So far, many numerical methods have been proposed to solve the optimization problem (1). Some of these methods are line search methods, trust-region methods, Newton's method and its modifications, quasi-Newton methods and conjugate gradient method [7, 12, 15].

Conjugate gradient (CG) methods to solve (1) starting from $x_0 \in \mathbb{R}^n$, an initial guess, and generate

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

in which $\alpha_k > 0$ is a step-size obtained by inexact line search conditions and the search direction d_k given by

$$d_k = \begin{cases} -g_k, & k = 0, \\ -g_k + \beta_k d_{k-1}, & k > 0, \end{cases} \tag{3}$$

where β_k is called CG-parameter and $g_k = \nabla f(x_k)$. The step-size α_k is usually obtained by weak Wolfe line search (WWLS) conditions [12]:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k g_k^T d_k, \tag{4}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq c_2 g_k^T d_k, \tag{5}$$

or the strong Wolfe line search (SWLS) conditions:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k g_k^T d_k, \tag{6}$$

$$\left|g(x_k + \alpha_k d_k)^T d_k\right| \leq c_2 |g_k^T d_k|, \tag{7}$$

in which $0 < c_1 < c_2 < 1$. Moreover, the search direction d_k satisfies the descent condition

$$g_k^T d_k < 0,$$

or the sufficient descent condition

$$g_k^T d_k < -c\|g_k\|^2, \qquad c > 0,$$

in which $\|\cdot\|$ is the Euclidean norm. The well-known CG parameters are Hestenes-Stiefel (HS) [9], Fletcher-Reeves (FR) [6], Polak-Ribière-Polak (PRP) [13, 14], conjugate descent (CD) [5], Dai-Yuan (DY) [3] and Hager-Zhang (HZ) [7]. These parameters are listed as follows:

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \qquad \beta_k^{CD} = -\frac{\|g_k\|^2}{g_{k-1}^T d_{k-1}}, \qquad \beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}},$$

$$\beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \qquad \beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \qquad \beta_k^{HZ} = \beta_k^{HS} - 2\frac{\|y_{k-1}\|^2 d_{k-1}^T g_k}{(d_{k-1}^T y_{k-1})^2},$$

where $y_{k-1} = g_k - g_{k-1}$.

The steepest descent direction $-g_k$ has verry small step-size. Therefore, the convergence of this method is slow. To overcome this drawback, Barzilai and Borwein (BB) [1] obtained the following step-sizes:

$$\lambda_k^1 = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}, \qquad \text{and} \qquad \lambda_k^2 = \frac{y_{k-1}^T y_{k-1}}{s_{k-1}^T y_{k-1}}.$$

In fact, λ_k^1 and λ_k^2 are the solution of the least-square problems:

$$\min \quad \|\lambda s_{k-1} - y_{k-1}\|^2, \qquad \lambda \in \mathbb{R},$$

and

$$\min \quad \|\lambda y_{k-1} - s_{k-1}\|^2, \qquad \lambda \in \mathbb{R}.$$

Finally, the BB parameter is

$$\lambda_k^{BB} = \min\left\{\lambda_k^1, \lambda_k^2\right\}.$$

2 Hybrid Conjugate Gradient Methods

Using the quasi-Newton equation, Dai and Liao obtained the following conjugate condition [2]:

$$d_k^T y_{k-1} = -t g_k^T s_{k-1}, \tag{8}$$

in which $t > 0$. Substituting (3) into (8), then

$$d_k^T y_{k-1} = -g_k^T y_{k-1} + \beta_k d_{k-1}^T y_{k-1} = -t g_k^T s_{k-1},$$

or

$$\beta_k = \frac{g_k^T (y_{k-1} - t s_{k-1})}{d_{k-1}^T y_{k-1}}.$$

Let $t = 1$. Hence, Dai-Liao (DL) conjugate gradient parameter β_k^{DL} is as follows:

$$\beta_k^{DL} = \frac{g_k^T (y_{k-1} - s_{k-1})}{d_{k-1}^T y_{k-1}}. \tag{9}$$

β_k^{LS} proposed by Liu and Storey [10] as following:

$$\beta_k^{LS} = -\frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}}. \tag{10}$$

If the exact line search is used, the LS method is equivalent to PRP method [8], which is efficient CG method in practical computation.

Rivaie et al. [17] proposed β_k^{RMIL} which the generated directions are sufficient descent. β_k^{RMIL} is denoted by

$$\beta_k^{RMIL} = \frac{g_k^T(g_k - g_{k-1})}{\|d_{k-1}\|^2}. \tag{11}$$

In [16], parameter β_k^{RMIL} is modified as follows

$$\beta_k^{RMIL+} = \frac{g_k^T(g_k - g_{k-1} - d_{k-1})}{\|d_{k-1}\|^2}. \tag{12}$$

The most important properties of these conjugate gradient methods are

- The DL method satisfies in conjugate condition (8) for $t = 1$ and has strong global convergence properties.
- The LS method is numerical efficiency and becomes PRP method with a strong numerical results by exact line search.
- The RMIL+ method has good convergence and the generated directions by it are sufficient descent.

Now, we combine the above conjugate gradient methods and obtain three hybrid conjugate gradient methods. Some of the most important characteristics of hybrid methods are

1. We combine strong convergence methods with numerically efficient methods. So, the new method has both convergence and numerical efficiency properties.
2. We use parameter BB to combine methods that accelerate convergence.
3. Two-dimensional optimization problems have many practical applications, so it is important to provide useful methods for solving them.
4. The methods presented in this paper are suitable for two-dimensional optimization problems, while they may not be suitable for higher dimensions.

We present hybrid algorithms by β_k^{DL}, β_k^{LS} and β_k^{RMIL+} parameters and introduce three hybrid methods to solve unconstrained optimization problems in two-dimensions.

Case I. Let $0 < \lambda_{\text{min}} < \lambda_{\text{max}}$. Consider the combination of parameters β_k^{DL} and β_k^{LS} which one has strong convergence and the other has good numerical efficiency

$$\begin{aligned}\beta_k^1 &= \hat{\lambda}\beta_k^{DL} + (1-\hat{\lambda})\beta_k^{LS} \\ &= \hat{\lambda}\frac{g_k^T(y_{k-1} - s_{k-1})}{d_{k-1}^T y_{k-1}} + (\hat{\lambda} - 1)\frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}},\end{aligned} \tag{13}$$

in which

$$\hat{\lambda} = \max\{\lambda_{\text{min}}, \min\{\lambda_k^{BB}, \lambda_{\text{max}}\}\}. \tag{14}$$

Hence, $\lambda_{\text{min}} \leq \hat{\lambda} \leq \lambda_{\text{max}}$.

Algorithm 1 Combination of β_k^{DL} and β_k^{LS} based on BB step-size (DL-LS).

$(S0)$ Compute the initial function value $f_0 = f(x_0)$ and the initial gradient vector $g_0 = g(x_0)$ and set $d_0 = -g_0$.

$(S1)$ If $\|g_k\| < \varepsilon$ or $k > k_{\text{max}}$, stop.

$(S2)$ Find α_k satisfying SWLS conditions resulting in $x_{k+1} = x_k + \alpha_k d_k$, $f_{k+1} = f(x_{k+1})$ and $g_{k+1} = g(x_{k+1})$.

$(S3)$ Calculate $\hat{\lambda}$, β_{k+1}^{DL} and β_{k+1}^{LS} and obtain the parameter β_{k+1}^{1} by (13) and $d_{k+1} = -g_{k+1} + \beta_{k+1}^{1} d_k$.

$(S4)$ Set $k = k+1$ and go to $(S1)$.

Case II. In this case, we consider the combination of parameters β_k^{DL} and β_k^{RMIL+} where both have the strong global convergence properties.

$$\begin{aligned} \beta_k^2 &= \hat{\lambda}\beta_k^{DL} + (1-\hat{\lambda})\beta_k^{RMIL+} \\ &= \hat{\lambda}\frac{g_k^T(y_{k-1} - s_{k-1})}{d_{k-1}^T y_{k-1}} + (1-\hat{\lambda})\frac{g_k^T(g_k - g_{k-1} - d_{k-1})}{\|d_{k-1}\|^2}, \end{aligned} \tag{15}$$

in which $\hat{\lambda}$ is obtained by (14).

Case III. In this case, we consider the combination of parameters β_k^{LS} and β_k^{RMIL+}. The first method has appropriate numerical results and the second method has strong convergence.

$$\begin{aligned} \beta_k^3 &= \hat{\lambda}\beta_k^{LS} + (1-\hat{\lambda})\beta_k^{RMIL+} \\ &= -\hat{\lambda}\frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}} + (1-\hat{\lambda})\frac{g_k^T(g_k - g_{k-1} - d_{k-1})}{\|d_{k-1}\|^2}, \end{aligned} \tag{16}$$

where $\hat{\lambda}$ is computed by (14).

Algorithm 2 Combination of β_k^{DL} and β_k^{RMIL+} based on BB step-size (DL-RMIL+).

$(S0)$ Compute the initial function value $f_0 = f(x_0)$ and the initial gradient vector $g_0 = g(x_0)$ and set $d_0 = -g_0$.

$(S1)$ If $\|g_k\| < \varepsilon$ or $k > k_{\max}$, stop.

$(S2)$ Find α_k satisfying SWLS conditions resulting in $x_{k+1} = x_k + \alpha_k d_k$, $f_{k+1} = f(x_{k+1})$ and $g_{k+1} = g(x_{k+1})$.

$(S3)$ Calculate $\hat{\lambda}$, β_{k+1}^{DL} and β_{k+1}^{RMIL+} and obtain the parameter β_{k+1}^2 by (15) and $d_{k+1} = -g_{k+1} + \beta_{k+1}^2 d_k$.

$(S4)$ Set $k = k + 1$ and go to $(S1)$.

Algorithm 3 Combination of β_k^{LS} and β_k^{RMIL+} based on BB step-size (LS-RMIL+).

$(S0)$ Compute the initial function value $f_0 = f(x_0)$ and the initial gradient vector $g_0 = g(x_0)$ and set $d_0 = -g_0$.

$(S1)$ If $\|g_k\| < \varepsilon$ or $k > k_{\max}$, stop.

$(S2)$ Find α_k satisfying SWLS conditions resulting in $x_{k+1} = x_k + \alpha_k d_k$, $f_{k+1} = f(x_{k+1})$ and $g_{k+1} = g(x_{k+1})$.

$(S3)$ Calculate $\hat{\lambda}$, β_{k+1}^{LS} and β_{k+1}^{RMIL+} and obtain the parameter β_{k+1}^3 by (16) and $d_{k+1} = -g_{k+1} + \beta_{k+1}^3 d_k$.

$(S4)$ Set $k = k + 1$ and go to $(S1)$.

3 Convergence Analysis

We consider some assumptions to investigate the convergence of Algorithms 1-3.

(H1). For any $x_0 \in \mathbb{R}^n$, the level set $L(x_0) = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ is bounded.

(H2). For all $x \in L(x_0)$ there exists a constant $\Lambda > 0$ such that

$$\|x\| \leq \Lambda.$$

(H3). The gradient of f is Lipschitz continuous, i.e., there exists constant $L_g > 0$ such that

$$\|g(x) - g(y)\| \leq L_g \|x - y\|, \quad \forall x, y \in L(x_0).$$

Theorem 1. The generated direction by Algorithm 1 is the sufficient descent direction.

Proof. Using Algorithm 1, we have

$$\begin{aligned} g_k^T d_k &= -\|g_k\|^2 + \beta_k^1 g_k^T d_{k-1} \\ &= -\|g_k\|^2 + \hat{\lambda}\frac{g_k^T(y_{k-1}-s_{k-1})}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1} + (\hat{\lambda}-1)\frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}} g_k^T d_{k-1} \\ &\leq -\|g_k\|^2 + \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1} - \frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}} g_k^T d_{k-1} \\ &\leq -\|g_k\|^2. \end{aligned}$$

□

Theorem 2. Let the hypotheses(H1)-(H3) hold, and let d_k be produced by the DL method. Then,

$$\lim_{k\longrightarrow\infty} \inf \|g_k\| = 0.$$

Proof. This result follows directly from Theorem 1 in [2]. □

Theorem 3. Assume that d_k be generated by Algorithm 2. Then

$$g_k^T d_k \leq 0.$$

Proof. From (15), we get

$$\begin{aligned} g_k^T d_k &= -\|g_k\|^2 + \beta_k^2 g_k^T d_{k-1} \\ &= -\|g_k\|^2 + \hat{\lambda}\frac{g_k^T(y_{k-1}-s_{k-1})}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1} + (1-\hat{\lambda})\frac{g_k^T(g_k - g_{k-1} - d_{k-1})}{\|d_{k-1}\|^2} g_k^T d_{k-1} \\ &\leq \frac{g_k^T(y_{k-1}-\alpha_{k-1}d_{k-1})}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1} - \frac{g_k^T(y_{k-1}-d_{k-1})}{\|d_{k-1}\|^2} g_k^T d_{k-1} \\ &\leq 0. \end{aligned}$$

□

Theorem 4. Let the hypotheses (H1)-(H3) hold. For the LS conjugate gradient method, either

$$\lim_{k\longrightarrow\infty} \|g_k\| = 0,$$

or

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty.$$

Proof. This result follows from Theorem 3.1 in [11]. □

Theorem 5. Let $0 < c_1 < \dfrac{1}{4}$. Then the generated direction by the RMIL+ method is a descent direction.

Proof. From equation (16), we obtain

$$
\begin{aligned}
g_k^T d_k &= -\|g_k\|^2 + \beta_k^3 g_k^T d_{k-1} \\
&= -\|g_k\|^2 - \hat{\lambda}\frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}} g_k^T d_{k-1} + (1-\hat{\lambda})\frac{g_k^T(y_{k-1}-d_{k-1})}{\|d_{k-1}\|^2} g_k^T d_{k-1} \\
&\leq -\|g_k\|^2 - \hat{\lambda}\frac{g_k^T y_{k-1}}{g_{k-1}^T d_{k-1}} g_k^T d_{k-1} - \frac{g_k^T(y_{k-1}-d_{k-1})}{\|d_{k-1}\|^2} g_k^T d_{k-1} \\
&\leq -\|g_k\|^2.
\end{aligned}
$$

□

Theorem 6. Let the hypotheses (H1)-(H3) hold. If the sequences $\{g_k\}$ and $\{d_k\}$ are generated by the RMIL+ method, then

$$\lim_{k\longrightarrow\infty} \inf \|g_k\| = 0.$$

Proof. This result follows from Theorem 3.1 in [4]. □

Since the hybrid CG methods are combined by constant parameter $\hat{\lambda}$, based on Theorems 2, 4 and 6, we conclude that the generated iterations sequence by the DL-LS, DL-RMIL+ and the LS-RMIL+ convergence to the optimal solutions of two-dimensional unconstrained optimization problems.

4 Numerical Results

We compare the numerical results of the DL-LS, DL-RMIL+ and LS-RMIL+ methods for solving two-dimensional unconstrained optimization problems. These results are contrasted with the DL, LS and RMIL+ algorithms as applicable.

The stopping criteria is either $\|g_k\| \leq \varepsilon$ or reaching a maximum iteration count $k_{\max} = 500$. For all methods, we use the following parameters: $c_1 = 0.15$, $c_2 = 0.85$, $\varepsilon = 10^{-6}$, $\lambda_{\min} = 0.001$ and $\lambda_{\max} = 100$. All codes are implemented in Matlab 2017a on a Laptop with an Intel Core i3 processor, 2.3 GHz, and 4 GB of RAM. To enable a fair comparison across all algorithms, we evaluate ten standard two-dimensional test problems, which are introduced in the next subsection.

4.1 Test Functions

We reference test problems from the Test Functions and Datasets page of the Virtual Library of Simulation Experiments: http://www.sfu.ca/~ssurjano/index.html.

- Beale test function

$$f(x,y) = (1.5 - x(1-y))^2 + (2.25 - x(1-y^2))^2 + (2.625 - x(1-y^3))^2,$$
$$x^* = (3, 0.5)^T, \quad x_0 = (1,1)^T.$$

- Booth test function

$$f(x,y) = (x+2y-7)^2 + (2x+y-5)^2,$$
$$x^* = (1,3)^T, \quad x_0 = (0,1)^T.$$

- Rastrigin test function

$$f(x,y) = x^2 + y^2 - 10\cos(2\pi x) - 10\cos(2\pi y) + 20,$$
$$x^* = (0,0)^T, \quad x_0 = (1,1)^T.$$

- Three Hump Camel test function

$$f(x,y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2,$$
$$x^* = (0,0)^T, \quad x_0 = (1,1)^T.$$

- Matyas test function

$$f(x,y) = 0.26(x^2+y^2) - 0.48xy,$$
$$x^* = (0,0)^T, \quad x_0 = (5,1)^T.$$

- Trid test function

$$f(x,y) = (x-1)^2 + (y-1)^2 - xy,$$
$$x^* = (2,2)^T, \quad x_0 = (3,3)^T.$$

- Six Hump Camel test function

$$f(x,y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4+4y^2)y^2,$$
$$x^* = (0.0898, -0.7126)^T, \quad x_0 = (0.01, 0.01)^T.$$

- Rosenbrock test function

$$f(x,y) = 100(y-x^2)^2 + (x-1)^2,$$
$$x^* = (1,1)^T, \quad x_0 = (1.3, 1.3)^T.$$

- Perm test function $(\beta = 0)$

$$f(x,y) = (x+2y-2)^2 + (x^2+2y^2-1.5)^2,$$
$$x^* = (1,0.5)^T, \quad x_0 = (0.95, 0.55)^T.$$

- Rotated Hyper-Ellipsoid test function

$$f(x,y) = 2x^2 + y^2,$$
$$x^* = (0,0)^T, \quad x_0 = (1,1)^T.$$

The comparative results on these test functions derived from various algorithms are presented in Tables 1-3. The total number of iterations are presented in Table 1, which shows LS-RMIL+ and DL-RMIL+ methods can solve the unconstrained two-dimensional optimization problems with less number of iterations, respectively. The comparison of the total number of function evaluations are also shown in Table 2. In this case, DL-RMIL+ and LS-RMIL+ methods are more efficient. Finally, the CPU times of six algorithms are presented in Table 3, which shows DL, LS-RMIL+ and DL-RMIL+ solve the two-dimensional unconstrained optimization problems faster than other methods, respectively.

Table 1: The number of iterations.

Test Function	**DL**	**LS**	**RMIL+**	**DL-LS**	**DL-RMIL+**	**LS-RMIL+**
Beal	285	643	385	144	174	85
Booth	214	149	69	191	110	96
Rastrigin	1	1	1	1	1	1
Three Hump Camel	21	500	65	177	77	112
Natyas	20	634	312	11	8	5
Trid	21	35	11	21	21	35
Six Hump Camel	9	500	27	4	23	4
Rosenbrock	20	500	500	500	13	26
Perm	4	500	175	500	234	24
Rotated Hyper-Ellipsoid	171	500	62	500	71	171
Avarage	76.6	396.2	160.7	204.9	73.2	55.9

Table 2: The number of function evaluations.

Test Function	DL	LS	RMIL+	DL-LS	DL-RMIL+	LS-RMIL+
Beal	287	645	387	146	176	124
Booth	216	151	71	193	112	98
Rastrigin	2	2	2	2	2	2
Three Hump Camel	22	504	67	179	79	114
Natyas	21	635	313	13	10	45
Trid	22	36	12	22	22	36
Six Hump Camel	51	543	29	47	25	46
Rosenbrock	62	504	504	504	26	65
Perm	4	540	177	540	236	62
Rotated Hyper-Ellipsoid	173	541	64	541	73	173
Avarage	90	410.1	194.8	218.7	76.1	76.5

Table 3: The CPU times for all algorithms.

Test Function	DL	LS	RMIL+	DL-LS	DL-RMIL+	LS-RMIL+
Beal	23.8015	65.8402	31.2997	21.1464	24.9714	14.7864
Booth	17.8149	11.9217	5.7808	27.0438	15.9553	13.8227
Rastrigin	0.3531	0.3634	0.3419	0.4322	0.4340	0.4317
Three Hump Camel	2.0672	82.2808	6.8952	30.5951	10.9724	16.1379
Natyas	2.0114	51.3565	26.8285	2.0884	1.5943	2.4374
Trid	2.1437	3.1208	1.3013	3.3345	3.3244	5.2041
Six Hump Camel	2.4208	57.7903	2.8750	2.3456	3.6615	2.2703
Rosenbrock	3.8371	42.0699	45.6841	43.5780	3.1579	6.0971
Perm	2.1029	82.4213	13.9331	52.1107	33.9148	4.9238
Rotated Hyper-Ellipsoid	14.9261	52.2383	5.3636	89.3391	11.0134	12.0363
Avarage	7.1515	44.9403	14.0303	27.2014	10.8999	7.8148

Finally, to compare the efficiency of conjugate gradient methods, we use the relative efficiency (R_i) which is the ratio of the total iterations for the DL, LS, RMIL+, DL-LS and DL-RMIL+ methods with respect to the number of iterations of the LS-RMIL+ method as follows

$$R_i = \frac{N_{\text{iter}}(i)}{N_{\text{iter}}(LS - RMIL+)}. \tag{17}$$

The relative efficiency of hybrid CG methods are given in Table 4.

Table 4: Relative efficiency for hybrid CG methods.

DL	LS	RMIL+	DL-LS	DL-RMIL+	LS-RMIL+
1.37	7.09	2.87	3.67	1.31	1

5 Conclusion

Conjugate gradient CG methods are among the most effective methods for solving unconstrained optimization problems; however, each method exhibits distinct advantages and limitations. Hybrid methods are commonly employed to enhance iterative CG-based algorithms for unconstrained optimization problems. In this study, we combined three CG methods in pairwise configurations to form hybrid CG methods. The results demonstrate that these hybrid solvers outperform their individual constituents in terms of iterations, function evaluations, and CPU time. By integrating complementary strengths, the hybrids mitigate the weaknesses of the constituent methods. Numerical experiments reported herein indicate that the proposed hybrid CG methods are well-suited for unconstrained two-dimensional optimization problems.

Declarations

Availability of Supporting Data

All data generated or analyzed during this study are included in this published paper.

Funding

The authors conducted this research without any funding, grants, or support.

Competing Interests

The authors declare that they have no competing interests relevant to the content of this paper.

Authors' Contributions

All authors contributed equally to the design of the study, data analysis, and writing of the manuscript, and share equal responsibility for the content of the paper.

Contribution of AI Tools

The authors used AI-based tools solely as part of their writing and editing workflow. Specifically, AI-assisted capabilities were employed to improve language quality, clarity, grammar, and stylistic consistency. The authors did not rely on AI to generate original scientific content, data, analyses, or interpretations.

References

[1] Barzilai, J., Borwein, J.M. (1988). "Two-point step size gradient methods". *IMA Journal of Numerical Analysis*, 8(1), 141-148, doi:https://doi.org/10.1093/imanum/8.1.141.

[2] Dai, Y.H., Liao, L.Z. (2001). "New conjugacy conditions and related nonlinear conjugate gradient methods". *Applied Mathematics and Optimization*, 43, 87-101, doi:https://doi.org/10.1007/s002450010019.

[3] Dai, Y.H., Yuan, Y. (1999). "A nonlinear conjugate gradient method with a strong global convergence property". *SIAM Journal on Optimization*, 10(1), 177-182, doi:https://doi.org/10.1137/S1052623497318992.

[4] Diphofu, T., Kaelo, P., Kooepile-Reikeletseng, S., Koorapetse, M., Sam, C.R. (2024). "Convex combination of improved Fletcher-Reeves and Rivaie-Mustafa-Ismail-Leong conjugate gradient methods for unconstrained optimization problems and applications". *Quaestiones Mathematicae*, 47(12), 2375–2397, doi:https://doi.org/10.2989/16073606.2024.2367715.

[5] Fletcher, R. (2000). "Practical Methods Of Optimization: Vol. 1 Unconstrained Optimization". *2nd ed., John Wiley and Sons, Chichester*, doi:https://onlinelibrary.wiley.com/doi/book/10.1002/9781118723203.

[6] Fletcher, R., Reeves, C. (1964). "Function minimization by conjugate gradients". *The Computer Journal*, 7(2), 149-154, doi:https://doi.org/10.1093/comjnl/7.2.149.

[7] Hager, W.W., Zhang, H. (2005). "A new conjugate gradient method with guaranteed descent and an efficient line search". *SIAM Journal on Optimization*, 16(1), 170-192, doi:https://doi.org/10.1137/030601880.

[8] Hager, W., Zhang, H. (2006). "A survey of nonlinear conjugate gradient methods". *Pacific Journal of Optimization*, 2(1), 35-58.

[9] Hestenes, M.R., Stiefel, E.L. (1952). "Methods of conjugate gradients for solving linear systems". *Journal of Research of the National Institute of Standards and Technology*, 49(6), 409-436, doi:https://doi.org/10.6028/jres.049.044.

[10] Liu, Y.L., Storey, C. (1991). "Efficient generalized conjugate gradient algorithms, Part 1: Theory". *Journal of Optimization Theory and Applications*, 69(1), 129-137, doi:https://doi.org/10.1007/BF00940464.

[11] Mustafa, A.A. (2023). "New spectral LS conjugate gradient method for nonlinear unconstrained optimization". *International Journal of Computer Mathematics*, 100(4), 838-846, doi:https://doi.org/10.1080/00207160.2022.2163165.

[12] Nocedal, J., Wright, S.J. (2006). "Numerical optimization". *Springer-Verlag, New York, NY*, doi:https://doi.org/10.1007/978-0-387-40065-5.

[13] Polyak, B.T. (1969). "The conjugate gradient method in extreme problems". *USSR Computational Mathematics and Mathematical Physics*, 9(4), 94-112, doi:https://doi.org/10.1016/0041-5553(69)90035-4.

[14] Polyak, E., Ribière, G. (1969). "Note sur la convergence de directions conjugées". *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 3(R1), 35-43.

[15] Rahpeymaii, F., Rostami, M. (2018). "A new hybrid conjugate gradient method based on eigenvalue analysis for unconstrained optimization problems". *Control and Optimization in Applied Mathematics*, 3(1), 27-43, doi:https://doi.org/10.30473/coam.2019.44564.1108.

[16] Rivaie, M., Mamat, M., Abashar, A. (2015). "A new class of nonlinear conjugate gradient coefficients with exact and inexact line searches". *Applied Mathematics and Computation*, 268, 1152-1163, doi:https://doi.org/10.1016/j.amc.2015.07.019.

[17] Rivaie, M., Mamat, M., June, L.W., Mohd, I. (2012). "A new class of nonlinear conjugate gradient coefficients with global convergence properties". *Applied Mathematics and Computation*, 218(22), 11323-11332, doi:https://doi.org/10.1016/j.amc.2012.05.030.