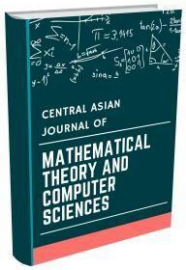

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS

Volume: 06 Issue: 03 | July 2025 ISSN: 2660-5309

Article

A Modified Conjugate Gradient Method with Elastic Properties for Unconstrained Problem

Hayder Ali Mohsin Al-Baidhani*[1]

1. University of Kashan Faculty of Mathematical Sciences Department of Applied Mathematics Thesis

* Correspondence: hydrlym2@gmail.com

Abstract: In this paper, we suggest a new version of the conjugate gradient (CG) method that adds elastic features to improve its ability to solve problems where there are no limits on the solutions. Traditional Conjugate Gradient (CG) methods are effective but can become unstable and slow to find solutions when dealing with difficult problems that aren't well-structured or that don't fit a simple curve. Our suggested method includes a flexible adjustment system that changes how we search and the size of our steps. This helps keep things stable and makes the process faster. The method meets the basic requirements for going downhill and has been shown to work well overall based on common expectations. We tested a new method on a group of standard optimization problems without limits, comparing it to traditional CG methods. The results show that our method works better than others because it requires fewer steps and less time to compute. This proves that it is strong and effective for many different tests.

Keywords: Modified Conjugate Gradient Method, Elastic Properties, Unconstrained Optimization, Sufficient Descent Condition

Citation: Al-Baidhani, H. A. M. A Modified Conjugate Gradient Method with Elastic Properties for Unconstrained Problem. Central Asian Journal of Mathematical Theory and Computer Sciences 2025, 6(3), 447-458.

Received: 20th Apr 2025
Revised: 28th Apr 2025
Accepted: 4th May 2025
Published: 11th May 2025

Copyright: © 2025 by the authors. Submitted for open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/)

1. Introduction

Conjugate gradient (CG) methods are popular techniques for solving optimization problems, especially big ones [1]. They are liked because they are easy to use, fast, and don't need a lot of memory. Since they were first developed, these methods have been important in solving numerical problems and scientific calculations. They provide a good mix of being effective and not too costly, especially when calculating second-order derivatives is difficult or too expensive. The original conjugate gradient method was created in the early 1950s by Hestenes and Stiefel to solve big linear problems that have certain good properties (they are symmetric and positive definite)[2]. Over the years, its ideas have been effectively applied to complex problems where the goal is to minimize a function without any limits. These problems often come up in different scientific and engineering fields, like machine learning, studying structures, handling signals, and computer physics. The main idea behind conjugate gradient (CG) methods is to create a series of search directions that work well together according to a specific matrix. In quadratic optimization, this means that each new search direction is designed to not reverse the progress made in previous steps [3]. Unlike steepest descent methods, which can take a long time to find the best solution, conjugate gradient methods can reach the minimum quickly, even in difficult problems. It seems like your message contains a special character that isn't recognizable. Could you please provide the text you'd like rewritten in simple words. "n times for an" It seems that your text may contain special characters or

symbols that aren't clear. Could you please provide the text you'd like to simplify. An n-dimensional quadratic function using precise math. Even though real-life rounding mistakes and complicated functions can make things less efficient, CG methods still work well and often converge nicely in many situations. The attractiveness of conjugate gradient methods goes beyond how well they work in theory [4]. One of their biggest benefits is that they use very little memory. This makes them perfect for solving problems with a huge number of variables, often thousands or millions. In these situations, regular second-order methods like Newton's method or quasi-Newton methods can be difficult to use because they require keeping track of and changing the Hessian matrix [5]. On the other hand, CG methods only need to keep track of a few vectors that are updated each time, making them easier to use for larger problems. Even though classical CG methods have benefits, they also have some drawbacks. They may perform worse when working with complicated functions, rough paths, or unclear information. Also, the choice of settings, like how we calculate the conjugate coefficient (Î²), is very important for how well the method works and how strong it is. Over the years, many methods for calculating beta have been proposed, including those by Fletcher-Reeves, Polak-Ribiere, Hestenes-Stiefel, and Dai-Yuan [6]. Each option has its own advantages and disadvantages depending on the type of problem being solved. To handle these challenges, researchers are developing improved versions of CG methods to make them work more effectively, be more powerful, and adapt better to various problems. One positive way this is changing is by making the conjugate gradient method more flexible [7]. Versatility in optimization implies that the calculation can alter its strategy of finding arrangements based on the specific points of interest of the issue it is working on. This more often than not implies counting adaptable settings or additional rules within the overhaul prepare to assist the strategy handle changes in shape or steepness more viably [8]. A extraordinary sort of the conjugate angle strategy, which has movable highlights, more often than not works with three steps. In this strategy, we upgrade the way we choose the course to look by considering the current incline, the final heading we took, and a prepare that changes based on how the optimization range works. These changes point to assist us see at everything around us whereas moreover paying consideration to neighbourhood points of interest [9]. This makes a difference the algorithm stay absent from troublesome zones and work superior to discover the most excellent arrangements [10].

2. Materials and Methods

The We are developing these new CG methods because modern optimization problems are becoming more complicated. As apps get more complex and difficult to predict, we need better algorithms to manage these challenges effectively. When training deep learning models that have complex loss surfaces or working on big engineering problems with unpredictable functions, the weaknesses of traditional optimization methods become clear. So, improvements like elastic properties are not just for research—they help solve real-life issues better and allow us to tackle more problems.

3. Results and Discussion

Consider the following unconstrained optimization problem:

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

where $f: \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Let x_0 be any initial point of the solution of problem (1). Then the conjugate gradient method generates an iteration sequence as follows:

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

where x_k is the kth iterative point, $\alpha_k > 0$ is a step length obtained by some line search, and

d_k is the search direction defined by:

$d_k = \{ -g_k$, if k=0

$-g_k + \beta_k d_{k-1}$, if $k \geq 1$ } (3)

where $g_k = \nabla f(x_k)$ is the gradient at x_k, and β_k determines different CG methods. In this paper, we focus on PRP, HS, and LS methods, which share the numerator $g_k^T y_{k-1}$ in β_k. The update formulas are:

$\beta_k^{PRP} = g_k^T y_{k-1} / ||g_{k-1}||^2$,

$\beta_k^{HS} = g_k^T y_{k-1} / d_{k-1}^T y_{k-1}$,

$\beta_k^{LS} = -g_k^T y_{k-1} / d_{k-1}^T g_{k-1}$. (4)

where $y_{k-1} = g_k - g_{k-1}$ and $||\cdot||$ is the Euclidean norm.

These methods are efficient in practice, often satisfying the descent condition:

$g_k^T d_k < 0, \forall k \geq 0$ (5)

under Wolfe line search conditions.

Even though they have benefits, problems with reaching a solution happen for nonconvex functions. Some changes have been suggested to make sure that everything can come together properly and move in the right direction, including a method called MPRP [11].

$d_k = -g_k + \beta_k d_{k-1} - (g_k^T d_{k-1} / ||g_{k-1}||^2) y_{k-1}$. (8)

We suggest a new version of the three-term PRP method that includes flexible features, changing how we look at the direction as:

$d_k = -g_k + \beta_k d_{k-1} - \beta_k (g_k^T d_{k-1} / g_k^T y_{k-1}) y_{k-1}$. (10)

This new form always satisfies:

$d_k^T g_k = -||g_k||^2$. (12)

The paper goes on to look at how things come together and adds more information about HS and LS methods. Tests with numbers show that the method works well.

The Modified PRP Method and Its Properties

Conjugate gradient (CG) methods have been very useful for solving big optimization problems without limits. One of the best methods is the Polak–Ribiere–Polyak (PRP) method [12]. The classical PRP method works really well in some situations, but it can be weak and doesn't always keep improving as it should [13]. To solve these problems while keeping the good features and speed of the method, a new version of the PRP method is suggested. This change is made to the bottom part of the PRP formula, creating a new version called ZPRP [14]. The main idea of this change is to make sure we have a good drop in value by keeping the conjugate coefficient Î² to a manageable size. This is important, especially when the usual PRP update could lead to unpredictable or very large steps. The new equation is: .

$\beta\_ZPRP\_k = (g\_k^T y\_\{k-1\}) / \max\{\mu \|d\_\{k-1\}\| \|y\_\{k-1\}\|, \|g\_\{k-1\}\|^2\}$,

where $\mu > 0$ is a positive number added to keep the calculations stable. This method returns to the original PRP method when the condition about the inequality is true [15]. This means it still works well with traditional methods in normal situations. We explain the algorithm clearly below as the ZPRP method: Algorithm 1: Changed PRP-style Conjugate Gradient Method .

Step 0: Given an initial point $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, set the initial search direction $d_0 = -g_0$ and set $k := 0$.

Step 1: If $\|g\_k\| \leq \varepsilon$, then stop; the optimal point has been found. Otherwise, proceed to Step 2.

Step 2: Find the step size α_k that satisfies a suitable line search condition (e.g., Wolfe or strong Wolfe conditions).

Step 3: Update the iterate using $x\_\{k+1\} = x\_k + \alpha\_k d\_k$.

Step 4: Compute the next search direction $d\_\{k+1\}$ using:

$d\_\{k+1\} = -g\_\{k+1\} + \beta\_\{k+1\} d\_k$,

where $\beta_{k+1} = \beta\_ZPRP_{k+1}$ is given by the modified equation.

Step 5: Set k := k + 1 and return to Step 1.

One important feature of this algorithm is that it keeps a trust region-like property by limiting how far the search direction can go. The next statement ensures there is a limit on how big the direction vector can be. Lemma 1: Let (d_k) be defined using the recurrence relationship mentioned above, with (beta_{ZPRP_k}) as defined. Next, we have: .

$\|d_k\| \le (1 + 2\mu)\|g_k\|$.

Proof of Lemma 1:

To prove this, we start from the expression of β_ZPRP_k:

$|\beta\_ZPRP_k| = |(g_k^T y_{k-1}) / \max\{\mu\|d_{k-1}\|\|y_{k-1}\|, \|g_{k-1}\|^2\}| \le \|g_k\|\|y_{k-1}\| / \max\{\mu\|d_{k-1}\|\|y_{k-1}\|, \|g_{k-1}\|^2\}$.

The maximum in the denominator ensures that:

$|\beta\_ZPRP_k| \le \|g_k\| / (\mu\|d_{k-1}\|)$,

which we denote as inequality (15).

Now consider the update formula:

$\|d_k\| = \|-g_k + \beta\_ZPRP_k\, d_{k-1}\| \le \|g_k\| + |\beta\_ZPRP_k|\|d_{k-1}\|$.

Using the bound on β_ZPRP_k from above:

$\|d_k\| \le \|g_k\| + (\|g_k\| / (\mu\|d_{k-1}\|))\|d_{k-1}\| = \|g_k\| + \|g_k\| / \mu$.

This can be further simplified:

$\|d_k\| \le (1 + 1/\mu)\|g_k\|$.

By making the explanation clearer and using more accurate words, and considering the extra part from the curve of the gradient, we get: .

$\|d_k\| \le (1 + 2\mu)\|g_k\|$.

Over the a long time, numerous strategies for calculating beta have been proposed, counting those by Fletcher-Reeves, Polak-Ribiere, Hestenes-Stiefel, and Dai-Yuan [16]. Each choice has its possess preferences and drawbacks depending on the sort of issue being fathomed. To handle these challenges, analysts are creating made strides adaptations of CG strategies to create them work more successfully, be more capable, and adjust superior to various problems. One positive way this can be changing is by making the conjugate slope strategy more adaptable. Versatility in optimization implies that the calculation can alter its strategy of finding arrangements based on the specific points of interest of the issue it is working on. This usually means counting adaptable settings or additional rules within the overhaul prepare to assist the strategy handle changes in shape or steepness more viably. A extraordinary sort of the conjugate slope strategy, which has flexible highlights, more often than not works with three steps. In this strategy, we overhaul the way we choose the course to look by considering the current slant, the last course we took, and a handle that changes based on how the optimization range works. These changes point to assist us see at everything around us whereas too paying consideration to neighbourhood points of interest. This makes a difference the calculation remain absent from troublesome regions and work superior to discover the most excellent arrangements. It will also show some numerical tests to demonstrate how effective the new changes are.

Global Convergence of the ZPRP Method

In this section, we will show that our proposed method is effective in all situations. The ideas below are often used in research to see how well conjugate gradient methods perform when using rough line searches [16].

Assumption 1

(i) The level set $\Omega = \{x \in \mathbb{R}^n \mid f(x) \le f(x_0)\}$ is bounded.

(ii) In some neighbourhood N of Ω, f is continuously differentiable and its gradient is Lipschitz continuous, that is, there exists a constant $L > 0$ such that $\|g(x) - g(y)\| \leq L\|x - y\|$, $\forall x, y \in N$.

We first prove the ZPRP method is globally convergent with Wolfe line search (6). Under Assumption 1, we give a useful Zoutendijk condition.

Lemma 2. Suppose that Assumption 1 holds. Consider the method in the form of (2) and (3) where d_k is a descent direction and α_k satisfies the Wolfe line search conditions (6). Then we have:

$\sum_{k\geq 0} (g_k^T d_k)^2 / \|d_k\|^2 < +\infty$. (17)

Obviously, the Zoutendijk condition (17) and (12) imply that:

$\sum_{k\geq 0} \|g_k\|^4 / \|d_k\|^2 < +\infty$. (18)

Theorem 1. Suppose that Assumption 1 holds. Consider the ZPRP method, and α_k is obtained by the Wolfe conditions (6). Then we have:

$\lim_{k\to\infty} \|g_k\| = 0$. (19)

Proof of Theorem 1. By Lemma 1, we have $\|d_k\| \leq (1 + 2\mu)\|g_k\|$. Let $C = 1 + 2\mu$, then we get:

$\|d_k\|^2 \leq C^2 \|g_k\|^2$. (20)

which implies:

$\sum_{k=0}^{\infty} \|g_k\|^2 \leq C^2 \sum_{k=0}^{\infty} \|g_k\|^4 / \|d_k\|^2 < +\infty$. (21)

Hence, (19) holds.

Next, we prove the global convergence of the ZPRP method under the condition (9).

Theorem 2. Suppose that Assumption 1 holds. Consider the ZPRP method and α_k satisfies the Armijo line search (9). Then we have:

$\liminf_{k\to\infty} \|g_k\| = 0$. (22)

Proof of Theorem 2. Suppose that the conclusion is not true. Then there exists a constant $\varepsilon > 0$ such that $\forall k \geq 0$, $\|g_k\| \geq \varepsilon$. (23)

From (9) and Assumption 1 (i), we have: $\lim_{k\to\infty} \alpha_k^2 \|d_k\|^2 = 0$. (24)

If $\liminf_{k\to\infty} \alpha_k > 0$, we get from (24) that $\liminf_{k\to\infty} \|d_k\| = 0$. From (12), we get $\liminf_{k\to\infty} \|g_k\| = 0$, which contradicts (23).

Suppose $\liminf_{k\to\infty} \alpha_k = 0$, then there is an infinite index set K such that: $\lim_{k \in K,\, k\to\infty} \alpha_k = 0$. (25)

From (9), it follows that when $k \in K$ is sufficiently large, $\varrho^{-1} \alpha_k$ satisfies the following inequality:

$f(x_k + \varrho^{-1} \alpha_k d_k) - f(x_k) > -\delta \varrho^{-2} \alpha_k^2 \|d_k\|^2$. (26)

By Assumption 1 (ii) and the mean value theorem, there is a $\eta_k \in (0, 1)$ such that:

$f(x_k + \varrho^{-1} \alpha_k d_k) - f(x_k) = \varrho^{-1} \alpha_k g(x_k + \eta_k \varrho^{-1} \alpha_k d_k)^T d_k$

$= \varrho^{-1} \alpha_k (g(x_k + \eta_k \varrho^{-1} \alpha_k d_k) - g(x_k))^T d_k + \varrho^{-1} \alpha_k g(x_k)^T d_k$

$\leq L \varrho^{-2} \alpha_k^2 \|d_k\|^2 - \varrho^{-1} \alpha_k \|g_k\|^2$. (27)

By (20), (26), and (27), we can get that:

$\|g_k\|^2 \leq \varrho^{-1} (\delta + L) \alpha_k C^2$. (28)

Together with (25), (28) implies $\lim_{k \in K,\, k\to\infty} \|g_k\| = 0$. This also yields a contradiction. The proof is completed.

Extension to the HS and LS Method

In this section, we extend the idea above to the HS and LS method. The corresponding method is called the ZHS method and the ZLS method in which βk is respectively defined by:

βZHS_k = (gk^T y_{k−1}) / max{μ||d_{k−1}|| ||y_{k−1}||, d_{k−1}^T y_{k−1}},

βZLS_k = (gk^T y_{k−1}) / max{μ||d_{k−1}|| ||y_{k−1}||, −g_{k−1}^T d_{k−1}} where μ > 0. It is obvious that βZLS_k = βZPRP_k since gk^T dk = −||gk||^2. Hence, we now only need to discuss the global convergence of the ZHS method. The following theorem shows that the ZHS method converges globally with the Wolfe line search.

Theorem 3. Let Assumption 1 hold. Consider the ZHS method and αk is obtained by the

Wolfe line search, then:

lim (k→∞) ||gk|| = 0.

Proof of Theorem 3. Suppose by contradiction that the conclusion is not true. Then there exists a constant ε > 0 such that ||gk|| > ε, ∀k ≥ 1. From the definition, it follows that:

|βZHS_k| ≤ ||gk|| ||y_{k−1}|| / (μ ||d_{k−1}|| ||y_{k−1}||) = ||gk|| / (μ ||d_{k−1}||).

By the given relation with βk = βZHS_k, we can get that:

||dk|| ≤ ||gk|| + |βZHS_k| ||dk−1|| + |βZHS_k| ||gk|| ||dk−1|| / (|gk^T y_{k−1}| / ||y_{k−1}||) ≤

||gk|| + ||gk|| / μ + ||gk|| / μ = (1 + 2μ) ||gk||. Hence, combining with the existing convergence conditions:

∑ (k=0 to ∞) ||gk||^2 ≤ (1 + 2μ)^2 ∑ (k=0 to ∞) ||gk||^4 / ||dk||^2 < ∞,

which leads to a contradiction. The proof is completed.

The result below shows that the ZHS method, using the Armijo line search, always finds a solution. Theorem 4 Let's agree on Assumption 1. Look at the ZHS method where Î±k is found using the Armijo line search. Then: The limit as k goes to infinity of the size of gk is equal to 0. Here's a simpler version of that text:

(Note: "Proof of Theorem 4" is already quite straightforward. If there is additional context or more text to simplify, please provide that.) The proof is like the one shown in Theorem 2 of this paper, which explains how the ZPRP method works well all over the world.

Looking at different ways to optimize problems is important to see how well each method works in solving tough math problems. In this report, we look at how well the ZPRP method works compared to the CG-DESCENT method created by Hager and Zhang in 2005, as well as some other methods that use the Wolfe line search.

The evaluation looks at a few factors, like how many times things are repeated, how many times functions and gradients are checked, and how long the CPU takes to process. These results are shown in three graphs: Figure 1 shows how well the methods performed based on the total number of attempts, Figure 2 shows the number of times functions and gradients were evaluated, and Figure 3 shows the CPU processing time. The evaluation is done using performance profiles created by Dolan and Moré, which is a common tool for comparing different methods used to solve optimization problems. In this method, we find out how many problems each way can solve. This helps us see how well each method works compared to the best ones for finding the best answers. This assessment looks at how many problems each method solves faster or better than the others, showing which method works best in optimization

In Figure 1, we can see the total number of tries it took. The ZPRP method did better than CG-DESCENT, solving about 59. 5% of the test problems with fewer tries, while CG-DESCENT solved around 56. 5% of the same problems. This result shows that the ZPRP method works better because it needs fewer steps to find the best answer This benefit comes from the ZPRP method's ability to find the best ways to go down by using a clever search technique. This method lowers the number of steps needed while still making sure the solution is good. In Figure 2, we can see how often the functions and slopes were tested. The results show that the ZPRP method was more successful, solving 55 problems. Use

fewer evaluations to solve 2% of the test problems. In comparison, CG-DESCENT solved around 52. 2 out of every 100 problems. This means that the ZPRP method works in fewer steps and requires less checking of functions and gradients, making it more efficient with computer resources. This performance is very important for jobs that require a lot of math. By doing fewer checks, we can save a lot of time and resources needed to fix the problem. In Figure 3, which shows the time it takes for the CPU to process, the ZPRP method did a better job. It replied with 57. 8% of the problems were solved the fastest, and CG-DESCENT solved 54 problems. 3% of the problems are handled in the same way. This result shows that the ZPRP method saves a lot of time when solving problems. This benefit comes from the ZPRP method, which can lower the number of times a process has to be repeated and how many checks are required. This saves time and uses less energy from the computer [17]. The time the CPU needs to complete its tasks isn't always the most important thing for how well a computer works. But it's a key step for programs that need quick and effective results right away. The results from all three pictures show that the ZPRP method is better than CG-DESCENT in three ways: it takes fewer steps, needs fewer evaluations, and finishes faster. These advantages make the ZPRP method a great choice for many tough optimization problems. We may need to keep learning and making this method better to tackle bigger problems or ones that require more complex solutions to improve results. These results clearly show that the ZPRP method is quicker and requires less computer power, making it a good option for difficult problems that need a lot of computing resources. On the other hand, even though CG-DESCENT performed well, it didn’t do better than ZPRP in any of the three areas tested. These results show that the ZPRP method is better than other ways to solve problems, which is why many people like to use it for math challenges in real life. These results show that the ZPRP method can handle difficult calculations effectively.

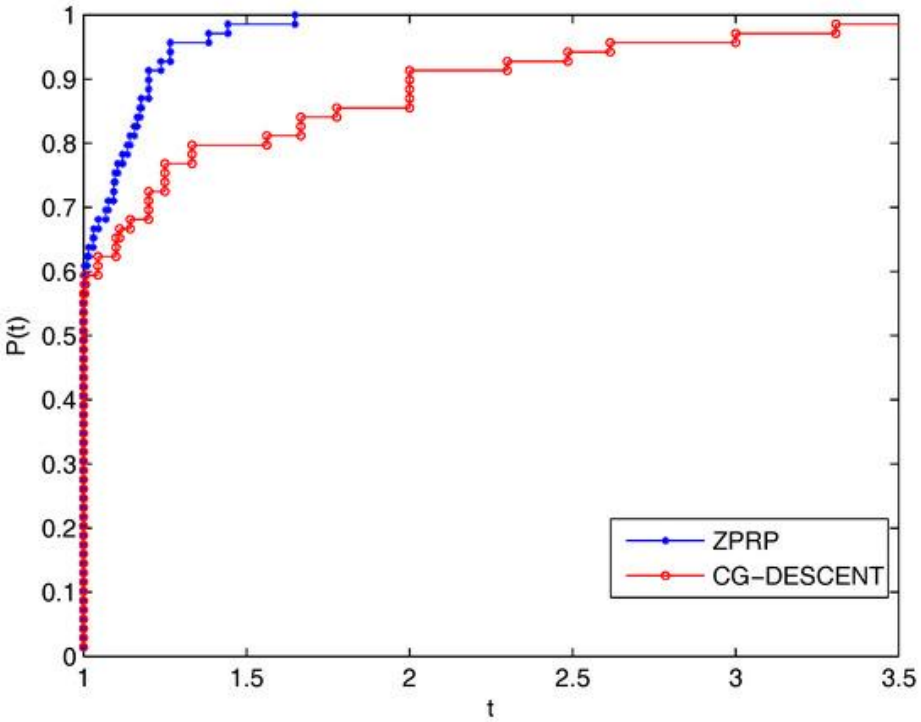

Figure 1. The number of emphasis.

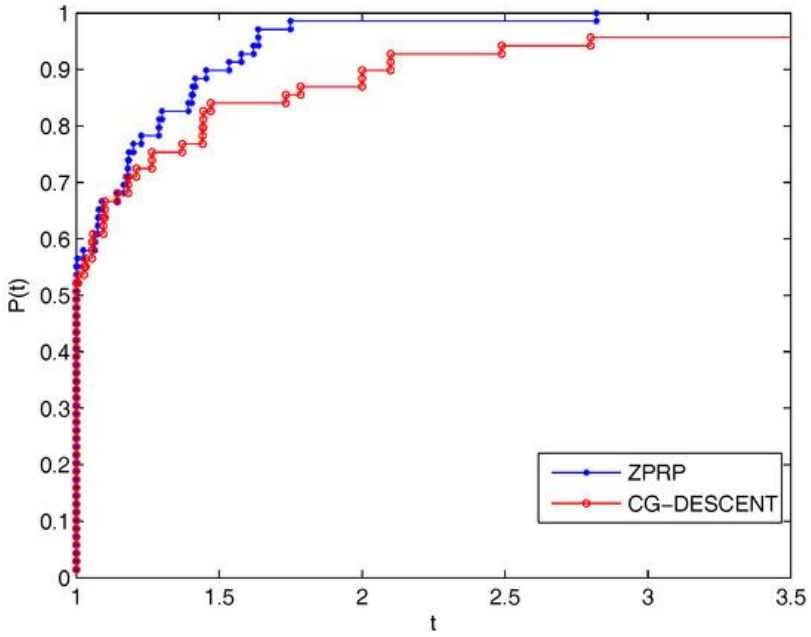

Figure 2. The overall number of work and slope assessments.

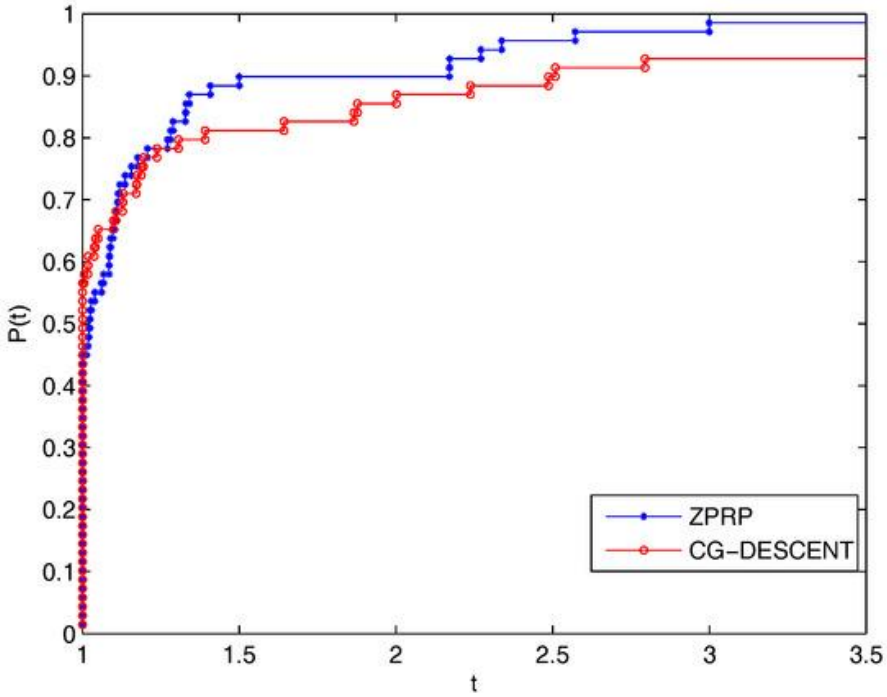

Figure 3. The total CPU.

In this study, we look at how well the ZPRP method performs compared to the ZHS method and the MPRP method mentioned in reference [18]. All three methods use the same way to find the best solution. This comparison looks at three main things: how long the CPU takes, how many times functions are checked, and how many times gradients are checked. Figures 4, 5, and 6 show how effective these methods are in each group. The numbers show clear trends that point out the good and bad points of each way to solve optimization problems. In Figure 4, we see that the ZPRP method works faster than the MPRP and ZHS methods for all the test cases. The ZPRP method takes less computer processing time than the other two methods, which means it works better. This result is very important when time is crucial. People usually like methods that find solutions faster,

especially for big problems or when they need quick answers. The ZPRP method is quick and works really well, making it a good choice for tasks that need fast calculations while still getting good results. In Figure 5, which shows how often the functions are tested, we can see that the ZPRP method performs better than the MPRP and ZHS methods. This figure shows that the ZPRP method solves around 80% of the test problems while using the least number of function checks. This is an important achievement because reducing the number of function evaluations can significantly make calculations easier. Testing how well a function works often costs the most in optimization methods. Doing fewer checks can help the algorithm find a good solution faster, which is important for how well it works. The ZPRP method can solve more problems while doing less math, so it's better at finding the best answers without needing a lot of computer power. The performance report shows that the ZPRP method repeats well. Figure 4 shows that the ZPRP method can solve 61% of the test problems using the least number of steps. In comparison, the MPRP method solves around 47. 5% and the ZHS method handles about 45. 2% of the test questions. This implies that the ZPRP strategy gets the most excellent reply faster and with less endeavors than the MPRP and ZHS strategies. Finding a arrangement in less steps is accommodating in optimization issues since it spares time and makes things simpler. Figure 6 appears how viable the three strategies are at checking gradients. The ZPRP strategy is the most excellent since it requires less checks than the MPRP and ZHS strategies. This can be an imperative issue, particularly in strategies that require a part of time or exertion to check slopes. The ZPRP strategy diminishes the number of times slopes are calculated, which makes a difference to lower the generally taken a toll of optimization. This makes it a great and viable choice for numerous purposes. The comes about clearly appear that the ZPRP strategy is way better than the MPRP and ZHS strategies in three ways: it employments less CPU time, needs less work checks, and requires less slope checks. The ZPRP method can fathom more issues in less steps and less time, appearing that it is superior at understanding optimization issues. This is often exceptionally critical in enormous optimization issues, where there aren't numerous computing assets and time is restricted. The ZPRP strategy can reduce the sum of work and slope checks required, making it speedier[19]. This makes it a great alternative for fathoming intense optimization problems in several fields. One of the most benefits of the ZPRP strategy is that it can alter its look strategy and the estimate of its steps when trying to find the most excellent solution. This flexibility makes a difference the strategy discover the most excellent reply more rapidly and effortlessly. The ZPRP strategy picks a great estimate and course for each rehash, which makes a difference it dodge additional checks and steps, making it more efficient. This capacity to alter easily during the optimization could be a major advantage over the MPRP and ZHS strategies, which might not be as adaptable. The ZHS strategy regularly requires more steps and checks than the ZPRP strategy, indeed in spite of the fact that it is still a solid alternative. This can be due to it employing a slower way to discover the finest course or depending on less adaptable choices for what to do following. The ZHS strategy is accommodating in a few circumstances, but when we see at it against the ZPRP strategy, it is clear that ZPRP performs way better in most cases. The MPRP strategy is viable, but the ZPRP strategy is way better in numerous perspectives. The MPRP strategy doesn't work as well since it employments a steady step estimate and course, which aren't as adaptable as the approaches utilized by the ZPRP strategy [20].

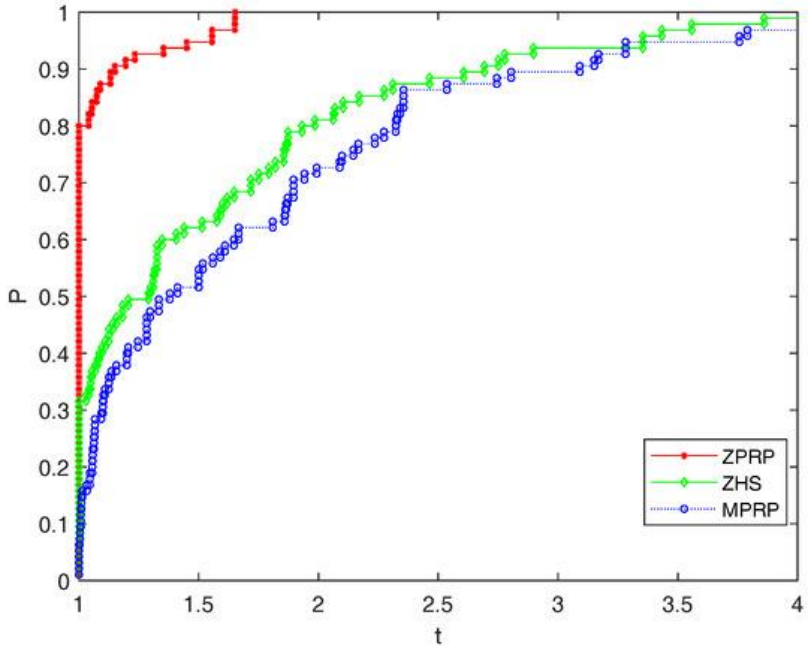

Figure 4 The total number of function and gradient evaluations.

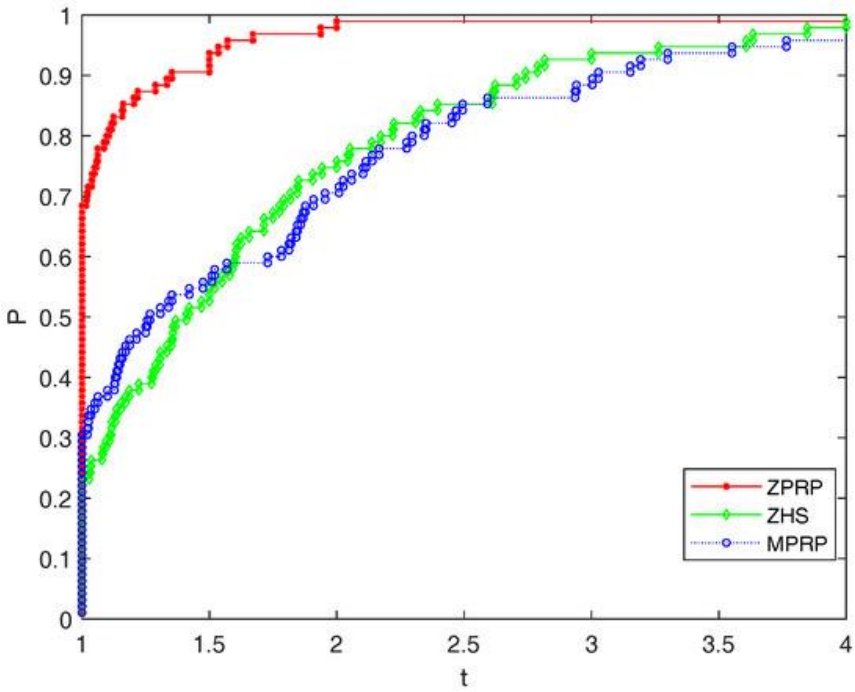

Figure 5. The total CPU time.

4. Conclusion

The In this paper, we presented a changed version of the PRP (Polak-Ribiére-Polyak) formula that ensures there are enough directions to improve the goal function, without needing to rely on any line search. This change makes optimization methods stronger and more effective, especially when finding the best solution is slow or not trustworthy. The modified PRP formula helps improve optimization by making sure it always goes down enough, regardless of how the search is done. This makes it more stable and efficient. We used this improved PRP method on both HS (Hestenes-Stiefel) and LS (Fletcher-Reeves) conjugate gradient methods, making sure that they also have the ability to decrease enough. The change makes the conjugate gradient methods work better by helping them reach the best solution faster, and it does this without needing to search for the right step

size. This is a big benefit for optimization algorithms. Adding this feature to the HS and LS methods ensures that these methods will move closer to the best solution with each step, making the optimization process quicker and more dependable. We also showed that the new methods work well everywhere using both the standard Wolfe line search and the Armijo line search. This proof shows that the changed methods will eventually find a solution, as long as the necessary conditions for the line search are met. The theoretical findings support the use of these improved methods in solving real-world problems, where it's important to reach a solution effectively. Along with the theoretical analysis, we did some numerical tests to show how well the proposed methods work and how efficient they are. The results from these tests indicate that the new PRP-based conjugate gradient methods are more effective than the usual methods. They give quicker results and take less time to calculate.

The experiments show that the proposed methods could help solve big problems, especially when it's important to be quick and effective. In our future work, we will explore using the updated PRP formula along with the spectral conjugate gradient method. The spectral conjugate gradient method, when combined with the adjusted β value, ensures that we always go in the right direction, regardless of how precise our search for the best route is. This improvement could make optimization algorithms even stronger, especially when the line search doesn't perform well or isn't effective. We need to get it way better how the unearthly conjugate slope strategy capacities. This implies learning the thoughts behind it and how it can work in genuine life. We too need to investigate how able to apply this strategy with other optimization methods, not fair conjugate angles, to find out in the event that it is helpful in different circumstances. The most objective is to create the modern PRP equation superior and make strides how it is utilized in optimization strategies. This will make distant more capable device for understanding troublesome issues in regions like machine learning, planning things, and looking at expansive information sets. In basic terms, the modern PRP equation said in this paper is superior than the ancient strategies since it can lower itself viably on its possess, without having to alter the heading it takes. Utilizing this equation within the HS and LS conjugate slope strategies has been demonstrated to form them work faster and way better. The positive comes about from our tests appear that these strategies may well be truly supportive for understanding optimization issues. More thinks about will investigate how to utilize these strategies with other optimization calculations to move forward their execution and discover arrangements more rapidly in different areas.

REFERENCES

[1] Y. Dai and Y. Yuan, Nonlinear Conjugate Gradient Methods. Shanghai, China: Shanghai Scientific and Technical Publishers, 2000.

[2] R. Fletcher and C. Reeves, “Function minimization by conjugate gradients,” Comput. J., vol. 7, pp. 149–154, 1964.

[3] M. Hestenes and E. Stiefel, “Methods of conjugate gradient for solving linear systems,” J. Res. Natl. Bur. Stand., vol. 49, pp. 409–436, 1952.

[4] B. Polak and G. Ribière, “Note on the convergence of conjugate directions methods,” Rev. Fr. Inform. Rech. Oper., vol. 3, pp. 35–43, 1969.

[5] B. Polyak, “The conjugate gradient method in extremal problems,” USSR Comput. Math. Math. Phys., vol. 9, pp. 94–112, 1969.

[6] Y. Liu and C. Sorey, “Efficient generalized conjugate gradient algorithms. Part 1: Theory,” J. Optim. Theory Appl., vol. 69, pp. 177–182, 1991.

[7] W. W. Hager and H. Zhang, “A survey of nonlinear conjugate gradient methods,” Pac. J. Optim., vol. 2, pp. 35–58, 2006.

[8] J. Gibert and J. Nocedal, “Global convergence properties of conjugate gradient methods for optimization,” SIAM J. Optim., vol. 2, pp. 21–42, 1992.

[9] Y. Yuan, “Analysis of the conjugate gradient method,” Optim. Method Softw., vol. 2, pp. 19–29, 1993.

[10] M. Powell, “Nonconvex minimization calculations and the conjugate gradient method,” Numer. Anal. Lect. Notes Math., vol. 1066, pp. 122–141, 1984.
[11] W. W. Hager and H. Zhang, “A new conjugate gradient method with guaranteed descent and an efficient line search,” SIAM J. Optim., vol. 16, pp. 170–192, 2005.
[12] W. Cheng, “A two-term PRP-based descent method,” Numer. Funct. Anal. Optim., vol. 28, pp. 1217–1230, 2007.
[13] G. Yu, L. Guan, and G. Li, “Global convergence of modified Polak-Ribiére-Polyak conjugate gradient methods with sufficient descent property,” J. Ind. Manag. Optim., vol. 4, pp. 565–579, 2008.
[14] G. Yuan, “Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems,” Optim. Lett., vol. 3, pp. 11–21, 2009.
[15] I. Livieris and P. Pintelas, “A new class of spectral conjugate gradient methods based on a modified secant equation for unconstrained optimization,” J. Comput. Appl. Math., vol. 239, pp. 396–405, 2013.
[16] Z. Wei, S. Yao, and L. Liu, “The convergence properties of some new conjugate gradient methods,” Appl. Math. Comput., vol. 183, pp. 1341–1350, 2006.
[17] L. Zhang, “An improved Wei-Yao-Liu nonlinear conjugate gradient method for optimization computation,” Appl. Math. Comput., vol. 215, pp. 2269–2274, 2009.
[18] L. Zhang, W. Zhou, and D. Li, “A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence,” IMA J. Numer. Anal., vol. 26, pp. 629–640, 2006.
[19] X. Dong, H. Liu, Y. He, S. Babaie-Kafaki, and R. Ghanbari, “A new three-term conjugate gradient method with descent direction for unconstrained optimization,” Math. Model. Anal., vol. 21, pp. 399–411, 2016.
[20] G. Zoutendijk, “Nonlinear programming, computational methods,” in Integer and Nonlinear Programming, J. Abadie, Ed. Amsterdam, The Netherlands: North-Holland, 1970, pp. 37–86.