TWMS J. App. and Eng. Math. V.16, N.3, 2026, pp. 409-427

# INVESTIGATION OF THE CONJUGATE GRADIENT METHODS IN SOLVING THE UNCONSTRAINED NONLINEAR OPTIMIZATION PROBLEM AND ITS APPLICATIONS

AMEL NASHAT SHAKIR[1*],FORAT MOHAMMED TAHA[1], §

Abstract. In this paper, we consider various types of methods such as; Newton, Quasi-Newton, Conjugate gradient, Trust region algorithm and etc. to solve an unconstrained nonlinear optimization problem. in most practical applications, the conjugate gradient method is the most efficient method to solve the large-scale optimization problems. numerical experimants show that, the conjugate gradient method requires less storage memory compared to that of existing ones. In this paper, we describe the solution of monotone nonlinear equations systems using the conjugate gradient methods.

Keywords: Unconstrained Nonlinear Optimization Problemes, Conjugate Gradient Method, Line Search Methods, Nonlinear Equation System, Globally Convergent

AMS 2020 Subject Classification: 90C30, 65K05, 49M37

## 1. Introduction

This study explores how conjugate gradient methods can be used to solve large nonlinear optimization problems and systems of monotone nonlinear equations. Many well-known optimization approaches, such as Newton, quasi-Newton, trust-region, and spectral gradient methods can converge very quickly, but they depend on Hessians or matrix approximations. For large problems, computing and storing these matrices can be expensive. Conjugate gradient methods avoid this cost because they do not require matrix calculations, making them attractive for high-dimensional settings. This motivation drives an examination of the various conjugate gradient method formulations, modified line-search techniques, and recent derivative-free methods. The numerical results suggest that conjugate gradient methods continue to be of a viable approach and often do even better than the rest, especially for large scale problems that fulfill the monotonicity and Lipschitz continuity constraint. The area of unconstrained nonlinear optimization is of great breadth because it appears in disciplines such as engineering, physics, and machine learning as well as inverse modeling. The early efforts on it dealt more with the Newton-type approaches,

[1] Department of Mathematics, Kirkuk Polytechnic College, Northern Technical University, Iraq.
e-mail: Umayaa75@ntu.edu.iq; ORCID: https://orcid.org/0000-0001-7936-4576 .
e-mail: firat@ntu.edu.iq; ORCID: https://orcid.org0009-0003-6863-566X.
∗ Corresponding Author.

§ Manuscript received: March 02, 2025; accepted: December 24, 2025.

which, although enjoyed better convergence, are computationally expensive as one has to evaluate and decompose the Hessian, an even more daunting task when working in high-dimensions. The quasi-Newton approaches that came about to remedy that considerably by approximating the Hessian, although works like [1] and [2] have strong convergence for BFGS-style updates, their matrix dependencies are dense which considerably hinders their scalability. This is where the conjugate gradient approach serves as a better fit as the matrix-free option. The classical formulations in [3] are gradient dependent which suites the problem well, although later works focused more on improving the rate of global convergence on such updates and the numerical stability. Some of these included [4] and [5] with better descent-ensured parameter choices while designing to a three-term version for better stability is [6] and [7]. More recently, these are incorporated with nonmonotone line search [8] for better convergence some classically called for monotone operators [9-10]. Solving monotone nonlinear equations is more than just an optimization problem. The earliest works that incorporated newton-type methods are [11]. The works in [12] are quasi-newton, while technique [13] utilizes spectral gradient. More recently, the literature has shifted towards projection-based methods, such as the one in [14] that established convergence via hyperplane projections without using Hessians. The works in [15-20] introduced several conjugate-gradient-based solvers for nonlinear equations, most of which exhibited good global convergence properties in the presence of monotonicity and Lipschitz conditions. In addition to optimization, sensitivity analysis is of great importance in mathematical modeling and nonlinear optimization. Saltelli et al. [21] is one of the most comprehensive studies in the field and provides a number of systematic approaches to model reliability assessment and uncertainty analysis. The study [22] has described one of the most relevant computational environments (MATLAB R2024a) and has become one of the standards in the implementation of modern optimization methods. The development of algorithms is still heavily influenced by benchmark functions, one of which is the Rastrigin functions introduced in [23], known for its multimodal nature and local minima. New conjugate gradient methods have incorporated newer benchmarks into the methodology as described in recent research. One such example is CG application to large scale optimization in image restoration in which stable convergence and computation efficiency were proved as Yuan et al. [24] did. In the same way, in the image denoising application, Xue et al. [25] introduced another DY-type method and reported better results with that method in the presence of realistic noise. New CG variants still focus on the same challenges of nonconvexity and promises of a descent. Liu and Du [26], on the other hand, modified the classical approaches of three terms to make them more globally and strongly convergent and were able to test the method on some engineering problems. CG method based on regression estimation by Dawahdeh et al. [27] also proved that statical conditions can have the methods iterative gradient methods obtain better accuracy and stability. The more recent [28] is able, we understand, to produce a nonconvex problem stagnation by providing theoretical grounds with three term methods that descent guarantee. CG search properties and line descent have always been a focus as in CG. Liu et al. [29] introduced three terms to inexact line search method to solve other methods computational exhausting problems while achieving the same results. Foundational contributions include Wolfe's conditions in [30], which still influence modern line search procedures, and Zoutendijk's classical inequality [31], a core tool in convergence proofs. Building on this groundwork, the present study analyzes several CG-based algorithms, including derivative-free approaches such as the NHZ method. These approaches combine efficient search directions, modified Armijo-type line searches, and projection techniques to solve nonlinear systems of the form $F(x) = 0$. Comparative experiments indicate improvements in iteration counts, function evaluations, and computation time on a wide range of large-scale benchmark problems.

## 2. Problem statement

The problem addressed in this study is solving large-scale unconstrained nonlinear optimization problems and monotone nonlinear equation systems without relying on Hessian information or matrix approximations. Many existing algorithms require storing or computing second-order information, making them inefficient for high-dimensional applications. The goal is to develop and analyze conjugate gradient- based methods that are matrix-free, memory-efficient, globally convergent, and capable of handling large monotone systems. This includes improving search directions, adopting modified line-search rules, and ensuring sufficient descent conditions so that numerical performance remains stable and competitive across a broad range of test problems.

## 3. Conjugate Gradient Methods for Solving Nonlinear Equations

In the first two chapters, we introduced conjugate gradient methods for solving unconstrained nonlinear optimization problems. However, it is important to note that conjugate gradient methods are not limited to nonlinear optimization problems alone. Other classes of problems, such as nonlinear equations, can also be solved using conjugate gradient approaches. In what follows, we discuss such problems in detail.

Consider the nonlinear equation

$$F\left(x\right) = 0 \tag{3.1}$$

where $F : \mathfrak{R}^n \to \mathfrak{R}^n$ is a continuous, Lipschitz, and monotone mapping. The objective is to solve equation (3.1), which, according to the first-order necessary optimality condition, is analogous to solving the optimization problem:

$$\min_{x \in \mathfrak{R}^n} f\left(x\right) \tag{3.2}$$

where

$$f\left(x\right) = \frac{1}{2} \left\|F\left(x\right)\right\|^r \tag{3.3}$$

Since the optimization problem (3.2) can be solved using methods such as Newton, quasi-Newton, conjugate gradient, and spectral gradient algorithms, the nonlinear equation (3.1) can likewise be solved using Newton methods [32], quasi-Newton methods [33], spectral gradient methods [34], and conjugate gradient methods [35,36,37,38,39,40,41,43,44], among others. Although Newton and quasi-Newton methods typically exhibit faster convergence, they require computation of the Hessian matrix (or its approximation) at each iteration to determine the search direction or step length. Consequently, for large-scale problems, these matrix operations become computationally expensive. However, conjugate gradient methods do not require any matrix computations and therefore demand significantly less storage memory. Consequently, they serve as more efficient approaches for solving equations of the form (3.1). In the following section, we present the conjugate gradient algorithm for solving equation (3.1).

3.1. **Conjugate Gradient Algorithms for Solving Nonlinear Equations.** Given that conjugate gradient methods for solving unconstrained nonlinear optimization problems are defined by relation (3.4),

$$d_{k+1} = \begin{cases} -g_{k+1} & k=-1 \\ -g_{k+1} + \beta_{k+1} \quad d_k + \gamma_{k+1} y_k, & k \geq 0 \end{cases} \tag{3.4}$$

and considering that according to (3.3) we have $\nabla f = g = F$, the structure of conjugate gradient methods for solving nonlinear equations of the form (3.1) can be expressed as

$$d_{k+1} = \begin{cases} -F_{k+1} & k=-1 \\ -F_{k+1} + \beta_{k+1}\delta_k + \gamma_{k+1} y_k & k \geq 0 \end{cases} \tag{3.5}$$

where $y_k = F_{k+1} - F_k, \delta_k = x_{k+1} - x_k$.

For searching directions defined by (3.5), a concept analogous to the notion of sufficient descent is defined as follows:

$$F_k^T d_k \leq -c \left\| F_k(x) \right\|^r, \qquad c > 0.$$

Let us assume that the point $z_k = x_k + \alpha_k d_k$ is generated through an appropriate line search. According to the definition of a descent direction $d_k$ for any $x_k$ we have

$$F(z_k)^T (x_k - z_k) > 0 \tag{3.6}$$

On the other hand, the definition of monotonicity of $F$ implies that for any solution $x^*$ satisfying $F(x^*) = 0$ *we* have

$$(z_k)^T (x^* - z_k) = -\left(F(z_k) - F(x^*)\right)^T (x_k - x^*) \leq 0 \tag{3.7}$$

Therefore, from relations (3.6) and (3.7), the hyperplane

$$H_k = \left\{ x \in \Re^n | F(z_k)^T (x - z_k) = 0 \right\}$$

exists, which strictly separates the point $x_k$ from the solution set of equation (3.1). Consequently, Solodov and Svaiter [32] defined the next iteration as

$$x_{k+1} = x_k - \frac{F(z_k)^T (x_k - z_k)}{\left\| F(z_k) \right\|^r} F(z_k) \tag{3.8}$$

which is the projection of $x_k$ onto the hyperplane $H_k$. For most iterative optimization algorithms, the line search plays a crucial role in both convergence analysis and numerical performance. Here, the step length $\alpha_k$ is determined using a *modified Armijo line search*, which improves upon the classical Armijo procedure. This enhanced version was first proposed by Zhang and Zheng in reference [44]. It is known that the classical Armijo line search is defined as

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \sigma_1 \alpha_k g_k^T d_k \tag{3.9}$$

where $\sigma_1 \in (0,1)$ is a given constant scalar. According to the standard backtracking line search procedure, $\alpha_k$ is chosen as the largest element from the set $\left(\beta, \beta\rho, \beta\rho^2, ...\right)$ which satisfies inequality (3.9), where $\rho \in (0,1)$ and $\beta > 0$ are fixed scalar parameters. Assuming that $g$ is Lipschitz continuous with Lipschitz constant $L$, Zhang and Zheng proposed a *modified Armijo line search* in which $\alpha_k$ is computed as

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \sigma \alpha_k \left( g_k^T d_k - \frac{1}{r} \alpha_k \mu L_k \left\| d_k \right\|^2 \right)$$

where $L_k$ is an approximation of $L$. The value of $L_k$ is obtained by solving the optimization problem:

$$\min \left\| L_k \delta_{k-1} - y_{k-1} \right\|$$

which yields

$$L_k = \frac{\delta_{k-1}^T y_{k-1}}{\|\delta_{k-1}\|^r}$$

where $\mu \in [0, \infty)$, $y_{k-1} = g_k - g_{k-1}\delta_{k-1} = x_k - x_{k-1}$ is a given constant scalar. The computation of the step size $\alpha_k$ using this *modified Armijo line search* for solving nonlinear equations of the form (3.1) has been presented by several researchers as follows:

1. $\alpha_k = \max\left\{\left(\beta\rho^i : i = 0, 1, ...\right)\right\}$ such that

$$-F\left(x_k + \alpha_k d_k\right)^T d_k \geq \sigma\alpha_k \|d_k\|^2 \tag{3.10}$$

where $\rho \in (0, 1)$, $\beta > 0$ ([45]).

Alternatively,

2. $\alpha_k = \max\left\{\beta\rho^i : i = 0, 1, ...\right\} d_k \geq \sigma\alpha_k \|d_k\|^2$

$$-F\left(x_k + \alpha_k d_k\right)^T d_k \geq \sigma\alpha_k \|F\left(x_k + \alpha_k d_k\right)\|^2 \|d_k\|^2 \tag{3.11}$$

where $\beta > 0$ and $\rho \in (0, 1)$, ([46]).

3. $\alpha_k = \max\left\{\beta\rho^i : i = 0, 1, ...\right\}$, where

$$-F\left(x_k\right)^T d_k \geq \sigma\alpha_k \min\left\{\|d_k\|^2 \|F\left(x_k + \alpha_k d_k\right)\| \|d_k\|^2 - F\left(x_k\right)^T d_k\right\} \tag{3.12}$$

where $\beta > 0$ and $\rho \in (0, 1)$, ([47]).

Among the various search directions proposed for solving equation (3.1), the following can be mentioned:

$$d_k = -F_k + \beta_k^{MHS} d_{k-1} + \theta_k^M \omega_{k-1}, \tag{3.13}$$

and, with a slight modification of direction (3.13), another form is given by

$$d_k = -F_k + \beta_k^{MHS}\left(I - \frac{F_k F_k^T}{\|F_k\|^T}\right) d_{k-1}$$

both of which were proposed in reference [35]. Here

$$\begin{aligned} \beta_k^{MHS} &= \frac{F_k^T \omega_{k-1}}{\omega_{k-1}^T d_{k-1}},\ \theta_k^M = \frac{F_k^T d_{k-1}}{\omega_{k-1}^T d_{k-1}},\ t = 1 + \|F_k\|^{-1} \max\left(0, \frac{y_k^T \delta_k^T}{\|\overline{\delta_k}\|^T}\right) \\ \overline{\delta_k} &= z_k - x_k = \alpha_k d_k, \omega_k = y_k + t\|F_k\|\,\overline{\delta_k} \end{aligned}$$

In these formulations, convergence was proven for monotone and Lipschitz continuous functions. Two additional directions are defined as

$$d_k = -F_k + \beta_k^{PRP} d_{k-1} + \theta_k y_{k-1},$$

and

$$d_k = -F_k + \beta_k^{PRP}\left(I - \frac{F_k F_k^T}{\|F_k\|^T}\right) d_{k-1}$$

where $\theta_k = \frac{F_k^T d_{k-1}}{\|F_{k-1}\|^T}$ and $\beta_k^{PRP} = \frac{F_k^T y_{k-1}}{\|F_{k-1}\|^T}$ as introduced in reference [36]. Their convergence was also established for monotone and Lipschitz continuous functions. From reference [37], two additional directions were derived, which similarly exhibit convergence for monotone and Lipschitz continuous mappings:

$$d_k = -F_k + \beta_k^{PRP} \omega_{k-1} - \theta_k y_{k-1},$$

where

$$\beta_k^{PRP} = \frac{F_k^T y_{k-1}}{\|F_{k-1}\|^T},\ \omega_{k-1} = z_{k-1} - x_{k-1} = \alpha_{k-1} d_{k-1}$$

and

$$\theta_k = \frac{F_k^T \omega_{k-1}}{\|F_{k-1}\|^T} + \frac{F_k^T \|y_{k-1}\|}{\|F_{k-1}\|^T}$$

and

$$\theta_k = \frac{\left(F_k^T y_{k-1}\right) \|\omega_{k-1}\|^T}{\|F_{k-1}\|^T}$$

**Algorithm 3.1: Derivative-Free Projection-Based Algorithm for Solving Nonlinear Equations**

Input: Initial point ($x_0$) and $\varepsilon > \circ$,

Output: Optimal (approximate) solution ($x^*$)

1. Set (k = 0).
2. While $\|F_k\| > \varepsilon$, do:
3. Find a descent direction ($d_k$).
4. Determine a suitable step size ($\alpha_k$).
5. Set $z_k = x_k + \alpha_k d_k$
6. Compute $F(z_k)$.
7. If $\|F(z_k)\| \le \varepsilon$,
8. then stop.
9. Else
10. $x_{k+1} = x_k - \frac{F(z_k)^1 (x_k - z_k)}{\|F(z_k)\|^k} F(z_k)$
11. End if
12. Set (k = k + 1).
13. End while

3.2. **3.3 The Derivative-Free NHZ Method for Solving Nonlinear Equations.** In this section, we present a recently developed method introduced in reference [47], which combines the HS method [48] with the projection technique [32]. This hybrid approach demonstrates improved performance compared to previous methods. In this algorithm, the step size parameter $\alpha_k$ is computed according to equation (3.12), and the search direction ($d\_k$) is defined as follows:

$$d_k = \begin{cases} -F_k & k = 0 \\ -F_k + \beta_k^{NHZ} d_{k-1} & k > 0 \end{cases} \tag{3.14}$$

where

$$\beta_k^{NHZ} = \frac{F_k^T y_{k-1}}{d_{k-1}^T \omega_{k-1}} - \mu \frac{\|y_{k-1}\|^T}{\left(d_{k-1}^T \omega_{k-1}\right)^T} F_k^T d_{k-1} \tag{3.15}$$

and the parametres are defined as follows:

$$y_{k-1} = F_k - F_{k-1} \cdot \overline{\delta_k} = z_{k-1} - x_{k-1} = \alpha_{k-1} d_{k-1},\ \omega_{k-1} = y_{k-1} + \gamma \bar{\delta}_{k-1},\quad \gamma > 0, \mu > \frac{1}{2}$$

The following theorem establishes that the derivative-free NHZ method satisfies the sufficient descent condition, which plays a crucial role in the convergence analysis of the algorithm.

**Theorem 3.1.** ([47]) *The search direction ( d_k) generated by (3.14) satisfies the *sufficient descent condition*. That is, if $d_k^T\omega_k \neq 0$, then*

$$F_k^T d_k \leq -\left(1-\frac{1}{\varphi\mu}\right)\|F_k\|^r, \mu > \frac{1}{4} \tag{3.16}$$

*Proof.* For $k=0$, we have

$$F_0^T d_0 = -\|F_0\|^T \leq \left(1-\frac{1}{\varphi\mu}\right)\|F_0\|^2 \; k \geq 1$$

so the sufficient descent condition for (3.14) clearly holds for $k=0$. Now, for $k>0$, from (3.14) we have:

$$\begin{aligned} F_k^T d_k &= -\|F_k\|^T + \beta_k F_k^T d_k - 1 = -\|F_k\|^T + \left\{\frac{F_k^T y_{k-1}}{d_{k-1}^T\omega_{k-1}} - \mu\frac{\|y_{k-1}\|^r}{\left(d_{k-1}^T\omega_{k-1}\right)^r}F_k^T d_{k-1}\right\}F_k^T d_{k-1} \\ &= \frac{F_k^T y_{k-1}\left(d_{k-1}^T\omega_{k-1}\right)\left(F_k^T d_{k-1}\right) - \|F_k\|^T\left(d_{k-1}^T\omega_{k-1}\right)^T - \mu\|y_{k-1}\|^T\left(F_k^T d_{k-1}\right)^T}{\left(d_{k-1}^T\omega_{k-1}\right)^T} \end{aligned}$$

If we define

$$u_k = \frac{1}{\sqrt{\Gamma\mu}}\left(d_{k-1}^T\omega_{k-1}\right)F_k, \;\; v_k = \sqrt{\Gamma\mu}\left(F_k^T d_{k-1}\right)y_{k-1} \tag{3.17}$$

then, using (3.17) and the inequality $u_k^T v_k \leq 1/r\left(\|u_k\|^r + \|v_k\|^r\right)$, we obtain

$$F_k^T d_k = \frac{u_k^T v_k - \frac{1}{r(\|u_k\|^r+\|v_k\|^r)}}{\left(d_{k-1}^T\omega_{k-1}\right)^r} - \left(1-\frac{1}{\varphi\mu}\right)\frac{\left(d_{k-1}^T\omega_{k-1}\right)^r}{\left(d_{k-1}^T\omega_{k-1}\right)^r}\|F_k\|^2 \leq \left(1-\frac{1}{\varphi\mu}\right)\|F_k\|^2$$

So the verdict is for $k \geq 0$. □

**Algorithm 3.2: Derivative-Free NHZ Method for Solving Nonlinear Equations**

Input: Initial point ($x_0$ ), constants $\varepsilon > 0, \sigma > 0., \beta > 0, \rho \in (0,1)$

Output: Optimal (approximate) solution ($x^*$)

1. Set (k = 0).
2. While $\|F_k\| > \varepsilon$, do:
3. Find a descent direction ($d_k$).
4. Determine a suitable step size ($\alpha_k$).
5. Set $z_k = x_k + \alpha_k d_k$
6. Compute $F(z_k)$.
7. If$\|F(z_k)\| \leq \varepsilon$,
8. then stop.
9. Else
10. $x_{k+1} = x_k - \frac{F(z_k)^1(x_k - z_k)}{\|F(z_k)\|^k} F(z_k)$
11. End if
12. Set (k = k + 1).
13. End while

3.2.1. *Convergence Analysis of the NHZ Method.* In this section, we present the global convergence of Algorithm 3.2, which was proven in reference [47]. First, the following lemma shows that the line search (3.12) is well-defined if the search direction ($d_k$) satisfies the sufficient descent condition.

**Lemma 3.1.** ([47]). *If the iterative sequences $\{x_k\}$ and $\{z_k\}$ are generated by Algorithm 3.2, then there exists a step size that satisfies the line search condition (3.12).*

*Proof.* Assume that for some nonnegative integer ($i$), the line search (3.12) fails for $\beta\rho^i$ in iteration $k_0$. Then we have

$$-F\left(x_0 + \beta\rho^i d_{k_0}\right)^T d_{k_0} < \sigma\beta\rho^i \min\left\{\|d_{k_0}\|^r, \left\|F\left(x_0 + \beta\rho^i d_{k_0}\right)\right\| \|d_{k_0}\|^r, -Fx_{k_0})^T d_{k_0}\right\}$$

Now, letting $i \to \infty, \rho \in (0,1)$, we obtain

$$-F\left(x_{k_0} + \beta\rho^i d_{k_0}\right)^T d_{k_0} < 0$$

which contradicts (3.16) (the sufficient descent condition for ($d_k$). Therefore, the statement of the lemma is valid. □

The next lemma shows that the line search (3.12) also has a positive lower bound for the step size ($\alpha_k$).

**Lemma 3.2.** ([47]). *If the iterative sequences $\{x_k\}$ and $\{z_k\}$are generated by Algorithm 3.2, then*

$$\alpha_k \geq \min\left\{\beta, \frac{\delta\rho}{(L+\sigma)} \frac{\|F_k\|^2}{\|d_k\|^2}\right\} \tag{3.18}$$

*where $\delta = 1 - \frac{1}{4\mu}$.*

*Proof.* Since $\alpha_k = \max\left\{\beta, \beta\rho, \beta\rho^2, ...\right\}$, if $\alpha_k = \beta_k$, then (3.18) clearly holds. Now, assume $\alpha_k \neq \beta$ Then, by the backtracking line search,

$$\alpha'_k = \rho^{-1}\alpha_k.$$

Since $\alpha_k \neq \beta$, it follows that $\alpha_k \notin \max\left\{\beta, \beta\rho, \beta\rho^2, ...\right\}$ does not satisfy (3.12). Hence

$$-F\left(z'_k\right)^T d_k < \sigma\alpha'_k \min\left\{\|d_k\|^2, \left\|F\left(z'_k\right)^T\right\|\|d_k\|^2, -F\left(z'_k\right)^T d_k\right\} \leq \sigma\alpha'_k\|d_k\|^2 \tag{3.19}$$

where $z'_k = x_k + \alpha'_k d_k$. From the sufficient descent condition (3.16), we have

$$\left(1 - \frac{1}{\varphi_\mu}\right)\|F_k\|^r = \delta\|F_k\|^r \leq -F_k d_k \tag{3.20}$$

Using the Lipschitz continuity of (F), (3.20), and (3.19), we get

$$-F^T d_k = \left(F\left(z'_k\right) - F\left(x_k\right)\right)^T d_k - F\left(z'_k\right)^T d_k \leq \left\|F\left(z'_k\right) - F\left(x_k\right)\right\|\|d_k\| + \sigma\alpha'_k\|d_k\|^2 \leq$$

$$\leq L\left\|z'_k - x_k\right\| + \sigma\alpha'_k\|d_k\|^2 \leq L\alpha'_k\|d_k\|^r + \sigma\alpha'_k\|d_k\|^2 = \rho^{-1}\alpha_k\left(L+\sigma\right)\|d_k\|^2 \tag{3.21}$$

From 3-20 and 3-21, we have:

$$\alpha_k \geq \frac{\delta\rho}{(L+\sigma)}\frac{\|F_k\|^2}{\|d_k\|^2}$$

which shows that the step size $\alpha_k$ has a positive lower bound. □

The next lemma for Algorithm 3.2 was proven by Solodov and Svaiter in [32]. Due to the similarity of its proof with Lemma 1.2 in [32], it is stated here without proof.

**Lemma 3.3.** ([47]) *Assume $(F)$ is monotone and Lipschitz continuous. If the iterative sequence $\{x_k\}$ is generated by Algorithm 3.2, then for any $(x^*)$ satisfying $F(x^*) = 0$,*

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \|x_{k+1} - x_k\|^2$$

*i.e. the sequence $\{x_k\}$ is bounded.*

**Remark 3.1.** *Lemma 3.2 shows that the sequence $\{\|x_k - x^*\|\}$ is non-increasing with $k$. Moreover, it implies that*

$$\lim_{k\to\infty}\|x_{k+1} - x^*\| = 0 \tag{3.22}$$

**Theorem 3.2.** *If the iterative sequence $x_k$ is generated by Algorithm 3.2, then*

$$\lim_{k\to\infty}\alpha_k\|d_k\| = 0 \tag{3.23}$$

*Proof.* From (3.8) and (3.12) for any $k$, we have

$$\|x_{k+1} - x_k\| = \frac{F\left(z_k\right)^T\left(x_k - z_k\right)}{\|F\left(z_k\right)\|} = \frac{-\alpha_k F\left(z_k\right)^T d_k}{\|F\left(z_k\right)\|} \geq \sigma\alpha_k^2\|d_k\|^2 \tag{3.24}$$

Using (3.24) and (3.22), we get $\lim_{k\to\infty}\alpha_k\|d_k\| = 0$. □

**Lemma 3.4.** ([47]). *If the iterative sequence $x_k$ is generated by Algorithm 3.2, and satisfies$F(x^*) = 0$ and $z'_k = x_k + \alpha'_k d_k$, then the sequences $\{\|F_k\|\}, \{\|F(z'_k)\|\}$ are bounded, i.e., there exists a constant $M > 0$ such that*

$$\left\|F\left(z'_k\right)\right\| \leq M, \quad \|F_k\| \leq M$$

*Proof.* Using Lemma 3.3, we have

$$\|x_k - x^*\| \le \|x_0 - x^*\|$$

From (3.23), there exists a constant $M_1 > 0$ such that $\alpha_k \|d_k\| \le M$. Therefore,

$$\left\|z_k' - x^*\right\| \le \|x_k - x^*\| + \alpha_k' \|d_k\| \le \|x_0 - x^*\| + \rho^{-1}\alpha_k \|d_k\| \le \|x_0 - x^*\| + M_1$$

Since $F(x)$ is Lipschitz continuous, we conclude that

$$\left\|F\left(z_k'\right)\right\| \le \left\|F\left(z_k'\right) - F\left(x^*\right)\right\| \le L\left(\|x_0 - x^*\| + \rho^{-1}M_1\right)$$

and

$$\|F_k\| \le \|F\left(x_k\right) - F\left(x^*\right)\| \le L\|x_k - x^*\| \le L\|x_0 - x^*\|$$

If we set

$$M = \max\left\{L\|x_0 - x^*\|, L\left(\|x_0 - x^*\| + \rho^{-1}M_1\right)\right\}$$

then we get the proof of the theorem. □

The following theorem proves the global convergence of the NHZ method.

**Theorem 3.3.** ([47]). *If the iterative sequence $\{x_k\}$ is generated by Algorithm 3.2, then*

$$\lim_{k\to\infty} \inf \|F_k\| = 0. \tag{3.25}$$

*Proof.* Assume, that (3.25) is false. Then there exists a constant $\varepsilon > 0$ such that $\|F_k\| > \varepsilon$. Since $F_k \neq 0$, from (3.16) we have $d_k \neq 0$. On the other hand, the monotonicity of $F$ and (3.14) imply

$$\overline{\delta}_{k-1}^T\omega_{k-1} = \langle F\left(z_{k-1}\right) - F\left(x_{k-1}\right), z_{k-1} - x_{k-1}\rangle + \gamma\overline{\delta}_{k-1}^T\overline{\delta}_{k-1} \ge \gamma\overline{\delta}_{k-1}^T\overline{\delta}_{k-1} \tag{3.26}$$

Now, since $\overline{\delta}_{k-1} = z_{k-1} - x_{k-1} = \alpha_{k-1}d_{k-1}$ relation (3.26) implies:

$$d_{k-1}^T\omega_{k-1} \ge \gamma\alpha_{k-1}\|d_{k-1}\|^2 \tag{3.27}$$

Using (3.15) and (3.27), we have

$$\left|\beta_k^{NHZ}\right| = \left|\frac{F_k^T y_{k-1}}{d_{k-1}^T\omega_{k-1}} - \mu\frac{\|y_{k-1}\|^T}{\left(d_{k-1}^T\omega_{k-1}\right)^r}F_k^T d_{k-1}\right| \le \frac{L\alpha_{k-1}\|d_{k-1}\|\|F_k\|}{\gamma\alpha_{k-1}\|d_{k-1}\|^2} +$$

$$+\mu\frac{L^2\alpha_{k-1}^2\|d_{k-1}\|^2}{\gamma^2\alpha_{k-1}^2\|d_{k-1}\|^{\varphi}}\|F_k\|\|d_{k-1}\| \le \left(\frac{L}{\gamma} + \mu\frac{L^2}{\gamma^2}\right)\frac{\|F_k\|}{\|d_{k-1}\|}$$

Using (3.14) and (3.27), we have

$$\|d_k\| \le \|F_k\| + \left|\beta_k^{NHZ}\right|\|d_{k-1}\| \le \left(1 + \frac{L}{\gamma} + \mu\frac{L^r}{\gamma^r}\right)M'$$

Letting $c = \left(1 + \frac{L}{\gamma} + \mu\frac{L^2}{\gamma^2}\right)M$, we get $\|d_k\| \le c$. Then, by Lemmas 3.2 and 3.3, together with $\|F_k\| > \varepsilon$ and $\|d_k\| > \varepsilon$, it follows that for all sufficiently large $k$,

$$\alpha_k\|d_k\| \ge \min\left\{\beta, \frac{\delta\rho}{L+\sigma}\frac{\|F_k\|^r}{\|d_k\|^r}\right\}\|d_k\| \ge \min\left\{\beta\varepsilon, \frac{\delta\rho\varepsilon^r}{(L+\sigma)c}\right\} \tag{3.28}$$

Clearly, inequality (3.28) contradicts (3.23). Hence, (3.52) holds, and the theorem is proven. □

## 4. Numerical Results and Experiments

4.1. **Concepts Related to the Numerical Comparison of Methods.** In this section, we demonstrate the performance of the methods TTRMIL, MCG2, MCG1, and TTMRMIL. The different algorithms are compared based on four metrics:

(k): the number of iterations performed by the algorithm,

($k_f$): the number of function evaluations,

($k_g$):): the number of gradient vector evaluations,

(t): the execution time of the algorithm in seconds.

The method for comparing algorithms was introduced in 2002 by Dolan and Moré [49]. Assume there are $n_p$ problems $\left\{p_1, p_\gamma, ..., p_{n_p}\right\}$ in the set $P$, and $n_q$ different methods $\left\{q_1, q_\gamma, ..., q_{n_q}\right\}$ in the set $Q$. The goal is to assign a number (or numbers) to each method that reflects its efficiency. Suppose we want to compare these methods in terms of execution time. Let $t_{p_l,q_i}$ denote the time required by method $q_i$ to solve problem $p_l$. First, for each method $q_i$ and problem $(p_l)$, we compute the performance ratio

$$r_{p_l,q_i} = \frac{t_{p_l,q_i}}{\min\left\{t_{p_l,q_j} : q_j \in Q\right\}},$$

Next, for each method $q_i$ and a given threshold $r \geq 1$, we assign the probability

$$P_{q_i}(\tau) = \frac{size\left\{p \in P : r_{p_l,q_i} < \tau\right\}}{n_p}$$

For example, $Pq_1(1) = 0.6$ means that method $q_1$ solved 60% of the problems in the shortest time compared to the other methods, and $Pq_1(2) = 0.7$ means that method $q_1$solved 70% of the problems within at most twice the best time. Note that $\lim_{\tau\to\infty} P_{q_i}(\tau)$.actually corresponds to the total number of problems that method $q_1$successfully solved.

A plot in which all $Pq_i$ curves are drawn is called a performance profile.

4.2. **Numerical Results for the Conjugate Gradient Methods MCG1 and MCG2.** Here, the methods MCG1 and MCG2 are numerically compared with some other conjugate gradient methods on two test sets. For these comparisons, 30 problems with their standard initial points were selected from [50]. All problems were tested numerically with dimensions ($n = [1000, 5000, 10000, 15000, 20000]$). For the WWP line search algorithm two-phase method similar to [5] was used, and its initial step size was set as

$$\alpha_0^0 = 1, \alpha_{k+1}^0 = \alpha_k \frac{\|d_k\|}{\|d_{k+1}\|} \quad \text{for } k = 0, 1, \zeta, ...$$

In all experiments, for the WWP line search technique, the parameters were set as $\sigma_1 = 10^{-5}$ and $\sigma_2 = 0.8$, and the stopping criterion for the conjugate gradient algorithm was $\varepsilon = 10^{-5}$. The algorithm was terminated if the number of iterations exceeded 4000 or if the number of function evaluations exceeded 20,000. Additionally, the line search loop was stopped after 15 iterations. For the MCG1 method, the parameter $\tau_1 = 1.1$ was used, while for MCG2 the parameters $(\tau_1, \tau_1) = (1.5, \circ\, 0.5)$ were selected. Two sets of experiments were conducted to compare MCG1and MCG2 with other conjugate gradient methods as follows:

1. In the first set of experiments, MCG1and MCG2 were compared with the following modified three-term conjugate gradient methods:

Table 4.1: Modified three-term methods used for comparison with MCG1 and MCG2 in the first set of experiments

| Name | Direction | Ref. |
|---|---|---|
| CGW | $d_{k+1}=\begin{cases}-g_{k+1}, & k=-1\\ -g_{k+1}+\dfrac{(g_{k+1}^Ty_k)d_k-(g_{k+1}^Td_k)y_k}{2\Vert g_k\Vert^2+5\Vert d_k\Vert\Vert y_k\Vert+3\Vert d_k\Vert\Vert g_k\Vert}, & k\geq 0\end{cases}$ | [55] |
| CGLFZ | $d_{k+1}=\begin{cases}-g_{k+1}, & k=-1\\ -g_{k+1}+\dfrac{(g_{k+1}^Ty_k)}{\Vert d_k\Vert^2}d_k-\dfrac{(g_{k+1}^Td_k)}{\Vert d_k\Vert^2}y_k, & k\geq 0\end{cases}$ | [29] |
| CGYN | $d_{k+1}=\begin{cases}-g_{k+1}, & k=-1\\ -g_{k+1}+\max\left(\dfrac{t_kg_{k+1}^Ty_k-g_{k+1}^Ts_k}{d_k^Ty_k},0\right)d_k+t_k\dfrac{g_{k+1}^Ts_k}{s_k^Ty_k}y_k, & k\geq 0\end{cases}$ <br> $t_k=\min\left\{\dfrac{(s_k^Ty_k)^2}{(s_k^Ty_k)^2+\Vert s_k\Vert^2\Vert y_k\Vert^2},\dfrac{s_k^Ty_k}{\Vert y_k\Vert^2}\right\}$ | [20] |

The results of these experiments are presented in the following figures:

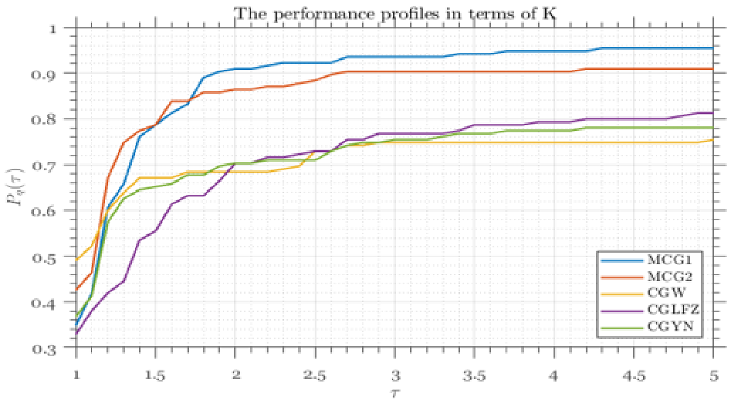


Fig. 4.1: Performance profile for $k$ (number of iterations)

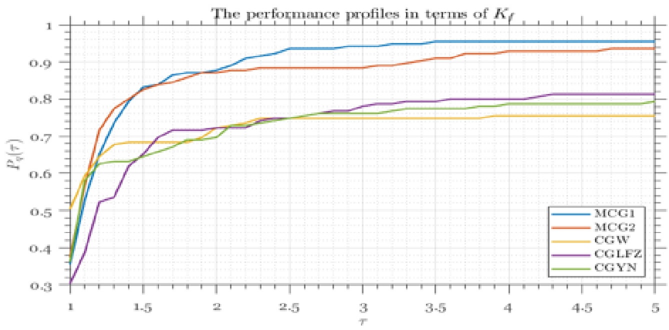


Fig. 4.2: Performance profile for $(k_f)$ (number of function evaluations)

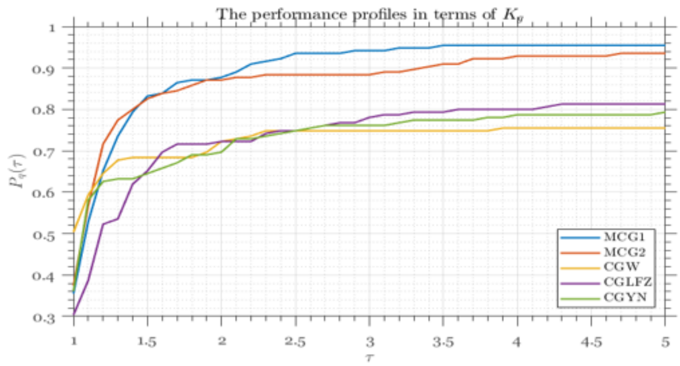


Fig. 4.3: Performance profile for ($k_g$ ) (number of gradient evaluations)

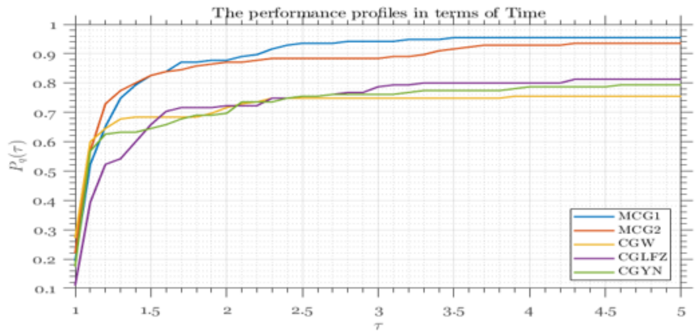


Fig. 4.4: Performance profile for $t$ (execution time)

2. In the second set of experiments, MCG1 and MCG2 were compared with four methods listed in the following table:

Table 4.2: Four conjugate gradient methods used for comparison with MCG1and MCG2 in the second set of experiments.

| Name | Direction | Ref. |
|---|---|---|
| CGHZ | $d_{k+1}=\begin{cases}-g_{k+1}, & k=-1\\ -g_{k+1}+\left(\frac{g_{k+1}^T y_k}{d_k^T y_k}-\left(2\frac{\Vert y_k\Vert^2}{s_k^T y_k}\right)\frac{g_{k+1}^T s_k}{d_k^T y_k}\right)d_k, & k\geq 0\end{cases}$ | [52] |
| CGF | $d_{k+1}=\begin{cases}-g_{k+1}, & k=-1\\ -\theta_k g_{k+1}+\left(\theta_k\frac{g_{k+1}^T y_k}{d_k^T y_k}-(1-\sigma_1)\left(\frac{s_k^T y_k}{\Vert s_k\Vert^2}\right)\frac{g_{k+1}^T s_k}{d_k^T y_k}\right)d_k, & k\geq 0\end{cases}$ <br> $\theta_k=\frac{\Vert s_k\Vert^2\left(\Vert g_{k+1}\Vert^2 s_k^T y_k-(g_{k+1}^T s_k)(g_{k+1}^T y_k)\right)}{s_k^T y_k(\Vert g_{k+1}\Vert^2\Vert s_k\Vert^2-(g_{k+1}^T s_k)^2)}$ | [15] |
| CGB | $d_{k+1}=\begin{cases}-g_{k+1}, & k=-1\\ -g_{k+1}+\left(\frac{g_{k+1}^T y_k}{d_k^T y_k}-\left(\sqrt{\frac{\Vert y_k\Vert(s_k^T y_k)}{\Vert s_k\Vert^3}}\right)\frac{g_{k+1}^T s_k}{d_k^T y_k}\right)d_k, & k\geq 0 \quad t_k\end{cases}$ | [53] |
| CGA | $d_{k+1}=\begin{cases}-g_{k+1}, & k=-1\\ -\theta_{k+1}g_{k+1}-\left[\left(1+\theta_{k+1}\frac{\Vert y_k\Vert^2}{s_k^T y_k}\right)\frac{g_{k+1}^T s_k}{s_k^T y_k}-\theta_{k+1}\frac{g_{k+1}^T y_k}{s_k^T y_k}\right]s_k+\theta_{k+1}\left(\frac{g_{k+1}^T s_k}{s_k^T y_k}\right)y_k, & k\geq 0\end{cases}$ <br> $\theta_{k+1}=\frac{\Vert s_k\Vert^2}{s_L^T y_k}$ | [14] |

Table 4.2: Four conjugate gradient methods used for comparison with MCG1and MCG2 in the second set of experiments

The results of these experiments are presented as follows

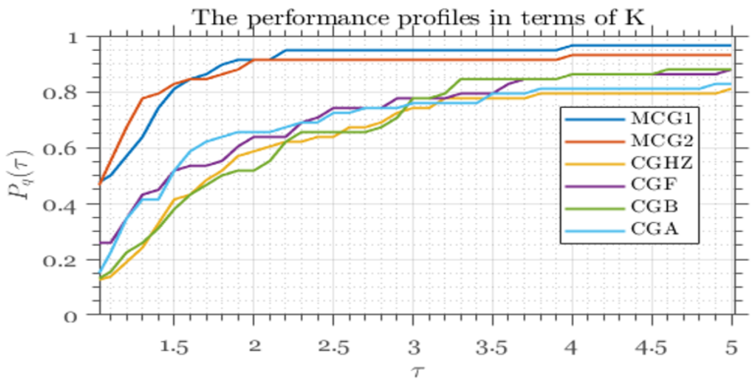


Fig. 4.5: Performance profile for ($k$) (number of iterations) in the second set of experiments for MCG1 and MCG2.

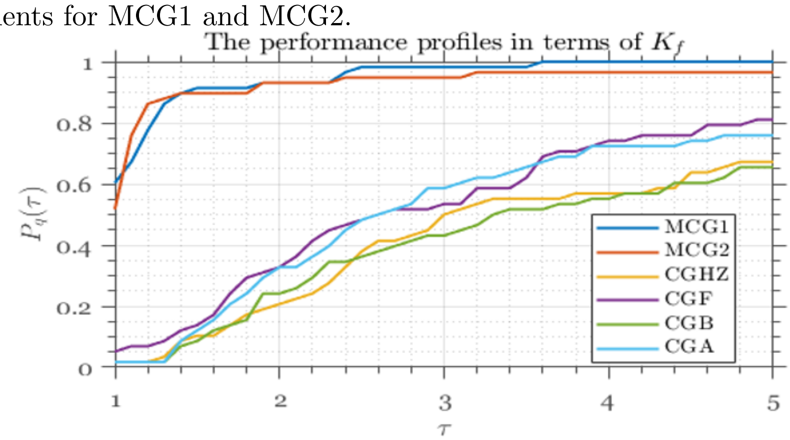


Fig. 4.6: Performance profile for ($k_f$) (number of function evaluations) in the second set of experiments for MCG1and MCG.

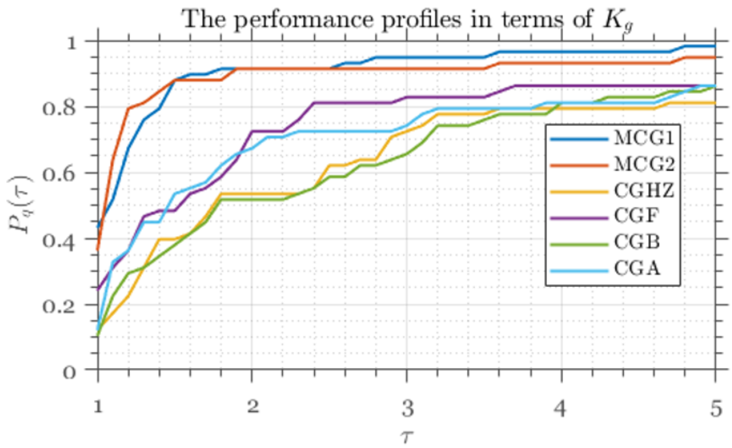


Fig. 4.7: Performance profile for ($k_f$) (number of gradient evaluations) in the second set of experiments for MCG1and MCG2.

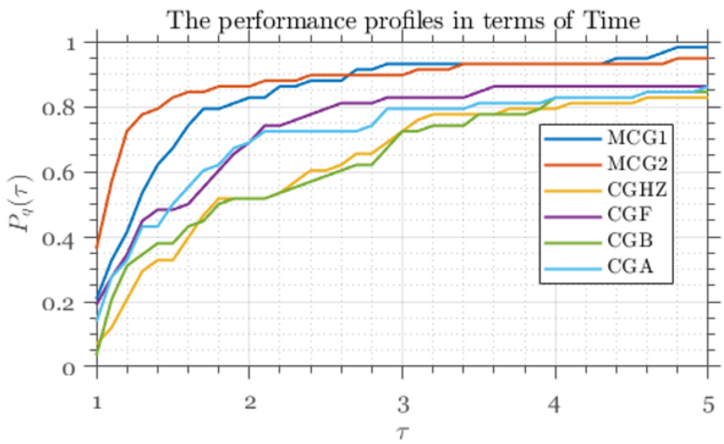


Fig. 4.8: Performance profile for $(t)$ (execution time) in the second set of experiments for MCG1 and MCG2 The following tables present all numerical results from the first set of experiments. The first column lists the problems, and the second column indicates the problem dimensions.

Table 4.3: Numerical results of MCG1 and MCG2 experiments

| No | Dim ($\times 10^9$) | MCG1 k/kf/kg/t | MCG2 k/kf/kg/t | CGW k/kf/kg/t | CGLFZ k/kf/kg/t | CGYN k/kf/kg/t |
|---|---|---|---|---|---|---|
| 1 | 1 | 38/327/109/5.005 | 34/291/97/4.27 | 85/582/194/8.39 | 56/462/154/6.65 | 91/699/203/8.78 |
| | 5 | 42/354/118/95.17 | 35/300/100/81.61 | 85/582/194/156.26 | 51/423/141/114.31 | 91/609/203/164.26 |
| | 10 | 38/327/109/333.10 | 35/300/100/296.01 | 85/582/194/592.30 | 54/444/148/451.74 | 91/609/203/620.02 |
| | 15 | 38/327/109/730.10 | 35/300/100/669.9 | 85/582/194/1300.02 | 53/447/149/997.33 | 91/609/203/1358.96 |
| | 20 | 38/327/109/1371.20 | 36/318/106/1333.4 | 85/582/194/2438.26 | 56/462/154/1936.5 | 91/609/203/2523.90 |
| 2 | 1 | 32/381/127/2.31 | 41/429/143/2.61 | 28/351/117/2.13 | 37/483/161/2.94 | 29/360/120/2.19 |
| | 5 | 118/1173/391/146.42 | 134/1485/495/185.12 | 100/816/272/101.76 | 8/141/47/72.73 | 6/96/32/11.95 |
| | 10 | 8/291/97/149.29 | 7/279/93/143.56 | 5/96/32/49.50 | - | 3/84/28/43.16 |
| | 15 | 6/264/88/294.48 | 8/294/98/327.55 | 3/84/28/93.49 | 3/111/37/126.51 | 4/90/30/104.64 |
| | 20 | 1/54/18/114.50 | 1/54/18/114.61 | 1/54/18/114.61 | 1/54/18/114.71 | 1/54/18/114.80 |
| 3 | 1 | 9/108/36/0.93 | 9/108/36/0.93 | 9/108/36/0.93 | 9/108/36/0.93 | 9/108/36/0.92 |
| | 5 | 9/108/36/16.26 | 9/108/36/16.17 | 9/108/36/16.24 | 9/108/36/16.32 | 9/108/36/16.27 |
| | 10 | 9/108/36/60.17 | 9/108/36/59.99 | 9/108/36/60.05 | 9/108/36/59.99 | 9/108/36/59.99 |
| | 15 | 9/108/36/139.91 | 9/108/36/140.77 | 9/108/36/129.15 | 9/108/36/129.62 | 9/108/36/129.18 |
| | 20 | 9/108/36/242.86 | 9/108/36/243.35 | 9/108/36/243.71 | 9/108/36/243.21 | 9/108/36/242.72 |
| 4 | 1 | 42/336/112/4.62 | 44/348/116/4.79 | 104/708/236/9.74 | 39/315/105/4.34 | 127/849/283/11.07 |
| | 5 | 65/480/160/107.72 | 68/633/211/141.82 | - | - | 272/1815/605/414.02 |
| | 10 | 76/645/215/520.45 | 89/870/290/702.76 | - | - | |
| | 15 | 76/639/213/1110.25 | 92/738/246/1282.68 | - | - | |
| | 20 | 86/723/241/2319.11 | 99/840/280/2693.83 | - | - | |
| 5 | 1 | 34/369/123/2.45 | 19/168/56/1.10 | - | 105/945/315/6.41 | 191/1185/395/7.92 |
| | 5 | 34/369/123/29.89 | 19/168/56/13.55 | - | 90/855/285/70.33 | 191/1185/395/7.92 |
| | 10 | 33/348/116/87.85 | 19/168/56/42.42 | - | 78/708/236/180.49 | 191/1185/395/298.60 |
| | 15 | 33/348/116/189.49 | 19/168/56/90.79 | - | 74/693/231/383.24 | 191/1185/395/448.48 |
| | 20 | 40/417/139/397.42 | 19/168/56/159.57 | - | 99/939/313/908.52 | 191/1185/395/1132.41 |

4.3. **Numerical Results for the Conjugate Gradient Methods TTRMIL and TTMRMIL.** In this section, the methods TTRMIL and TTMRMIL are compared in terms of performance with the following two methods:

Table 4.4: Two three-term conjugate gradient methods used for comparison with TTRMIL and TTMRMIL

| Name | Direction | Ref. |
|---|---|---|
| TTPRP | $d_k = \begin{cases} -g_k, & k = 0 \\ -g_k + \beta_k^{PRP} d_{k-1} - \theta_k y_{k-1}, & k \geq 1 \end{cases}$ <br> $\theta_k = \frac{g_k^T d_{k-1}}{\\|g_{k-1}\\|^2}, \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\\|g_{k-1}\\|^2}$ | [18] |
| MRMIL | $d_k = \begin{cases} -g_k, & k = 0 \\ -g_k + \beta_k^{MRMIL} d_{k-1}, & k \geq 1 \end{cases}$ <br> $\theta_k \beta_k^{MRMIL} = \frac{g_k^T (g_k - g_{k-1} - d_{k-1})}{\\|d_{k-1}\\|^2}$ | [54] |

For these comparisons, 27 problems with their standard initial points were selected from [50], All problems were tested in five dimensions: ([1000, 6000, 11000, 15000, 20000]). In all experiments, for the WWP line search technique, the parameters were set as $\sigma_1 = 10^{-4}$ and $\sigma_2 = 0.8$ and the stopping criterion for the conjugate gradient algorithm was $\varepsilon = 10^{-6}$ The CG algorithm was terminated if the number of iterations exceeded 10,000. For the WWP line search algorithm, a two-phase method similar to [5] was used, and the initial step size was calculated as

$$\alpha_k = \begin{cases} -1 & k = 0 \\ -\alpha_{k-1} \frac{\|d_{k-1}\|}{\|d_k\|} & k \geq 1 \end{cases}$$

The performance profiles of the two methods, TTRMIL and TTMRMIL, in terms of execution time and number of iterations, are presented as follows:

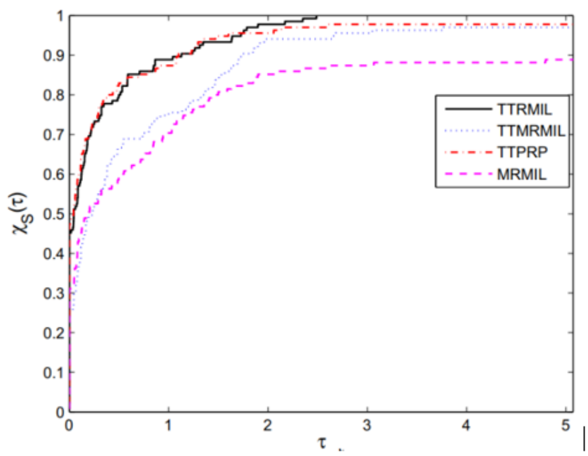


Fig.4.9: Performance profile for $(k)$ (number of iterations) in the experiments for TTRMIL and TTMRMIL

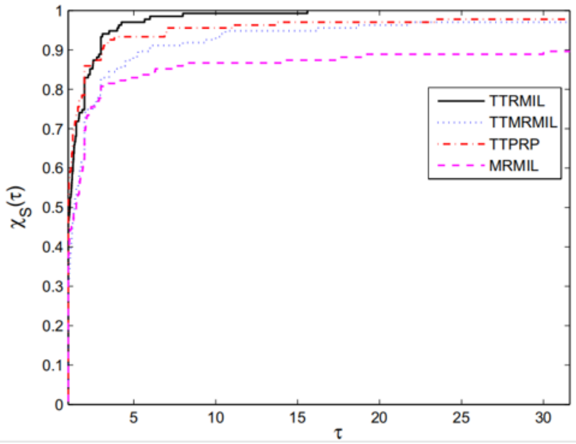


Fig.4.10: Performance profile for (L) (number of iterations) in the experiments for TTRMIL and TTMRMIL The numerical results of these experiments are presented in the following tables

## 5. Conclusion

This research focused on nonlinear optimization problems and systems of monotone nonlinear equations and the design and assessment of an augmented conjugate-gradient procedure intended for addressing large-scale cases. This method is characterized by an efficient search direction, modified line search, and sufficient descent mechanisms, and is, therefore, the design is balanced along the theoretical guarantees and the performance in practice. From the theoretical perspective, global convergence is proved under standard assumptions and conditions of the Wolfe and Zoutendijk type, classical convergence. There is substantial evidence from a number of large-scale sets that the method is effective and outperformed the competition on each one, requiring fewer overall iterations, less overall computation time, and less function and gradient evaluations of the method. This points to the merits of employing a conjugate-gradient design for cases problematic to convergence in terms of high dimensional, nonlinear, and nonconvex, as well as cases memory-efficient and high stability needed for emerging problem types. The impact of component derivative-free of the algorithm and restarted strategies, in particular, is to help convergence, and moving past stagnation, on systems of difficult nonlinearity. The research opens an important area of conjugate gradient methods for other problems of large-scale optimization and offers a fitting alternative from the increasing bodies of hybrid and three-term CG algorithms for image restoration, machine learning, and a number of other applications

## References

[1] Al-Baali, M., (1985), Descent property and global convergence of the Fletcher-Reeves method with inexact line search, IMA Journal of Numerical Analysis, 5(1), pp. 121-124.
[2] Andrei, N., (2008), An unconstrained optimization test functions collection', Advanced Modeling and Optimization, 10, pp. 147-161.
[3] Babaie-Kafaki, S., (2011), On optimality of two adaptive choices for the parameter of Dai–Liao method, Optimization Methods and Software, 31(1), pp. 1-15.
[4] Birgin, E. G. and Martínez, J. M., (2001), A spectral conjugate gradient method for unconstrained optimization, Applied Mathematics and Optimization, 43(2), pp. 117-128.
[5] Dai, Y. H. and Liao, L. Z., (2001), New conjugacy conditions and related nonlinear conjugate gradient methods, Applied Mathematics and Optimization, 43(1), pp. 87-101.

[6] Dai, Y. H. and Yuan, Y.,(1999), A nonlinear conjugate gradient method with a strong global convergence property, SIAM Journal on Optimization, 10(1), pp. 177-182.

[7] Dai, Z. and F., (2012), Global convergence of a modified Hestenes-Stiefel nonlinear conjugate gradient method with Armijo line search, Numerical Algorithms, 59(1), pp. 79-93.

[8] Dai, Z. and Zhu, H., (2020), A modified Hestenes-Stiefel-type derivative-free method for large-scale nonlinear monotone equations, Mathematics, 8(2), p. 168.

[9] Dolan, E. D. and Moré, J. J. ,(2002), Benchmarking optimization software with performance profiles, Mathematical Programming, 91(2), pp. 201-213.

[10] Fletcher, R., (1987), Practical Methods of Optimization. 2nd edn. New York: John Wiley & Sons.

[11] Fletcher, R. and Reeves, C. M., (1964), Function minimization by conjugate gradients', The Computer Journal, 7(2), pp. 149-154.

[12] Gilbert, J. C. and Nocedal, J., (1992), Global convergence properties of conjugate gradient methods for optimization, SIAM Journal on Optimization, 2(1), pp. 21-42.

[13] Hager, W. W. and Zhang, H., (2005), A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM Journal on Optimization, 16(1), pp. 170-192.

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

[15] Mosic, D., Stanimirovi P.S., Kazakovtsev L.A. Solving Minimization Problems by Weighted Weak Group Inverse

[16] Hestenes, M. R. and Stiefel, E., (1952), Method of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards, 49(6), pp. 409-436.

[17] Liu, Y. and Storey, C.,(1991), Efficient generalized conjugate gradient algorithms, part 1: theory, Journal of Optimization Theory and Applications, 69(1), pp. 129-137.

[18] Nocedal, J. and Wright, S. J., (2006), Numerical Optimization. 2nd edn. New York: Springer.

[19] Polak, E. and Ribiere, G., (1969), Note sur la convergence de méthodes de directions conjuguées, Revue française d'informatique et de recherche opérationnelle, 3(16), pp. 35-43.

[20] Polyak, B. T., (1969), The conjugate gradient method in extremal problems, USSR Computational Mathematics and Mathematical Physics, 9(4), pp. 94-112.

[21] Powell, M. J. D., (1977), Restart procedures for the conjugate gradient method, Mathematical Programming, 12(1), pp. 241-254.

[22] Shakir, A. N., (2021), Optimizing the modified conjugate gradient algorithm', International Journal of Nonlinear Analysis and Applications, 12 (Special Issue), pp. 1-10.

[23] Shanno, D. F., (1978), Conjugate gradient methods with inexact searches', Mathematics of Operations Research, 3(3), pp. 244-256.

[24] Touati-Ahmed, D. and Storey, C., (1990), Efficient step-size codes for optimization algorithms, Journal of Optimization Theory and Applications, 66(2), pp. 269-278.

[25] Wu, Y., (2017), A modified three-term PRP conjugate gradient algorithm for optimization models, Journal of Inequalities and Applications, 2017(1), pp. 1-14.

[26] Yuan, G. and Zhang, M., (2015), A three-term Polak-Ribière-Polyak conjugate gradient algorithm for large-scale unconstrained optimization, Applied Mathematics and Computation, 263, pp. 284-293.

[27] Zhang, L., Zhou, W. and Li, D., (2006), A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence, IMA Journal of Numerical Analysis, 26(4), pp. 629-640.

[28] Zhang, L., Zhou, W. and Li, D., (2007), Global convergence of a modified Fletcher–Reeves conjugate gradient method with Armijo-type line search, Numerische Mathematik, 104(4), pp. 561-572.

[29] Li, G. and Tang, C., (2010), A new hybrid conjugate gradient method for unconstrained optimization, Applied Mathematics and Computation, 216(10), pp. 2862-2870.

[30] Wei, Z., Yao, S. and Liu, L., (2006), The convergence properties of a conjugate gradient method, Applied Mathematics and Computation, 181(2), pp. 1102-1106.

[31] Liu, J. and Li, J., (2015), 'A modification of the PRP conjugate gradient method with implicit restart, Journal of Computational and Applied Mathematics, 280, pp. 199-211.

[32] Yao, S., Wei, Z. and Huang, H., (2008), A new conjugate gradient method with global convergence, Computational & Applied Mathematics, 27(1), pp. 1-15.

[33] Zhou, B. and Zhang, L., (2006), A descent Polak–Ribière–Polyak conjugate gradient method for unconstrained optimization, Applied Mathematics and Computation, 175(2), pp. 1195-1205.

[34] Shi, Z. J., (2002), Convergence of conjugate gradient method with a new line search, Applied Mathematics-A Journal of Chinese University, 17(2), pp. 183-188.

[35] ] Andrei, N., (2011), A hybrid conjugate gradient algorithm for unconstrained optimization as a convex combination of Fletcher–Reeves and Polak–Ribière–Polyak algorithms, Optimization Methods and Software, 24(1), pp. 55-70.

[36] Ghamisi, P. et al.,(2012), A survey on spectral-spatial classification methods based on statistical machine learning for hyperspectral data, International Journal of Remote Sensing, 33(23).

[37] Livieris, I. E. and Pintelas, P., (2012), A new spectral conjugate gradient method for training neural networks, Hellenic Conference on Artificial Intelligence, pp. 226-234.

[38] Rivaie, M. et al., (2012), A new class of nonlinear conjugate gradient coefficients with global convergence properties, Applied Mathematics and Computation, 218(22), pp. 11323-11332.

[39] Sugiki, A. et al., (2012), A new conjugate gradient method with a descent property, Journal of Industrial and Management Optimization, 8(3), pp. 549-568.

[40] Zhang, J., (2012), A new descent conjugate gradient method for unconstrained optimization, Applied Mathematics and Computation, 218(17), pp. 8830-8837.

[41] Mo, J., Liu, C. and Yan, S., (2011), A new spectral conjugate gradient method for unconstrained optimization, Applied Mathematics and Computation, 217(24), pp. 10090-10097.

[42] Narushima, Y., Yabe, H. and Ford, J. A., (2011), A three-term conjugate gradient method with sufficient descent property for unconstrained optimization, SIAM Journal on Optimization, 21(1), pp. 212-230.

[43] Liu, X. and Li, G., (2011), A new spectral conjugate gradient method for unconstrained optimization, Journal of Computational and Applied Mathematics, 235(12), pp. 3381-3389.

[44] Wan, Z. et al., (2011), A modified Polak–Ribière–Polyak conjugate gradient method with global convergence properties, Journal of Computational and Applied Mathematics, 235(12), pp. 3514-3522.

[45] Mosic D., Stanimirovi P.S., Kazakovtsev L.A. Solving Minimization Problems by Weighted Weak Group Inverse, Appl. Comput. Math., V.23, N.2, 2024, pp.228-243.

[46] Yu, G., Guan, L. and Chen, W., (2011), Global convergence of a modified PRP conjugate gradient method with sufficient descent property, Procedia Engineering, 15, pp. 3433-3437.

[47] Yuan, G., Lu, X. and Wei, Z., (2011), A conjugate gradient method with descent direction for unconstrained optimization, Journal of Computational and Applied Mathematics, 235(6), pp. 1507-1520.

[48] Zhang, L., (2011), An efficient three-term conjugate gradient method for unconstrained optimization, IMA Journal of Numerical Analysis, 31(4), pp. 1625-1635.

[49] Dai, Z. and Zhu, H., (2020), A modified Hestenes-Stiefel-type derivative-free method for large-scale nonlinear monotone equations, Mathematics, 8(2), p. 168.

[50] Dai, Z. and Wen, F. (2012) 'Global convergence of a modified Hestenes-Stiefel nonlinear conjugate gradient method with Armijo line search', Numerical Algorithms, 59(1), pp. 79-93.

[51] Dolan, E. D. and Moré, J. J., (2002), Benchmarking optimization software with performance profiles, Mathematical Programming, 91(2), pp. 201-213.

[52] Andrei, N., (2008), An unconstrained optimization test functions collection, Advanced Modeling and Optimization, 10(1), pp. 147-161.

[53] Wu, Y., (2017), A modified three-term PRP conjugate gradient algorithm for optimization models, Journal of Inequalities and Applications, 2017(1), pp. 1-14.

[54] Hager, W. W. and Zhang, H., (2005), A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM Journal on Optimization, 16(1), pp. 170-192.

[55] Babaie-Kafaki, S., (2016), On optimality of two adaptive choices for the parameter of Dai–Liao method, Optimization Methods and Software, 31(1), pp. 1-15.

[56] Rivaie, M., Mamat, M., & Abashar, A., (2015), A new class of nonlinear conjugate gradient coefficients with exact and inexact line searches. Applied Mathematics and Computation, 268, 1152-1163.

[57] Havvaei A., Ejlali R.G., Lakestani M., Sarand F.B. Solving the Exact Nonlinear Consolidation Equation using Chebyshev Cardinal Functions, Appl. Comput. Math., Vol.24, No.3, 2025, pp.469-485.

[58] Shakir, Amel Nashat., (2025), A New multi-objective arithmetic optimization algorithm, International Journal of Applied Mathematics, 38(8s), 577-596.

**A**mel Nashat Shakir is a lecturer in the Department of Mathematics at Kirkuk Polytechnic College, Northern Technical University, Iraq. She received her MSc degree in Applied Mathematics with a specialization in optimization and numerical methods. Her research interests include unconstrained nonlinear optimization, conjugate gradient methods, large-scale computational problems, and numerical analysis. She has published research in international journals and is committed to advancing efficient optimization algorithms with practical applications in engineering and applied sciences.

Her current focus is on improving the convergence properties and computational performance of conjugate gradient–based methods for large-dimensional nonlinear problems.

**F**orat Mohammed Taha is an Assistant Lecturer at Northern Technical University, Iraq. He received his academic training in mathematics and specializes in nonlinear optimization and numerical analysis. His research focuses on the development and analysis of efficient iterative algorithms, particularly conjugate gradient methods for solving unconstrained nonlinear optimization problems. He is interested in convergence analysis, large-scale optimization techniques, and applied computational mathematics. His work aims to enhance the performance and robustness of optimization algorithms for engineering and applied scientific applications.