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

www.csmj.uomosul.edu.iq

A Hybrid Conjugate Gradient Algorithm Using the Golden Section Ratio for Unconstrained Optimization Problems

Sara Sahib Mohammed Zaki[1], Hawraz Nadhim Jabbar[2] and Sozan Saber Haider[3]

[1] Department of Mathematics, College of Science, University of Sulaimani, Sulaimani, Iraq
[2] Department of Mathematics, College of Science, University of Kirkuk, Kirkuk, Iraq
[3] College of Administration and Economics, University of Sulaimani, Sulaimani, Iraq
Email: sara.mohammedzaki@univsul.edu.iq[1] , hawrazmath@uokirkuk.edu.iq[2] and sozan.haider@univsul.edu.iq[3]

Article information

Article history:
Received 03 May, 2025
Revised 14 July, 2025
Accepted 05 August, 2025
Published 25 December, 2025

Keywords:
Unconstrained Optimization,
Hybrid Conjugate Gradient,
The Golden Section Ratio,
Convex Combination,
Global Convergence,
Sufficient Descent Condition.

Correspondence:
Sara Sahib Mohammed Zaki
Email:
sara.mohammedzaki@univsul.edu.iq

Abstract

Conjugate gradient techniques are highly effective for addressing large-scale nonlinear optimization problems. Hybridization is a prevalent strategy for improving the conjugate gradient method. This study presents a novel hybrid conjugate gradient (CG) algorithm that incorporates the golden section ratio for solving unconstrained optimization problems. Hence, we improve the efficiency and robustness of traditional CG methods by taking advantage of the properties of the golden section ratio, which is known for its optimality in line search procedures. Therefore, this study explores the novel use of the golden section ratios (0.382) and (0.618) as a weighting factor (β) in a hybrid convex combination under Dai-Liao condition of two pairs of standard conjugate gradient methods: (β^{HS}, β^{DY}) and (β^{LS}, β^{CD}) separately. These formulas are fundamental to conjugate gradient methods and provide clear benefits in optimization situations. The suggested strategies seek to capitalize on the advantages of the approaches while minimizing their drawbacks by including the golden section ratio which is known for reducing computational cost, improving step size selection, and ensuring robust convergence especially in ill-conditioned problems in optimization. The numerical results show how effective the suggested approaches are.

DOI: 10.33899/csmj.2025.159795.1187, ©Authors, 2025, College of Computer Science and Mathematics, University of Mosul, Iraq.

1. Introduction

The process of minimizing or maximizing an objective function is known as optimization. A subfield of optimization known as "unconstrained optimization" involves minimizing an objective function that is dependent on actual variables while completely removing any constraints on those variables' values.

In unconstrained optimization, consider the following objective function [1]:

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

when $f: R^n \longrightarrow R$ is a continuously differentiable function.

Mathematicians have created a number of numerical techniques throughout the years to address this type of problem, including the Newton method, CG, quasi-Newton method, and steepest descent.

The conjugate gradient approach is the main emphasis of this work due to its ease of use, minimal memory requirements and particularly its usefulness when the dimension is big.

As is well known, there is a beginning point for solving this issue $\{x_n\}$ is a sequence produced by a nonlinear CG [2] as:

$$x_{k+1} = x_k + \alpha_k d_k \qquad (2)$$

Where:

x_k is the current iterating, $\alpha_k > 0$ is the step size is usually determined by line search to fulfill the standard Wolfe line search conditions [3]

$$f(x_k + \alpha_k d_k) - f(x_k) \le \delta \alpha_k\, g_k^T d_k\,, \qquad (3)$$

$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k$, (4)

or stronger version of the Wolfe line search conditions, given by (4) and

$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma| g_k^T d_k |$, (5)

Where $0 < \delta \leq \sigma < 1$, d_k is the search direction. In CG methods, the search direction d_k is computed as:

$$d_{k+1} = -g_{k+1} + \beta_k s_k, \text{and } d_0 = -g_0 \quad (6)$$

where , $s_k = x_{k+1} - x_k$, $g_k = \nabla f(x_k)$ and β_k is known as conjugate gradient parameter throughout history it has been created in various ways.

The scalar parameters β_k are chosen differently for each conjugate gradient technique.

These are a few well-known beta formulas:

$\beta^{HS} = \frac{g_k^T y_{k-1}}{y_{k-1}^T d_{k-1}}$ [4] (Hestenes and Stiefel, 1952)

$\beta^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}$ [5] (Flecher and Reeves, 1964)

$\beta^{PR} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}$ [6] (Polak-Ribiere,1969)

$\beta^{CD} = \frac{\|g_k\|^2}{-g_{k-1}^T d_{k-1}}$ [7] (Conjugate Descent,1987)

$\beta^{LS} = \frac{g_{k+1}^T y_k}{-g_k^T s_k}$ [8] (Liu and Storey, 1991)

$\beta^{DY} = \frac{\|g_k\|^2}{y_{k-1}^T d_{k-1}}$ [9] (Dai-yuan, 1999)

Where $y_{k-1} = g_k - g_{k-1}$ and $\|.\|$ represent the Euclidean norm.

In conjugate gradient method, the search direction d_k is determined in such a way that the following conjugacy condition holds

$$d_i^T G d_j = 0, \qquad i \neq j \qquad (7)$$

where G is the Hessian of the objective function. On the other hand, according to the mean value theorem, there exists some $\omega \in (0, 1)$ such that

$d_{k+1}^T y_k = \alpha_k d_{k+1}^T g(x_k + \omega \alpha_k d_k)^T d_k$, (8)

Now, by combining (7), (8) the following conjugate condition can be deduced

$d_{k+1}^T y_k = 0$

Dai and Liao (DL) [10] with modification of conjugate condition, presented a family of CG methods, denoted by β_k^{DL}, is decided by the extended conjugacy condition

$d_{k+1}^T y_k = -t g_{k+1}^T s_k$, (9) $t > 0$, is scalar

in the DL method the CG coefficient is computed by

$$\beta_k^{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{y_k^T d_k} \qquad (10)$$

The procedure (2), (6) with β_k^{DL} in (10) is called the Dai-Liao method. In recent years, much efforts has been made to find the proper choice for the nonnegative parameter t in (10), see [11], [12], [13], [14], [15]. Based on a singular value study on the DL method, Babaie-Kafaki proposed:

$t = \frac{y_k^T s_k}{\|s_k\|^2} + \frac{\|y_k\|}{\|s_k\|}$, Ghanbari [16] proposed the following adaptive choices for t

$$t = \frac{\|y_k\|}{\|s_k\|}.$$

And Andrei [17] suggested the following value for t

$$t = \frac{y_k^T s_k}{\|s_k\|^2}$$

The hybrid conjugate gradient method is a significant class of conjugate gradient algorithms, it is a projection of several conjugate gradient algorithms, primarily designed to prevent jamming phenomena. Therefore, many researchers have focused on hybrids, for example, the following hybrids have been created:

$\beta_k^{hDY} = \theta_k \beta_k^{FR} + (1 - \theta_k)\beta_k^{HS}$ [18]

$\beta_k^{hDY} = \theta_k \beta_k^{LS} + (1 - \theta_k)\beta_k^{CD}$ [19]

$\beta_k^{N} = \theta_k \beta_k^{MMWU} + (1 - \theta_k)\beta_k^{FR}$ [20]

$\beta_k^{hyb} = \lambda_k \beta_k^{DY} + (1 - \lambda_k)\beta_k^{HS}$ [21]

Among the many advantages of Golden Section Ratio are it is optimal for line search in hybrid algorithms because it allows reusing one function evaluation per iteration, minimizing computational cost. It also provides the smallest possible worst-case interval shrinkage, ensuring faster convergence in unimodal minimization. Additionally, when used in convex combinations, it balances direction updates effectively, improving stability and reducing zigzagging.

Hence, Golden Section Ratio will be used in this paper to create a new hybrid conjugate gradient algorithm under Dai-Liao condition, as a convex combination of HS and DY for which we set (θ = 0.382) initially, and then, set (θ = 0.618) and the convex combination becomes:

$$\beta_k^{hSSH1} = (1 - 0.382)\beta_k^{HS} + 0.382\beta_k^{DY} \qquad (11)$$

$$\beta_k^{hSSH3} = (1 - 0.618)\beta_k^{HS} + 0.618\beta_k^{DY} \qquad (12)$$

Same process will be repeated for the convex combination of LS and CD, and the convex combination becomes:

$\beta_k^{hSSH2} = (1-0.382)\beta_k^{LS} + 0.382\beta_k^{CD}$ (13)

$\beta_k^{hSSH4} = (1-0.618)\beta_k^{LS} + 0.618\beta_k^{CD}$ (14)

This paper is organized as follows: Our hybrid conjugates gradient approaches, the applied algorithm, and under certain conditions, descent directions are generated that satisfy the sufficient descent condition are shown in Section 2. In the next section, we analyze the convergence property. Then, in section 4, we provide some numerical comparisons with some traditional methods to demonstrate the effectiveness of the algorithm. Finally, section 5 provides a brief conclusion.

2. Proposed Method

2.1. A Hybrid Conjugate Gradient Algorithm by Using Golden Section Ratio Under the Dai-Liao Condition

Hybrid techniques combine two or more approaches. While some of them have high comprehensive convergence properties, others have good computational properties.

$\beta_{k=}(1-\theta_k)\beta_k^{HS} + \theta_k\beta_k^{DY}$

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

By multiplying both sides by y_k, we get:

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

From Dai-Liao condition, $t \geq 0$

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

By using golden section ratio:

Case I: Let $\theta = 0.382$

$\beta_k^{hSSH1} = (1-0.382)\beta_k^{HS} + 0.382\, \beta_k^{DY}$

$= 0.618\beta_k^{HS} + 0.382\, \beta_k^{DY}$

From (5)

$d_{k+1} = -g_{k+1} + \beta_k^{hSSH1} s_k^T$

Multiple both sides by y_k and using (9)

$-t_1 g_{k+1}^T s_k = -g_{k+1}^T y_k + \beta_k^{hSSH1} s_k^T\, y_k$

$-t_1 g_{k+1}^T s_k = -g_{k+1}^T y_k + \left(\left(0.618\, \frac{g_{k+1}^T y_k}{y_k^T s_k}\right) + \left(0.382 \frac{g_{k+1}^T g_{k+1}}{y_k^T\, s_k}\right)\right) s_k^T\, y_k$

After some algebra operations, we get:

$t_1 = \frac{0.382(y_k - g_{k+1})g_{k+1}}{g_{k+1}^T s_k}$ (15)

Case II: Let $\theta = 0.618$

By the same way we can get

$t_2 = \frac{0.618(y_k - g_{k+1})g_{k+1}}{g_{k+1}^T s_k}$ (16)

Put (15) in (10) we get:

$$\beta_k^{hSSH1} = \frac{g_{k+1}^T\,(y_k - (0.382(y_k - g_{k+1})\frac{1}{s_k})s_k}{y_k^T d_k}$$

$$= \frac{g_{k+1}^T\,(y_k - 0.382 y_k + 0.382 g_{k+1}^T)}{y_k^T d_k}$$

$$= \frac{g_{k+1}^T\, y_k}{y_k^T d_k} - \frac{0.382 g_{k+1}^T\, y_k}{y_k^T d_k} + \frac{0.382 g_{k+1}^T\, g_{k+1}}{y_k^T d_k}$$

$$= \beta_k^{HS} - 0.382\beta_k^{HS} + 0.382\beta_k^{DY}$$

$$= 0.618\beta_k^{HS} + 0.382\beta_k^{DY} \quad (17)$$

In the same way:

$$\beta_k^{hSSH2} = \frac{0.618 g_{k+1}^T y_k + 0.382\|g_{k+1}\|^2}{-g_k^T d_k}$$

$$= 0.618\beta_k^{LS} + 0.382\beta_k^{CD} \quad (18)$$

Put (16) in (10) we get:

$$\beta_k^{hSHH3} = \frac{g_{k+1}^T(y_k - (\frac{0.618(y_k - g_{k+1})}{s_k})s_k)}{y_k^T d_k}$$

$$= \frac{g_{k+1}^T(y_k - 0.618 y_k + 0.618 g_{k+1})}{y_k^T d_k}$$

$$= 0.382\beta_k^{HS} + 0.618\beta_k^{DY} \quad (19)$$

In the same way:

$$\beta_k^{hSSH4} = \frac{0.382 g_{k+1}^T y_k + 0.618\|g_{k+1}\|^2}{-g_k^T d_k}$$

$$= 0.382\beta_k^{LS} + 0.618\beta_k^{CD} \quad (20)$$

Therefore; we obtained 4 new betas:

1- $\beta_k^{hSSH1} = 0.618\beta_k^{HS} + 0.382\beta_k^{DY}$

2- $\beta_k^{hSSH2} = 0.618\beta_k^{LS} + 0.382\beta_k^{CD}$

3- $\beta_k^{hSSH3} = 0.382\beta_k^{HS} + 0.618\beta_k^{DY}$

4- $\beta_k^{hSSH4} = 0.382\beta_k^{LS} + 0.618\beta_k^{CD}$

2.2. The Hybrid Algorithm

Step (1): Initialization: Select $x_0 \in R^n$, compute: $f(x_0),\ g_0 = \nabla f(x_0).\ Consider\ d_0 = -g_0,$ select ε (e.g $\varepsilon = 10^{-6}$)

Step (2): If $\|g_k\| \leq \varepsilon$, then stop. else go to step (3).

Step (3): Compute α_k by using (3) and (4)

Step (4): Generate $x_{k+1} = x_k + \alpha_k d_k$, compute $f(x_{k+1}),\ and\ g_{k+1} = \nabla f(x_{k+1})$

Step (5): Calculate β_k in (11), (12), (13) or (14) and search direction

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

Step (6): Test the convergence: if $f(x_{k+1}) \leq f(x_k)$ and $\|g_k\| \leq \varepsilon$, then stop.

Otherwise, $k = k + 1$

2.3. A Hybrid Conjugate Gradient Algorithm as a Convex Combination of (HS, DY) and (LS, CD) Algorithms When Θ =0.382 and Θ=0.618

We shall demonstrate that the descent property is satisfied by our conjugate gradient approaches.

Theorem (1): let $\{d_k\}$ be a sequence of directions generated by the new algorithm and α_k in $x_{k+1} = x_k + \alpha_k d_k$ be a step size determined by Strong Wolfe line search, then d_k satisfy sufficient descent conditions.

Proof:

1) $\beta_k^{hSSH1} = (1 - 0.382)\beta_k^{HS} + 0.382\, \beta_k^{DY}$

$$d_{k+1} = -g_{k+1} + \beta_k^{hSSH1}\, d_k$$

$$d_{k+1} = -g_{k+1} + (1 - 0.382)\beta_k^{HS} d_k + 0.382\beta_k^{DY} d_k$$

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + 0.618\frac{g_{k+1}^T y_k}{y_k^T d_k} g_{k+1}^T d_k + 0.382\frac{\|g_{k+1}\|^2}{y_k^T d_k} g_{k+1}^T d_k$$

Since $\sigma g_k^T d_k \le g_{k+1}^T d_k \le -\sigma g_k^T d_k$

$$g_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2 + 0.618\frac{g_{k+1}^T y_k}{y_k^T d_k}(-\sigma g_k^T d_k) + 0.382\frac{\|g_{k+1}\|^2}{y_k^T d_k}(-\sigma g_k^T d_k)$$

And $y_k^T d_k = (g_{k+1} - g_k)^T d_k \ge (\sigma - 1) g_k^T d_k$

$$g_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2 + 0.618\frac{g_{k+1}^T y_k}{(\sigma-1) g_k^T d_k}(-\sigma g_k^T d_k) + 0.382\frac{\|g_{k+1}\|^2}{(\sigma-1) g_k^T d_k}(-\sigma g_k^T d_k)$$

$$g_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2 + 0.618\frac{g_{k+1}^T y_k}{(\sigma-1)}(-\sigma) + 0.382\frac{\sigma}{(1-\sigma)}\|g_{k+1}\|^2$$

$$g_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2 + 0.618\frac{\sigma}{(1-\sigma)}\|g_{k+1}\|^2 + 0.382\frac{\sigma}{(1-\sigma)}\|g_{k+1}\|^2 - 0.618\frac{\sigma}{(1-\sigma)} g_{k+1}^T g_k$$

$$g_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2 + \frac{\sigma}{(1-\sigma)}\|g_{k+1}\|^2 - 0.618\frac{\sigma}{(1-\sigma)} g_{k+1}^T g_k$$

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

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

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

Where $c_1 = \frac{1-2\sigma}{1-\sigma}$

By the same way, we can prove that $\beta_k^{hSSH3} = (1 - 0.618)\beta_k^{HS} + 0.618\beta_k^{DY}$ satisfies the sufficient descent condition.

2) $\beta_k^{hSSH2} = (1 - 0.382)\beta_k^{LS} + 0.382\, \beta_k^{CD}$

$$d_{k+1} = -g_{k+1} + \beta_k^{hSSH2}\, d_k$$

$$d_{k+1} = -g_{k+1} + (1 - 0.382)\beta_k^{LS} d_k + 0.382\beta_k^{CD} d_k$$

$$d_{k+1} = -g_{k+1} + 0.618\frac{g_{k+1}^T y_k}{-g_k^T d_k} d_k + 0.382\frac{\|g_{k+1}\|^2}{-g_k^T d_k} d_k$$

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + 0.618\frac{g_{k+1}^T y_k}{-g_k^T d_k} g_{k+1}^T d_k + 0.382\frac{\|g_{k+1}\|^2}{-g_k^T d_k} g_{k+1}^T d_k$$

Since $g_{k+1}^T d_k \le -\sigma g_k^T d_k$

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + 0.618\frac{g_{k+1}^T (g_{k+1} - g_k)}{-g_k^T d_k} g_{k+1}^T d_k + 0.382\frac{\|g_{k+1}\|^2}{-g_k^T d_k} g_{k+1}^T d_k$$

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + 0.618\frac{g_{k+1}^T g_{k+1}}{-g_k^T d_k}(-\sigma g_k^T d_k) - 0.618\frac{g_{k+1}^T g_k}{-g_k^T d_k}(-\sigma g_k^T d_k) + 0.382\frac{\|g_{k+1}\|^2}{-g_k^T d_k}(-\sigma g_k^T d_k)$$

$$g_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2 + \sigma\|g_{k+1}\|^2 + 0.618\,\sigma g_{k+1}^T g_k$$

$$g_{k+1}^T d_{k+1} \le -(1 - \sigma)\|g_{k+1}\|^2 + 0.1236\,\sigma\|g_{k+1}\|^2$$

By using Powell restart

$$g_{k+1}^T d_{k+1} \le -(1 - 1.1236\sigma)\|g_{k+1}\|^2$$

$$g_{k+1}^T d_{k+1} \le -c_2\|g_{k+1}\|^2$$

Where $c_2 = 1 - 1.1236\,\sigma$

By the same way, we can prove that $\beta_k^{hSSH4} = (1 - 0.618)\beta_k^{LS} + 0.618\beta_k^{CD}$, satisfies the sufficient descent condition.

3. Convergence Analysis

Assumption (1): [22] The level set $T = \{x \in R^n : f(x) \le f(x_0)\}$ is bounded, i.e., there is a positive constant $B > 0$ $such\ that\ \|x\| \le B, \forall\, x \in T$

Assumption (2): [22] In a neighborhood N of T, $f(x)$ is continuously differentiable and its gradient is Lipschitz continuous, i.e.

$\exists\, L \ \ge\ 0\ such\ that\ \|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \forall\, x, y \in N$

at stated by assumptions (1) and (2), there is a non-negative constant $Ꚍ \ge 0$ such that:

$\|\nabla f(x)\| \le Ꚍ\ \ \forall x \in T$

The Zoutendijk criterion is commonly used to illustrate the global convergence of the conjugate gradient method.

Lemma (1): [23] Suppose that assumption equations (1) and (2) hold and $x_{k+1} = x_k + \alpha_k d_k$, where d_k is descent direction and α_k is a step size computed by using strong Wolf condition then:

$\sum_{k\ge1} \frac{(g_{k+1}^T d_{k+1})^2}{\|d_{k+1}\|^2} < \infty$ holds.

Lemma (2): [22] Suppose assumptions (1) and (2) hold, and let $x_{k+1} = x_k + \alpha_k d_k$ and $d_k = -g_k + \beta_{k-1}\, s_{k-1}$ ($k \ge 1$) where d_k is a descent direction and α_k is a step size determined by Strong Wolf line search, if $\sum_{k\ge1} \frac{1}{\|d_k\|^2} = \infty$ then: $\lim_{k\to\infty} inf\|g_k\| = 0$

Theorem (2): Consider the assumptions 1 and 2 hold and $\{x_k\}$ be generated by the new algorithm, then:
$\lim_{k\to\infty} inf\|g_{k+1}\| = 0$

Proof: We prove this theorem by using contradiction.
Suppose the theorem is false, then: $\exists\ r > 0$ such that: $\|g_k\| \geq r\ for\ all\ k$

From the theorem (1):
$g_{k+1}^T d_{k+1} \leq -K\|g_{k+1}\|^2\ for\ all\ k$

By using strong Wolfe condition, we obtain:

$$y_k^T d_k = g_{k+1}^T d_k - g_k^T d_k \geq \sigma g_k^T d_k - g_k^T d_k \geq -(1-\sigma) g_k^T d_k \geq K(1-\sigma)\|g_k\|^2$$

Multiplying both sides by $\alpha_k\ where\ \alpha_k > \lambda,\ for\ \lambda > 0, \forall k \geq 0$ then:

$$y_k^T s_k \geq K(1-\sigma)\alpha_k\|g_k\|^2 \geq K(1-\sigma)\lambda r^2$$

And since:

$$\|y_k\| = \|g_{k+1} - g_k\| \leq L\|x_{k+1} - x_k\| \leq LD\ \ (D\ is\ diameter\ of\ the\ level\ set\ S)$$

$$d_{k+1} = -g_{k+1} + \beta_k^{new} s_k$$

$$\|d_{k+1}\| \leq \|g_{k+1}\| + |\beta_k^{new}|\ \|s_k\|$$

1) $\boldsymbol{\beta_k^{SSH1} = 0.618\beta_k^{HS} + 0.382\beta_k^{DY}}$ **[21]**

$$|\beta_k^{new}| \leq 0.618|\beta_k^{HS}| + 0.382|\beta_k^{DY}|$$

$$(0.618)\beta_k^{HS} = (0.618)\frac{g_{k+1}^T y_k}{y_k^T s_k} \leq (0.618)\frac{\|g_{k+1}\|\ \|y_k\|}{\|y_k\|\ \|s_k\|} = (0.618)\frac{Ʈ LD}{K(1-\sigma)\lambda r^2}$$

$$0.382\beta_k^{DY} = (0.382)\frac{\|g_{k+1}\|^2}{y_k^T s_k} \leq (0.382)\frac{\|g_{k+1}\|^2}{\|y_k\|\ \|s_k} \leq (0.382)\frac{Ʈ^2}{K(1-\sigma)\lambda r^2}$$

$$|\beta_k^{SSH1}| \leq \frac{(0.618)Ʈ LD + (0.382)Ʈ^2}{K(1-\sigma)\lambda r^2} = M\ \ and\ since\ \alpha_k > \lambda\ then\ \frac{1}{\alpha_k} < \frac{1}{\lambda}$$

$$Since\ \|d_{k+1}\| \leq \|g_{k+1}\| + |\beta_k^{SSH1}|\ \|s_k\| \leq \|g_{k+1}\| + \frac{|\beta_k^{SSH1}|\|x_{k+1} - x_k\|}{\alpha_k} \leq Ʈ + \frac{MD}{\lambda} = W$$

Hence:

$$\|d_{k+1}\| \leq W\ then\ \sum_{k\geq1}\frac{1}{\|d_k\|^2} = \infty$$

From Zoutendijk condition we have:

$$\sum_{k\geq1}\frac{(g_{k+1}^T d_{k+1})^2}{\|d_{k+1}\|^2} < \infty$$

$$since\ \|g_{k+1}\| \geq r\ and\ g_{k+1}^T d_{k+1} \leq -K\|g_k\|^2$$

$$K^2 r^4 \sum_{k\geq1}\frac{1}{\|d_k\|^2} \leq \sum_{k\geq1}\frac{K^2\|g_k\|^4}{\|d_k\|^2} \leq \infty$$

Which is contradiction with $\sum_{k\geq1}\frac{1}{\|d_{k+1}\|^2} = \infty$

Then, we get $\lim_{k\to\infty} inf\|g_{k+1}\| = 0$

2) $\boldsymbol{\beta_k^{SSH2} = 0.618\beta_k^{LS} + 0.382\beta_k^{CD}}$[24]

$$d_{k+1} = -g_{k+1} + 0.382 g_{k+1} - 0.382 g_{k+1} + 0.618\beta_k^{LS} s_k + 0.382\beta_k^{CD} s_k$$

$$= -0.382 g_{k+1} + 0.382\beta_k^{CD} s_k - 0.618 g_{k+1} + 0.618\beta_k^{LS} s_k$$

$$= (-g_{k+1} + \beta_k^{CD} s_k)0.382 + (-g_{k+1} + \beta_k^{LS} s_k)0.618$$

$$d_{k+1} = 0.382 d_{k+1}^{CD} + 0.618 d_{k+1}^{LS}$$

$$\|d_{k+1}\| \leq \|d_{k+1}^{CD}\| + \|d_{k+1}^{LS}\|$$

Furthermore;

$$\|d_{k+1}^{LS}\| \leq \|g_{k+1}\| + |\beta^{LS}|\|s_k\|$$

From assumption (1) and (2), $\|g_{k+1}\| \leq Ʈ$, since D is diameter of the level set S and by descent condition, we have:

$$-g_k^T d_k \geq K\|g_k\|\ \text{ then } \frac{1}{-g_k^T d_k} \leq \frac{1}{K\|g_k\|}$$

$$hence\ |\beta^{LS}| = \frac{y_k^T g_{k+1}}{-d_k^T g_k} \leq \frac{y_k^T g_{k+1}}{K g_k} \leq \frac{\|y_k\|\|g_{k+1}\|}{K\|g_k\|} \leq \frac{L\|s_k\|}{K} = \frac{LD}{K}$$

$$similarly\ |\beta^{CD}| = \frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k} \leq \frac{g_{k+1}^T g_{k+1}}{K\|g_k\|} \leq \frac{\|g_{k+1}\|^2}{K\|g_k\|} \leq \frac{Ʈ}{K}$$

$$|\beta^{SSH2}| \leq 0.618|\beta_k^{LS}| + 0.382|\beta_k^{CD}|$$

Now, $\|d_{k+1}^{SSH2}\| \leq \|g_{k+1}\| + |\beta^{SSH2}|\|s_k\|$

Then,

$$\|d_{k+1}^{SSH2}\| \leq 2Ʈ + \left(0.618\frac{LD}{K} + 0.382\frac{Ʈ}{K}\right)\|s_k\| = 2Ʈ + MD\ where\ M = 0.618\frac{LD}{K} + 0.382\frac{Ʈ}{K}$$

$$\|d_{k+1}^{SSH2}\| \leq W\ where\ W = 2Ʈ + MD$$

We get: $\sum_{k\geq0}\frac{1}{\|d_{k+1}\|} = \infty$

By using Lemma (2), we obtain:
$\lim_{k\to\infty} inf\|g_{k+1}\| = 0$

3) $\boldsymbol{\beta_k^{SSH3} = 0.382\beta_k^{HS} + 0.618\beta_k^{DY}}$

Similar to part (1)

4) $\boldsymbol{\beta_k^{SSH4} = 0.382\beta_k^{LS} + 0.618\beta_k^{CD}}$

Similar to part (2)

The use of the golden section ratio in the line search process is a novel approach that enhances the algorithm's efficiency by reducing the number of function evaluations required.

4. Numerical Experiments

This section focuses on testing the new methods' implementation. Based on this, we evaluate the computing performance of the suggested approaches with several known algorithms such as DY, HS, and LS conjugate gradient algorithms. We consider 110 unconstrained optimization test problems, some of which are selected from the CUTE (constrained and unconstrained test environment) library [25] and the rest are from the unconstrained problems collections [26], [27]. The sizes of the test issues (denoted by n in the tables) range from 2 to 200000. To be fair, all comparison methods employ the strong Wolfe line search method to compute the step length $\propto_k$. The hybridization parameter θ equals to 0.382 for the creation of (β^{hSSH1} , β^{hSSH2}), and equals to 0.618 for the creation of (β^{hSSH3} , β^{hSSH4}). The relevant parameters are set to be δ=0.0001 and σ=0.9 for the proposed methods. The termination criterion is either (1) $\|g_k\|_\infty \leq 10^{-6}$ or (2) number of iteration (NOI) >2000. When (2) happens, the relevant algorithm is claimed to be invalid for the corresponding test problem, which is denoted by "NaN".

All codes are written in MATLAB (as a tool for data analysis) 2024b and run on a Lenovo PC with a 360GHz CPU (Central Processing Unit) processor, 8 GB of RAM memory, and the Windows 10 operating system.

The comparison of various methodologies is offered in the following context for example, let fi^{H1} and fi^{H2} be the optimal values determined by H_1 and H_2 for problems i=1…110 respectively. It is considered that in the specific problem i, if the performance of H_1 was better than the performance of H_2:

$$|f_i^{H1} - f_i^{H2}| < 10^{-3}$$

The number of iterations (NOI), or the number of function gradient-evaluation (NOF), or CPU time of H_1 methods is less than that of H_2 methods. To get comprehensive comparisons, the profile of Dolan and More [28] is utilized to evaluate and compare the performance of the collection of approaches.

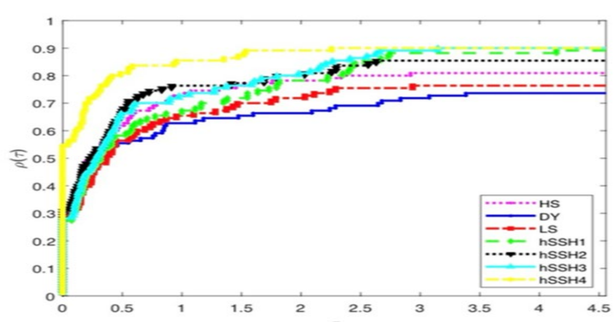

Figure 1. Number of Iteration.

Based on the extensive numerical data presented in table.1 and illustrated in **Figure.1**, the results clearly demonstrate that our proposed methods outperform the classical approaches in terms of number of iteration, with the fourth method (hSSH4) exhibiting the highest efficiency.

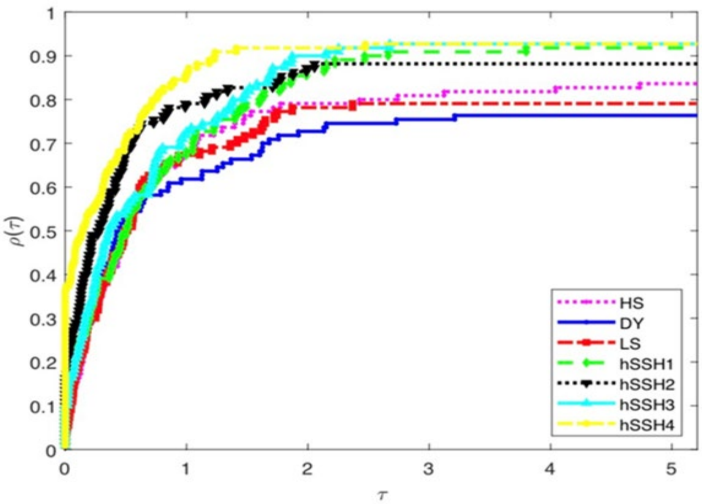

Figure 2. CPU Time.

The detailed numerical data presented in **Table 2** and illustrated in **Figure 2** provide performance profiles comparing the proposed methods with classical approaches (HS, DY, and LS). The new methods exhibit superior efficiency by solving the problems more quickly, with the fourth method (hSSH4) emerging as the most efficient.

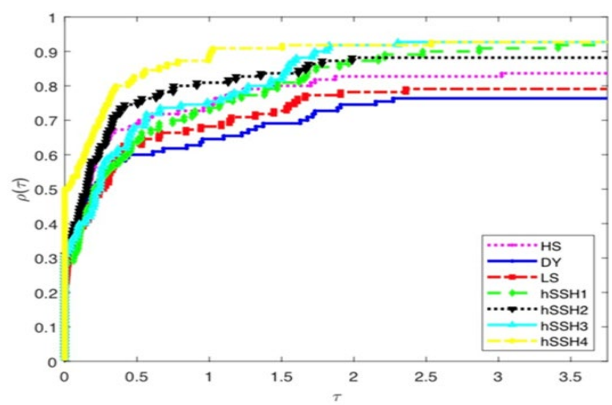

Figure 3. Gradient Function.

Figure 3 demonstrates how these algorithms perform in terms of the number of function gradient evaluations, highlighting that the proposed methods are capable of solving more functions than the classical approaches. It is worth noting that the fourth method (hSSH4) consistently demonstrates the highest efficiency across all three performance factors.

Table 1. Illustrates a numerical comparison between the classical methods and the proposed methods based on Number of Iteration.

#	Problem	n	HS	DY	LS	hSSH1	hSSH2	hSSH3	hSSH4
1	ARGLINB	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2		500	15	14	NaN	28	15	15	67
3		1000	18	27	20	79	20	19	15
4	BV	500	613	1593	1019	1342	1822	801	526
5		1000	250	229	168	165	140	133	101
6		10000	0	0	0	0	0	0	0
7		20000	0	0	0	0	0	0	0
8		30000	0	0	0	0	0	0	0
9	COSINE	10	12	12	11	11	11	12	11
10		100	11	11	11	10	11	11	10
11		1000	16	118	30	14	19	18	18
12		10000	14	14	14	16	15	15	16
13		100000	12	13	12	13	13	14	14
14	DIXMAANA	300	9	10	9	9	8	11	9
15		30000	9	9	10	9	9	10	9
16		60000	10	10	9	12	10	9	10
17		90000	11	10	10	10	10	9	10
18		120000	10	10	10	9	10	10	9
19	DIXMAANL	3000	2733	2613	1439	NaN	2283	2657	1431
20		9000	1572	NaN	2601	2698	1033	2326	774
21		30000	1649	NaN	NaN	1885	1218	NaN	1199
22		60000	NaN	NaN	NaN	NaN	NaN	NaN	NaN
23	DIXMAANK	3000	1617	NaN	2278	NaN	1222	1846	937
24		9000	1834	NaN	2602	2939	1092	2604	1003
25		30000	NaN	NaN	NaN	NaN	1474	NaN	1530
26		60000	NaN	NaN	NaN	NaN	NaN	NaN	NaN
27		120000	NaN	NaN	NaN	NaN	NaN	NaN	NaN
28	DIAG4	100	106	219	58	36	130	93	21
29		1000	54	189	138	204	129	139	40
30		10000	95	211	135	210	51	69	33
31		100000	105	132	40	208	100	131	31
32		200000	55	264	181	198	151	171	37
33	DIAG5	100	1	1	1	1	1	1	1
34		1000	7	7	7	7	7	7	7
35		10000	8	8	8	12	8	12	12
36		50000	8	8	8	8	8	8	8
37		200000	8	8	8	8	8	8	8
38	DIAG6	100	1	1	1	1	1	1	1
39		1000	8	8	8	8	8	8	8
40		10000	12	12	12	12	12	12	12
41		50000	8	8	8	8	8	8	8

42		100000	12	NaN	NaN	14	12	15	15
43	DIAG8	1000	9	9	9	11	9	11	11
44		5000	7	7	7	7	7	7	7
45		10000	8	8	8	8	8	8	8
46	DQRTIC	5000	17	17	17	38	32	54	33
47		10000	84	88	85	56	86	18	50
48		50000	21	36	21	98	56	21	61
49		100000	22	23	95	146	130	87	42
50		150000	63	23	176	126	113	23	66
51	EDENSCH	2	8	7	9	8	8	17	8
52		100	48	49	44	43	39	40	29
53		1000	43	67	50	67	44	75	31
54		10000	80	46	67	43	38	38	43
55		100000	45	65	49	45	38	46	36
56	DENSCHNF	10	18	18	15	19	16	18	16
57		100	15	16	20	16	18	17	16
58		1000	17	19	18	20	23	15	22
59		10000	20	22	19	20	16	20	21
60		100000	19	17	18	18	19	21	21
61	EDENSCHNB	2	13	12	13	12	13	13	11
62		100	12	12	13	14	11	12	12
63		10000	14	14	14	15	13	13	14
64		100000	18	17	18	14	17	15	16
65	HIMMELBG	10000	2	2	2	2	2	2	2
66		100000	2	2	2	2	2	2	2
67	IE	100	8	9	8	8	9	9	9
68		500	9	8	9	9	9	9	9
69	ENGVALI	10	41	40	39	39	37	37	NaN
70		1000	43	41	42	42	38	43	23
71		10000	40	41	42	42	36	37	28
72	EVF	2	38	34	40	34	38	39	26
73	EXPENALTY	100	6	6	6	6	6	6	6
74		1000	14	NaN	NaN	11	14	11	11
75		25000	12	12	12	12	12	12	15
76		50000	11	11	11	11	11	11	11
77	EXTROSNB	2	105	388	167	1096	304	62	64
78		500	64	66	71	64	61	59	42
79		1000	NaN	NaN	NaN	NaN	NaN	NaN	NaN
80		10000	1649	NaN	NaN	1161	265	666	218
81	EXROSEN	10	NaN	NaN	NaN	832	NaN	1533	319
82		100	NaN	NaN	NaN	914	NaN	1729	195
83		1000	NaN	NaN	NaN	661	NaN	1326	295
84		5000	NaN	NaN	NaN	1077	NaN	1903	324
85		10000	NaN	NaN	NaN	1115	NaN	1954	316

86		50000	NaN	NaN	NaN	1100	NaN	2007	317
87		100000	NaN	NaN	NaN	1149	NaN	2087	392
88	EXHIMMELBLAU	10	18	18	20	19	21	19	21
89		1000	16	18	22	18	21	15	21
90		10000	16	15	20	17	19	16	22
91	GENHUMPS	2	6	6	6	6	6	6	6
92		100	11	NaN	13	12	11	12	11
93		500	NaN	NaN	NaN	NaN	NaN	NaN	NaN
94		1500	11	12	10	11	13	12	NaN
95	GENQUARTIC	1000	24	18	21	16	17	19	18
96		25000	13	13	12	10	14	14	12
97		75000	12	13	14	15	14	13	12
98	GQUARTIC	50	NaN	NaN	NaN	161	165	180	148
99		100	NaN	NaN	NaN	262	280	302	220
100		500	NaN	NaN	NaN	934	1081	1141	767
101		1000	NaN	NaN	NaN	1835	2092	2169	1596
102	HARKERP	100	9	9	9	12	9	12	12
103		1000	10	10	10	58	10	13	13
104		10000	9	25	11	13	16	10	23
105		50000	19	16	16	13	19	14	14
106	HIMMELBH	100	18	16	19	19	14	19	14
107		10000	22	21	23	23	22	24	21
108		100000	25	24	25	22	23	22	22
109	HIMMELBG	100	2	2	2	2	2	2	2
110		1000	2	2	2	2	2	2	2

Table 2. Demonstrates a numerical comparison between the classical methods and the proposed methods based on CPU time.

#	Problem	n	HS	DY	LS	hSSH1	hSSH2	hSSH3	hSSH4
1	RGLINB	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2		500	0.322	0.278	NaN	0.654	0.301	0.326	1.544
3		1000	1.322	2.08	1.459	6.37	1.491	1.481	1.152
4	BV	500	4.444	10.56	7.099	8.99	10.771	5.499	3.294
5		1000	5.119	4.247	3.443	3.204	2.693	2.768	1.944
6		10000	0.385	0.435	0.376	0.352	0.376	0.331	0.352
7		20000	4.189	3.289	2.876	2.305	1.524	1.505	1.588
8		30000	6.17	5.014	5.1	5.036	4.706	4.789	4.286
9	COSINE	10	0.355	0.041	0.017	0.023	0.013	0.018	0.023
10		100	0.028	0.016	0.022	0.015	0.021	0.037	0.018
11		1000	0.048	0.211	0.07	0.057	0.05	0.048	0.048
12		10000	0.192	0.174	0.195	0.199	0.172	0.203	0.211
13		100000	0.981	1.119	1.265	1.043	0.864	1.034	1.044
14	AANA	300	0.047	0.03	0.031	0.034	0.035	0.05	0.048

15		30000	1.049	1.029	1.096	1.147	1.067	1.261	0.993
16		60000	2.223	2.155	1.988	2.562	2.143	1.781	2.235
17		90000	3.123	2.86	2.952	2.966	2.868	2.786	3.066
18		120000	4.15	4.268	5.11	4.392	4.231	4.469	3.657
19	AANL	3000	27.198	28.529	10.183	NaN	13.771	19.017	9.839
20		9000	30.659	NaN	47.928	47.545	16.31	41.997	13.107
21		30000	85.146	NaN	NaN	86.94	48.802	NaN	50.972
22		60000	NaN	NaN	NaN	NaN	NaN	NaN	NaN
23	AANK	3000	12.152	NaN	16.639	NaN	8.311	13.318	6.481
24		9000	33.618	NaN	43.654	48.192	16.34	44.005	15.666
25		30000	NaN	NaN	NaN	NaN	58.934	NaN	65.255
26		60000	NaN	NaN	NaN	NaN	NaN	NaN	NaN
27		120000	NaN	NaN	NaN	NaN	NaN	NaN	NaN
28	DIAG4	100	0.142	0.245	0.082	0.04	0.11	0.086	0.026
29		1000	0.069	0.211	0.158	0.213	0.136	0.15	0.068
30		10000	0.447	0.797	0.66	0.923	0.294	0.413	0.21
31		100000	2.469	3.058	1.187	4.342	2.144	3.093	1.192
32		200000	3.133	10.426	7.86	8.247	6.022	7.789	2.381
33	DIAG5	100	0.009	0.001	0.003	0.001	0.001	0.002	0.001
34		1000	0.035	0.036	0.044	0.034	0.033	0.035	0.039
35		10000	0.234	0.189	0.154	0.258	0.156	0.248	0.253
36		50000	0.538	0.527	0.497	0.497	0.505	0.517	0.508
37		200000	2.419	2.419	2.435	2.419	2.432	2.414	2.37
38	DIAG6	100	0.007	0.001	0	0	0.001	0.001	0.001
39		1000	0.013	0.015	0.016	0.017	0.013	0.016	0.019
40		10000	0.091	0.078	0.066	0.067	0.077	0.074	0.07
41		50000	0.141	0.135	0.117	0.12	0.116	0.117	0.115
42		100000	0.277	NaN	NaN	0.383	0.285	0.406	0.389
43	DIAG8	1000	0.021	0.019	0.032	0.035	0.023	0.033	0.045
44		5000	0.063	0.054	0.049	0.045	0.04	0.047	0.042
45		10000	0.077	0.07	0.073	0.064	0.059	0.067	0.063
46	DQRTIC	5000	0.295	0.264	0.273	0.547	0.437	0.63	0.482
47		10000	1.796	1.777	1.761	1.234	1.755	0.526	1.115
48		50000	2.502	3.916	2.516	8.511	5.545	2.565	5.014
49		100000	5.229	5.252	16.016	24.417	20.742	15.004	8.538
50		150000	16.679	7.962	40.752	31.306	27.065	7.895	15.983
51	EDENSCH	2	0.013	0.011	0.008	0.009	0.007	0.019	0.009
52		100	0.055	0.073	0.053	0.056	0.046	0.062	0.038
53		1000	0.182	0.319	0.216	0.313	0.183	0.342	0.134
54		10000	2.277	1.065	1.057	0.83	0.714	0.694	1.292
55		100000	8.402	16.756	10.034	9.357	6.805	8.764	8.622
56	DENSCHNF	10	0.039	0.031	0.032	0.029	0.044	0.032	0.036
57		100	0.031	0.038	0.035	0.032	0.036	0.036	0.038
58		1000	0.057	0.064	0.054	0.068	0.064	0.052	0.075

59		10000	0.256	0.25	0.262	0.253	0.233	0.237	0.264
60		100000	1.923	1.973	1.837	1.758	1.925	2.012	2.016
61	DENSCHNB	2	0.009	0.011	0.012	0.011	0.008	0.014	0.013
62		100	0.023	0.029	0.026	0.028	0.016	0.021	0.027
63		10000	0.166	0.177	0.159	0.164	0.114	0.116	0.119
64		100000	0.824	0.813	0.854	0.743	0.86	0.784	0.771
65	HIMMELBG	10000	0.015	0.012	0.013	0.011	0.013	0.013	0.011
66		100000	0.077	0.08	0.075	0.077	0.079	0.075	0.074
67	IE	100	0.301	0.249	0.227	0.208	0.227	0.255	0.248
68		500	5.311	5.065	6.113	5.793	5.641	5.606	5.461
69	ENGVALI	10	0.029	0.026	0.031	0.035	0.033	0.031	NaN
70		1000	0.032	0.026	0.033	0.03	0.028	0.032	0.019
71		10000	0.107	0.105	0.126	0.13	0.113	0.107	0.099
72	EVF	2	0.033	0.023	0.028	0.032	0.029	0.036	0.029
73	M)ENALTY	100	0.009	0.008	0.006	0.006	0.007	0.005	0.007
74		1000	0.015	NaN	NaN	0.017	0.018	0.013	0.012
75		25000	0.115	0.11	0.097	0.092	0.095	0.099	0.124
76		50000	0.135	0.124	0.126	0.123	0.12	0.126	0.117
77	EXTROSNB	2	0.079	0.312	0.138	0.655	0.173	0.051	0.047
78		500	0.044	0.045	0.048	0.045	0.045	0.039	0.032
79		1000	NaN	NaN	NaN	NaN	NaN	NaN	NaN
80		10000	4.013	NaN	NaN	2.67	0.654	1.794	0.601
81	EXROSEN	10	NaN	NaN	NaN	0.665	NaN	1.367	0.376
82		100	NaN	NaN	NaN	0.761	NaN	1.605	0.251
83		1000	NaN	NaN	NaN	0.654	NaN	1.315	0.413
84		5000	NaN	NaN	NaN	2.177	NaN	3.68	1.048
85		10000	NaN	NaN	NaN	3.277	NaN	5.469	1.229
86		50000	NaN	NaN	NaN	8.535	NaN	15.431	3.238
87		100000	NaN	NaN	NaN	14.384	NaN	24.892	6.813
88	LBLAU	10	0.066	0.036	0.034	0.036	0.033	0.048	0.047
89		1000	0.047	0.058	0.064	0.064	0.069	0.046	0.092
90		10000	0.248	0.214	0.216	0.176	0.173	0.164	0.219
91	GENHUMPS	2	0.01	0.003	0.004	0.01	0.005	0.004	0.005
92		100	0.01	NaN	0.016	0.012	0.011	0.011	0.015
93		500	NaN	NaN	NaN	NaN	NaN	NaN	NaN
94		1500	0.03	0.036	0.029	0.027	0.037	0.038	NaN
95	NQUARTIC	1000	0.026	0.02	0.026	0.017	0.03	0.025	0.027
96		25000	0.107	0.11	0.107	0.081	0.09	0.099	0.079
97		75000	0.177	0.193	0.191	0.205	0.196	0.199	0.183
98	GQUARTIC	50	NaN	NaN	NaN	0.13	0.131	0.14	0.118
99		100	NaN	NaN	NaN	0.239	0.26	0.272	0.202
100		500	NaN	NaN	NaN	0.986	1.154	1.417	0.867
101		1000	NaN	NaN	NaN	5.685	6.42	4.484	2.704
102	RKERP	100	0.016	0.009	0.012	0.013	0.006	0.015	0.015

103		1000	0.031	0.022	0.024	0.142	0.026	0.027	0.03
104		10000	0.061	0.186	0.095	0.122	0.107	0.078	0.163
105		50000	0.463	().404	0.384	0.316	0.488	0.359	0.346
106	MELBH	100	0.019	0.019	0.02	0.021	0.014	0.027	0.016
107		10000	0.198	0.202	0.161	0.161	0.174	0.182	0.154
108		100000	1.175	1.127	1.324	1.034	1.133	1.207	1.167
109	MELBG	100	0.001	0.001	0.001	0.001	0.002	0.002	0.002
110		1000	0.001	0.002	0.002	0.003	0.002	0.003	0.003

5. Conclusion

Conjugate gradient methods are frequently used to address unconstrained optimization problems, particularly in large scales. Hybrid strategies, which integrate classical approaches, are among the most useful. To create a practical and effective technique, this paper presents well-designed hybrid conjugate gradient algorithms that effectively leverage the golden section ratio to enhance the performance of traditional CG methods. The theoretical and empirical results are compelling, suggesting that the proposed algorithms represent a valuable contribution to the methods available for solving unconstrained optimization problems. The new algorithms, in particular, showed superior performance based on key factors such as CPU time, number of gradient function evaluations, and number of iterations. One of the main challenges faced during this research was selection and integration of the most effective features of the classical methods, particularly those derived from the DY-HS and LS-CD approaches. It is recommended for future work, to explore the use of these new hybrid algorithms in wider domains such as fuzzy logic systems, time series analysis, and finite difference methods.

Conflict of interest

None.

References

[1] Y. Dai, J. Han, G. Liu, D. Sun, H. Yin, and Y. Yuan, “Convergence Properties of Nonlinear Conjugate Gradient Methods,” *SIAM Journal on Optimization,* vol. 10, no. 2, pp. 345-358, 2000, DOI: 10.1137/S1052623494268443.

[2] Y. N. Huda, and I. A. Huda, “A Modification of Dai-Yuan’s Conjugate Gradient Algorithms for Solving Unconstrained Optimization,” Bulletin of the South Ural State University. Ser. Mathematical Modelling, Programming & Computer Software, vol. 15, no. 3, pp. 127-133, 2022, DOI:10.14529/mmp220309.

[3] S. B. Hanachi, B. Sellami, and M. Belloufi, “A New Family of Hybrid Conjugate Gradient Method for Unconstrained Optimization and Its Application to Regression Analysis,” RAIRO-Oper. Res, vol. 58, pp. 613-627, 2024, DOI:10.1051/ro/2023196.

[4] M. R. Hestenes, and E. Stiefel, “Method of conjugate gradient for solving linear equations,” J. Res. Nat. Bur. Stand, vol. 49, pp. 409-436, 1952, DOI:10.6028/JRES.049.044.

[5] R. Fletcher, and C. Reeves, “Function minimization by conjugate gradients,” Compute. J., vol. 7, pp. 149-154, 1964, DOI:10.1093/COMJNL/7.2.149.

[6] S. S. Djordjevic, “New hybrid conjugate gradient method as a convex combination of FR and PRP Methods,” Filomat, vol. 30, no. 11, pp. 3083-3100, 2016, DOI:10.2298/FIL1611083D.

[7] R. Fletcher, Practical Methods of Optimization: Unconstrained Optimization, vol. 1, 2nd ed. New York, NY, USA: Willey, 1997, ISBN:9780471915478.

[8] Y. Liu, and C. Storey, “Efficient generalized conjugate gradient algorithms,” J. Optim., vol. 69, no. 1, pp. 129-137, 1991, DOI: 10.1007/BF00940464.

[9] Y. Dai, and Y. Yuan, “A nonlinear conjugate gradient with a strong global convergence property,” SIAM J. Optim., vol. 10, pp. 177-182, 2000, DOI:10.1137/S1052623497318992.

[10] Y. H. Dai, and L. Z. Liao, “New Conjugacy Conditions and Related Nonlinear Conjugate Gradient Methods,” Applied Mathematics & Optimization, vol. 43, no. 1, pp. 87-101, 2001, DOI: 10.1007/s002450010019.

[11] S. Babaie-Kafaki, and R. Ghanbari, “A descent family of Dai-Liao conjugate gradient methods,” Optimization Methods and Software, vol. 29, no. 3, pp. 583-591, 2014, DOI: 10.1080/10556788.2013.833199.

[12] S. Babaie-Kafaki, and R. Ghanbari, “Two optimal Dai-Liao conjugate gradient methods,” Optimization, vol. 64, no. 11, pp. 2277-2287, 2015, DOI: 10.1080/02331934.2014.938072.

[13] S. Babaie-Kafaki, and R. Ghanbari, “A class of adaptive Dai-Liao conjugate gradient methods based on the scaled memoryless BFGS update,” 4OR, vol. 15, no. 1, pp. 85-92, 2017, DOI: 10.1007/s10288-016-0323-1.

[14] M. Fatemi, “A new efficient conjugate gradient method for unconstrained optimization,” Journal of Computational and Applied Mathematics, vol. 300, pp. 207-216, 2016, DOI: 10.1016/j.cam.2015.12.035.

[15] M. Fatemi, “An Optimal Parameter for Dai-Liao Family of Conjugate Gradient Methods,” Journal of Optimization Theory and Applications, vol. 169, no. 2, pp. 587-605, 2016, DOI: 10.1007/s10957-015-0786-9.

[16] S. Babaie-Kafaki, and R. Ghanbari, “The Dai-Liao nonlinear conjugate gradient method with optimal parameter choices,” European Journal of Operational Research, vol. 234, no. 3, pp. 625-630, 2014, DOI: 10.1016/j.ejor.2013.11.012.

[17] N. Andrei, “A Dai-Liao conjugate gradient algorithm with clustering of eigenvalues,” Numerical Algorithms, vol. 77, no. 4, pp. 1273-1282, 2018, DOI: 10.1007/s11075-017-0362-5.

[18] S. S. Djordjevic, “New Hybrid Conjugate Gradient Method as a Convex Combination of HS and FR Methods,” Journal of Applied Mathematics and Computation, vol. 2, no. 9, pp. 366-378, 2018, DOI: 10.26855/jamc.2018.09.002.

[19] S. S. Djordjevic, "New Hybrid Conjugate Gradient Method as a Convex Combination of LS and CD methods," Filomat, vol. 31, no. 6, pp. 1813-1825, 2017, DOI:10.2298/FIL1706813D.

[20] S. Li, and Z. Sun, "A new hybrid conjugate gradient method and its global convergence for unconstrained optimization," International Journal of Pure and Applied Mathematics, vol. 63, no. 3, pp. 285-296, 2010, DOI: 10.1109/CCDC.2012.6244138.

[21] N. Andrei, "Another hybrid conjugate gradient algorithm for unconstrained optimization," Numer. Algor., vol. 47, pp. 143-156, 2008, DOI:10.1007/s11075-007-9152-9.

[22] S. B. Hanachi, B. Sellami, and M. Belloufi, "New Iterative Conjugate Gradient Method for Nonlinear Unconstrained Optimization," RAIRO Operations Research, vol. 56, no. 3, pp. 2315-2327, 2022, DOI: DOI:10.1051/ro/2022109.

[23] Y. A. Laylani, H. M. Khudhur, E. M. Nori, and K. K. Abbo, "A hybridization of the Hestenes-Stiefel and Dai-Yuan Conjugate Gradient Methods," European Journal of Pure and Applied Mathematics, vol. 16, no. 2, pp. 1059-1067, 2023, DOI: 10.29020/nybg.ejpam.v16i2.4746.

[24] K. K. Abbo, and N. H. Hameed, "New Hybrid Conjugate Method as a Convex Combination of Liu-Storey and Dixon Methods," Journal of Multidisciplinary Modeling and Optimization, vol. 1, no. 2, pp.91-99, 2018, URL: https://journals.indexcopernicus.com/search/article?articleId=2631251.

[25] M. J. D. Powell, "Restart procedures of the conjugate gradient method," Math. Program, vol. 12, no.1, pp. 241-254, 1977, DOI: 10.1007/BF01593790.

[26] D. Touati-Ahmed, and C. Storey, "Efficient hybrid conjugate gradient techniques," J. Optim. Theory Appl., vol. 64, pp. 379-397, 1990, DOI:10.1007/BF00939455.

[27] G. Zoutendijk, "Nonlinear programming computational methods," in Integer and Nonlinear Programming, J. Abadie, Ed. Amsterdam, Netherlands: North-Holland, 1970, pp. 37-86, ISBN:0-444-10000-8.

[28] E. D. Dolan, and J. J. More, "Benchmarking optimization software with performance profiles," Math. Program., vol. 91, pp. 201-213, 2002, DOI:10.1007/s101070100263.