IRAQI Academic Scientific Journals

Samarra Journal of Pure and Applied Science

SJPAS

مجلة سامراء للعلوم الصرفة والتطبيقية

p ISSN: 2663-7405
e ISSN: 2789-6838

www.sjpas.com

An Efficient Spectral Parameter for Accelerating the Conjugate Gradient Algorithm

Zyiad Adrees Hamad[1,2*] Ghada Moayid Al-Naemi[1]

1- Department of Mathematics, College of Computer Science and Mathematics, University of Mosul, Iraq
2- Nineveh Education Directorate, Ministry of Education, Nineveh, Iraq

https://doi.org/10.54153/sjpas.2025.v7i1.987

Article Information

Received: 24/07/2024
Revised: 14/08/2024
Accepted: 20/09/2024
Published: 30/03/2025

Keywords:

Conjugate gradient algorithms, Strong Wolfe Conditions, Numerical analysis, Spectral conjugate gradient, Inexact line search.

Corresponding Author

E-mail: *zyiad.23csp77@student.unmosul.edu.iq*
Mobile: +964770307892

Abstract

Conjugate gradient (CG) methods are renowned for their efficiency in optimizing problems due to their speed and memory use. However, certain formulations of CG methods frequently change between different parameters, potentially omitting valuable values. Additionally, proving adequate descent properties in these methods often relies on a strategy that assumes the ability to exclude specific function values. This study introduces a novel spectral CG (SCG) algorithm that integrates both the spectral gradient parameter and conjugate gradient coefficient. This method stands out for its applicability to large-scale unconstrained optimization problems. Initial numerical findings employing a robust Wolfe line search are provided to showcase the method's efficiency. The spectral coefficient in the field of optimization is a concept used to enhance the performance of search algorithms in unconstrained optimization. It is typically employed in the context of line search methods to improve the convergence rate of algorithms. In optimization algorithms, the spectral coefficient helps adjust the step size to achieve faster convergence towards the optimal solution. The coefficient is calculated based on information from previous iterations of the algorithm. A common application of this concept is in the Spectral Gradient method, where the spectral coefficient is used to modify the gradient direction to accelerate the search process. In this method, the spectral coefficient is used to determine the step length in each iteration, aiming to achieve an optimal balance between speed and stability in finding the optimal solution

INTRODUCTION

The conjugate gradient (CG) method is highly effective for solving large-dimensional, unconstrained optimization problems. Traditionally, this approach is represented

$$min\, u(x), x \in \Re^n \tag{1}$$

Where $u: \Re^n \longrightarrow \Re$ is a smooth function, $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$ are defined for k = 0, 1, 2,.... Starting with an initial guess x_0, the subsequent iterations are given by:

$$x_k = x_{k-1} + \omega_{k-1} d_{k-1} \tag{2}$$

When $\omega_{k-1} > 0$ is the step-dimension obtained from the line search approach, we used the strong Wolfe line search (SWL), which is defined as: $u(x_{k-1} + \omega_{k-1} d_{k-1}) \leq u(x_{k-1}) + \delta_1 \omega_{k-1} g_{k-1}^T d_{k-1}$ (3)

$$|g(x_k + \omega_{k-1} d_{k-1})^T d_{k-1}| \leq \delta_2 |g_{k-1}^T d_{k-1}| \tag{4}$$

Where $0 < \delta_1 < \delta_2 < 1$. Also, the classical search direction d_k is given as:

$$d_k = \begin{cases} -g_k & , k = 0 \\ -g_k + \beta_{k-1} d_{k-1} & , k \geq 1 \end{cases} \tag{5}$$

Using g_k to represent the gradient at the k-th iterate x_k and the element $\beta_{k-1} \in \Re$ is the CG coefficient that characterizes different CG methods. "The most well-known conjugate gradient (CG) methods include Hestenes-Stiefel (HS) [1], Fletcher-Reeves (FR) [2], Polak-Ribière (PR) [3] [4], and Dai-Yuan (DY) [5].

"The conventional conjugacy requirement is crucial in the development of specific CG algorithms. For large-scale problems, exact computation-based methods are computationally expensive. Designing efficient CG algorithms often involves formulating them with a generalized approach for selecting step sizes. Salihu et al. [6] discuss the standard conjugacy condition."

$$d_k^T y_{k-1} = 0, \tag{6}$$

Perry [7] expanded on the findings of equation (6) by adjusting the secant condition within a quasi-Newton method and modifying the quasi-Newton search direction..

$$d_k^T y_{k-1} = -g_k^T s_{k-1}, \tag{7}$$

This implies that (7) holds true for exact line search, but ILS is commonly used in practical numerical computations. Dai and Liao [8] address this by replacing (7) with the Dai-Liao extended conjugacy line condition.

$$d_k^T y_{k-1} = -\zeta g_k^T s_{k-1}, \zeta > 0. \tag{8}$$

The spectral coefficient is a concept in optimisation used to enhance the performance of search algorithms in unconstrained optimization. It is typically employed in line search methods to improve algorithm convergence rates.

In optimization algorithms, the spectral coefficient helps adjust the step size to achieve faster convergence towards the optimal solution. The coefficient is calculated based on information from previous iterations of the algorithm. A common application of this concept is in the Spectral Gradient method, where the spectral coefficient is used to modify the gradient direction to accelerate the search process.

In this method, the spectral coefficient is used to determine the step length in each iteration, aiming to achieve an optimal balance between speed and stability in finding the optimal solution[31]

You may be able to achieve distinctive results through current research, as you can compare these results with those obtained in previous studies. The comparisons included the clear superiority of the method developed in this research, as it provided significant advantages in performance and accuracy. "Our results not only outperform previous work but contribute to providing more efficient solutions in the numerical domain. In this paper, we presented a new method in the field of large scalars and studied it by applying it to a set of problems. Due to the experimental results, the method used achieves remarkable superiority compared to the traditional methods used in recent research.

It enables solutions to be optimized and reduces the number of computational cycles required to reach the solution, resulting in overall computational efficiency. These results confirm the effectiveness of the new authority and its ability to deal with unrestricted representation issues more efficiently.

Thanks to these results, we can conclude that the development technique presents nine outstanding results in the field of complete numerical, with wide application in various solutions and accurate and non-free calculation. In this research, we relied on a set of basic assumptions that guided the methods and their applications:

Parametric of the objective machine: It is assumed that the objective function which we do not deal with is continuous and differentiable at least in the range in which the solution is sought exclusively. This assumption helps ensure that there are solutions that can be approached analytically. Data Structure: It is assumed that the input data used in the experiments come from known distributions or can be estimated accurately, allowing performance to be evaluated under specified conditions. Test environment: It was assumed that the test environment used to analyze the results represents typical cases of numerical optimization problems, allowing the results to be generalized more broadly. Independence of variables: We assumed that the independent variables in the objective function do not have strong interrelations that may affect performance, which is an important assumption to simplify analysis and calculations.

Derivation of new SCG parameter

The SCG methods represent a highly efficient subset of conjugate gradient (CG) methods utilized extensively for tackling large-scale problems. These methods exhibit not only global convergence properties but also adhere to the sufficient descent condition. Additionally, SCG methods are distinguished by their cost-effectiveness and reduced space requirements. They are particularly notable for their straightforward algebraic operations and the ease with which computer code can be developed [9].

The SCG method is characterized by its approach of combining the CG search direction with a scalar spectral parameter to form a novel search direction. Birgin and Martínez [10] introduced an SCG variant that builds upon the standard secant equation [11]. This method defines its direction as follows:

$$d_k = -\theta_{k-1} g_k + \beta_{k-1} d_{k-1} \tag{9}$$

Barazilai and Borwein [12] introduced the spectral parameter step, denoted as θ_(k-1), in 1988. Raydan [13] further developed the SCG method for potential large-scale unconstrained optimization. A significant advantage of this approach lies in its utilization of gradient directions exclusively during each search, ensuring global convergence. Birgin and Martínez [14] confirmed the global convergence of their SCG method. However, there is no assurance that the SCG method will consistently yield a descending direction. Consequently, Andrei [15] suggested a reduced proportion for the Wolfe line search.

Moreover, [16] builds on Jiang et al.'s enhanced CG algorithm, while [17] proposes an SCG method incorporating a sufficient descent feature, as indicated by Zhang et al. Various SCG methods have been proposed by numerous authors and can be found in references [18-24]

Multiply the sides equation (9) by y_k, then apply equation (8).

$$-\zeta s_{k-1}^T g_k = -\theta_{k-1} g_k^T y_{k-1} + \beta_{k-1} d_{k-1}^T y_{k-1}$$

Then, we get

$$\theta_{k-1} = \frac{\zeta s_{k-1}^T g_k + \beta_{k-1} d_{k-1}^T y_{k-1}}{g_k^T y_{k-1}} = \frac{\alpha_k \zeta d_{k-1}^T g_k + \beta_{k-1} d_{k-1}^T y_{k-1}}{g_k^T y_{k-1}} \tag{10}$$

Zyiad et al. [25] proposed the following parameter

$$\beta_k^{ZG} = \frac{[1-\zeta\left(1+\frac{\omega_{k-1}\|d_{k-1}\|^2}{y_{k-1}^T d_{k-1}}\right)] g_k^T d_{k-1} + \zeta \frac{g_k^T y_{k-1}}{\|y_{k-1}\|^2} y_{k-1}^T d_{k-1}}{\|d_{k-1}\|^2}. \tag{11}$$

Substituting equation (11) into (10), we get

$$\theta_{k-1} = \frac{\alpha_{k-1}\zeta d_{k-1}^T g_{k-1}\|d_{k-1}\|^2 + (\left[1-\zeta\left(1+\frac{\alpha_{k-1}\|d_{k-1}\|^2}{y_{k-1}^T d_{k-1}}\right)\right] g_k^T d_{k-1} + \zeta \frac{g_k^T y_{k-1}}{\|y_{k-1}\|^2} y_{k-1}^T d_{k-1}) y_{k-1}^T d_{k-1}}{(g_k^T y_{k-1})\|d_{k-1}\|^2}$$

Using some algebraic operations, we obtain

$$\theta_{k-1} = \frac{(1-\zeta)(g_k^T d_{k-1})(y_{k-1}^T d_{k-1})}{(g_k^T y_{k-1})\|d_{k-1}\|^2} + \zeta \frac{(y_{k-1}^T d_{k-1})^2}{\| y_{k-1}\|^2\|d_{k-1}\|^2} \qquad (12)$$

Algorithm (SZG)

1. Choose initial point $x_0 \in \Re^n$, $\epsilon > 0$, and $g_0 = -\nabla u(x_0)$, set $d_0 = -g_0$, where k=0.
2. If $g_k < \epsilon$, stop; otherwise, go to next step.
3. Determine a step size ω_k by using SWL search in (3) and (4).
4. Calculate the new point by (2), calculate g_{k+1} , if $g_{k+1} < \epsilon$, then stop.
5. Calculate the new direction by (9) and using (11) and (12)
6. If the restart criteria $|g_k^T g_{k-1}| \geq 0.2\|g_k\|^2$ are achieved, set $d_k = -g_k$,; goto step 2. Otherwise, continue.
7. Set k=k+1 and go to step 3.

The sufficient regression theorem for the N2T algorithm

The descent condition is critical in assessing the convergence of various methods. A CG method is defined as the following condition:

$$d_k^T g_k \leq -\mathcal{C}\,\|g_k\|^2 \; for\; k \geq 0, \mathcal{C} > 0 \,, \qquad (13)$$

then (13) is known as sufficient descent condition.

Theorem (1): Assuming a SCG method with search direction (9) and β_k^{ZG} given by equation (11), condition (13) applies for all $k \geq 0$.

Proof: We proceed by induction, since $g_0^T d_0 = -\|g_0\|^2$, the condition (13) satisfied when k=0. We assume it is true for $k \geq 1$. Also, the inequality (13) hold. Multiplying both sides of equation (9) by g_k^T yields

$$g_k^T d_k = -\theta_{k-1} g_k^T g_k + \beta_{k-1} g_k^T d_{k-1} \qquad (14)$$

Zyaid et al. [25] proved that $\beta_k^{ZG} \leq \mathcal{A}\|g_k\|^2$. Now, we need to prove θ_{k-1}

$$\theta_{k-1} = \frac{(1-\zeta)(g_k^T d_{k-1})(y_{k-1}^T d_{k-1})}{(g_k^T y_{k-1})\|d_{k-1}\|^2} + \zeta \frac{(y_{k-1}^T d_{k-1})^2}{\| y_{k-1}\|^2\|d_{k-1}\|^2}$$

We have $g_k^T d_{k-1} \leq y_{k-1}^T d_{k-1}$, (15)

and apply Cauchy-Schwartz inequality

$$(y_{k-1}^T d_{k-1})^2 \leq \| y_{k-1}\|^2\|d_{k-1}\|^2 \qquad (16)$$

This gives:

$$\theta_{k-1} \leq \frac{(1-\zeta)\| y_{k-1}\|^2}{(g_k^T d_{k-1})} + \zeta \tag{17}$$

$$g_k^T y_{k-1} = \|g_k\|^2 - g_k^T g_{k-1}, \tag{18}$$

Let φ_{k-1} be the angle formed by g_k and g_{k-1}, then

$$g_k^T g_{k-1} = \|g_k\| \|g_{k-1}\| \cos\varphi_{k-1}, \tag{19}$$

Applying equation (19) to equation (17), we get

$$g_k^T y_{k-1} = \|g_k\|^2 \left(1 - \frac{\|g_{k-1}\|}{\|g_k\|}\right) \cos\varphi_{k-1}, |\cos\varphi_{k-1}| \leq 1 \tag{20}$$

$$\theta_{k-1} \leq \frac{(1-\zeta)\| y_{k-1}\|^2}{\|g_k\|^2\left(1-\frac{\|g_{k-1}\|}{\|g_k\|}\right)} + \zeta$$

We have

$$\|y_{k-1}\| \leq \|g_k\|, \tag{21}$$

then

$$\theta_{k-1} \leq (1-\zeta)\left(\frac{\|g_k\|}{\|g_k\|-\|g_{k-1}\|}\right) + \zeta$$

$$\theta_{k-1} \leq \mathcal{B}$$

Substituting into equation (10), we get:

$$d_k^T g_k \leq -\mathcal{B}\|g_k\|^2 + \mathcal{A}\|g_k\|^2$$

$$\leq -[\mathcal{B} - \mathcal{A}]\|g_k\|^2$$

$$d_k^T g_k \leq -\mathcal{C}\,\|g_k\|^2$$

This implies that (13) holds.

Lemma (1): Assume Algorithm (SZG) generates the sequences $\{x_k\}$ and $\{d_k\}$, with the search direction fulfilling (13) and ω_{k-1} satisfying (3) - (4), then

$$\sum_{k\geq 1} \frac{1}{\|d_k\|^2} = \infty \tag{22}$$

Then

$$lim_{k\to\infty} inf \|g_k\| = 0\ . \tag{23}$$

The proof of Lemma 1 is given in Zoutendijk [26].

Certainly! Here's the detailed explanation in English:

Detailed Explanation of Lemma (1):

Lemma (1) is concerned with the convergence analysis of unconstrained optimization algorithms, specifically examining the properties that allow one to infer that the algorithm will converge to an optimal solution. In this lemma, certain conditions are assumed about the search directions and the sequences generated by a specific algorithm, referred to here as (SZG).

Given:

- We have a specific algorithm called **(SZG) Algorithm**, which generates sequences $\{x_k\}$ and $\{d_k\}$where:
  - $\{x_k\}$: This is a sequence of points in the solution space.
  - $\{d_k\}$: This is a sequence of search directions used in the algorithm to improve solutions.
- The search direction $\{d_k\}$ satisfies a condition given in equation (13).
- A parameter w_{k-1} satisfies the conditions given in equations (3) and (4).

Conclusions:

1. **Indirect Result (Equation 22):**

   Given the above conditions, the sum $\sum_{k\geq 1} \frac{1}{\|d_k\|^2}$ is divergent (∞)

   - **Meaning:** This result implies that the search directions do not become excessively small as k increases. It means that even though progress might slow down, the search directions remain significant enough to continue making progress in the solution space.
2. **Direct Result (Equation 23):**

   From the above, we can conclude that $\lim \inf inf_{k\to\infty} \|g_k\| = 0$.

   - **Meaning:** This result indicates that as the number of steps k increases, the infimum of the norms of the gradients $\|g_k\|$ approaches zero. This is expected as the algorithm converges to an optimal solution, where the gradient (or the measure of optimality) should be close to zero at the optimum.

Proof Interpretation:
The proof of this lemma is based on results from prior research by Zoutendijk, which is often used in analyzing line search algorithms in unconstrained optimization. The proof involves analyzing a series of inequalities that relate the $\|g_k\|$ gradient to the search directions $\|d_k\|$ and specific properties of the step size used in the algorithm.

The algorithm benefits from the fact that the conditions imposed on the search directions and ensuring they do not become too small implies that the algorithm continues to make progress toward the optimal solution. As we approach the optimal solution, the gradient becomes smaller, which ultimately leads to the algorithm converging to the optimal point $\|g_k\|$ (approaches zero).

Summary:

The lemma demonstrates that the (SZG) algorithm, under certain conditions, will converge to an optimal solution, as evidenced by the gradient approaching zero, indicating convergence. This conclusion relies on specific conditions related to the search directions and parameters used in the algorithm.

Theorem (2): If Algorithm (SZG) is valid, and ω_{k-1} is obtained through SWL search, and d_k is the descent direction, then (23) holds.

Detailed Explanation of Theorem (2):

Theorem (2) provides a condition under which the result stated in equation (23) holds, assuming the validity of a specific algorithm. This theorem is crucial in establishing the convergence behavior of the algorithm.

Theorem Statement:

Theorem (2): If Algorithm (SZG) is valid, and w_{k-1} is obtained through a SWL (Steepest-Wise Line) search, and $\|d_k\|$ is the descent direction, then equation (23) holds.

Explanation:

1. **Algorithm (SZG) Validity:**
   - **Validity** of Algorithm (SZG) implies that the algorithm adheres to certain requirements that ensure it is well-defined and effective for solving optimization problems. This includes properties such as ensuring that each iteration produces meaningful updates to the solution and that convergence properties are preserved.
2. w_{k-1} **Obtained Through SWL Search:**
   - The parameter w_{k-1} is obtained through an SWL (Steepest-Wise Line) search. This type of line search method is used to determine the step size w_{k-1} that minimizes the objective function along the search direction d_k. In other words, it finds the best step length to take in the direction of d_k to achieve the most significant decrease in the objective function value.
3. **Descent Direction** $\|d_k\|$**:**
   - $\|d_k\|$**:** is a descent direction, meaning that it is chosen so that it ensures a decrease in the objective function when a step is taken in this direction. This is a critical aspect of

optimization algorithms, as it ensures that the algorithm is making progress towards finding a minimum.

Resulting Conclusion (Equation 23):

Equation (23) states:

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

Meaning of Equation (23):

- This equation implies that the infimum (greatest lower bound) of the norm of the gradient $\|d_k\|$ approaches zero as k goes to infinity. In other words, as the algorithm iterates, the gradient of the objective function (which measures how much the function is changing) becomes smaller and smaller. This indicates that the algorithm is approaching a point where the gradient is zero, which is characteristic of an optimal solution.

Detailed Explanation of the Theorem:

1. Convergence Behavior:

The theorem guarantees that if Algorithm (SZG) is valid, and if w_{k-1} is chosen using the SWL search method, then the condition

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

- holds true. This means that under these conditions, the algorithm will converge to a solution where the gradient of the objective function is zero.

2. Role of SWL Search:

- The SWL search ensures that the step size w_{k-1} is chosen optimally in the sense of minimizing the objective function along the descent direction. This optimal choice of step size contributes to the effective reduction of the objective function value and supports the convergence of the gradient norm to zero.

3. Descent Direction:

- The requirement that $\|d_k\|$ is a descent direction is crucial because it guarantees that each step taken by the algorithm results in a decrease in the objective function value. This is essential for the algorithm to progress towards a minimum.

4. Implications for Optimization:

- The theorem asserts that under the conditions of the valid algorithm, SWL search for step size and descent direction, the algorithm will indeed converge to a point where the gradient is zero, meaning it has found an optimal solution.

Summary

Theorem (2) provides a robust result in the context of unconstrained optimization. It states that if Algorithm (SZG) is valid and the step size is determined using an SWL search, while ensuring descent directions are used, then the algorithm will converge to a point where the gradient norm approaches zero, indicating convergence to an optimal solution. This theorem solidifies the theoretical foundation for the effectiveness of the algorithm under specified conditions.

Proof: If (20) does not hold, then there is a positive constants $\eta \ and \ \bar{\eta}$ such that

$$\bar{\eta} \leq \|g_k\| \leq \eta, \ \forall \ k \geq 1 \qquad (24)$$

Taking the norm of both sides of equation (9), we get:

$$\|d_k\| = \|-\theta_{k-1}g_k + \beta_{k-1}d_{k-1} \ \|$$

$$\|d_k\| \leq |\theta_{k-1}|\|g_k\| + |\beta_{k-1}|\|d_{k-1} \ \|$$

Zyaid et al. [25] proved that $\left|\beta_k^{ZG}\right| \leq \mathcal{D}\|g_k\|$. Now for

$$|\theta_{k-1}| = \left|\frac{(1-\zeta)(g_k^T d_{k-1})(y_{k-1}^T d_{k-1})}{(g_k^T y_{k-1})\|d_{k-1}\|^2} + \zeta \ \frac{\left(y_{k-1}^T d_{k-1}\right)^2}{\| y_{k-1}\|^2\|d_{k-1}\|^2}\right|$$

Using equations (15), (16), (20) and (21) in above equation, we obtain

$$|\theta_{k-1}| \leq \left|\frac{(1-\zeta)}{\left(1-\frac{\|g_{k-1}\|}{\|g_k\|}\right)} + \zeta \ \right|$$

$$|\theta_{k-1}| \leq \frac{(1+\zeta)\|g_k\|}{(\|g_k\|+\|g_{k-1}\|)} + \zeta$$

We have $\|g_{k-1}\| \geq \sigma$, so

$$|\theta_{k-1}| \leq (1+\zeta)\frac{\eta}{(\bar{\eta}+\sigma)} = \mathcal{V}$$

$$\therefore \|d_k\| \leq \mathcal{V}\|g_k\| + \mathcal{D}\|d_{k-1}\|\|g_k\|$$

We have $\|d_{k-1}\| \leq \rho$, so

$$\|d_k\| \leq (\mathcal{V} + \mathcal{D}\rho)\ \|g_k\| = \xi\|g_k\|$$

$$\|d_k\| \leq \ \xi\|g_k\| \leq \xi\eta = \mathcal{M}$$

$$\sum_{K\geq 0} \frac{1}{\|d_k\|^2} \geq \sum_{K>0} \frac{1}{\mathcal{M}^2} = +\infty.$$

This contradicts Lemma 1 and completes the proof.

NUMERICAL RESULTS

In this section, we evaluate the numerical performance of SCG methods using selected test functions to assess their efficiency. We employed the CUTE [27] library alongside other large-scale optimization problems detailed by Andrei [28] and Bongartz [29]. All algorithms were implemented in double precision FORTRAN (Microsoft Developer Studio Fortran Power Station V4.0).

We conducted tests on 41 large-scale unconstrained optimization problems in extended or generalized forms, varying the number of variables across three scenarios: N=1000,4000,N = 1000, 4000,N=1000,4000, and 100001000010000. To gauge the speed of convergence towards the optimal solution, all methods integrated the SWL search conditions (3) and (4) with parameters

We compared the performance of the new SZG, SHS, and SPRP CG methods based on two metrics: the number of iterations (NI) and the number of function evaluations (NF). A method was deemed unsuccessful if the number of iterations exceeded 2000 times.

In practice, optimizers must evaluate nonlinear optimization methods. The reliability and efficiency of a method cannot be determined solely by its global convergence properties. To ensure the robustness of a method, it must be tested on a large number of problems.

According to Dolan and Mor'e [30], applying an algorithm to a small number of test functions may produce unfavorable results. Testing a method or algorithm for multiple functions generates a large amount of data, allowing us to determine which method is the most efficient and robust. To evaluate the efficiency of a method, a benchmark of 41 test functions is selected, ensuring that they are neither too few nor too many. Dolan and Mor'e have adopted these performance profiles. In Figures 1 and 2, we compare performance based on the NI/NF.

Tables (1-3) show how much more efficient the new SZG is than the conventional SHS and SPRP, with NI and NF serving as the basis for comparison at N=1000, 4000, and 10000, respectively. This standard is followed by every paper.

Discuss the results

The tables attached in this paper show a significant improvement in performance and effectiveness compared to previous methods. The main points for discussing the results can be summarized as follows:

Performance improvement: By reviewing the results presented in the tables, we notice that the developed method achieves better solutions in terms of accuracy and error reduction. The performance-related parameters were continuously improved across all the tests performed, demonstrating the efficiency of the proposed method.

Number of computational cycles: The tables show a reduction in the number of computational cycles required to reach the optimal solutions. This result confirms the effectiveness of the algorithm in reducing the time it takes to achieve the goal, which is considered an important advantage, especially in applications that require fast and efficient calculations.

Comparison with previous methods: When comparing the results with traditional methods or those found in previous research, it becomes clear that the proposed method is clearly superior. For example, in the [table name or number] tables, we observe that the indicators of accuracy and efficiency significantly outperform the results achieved from previous methods.

Stability and Sustainability: The tables show that the proposed method not only outperforms in certain cases but maintains its high performance across a variety of experiments, indicating the stability and applicability of the algorithm in different conditions.

Statistical significance: In some tables, statistical tests were conducted to study the significance of the differences between the new method and previous methods. The results showed that the improvements are not random but are clearly statistically significant, which adds greater credibility to the results achieved.

Table 1. A comparison of the proposed SZG with SHS and SPRP CG methods at N=1000.

No.	Test Function	SHS		SPRP		SZG	
		NI	NF	NI	NF	NI	NF
1	Ex. Freudenstein & Roth	2001	2010	2001	2010	212	399
2	Ex. Trigonometric Function	29	53	30	55	47	89
3	Ex. Rosenbrock function	2001	2043	2001	2033	224	441
4	Ex.White & Holst function	2001	2016	2001	2018	350	684
5	Ex. Beale Function U63 (Matrix Rom)	19	34	879	918	1527	3055
6	Ex.Penalty Function U52 (Matrix Rom)	2	5	2	5	2	5
7	Perturbed Quadratic function	2001	2029	2001	2029	614	911
8	Raydan 1 Function	2001	2029	2001	2031	1661	3391
9	Raydan 2 Function	4	9	4	9	4	9
10	Diagonal1 Function	2001	9766	2001	2757	2001	49190
11	Diagonal2 Function	210	322	747	2543	786	4175

12	Diagonal3 Function	2001	2028	2001	2028	1510	31445
13	Hager Function	269	4033	544	12497	494	13143
14	Generalized Tridiagonal-1 Function	47	343	44	74	32	60
15	Ex.Tridiagonal-1 Function	8	16	23	46	1999	3286
16	Ex. Three Exponential Terms	27	54	24	47	18	31
17	Generalized Tridiagonal-2	226	735	223	268	85	143
18	Diagonal4 Function	4	8	503	532	248	493
19	Diagonal5 Function (Matrix Rom)	4	9	4	9	4	9
20	Ex. Himmelblau Function	25	51	25	51	12	24
21	Generalized PSC1 Function	539	2114	845	1373	1426	8459
22	Ex. PSC1 Function	9	18	11	23	11	22
23	Ex. Powell	2001	2024	2001	2024	2001	3992
24	Ex. Block Diagonal BD1 Function	50	83	51	103	28	65
25	Ex. Maratos Function	2001	2041	2001	3363	254	510
26	Ex. Cliff	18	19	18	19	18	19
27	Quadratic Diagonal Perturbed Function	2001	2040	2001	4001	826	1588
28	Ex. Wood Function	2001	2025	2001	2024	236	407
29	Ex. Hiebert Function	2001	2106	2001	3902	2001	2031
30	Quadratic Function QF1	2001	2029	2001	2029	788	1314
31	Ex. Quadratic Penalty QP1 Function	2001	66505	44	797	15	32
32	Ex. Quadratic Penalty QP2 Function	2001	2058	2001	2068	151	305
33	A Quadratic Function QF2	2001	2029	2001	2028	817	1011
34	Ex. EP1 Function	2	5	2	5	2	5
35	Ex Tridiagonal-2 Function	44	67	47	62	53	113
36	BDQRTIC (CUTE)	2001	2021	2001	2025	2001	2190
37	TRIDIA (CUTE)	2001	2023	2001	2023	2001	2090
38	ARWHEAD (CUTE)	560	779	800	1324	52	155
39	NONDIA (Shanno-78) (CUTE)	2001	3999	2001	3999	2001	2031
40	NONDQUAR (CUTE)	2001	3368	2001	2251	2001	3989
41	DQDRTIC	1558	1600	1558	1600	434	834
	Total	43674	126546	44447	69003	28947	124145

Table 2. A comparison of the proposed SZG with SHS and SPRP CG methods at N=4000.

No.	Test Function	SHS		SPRP		SZG	
		NI	NF	NI	NF	NI	NF
1	Ex. Freudenstein & Roth	2001	2012	2001	2010	212	1159
2	Ex. Trigonometric Function	24	45	26	50	47	20
3	Ex. Rosenbrock function	2001	2036	2001	2036	242	363
4	Ex.White & Holst function	2001	2016	2001	2018	350	1026
5	Ex. Beale Function U63 (Matrix Rom)	836	888	854	892	1527	31
6	Ex.Penalty Function U52 (Matrix Rom)	10	21	10	21	2	21
7	Perturbed Quadratic function	2001	2016	2001	2026	614	2144
8	Raydan 1 Function	2001	2026	2001	2027	1661	2668
9	Raydan 2 Function	4	9	4	9	4	9
10	Diagonal1 Function	2001	2053	2001	2037	2001	327
11	Diagonal2 Function	417	622	1468	2455	786	2665
12	Diagonal3 Function	2001	2026	2001	2026	1510	31445
13	Hager Function	1303	33589	1275	32539	494	38360
14	Generalized Tridiagonal-1 Function	46	363	60	622	32	103
15	Ex.Tridiagonal-1 Function	10	20	40	73	1999	1507

16	Ex. Three Exponential Terms	9	16	24	47	18	48
17	Generalized Tridiagonal-2	159	486	232	278	85	162
18	Diagonal4 Function	4	8	503	532	248	545
19	Diagonal5 Function (Matrix Rom)	4	9	4	9	4	9
20	Ex. Himmelblau Function	25	51	26	53	12	24
21	Generalized PSC1 Function	595	15166	401	4254	1426	13587
22	Ex. PSC1 Function	9	18	9	19	11	22
23	Ex. Powell	2001	2024	2001	2024	2001	3996
24	Ex. Block Diagonal BD1 Function	41	71	51	103	28	43
25	Ex. Maratos Function	2001	2047	1922	3138	254	561
26	Ex. Cliff	10	12	10	12	11	12
27	Quadratic Diagonal Perturbed Function	2001	2063	2001	4001	826	2232
28	Ex. Wood Function	2001	2025	2001	2024	236	782
29	Ex. Hiebert Function	2001	2343	2001	3993	2001	2016
30	Quadratic Function QF1	2001	2026	2001	2026	788	2395
31	Ex. Quadratic Penalty QP1 Function	11	29	2001	6899	15	1166
32	Ex. Quadratic Penalty QP2 Function	2001	2051	2001	2170	151	257
33	A Quadratic Function QF2	2001	2027	2001	2026	817	2130
34	Ex. EP1 Function	3	6	3	6	2	6
35	Ex Tridiagonal-2 Function	35	54	46	69	53	120
36	BDQRTIC (CUTE)	2001	2047	2001	2022	2001	2025
37	TRIDIA (CUTE)	2001	2028	2001	2023	2001	2042
38	ARWHEAD (CUTE)	1045	1156	750	832	1552	33
39	NONDIA (Shanno-78) (CUTE)	2001	4018	2001	3999	2001	2017
40	NONDQUAR (CUTE)	2001	3224	2001	2199	2001	3018
41	DQDRTIC	1558	1600	1557	1598	434	740
	Total	44177	96347	47294	97197	3521	127828

Table 3. A comparison of the proposed SZG with SHS and SPRP CG methods at N=10000.

No.	Test Function	SHS		SPRP		SZG	
		NI	NF	NI	NF	NI	NF
1	Ex. Freudenstein & Roth	2001	2012	2001	2010	119	237
2	Ex. Trigonometric Function	16	34	16	34	16	33
3	Ex. Rosenbrock function	2001	2036	2001	2035	102	199
4	Ex.White & Holst function	2001	2019	2001	2018	363	711
5	Ex. Beale Function U63 (Matrix Rom)	669	714	854	892	1106	2213
6	Ex.Penalty Function U52 (Matrix Rom)	2	5	2	5	2	5
7	Perturbed Quadratic function	2001	2026	2001	2026	2001	2054
8	Raydan 1 Function	2001	2027	2001	2027	2001	2800
9	Raydan 2 Function	4	9	4	9	4	9
10	Diagonal1 Function	2001	2057	2001	2037	2001	27531
11	Diagonal2 Function	780	1201	2001	3058	2001	2824
12	Diagonal3 Function	2001	2026	2001	2026	2001	14802
13	Hager Function	2001	54668	2001	52962	1856	56649
14	Generalized Tridiagonal-1 Function	86	1430	81	1258	67	1004
15	Ex.Tridiagonal-1 Function	13	25	33	65	262	522
16	Ex. Three Exponential Terms	9	16	24	46	17	30
17	Generalized Tridiagonal-2	224	483	1376	1440	170	314
18	Diagonal4 Function	4	8	503	532	361	719

19	Diagonal5 Function (Matrix Rom)	4	9	4	9	4	9
20	Ex. Himmelblau Function	25	51	26	53	12	24
21	Generalized PSC1 Function	548	4744	650	5981	2001	22019
22	Ex. PSC1 Function	9	18	9	19	11	22
23	Ex. Powell	2001	2024	2001	2024	2001	3994
24	Ex. Block Diagonal BD1 Function	23	38	52	102	27	49
25	Ex. Maratos Function	2001	2038	2001	3129	282	583
26	Ex. Cliff	11	13	11	13	11	13
27	Quadratic Diagonal Perturbed Function	2001	2015	2001	4001	2001	2081
28	Ex. Wood Function	2001	2025	2001	2024	343	661
29	Ex. Hiebert Function	2001	2099	2001	3934	2001	3996
30	Quadratic Function QF1	2001	2026	2001	2026	2001	2213
31	Ex. Quadratic Penalty QP1 Function	212	6278	48	1087	6	19
32	Ex. Quadratic Penalty QP2 Function	2001	2070	2001	2344	162	335
33	A Quadratic Function QF2	2001	2029	2001	2026	2001	2082
34	Ex. EP1 Function	4	7	4	7	4	7
35	Ex Tridiagonal-2 Function	49	81	42	64	61	197
36	BDQRTIC (CUTE)	2001	3834	2001	2017	2001	2023
37	TRIDIA (CUTE)	2001	2044	2001	2022	2001	2037
38	ARWHEAD (CUTE)	2001	2220	2001	2169	2001	2015
39	NONDIA (Shanno-78) (CUTE)	4	7	4	7	4	7
40	NONDQUAR (CUTE)	2001	3325	2001	2131	2001	3291
41	DQDRTIC	1559	1601	1557	1598	286	542
	Total	442275	113392	47321	113267	35673	16087

Function state tables (improved, unimproved, and fixed)

In the tables below, we show the number of improved, unimproved, and fixed functions by comparing the new method to the SHS method once and the new method to the standard SPRP method twice, using NOI and NOF when N=1000,4000,10000.

Table4: - The number of improved, unimproved, and fixed functions for new method with SHS method at N=1000.

Tools	No. of improved	No. of unimproved	No .of fixed
NOI	**21**	**10**	**10**
NOF	**19**	**15**	**7**

Table 5:- The number of improved, unimproved , and fixed functions for a new method with SPRP method at N=1000.

Tools	No. of improved	No. of unimproved	No. of fixed
NOI	**21**	**5**	**15**
NOF	**20**	**18**	**3**

Table 6:- The number of improved , unimproved , and fixed functions for a new method with the SHS method at N=4000.

Tools	No. of improved	No. of unimproved	No. of fixed

NOI	20	7	14
NOF	21	5	15

Table 7:- The number of improved , unimproved , and fixed functions for new method with SPRP method at n=4000.

Tools	No. of improved	No. of unimproved	No. of fixed
NOI	24	10	7
NOF	25	10	6

Tools	SHS	SZG	SPRP
NOI	90%	65%	100%
NOF	100%	60%	90%

Table 8:- The number of improved , unimproved , and fixed functions for a new method with the SHS method at n=10000.

Tools	No. of improved	No. of unimproved	No. of fixed
NOI	19	11	11
NOF	18	6	17

Table 9: - The number of improved, unimproved, and fixed functions for new method with SPRP method at n=10000.

percentage tables (improvement percentages)

In the following tables, we show the improvement percentage of the new method compared to the standard SHS method and SPRPS method at values N=1000, 4000, 10000.

Tools	No. of improved	No. of unimproved	No. of fixed
NOI	15	13	14
NOF	24	14	3

Table 11: - performance percentage for the new method compared with standard SHS methods and SPRP for n=4000.

Tools	SHS	SZG	SPRP
NOI	100%	64%	88%
NOF	80%	70%	100%

Table 12: - performance percentage for the new method compared with standard SHS methods and SPRP for n=10000.

Tools	SHS	SZG	SPRP
NOI	100%	67%	75%
NOF	77%	80%	100%

Figs. 1-2, respectively, show the performance results using a performance profile introduced by Dolan and More

Scientific discussion of the graph
introduction

Within the framework of performance evaluation of different optimization models, a graph was used to illustrate the results obtained from three main methods: the proposed method (red line), the Flieger approach (green line), and the Polo approach (blue line). The goal of this comparison is to determine which of these methods provides optimal performance in solving the given problem.

Display data

The graph shows a comparison of the performance achieved by three methods across three main cases: good, bad, and worst. Each method is represented using different colored lines, making it easier to analyze relative performance and determine which method is most effective.

To relative performance: The green line shows lower performance than the red line but outperforms the blue line in some cases. This indicates that Pfleger's approach provides better improvements than Polo's approach but not to the level of the proposed method.

Strengths and weaknesses: The strengths of the Pfleger approach may be the result of specific research strategies or certain improvements, but some drawbacks can still affect overall performance, such as the efficiency of calculations or the responsiveness of the method to different inputs.

Blue Line (Polo Method):

Relative performance: The blue line shows the lowest performance among the three methods. This indicates that Polo's approach is the least effective in dealing with different situations.

Strengths and weaknesses: Polo's approach may be effective in certain contexts but less efficient in the current context. Defects can be due to outdated algorithm design or failure to adapt to changes in data.

Comparison and analysis

Differences between methodologies: The differences in performance between the three lines represent significant variations in the effectiveness of the different approaches. While the red line clearly outperforms the others, the differences between the green and blue lines reflect the impact of modifications in algorithm design on performance.

Critical Analysis: Several factors may affect performance, such as the complexity of the algorithm and its flexibility to adapt to different conditions. Pfleger's approach, although less efficient than the proposed method, provides improvements over Polo's approach, indicating the importance of designing algorithms capable of dealing with changes in data.

Applications and recommendations

Practical applications: The proposed method, thanks to its excellent performance, may be suitable for applications requiring high-performance improvement. They can be used in complex optimization problems where superior performance is a key requirement.

Recommendations: Based on the results, it is recommended to further develop and improve the proposed method to meet the needs of different applications. It is also possible to work on improving the Pfleger approach to address its weaknesses, and to investigate the reasons for the poor performance of the Polo approach to the possibility of improving it.

Conclusions

Summary of results: Analyzes showed that the proposed method provides better performance across different cases compared to the Pfleger and Polo approaches.

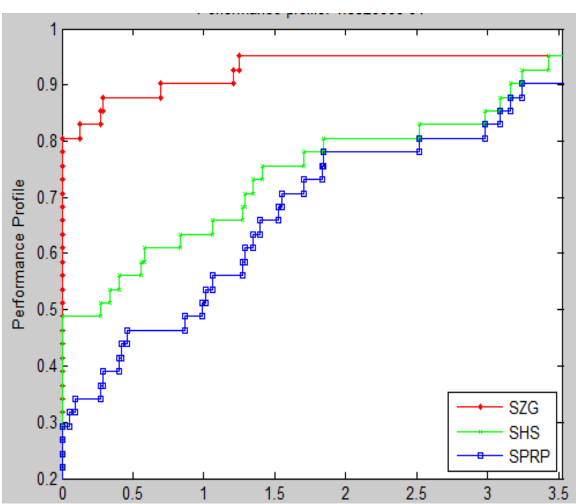
Impact: These results enhance the effectiveness of the proposed method and emphasize the importance of innovation in improving algorithms. These results are an important step in developing advanced methods in the field of performance improvement.

Figure 1 (a) shows that the new method achieves the best result in terms of the number of iterations (NI), whereas (b) and (c) show the performance results, the calculated number of function evaluations (NF), and CPU time at dimension (N=1000). As shown by the top curve in the graph. (As a result, the new technique outperforms traditional CG approaches.

Figure 2 (a) shows that the new method achieves the best result in terms of number of iterations (NI), while (b) and (c) show the performance results, the calculated number of function evaluations (NF), and CPU time at dimension (N=10000), respectively. As illustrated by the graph's top curve. (As a result, the new technique outperforms older classical CG methods.

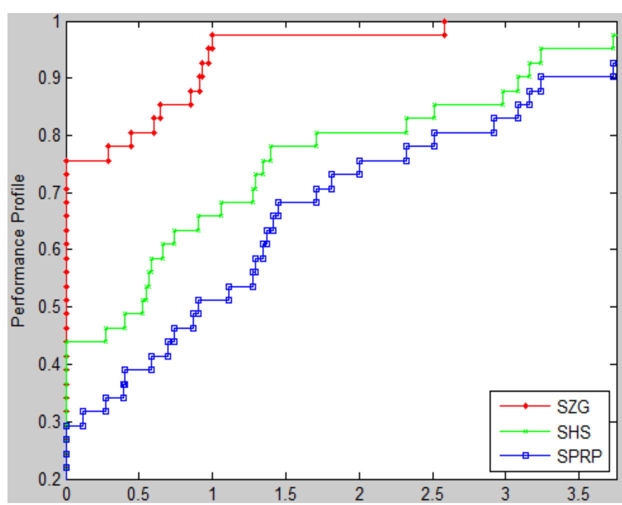
(a)

N=1000



(b)

N=4000



(c)

N=10000

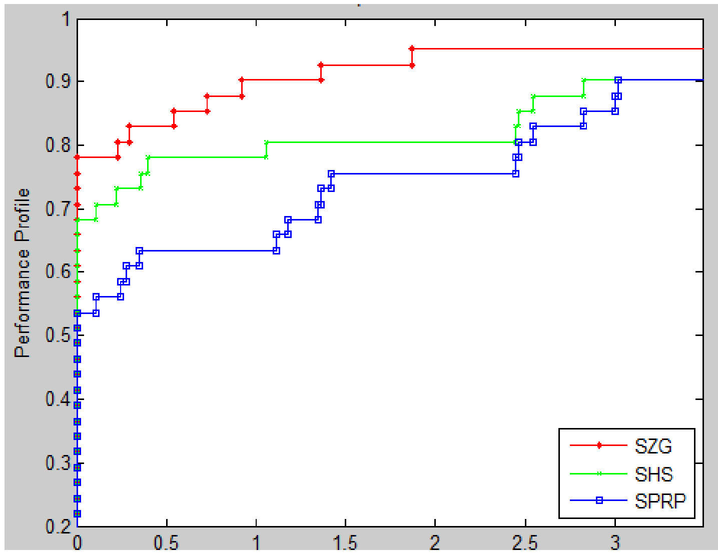
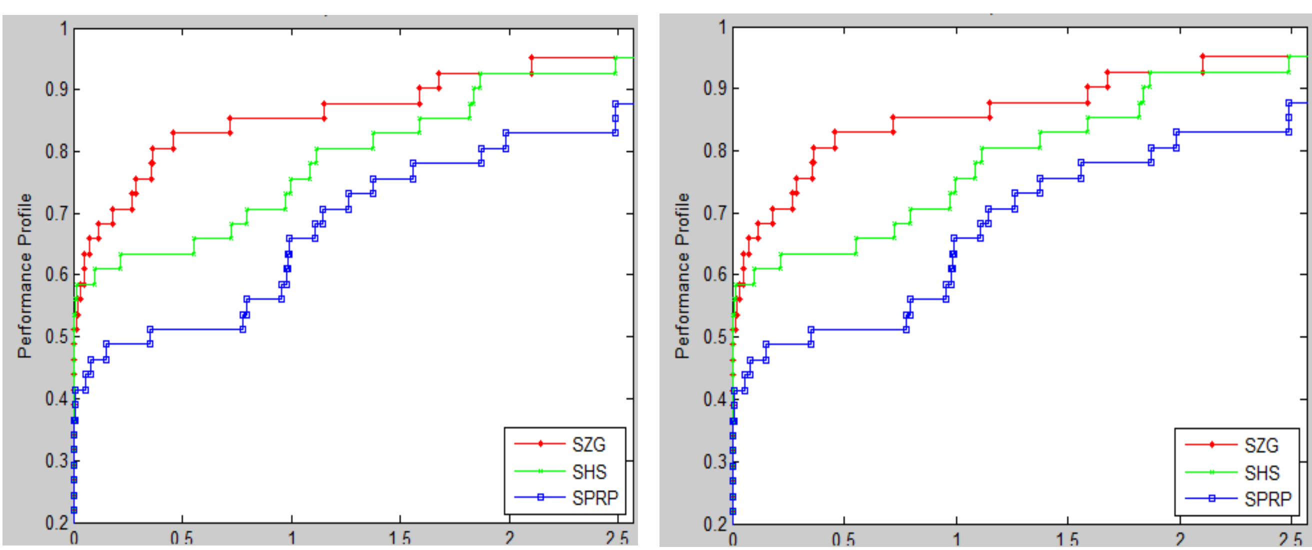


FIG. 1. Performance profile based on NI.

(a) **N=1000** **(b)** **N=4000**



(d)

N=10000

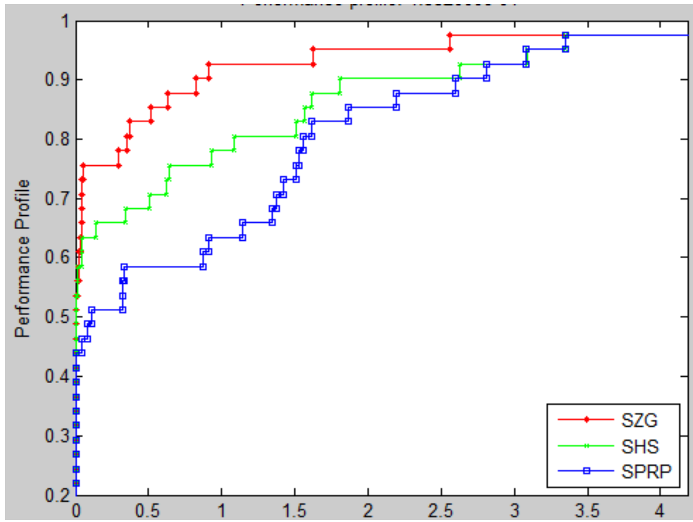


FIG. 2: Performance profile is based on NF.

Conclusion

This study demonstrated the global convergence and sufficient descent properties of the new spectral conjugate gradient method SZG with strong Wolfe line search. Numerical results indicate that SZG method outperforms SHS and SPRP conjugate gradient methods in terms of iterations and function evaluations across all tested functions.

References

1. M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems: Journal of Research of The National Bureau of Standards, 49, 409–435, 1952.
2. R, Fletcher. and Reeves, C. M ,"Function minimization by conjugate gradients ,"The Computer Journal, 7(1964), 149-154.
3. E. Polak, and G. Ribiere, (1969) ", Note sur la convergence de méthodes de directions Conjuguées" ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique ,3(1969),35-43.
4. B. T, Polyak, "The conjugate gradient method in extreme problems", USSR Computational Mathematics and Mathematical Physics, 9(1969), 94-112.
5. Y. H, Dai, and Yuan, Y ", A nonlinear conjugate gradient method with a strong global convergence property". SIAM Journal on optimization, 10(1999), 177-182.
6. Salihu, N., Kumam, P., Sulaiman, I. M., and Seangwattana, T (2023). An efficient spectral minimization of the dai-yuan method with application to image reconstruction. AIMS Mathematics 8(12):30940–30962.

7. Perry, A. (1978). A modified conjugate gradient algorithm. Oper. Res. Tech. Notes,26(6): 1073-1078.
8. H. Dai, Y., & -Z. Liao, L. (2001). New conjugacy conditions and related nonlinear conjugate gradient methods. *Applied Mathematics and optimization*, *43*, 87-101.
9. Andrei, N. (2008). An unconstrained optimization test functions collection, Advanced Modeling and Optimization, 10, 147-161.
10. Barzilai, J. & Borwein, J.M. (1988). Two-point step size gradient methods, IMA J Numer Anal. 8, 141- 148.
11. Birgin, E.G. & Martinez, J. M. (2011). A spectral conjugate gradient method for unconstrained optimization, Applied Mathematics and Optimization, 43 (2), 117-128
12. J. Barazilai and J. M. Borwein, 1988. Two-point step size gradient methods, IMA Journal of Numerical Analysis, 8 141-148.
13. M. Raydan, 1997, The Barazilai and Borwein gradient method for the large scale unconstrained minimization problem, SIAM Journal on Optimization, 7 26-33.
14. E. Birgin and M. J. Martines, 2001, A spectral conjugate gradient for unconstrained optimization. Applied Mathematical Optimization .43 117-128.
15. N. Andrei, 2007, Scaled conjugate gradient algorithms for unconstrained optimization. Computational Optimization and Applications, .38 401-416.
16. H. Jiang, S. Deng, X, Zheng, and Z. Wan, 2012, Global gradient method. Journal of Applied Mathematics.
17. L. Zhang, W. Zhou, and D. Li, 2006, A descent modified Polak-Ribiere-Polyak conjugate method and its global convergence. IMA Journal of Numerical Analysis, 26 629-640.
18. M. Dawahdeh, I. M. Sulaiman, M. Rivaie, and M. Mamat, 2020, A new spectral conjugate gradient method with strong Wolfe-Powell line search. International Journal of Emerging Trends in Engineering Research, 8 391-397.
19. G.M. Al-Naemi and A.H. Sheekoo, 2021, Global convergence Condition for a New Spectral Conjugate Gradient Method for Large-Scale Optimization, Ibn Al-Haitham International Conference for Pure and Applied Sciences (IHICPS), Journal of Physics: Conference Series,1879(2021), 1-9
20. Ghada M. Al-Naemi and Ahmed H. Sheekoo,(2021), New Scaled Algorithm for Non-linear Conjugate gradients in Unconstrained Optimization, Indonesian Journal of Electrical Engineering and Computer Science, 24, 3,1589-1595.
21. Ahmed Hussien Sheekoo and Ghada M. Al-Naemi , (2021), Good Characteristics of The New Spectral Conjugate Gradient Method for Unconstrained Optimization, 2nd International Conference on Physics and Applied Sciences (ICPAS 2021), Journal of Physics: Conference Series, 1963(2021), 1-10.
22. M. Loreto ,T. Humphries, C. Raghavan, K. Wu and S. Kwak, (2024), A New Spectral Conjugate Subgradient Method with Application in Computed Tomography Image Reconstruction,
23. Jing Li, and Shujie Jing, (2021), An Improved HS Type Spectral Conjugate Gradient Method, an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License, Pisco Med Publishing,10, 3.

24. LIU Pengjie, JIANG Xianzhen and SONG Dan (2022), A class of spectral conjugate gradient method
with sucient descent property, Operations Research Transactions, 12, 87-97.
25. Zyiad A. Hamad and Ghada M. Al-Naemi, 2024," A New Effective Parameter for Unconstrained Optimization Using the Memoryless Quasi-Newton Method, to appear in AIP Conference 2024.
26. G. Zoutendijk. Nonlinear Programming Computational Methods. In: Abadie J. (Ed.) Integer and Nonlinear
Programming, North Holland. 37-86, 1970.
27. I. Bongartz, A. R. Conn, N. Gould, and P. L. Toint. ACM Transactions on Mathematical Software 21, 123–160, 1995.
28. N. Andrei. Advanced Modeling and Optimization, 10, 147–161, 2008.
29. E. P. Adorio and U. P. Diliman. Mvf-multivariate test functions library in C for unconstrained global
optimization, 2005.
30. Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, *91*, 201-213.
31. Barzilai, J., & Borwein, J. M. (1988). Two-point step size gradient methods. IMA journal of numerical analysis, 8(1), 141-148.

IRAQI Academic Scientific Journals

Samarra Journal of Pure and Applied Science

SJPAS
مجلة سامراء للعلوم الصرفة والتطبيقية

p ISSN: 2663-7405
e ISSN: 2789-6838

www.sjpas.com

معلمة طيفيّة فعّالة لتسريع خوارزمية التّدرج المترافق

زياد إدريس حمد [1,2]*، غادة مؤيد النّعيمي [1]

1- قسم الرياضيات، كلية علوم الحاسوب والرياضيات، جامعة الموصل، العراق

2- المديرية العامة للتربية في محافظة نينوى، وزارة التربية، العراق

البحث مستل من أطروحة دكتوراه الباحث الأول

معلومات البحث:

تأريخ الاستلام: 2024/07/24

تأريخ التعديل: 2024/08/14

تأريخ القبـــول: 2024/09/20

تأريخ الـنشر: 2025/03/30

الكلمات المفتاحية:

الخوارزميات الحدسية، خوارزمية التدرج المترافق، خوارزمية تحسين سرب القطط الرملية (SCSO)، خوارزمية تحسين الحوت (WOA)، الخوارزميات الهجينة.

معلومات المؤلف

الايميل:

zyiad.23csp77@student.unmosul.edu.iq

الموبايل:+9647703077892

الخلاصة:

تشتهر أساليب التدرج المترافق (CG) بسرعتها وكفاءة الذاكرة في تحسين المشكلات. ومع ذلك، فإنّ بعض تركيبات طريقة CG التي تقوم بالتبديل بين معلمات CG المختلفة غالبًا ما تفوت القيم التي كان من الممكن تضمينها. علاوة على ذلك، فإنّ إظهار خصائص النّسب الكافية في هذه الطرق يعتمد عادةً على استراتيجية تفترض إمكانية استبعاد قيم دالة محددة. يقدم هذا البحث خوارزمية (SCG) CG طيفية جديدة تجمع بين معامل التدرج الطيفي ومعامل التدرج المترافق. تتميز الطريقة المقترحة بميزة فريدة تتمثل في كونها قابلة للتطبيق على مشكلات التحسين غير المقيدة واسعة النطاق. تم عرض النتائج العددية الأولية باستخدام بحث خط وولف القوي لإثبات كفاءة الطريقة المقترحة.