Applied Mathematics and Computational Intelligence
Volume 14, No. 2, 2025 [19 – 36]

A New Family of Hybrid Three-Term Conjugate Gradient BNC-BTC Based on Scaled Memoryless BFGS Update for Unconstrained Optimization Problems

Muhammad Aqiil Iqmal Ishak[1], Yuhanis Ayub[1], Siti Mahani Marjugi[1*]

[1]Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.

* Corresponding author: smahani@upm.edu.my

Received: 19 November 2024
Revised: 6 February 2025
Accepted: 11 April 2025

ABSTRACT

Conjugate gradient (CG) methods are an important enhancement to the category of techniques utilized for resolving unconstrained optimization problems. However, some of the existing CG algorithms are not the most effective solution for all different kinds of problems. Particularly, for some problems, traditional CG techniques may show slower convergence rates or even fail to converge. These inefficiencies frequently result from large-scale issues' incapacity to maintain suitable descent directions or to accurately approximate the Hessian matrix. Hence, this paper introduces a new hybrid CG method for solving unconstrained optimization problems. The method proposed in this study incorporates two parameters, as proposed by Hassan and Alashoor, and aligns with the memoryless Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton approach. This approach satisfies the descent requirement and has the potential to achieve global convergence, presuming that the Wolfe and Armijo-like conditions and any other prerequisite assumptions are satisfied. Numerical experiments on certain benchmark test issues are performed, and the results show that the proposed method is more efficient than other existing methods.

Keywords: Global convergence, Line search, Memoryless BFGS, Three-term conjugate gradient method, Unconstrained optimization problem.

1 INTRODUCTION

Consider unconstrained optimization problem as follows:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable function. Then, the conjugate gradient (CG) method would build incrementally in accordance with

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

This indicates that, scaled by the factor α_k, the method gradually improves the solution by moving steps from the current point α_k in the direction d_k. With each iteration, the objective is to get closer to the best solution, which improves the estimate progressively.

The search direction, denoted as d_k, is defined by

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

In addition, x_k is the current iterate point, α_k is positive stepsize, g_k assigned as gradient coefficient and β_k denotes as CG coefficient. Besides, the sufficient descent property described by

$$g_k^T d_k \leq -c||g_k||^2, \quad c > 0. \tag{4}$$

There are several well-known formulas for the CG coefficient. The formulas presented in the literature [1], [2], [3], [4], [5], [6] and [7] exhibit variations primarily in the selection of the coefficient, denoted as β_k, as indicated below:

$$\beta_k^{\text{HS}} = \frac{r_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{\text{PRP}} = \frac{r_k^T y_{k-1}}{r_{k-1}^T r_{k-1}}, \quad \beta_k^{\text{LS}} = \frac{r_k^T y_{k-1}}{-d_{k-1}^T r_{k-1}}, \tag{5}$$

$$\beta_k^{\text{DY}} = \frac{r_k^T r_k}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{\text{FR}} = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}, \quad \beta_k^{\text{CD}} = \frac{r_k^T r_k}{-d_{k-1}^T r_{k-1}}, \tag{6}$$

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

For the purpose of making advancements on the traditional two-term CG approaches described above, three-term and hybrid CG techniques were developed. For instance, Andrei [8] suggested a three-term CG approach to addressing (1). The BFGS formula of the inverse Hessian approximation was modified in order to develop the proposed approach. The requirements for conjugacy and descent are met by the search direction. Assumptions like the boundedness of the level set and the Lipschitz continuity of the gradient were used to obtain the convergence. Liu and Li [9] suggested a new hybrid CG approach as a way to solve (1). Utilising a convex combination approach allows for the LS and DY procedures to be combined together in such a way as to get the desired results. In addition to meeting the Newton direction conditions, the search direction is also designed to satisfy the Dai-Liao (DL) conjugacy conditions. This dual capability ensures that the search direction not only approximates the Newton direction effectively but also maintains the desirable properties of conjugacy, which is crucial for achieving efficient and reliable convergence in unconstrained optimization problems. In order to achieve a global convergence with the hybrid strategy, it was important to make use of a strong Wolfe line search.

A hybrid CG technique to solve (1) based on earlier classical methods was also proposed by Jian et al. [10] which generates a descent direction at each iteration and is independent of any line search. To achieve global convergence, the consideration of the Wolfe condition and the incorporation of medium-scale concerns were deemed necessary for the numerical experiments.

A three-term PRP CG method with a search direction similar to the memoryless BFGS quasi-Newton method was developed by Li [11]. When the consideration of an exact line search is incorporated, the approach converges to the standard PRP method. The approach meets the descent requirement without the need for any line searches. For the purpose of determining the global convergence, an appropriate line search was performed. The strategy was effective for the typical unconstrained optimization issues in the CUTEr library based on the numerical findings.

Li [12] also suggested a nonlinear CG approach that produces a search direction similar to the memoryless BFGS quasi-Newton method. Additionally, the search direction meets the requirement for descent. Under the strong Wolfe line search, global convergence for both strongly convex and non-convex functions was established. For the purpose of solving (1), the modified Dai-Kou (DK) approach was proposed by Liu et al.[13]. A quasi-Newton approach was created in the paper to enhance the orthogonality of the gradient. In order to demonstrate global convergence, the researchers made fundamental assumptions and provided numerical results.

Furthermore, Iiduka and Narushima [14] and Hassan [15] have proposed a new CG method that utilizes both gradient and function values to enhance accuracy in approximating curvature. These algorithms demonstrate an impressive level of computational efficiency that exhibit a highly effective method in various application. Both study reports an enhancement in unconstrained problem performance through a quadratic model modification with the aim of optimizing the benefits of the original CG methods.

Recent work by Hassan and Alashoor [16] resulted in the development of new coefficients that are known as BNC and BTC. The β_k are defined as follows:

$$\beta_k^{\text{BNC}} = \frac{||g_k||^2}{(f_k - f_{k-1})/\alpha_k - 3/2 d_{k-1}^T g_{k-1}}, \tag{7}$$

$$\beta_k^{\text{BTC}} = \frac{||g_k||^2}{(f_k - f_{k-1})/\alpha_k + 3/2 d_{k-1}^T y_{k-1}}. \tag{8}$$

This method employs an alternative denominator approach based on the quadratic model of the noise function. The proposed methodology demonstrated significant improvements in numerical performance and efficiency for image restoration.

The proposed method integrates both the BNC and BTC parameters, as introduced by [16], and is inspired by the works of [11, 12] on modified and three-parameter conjugate gradient methods for addressing unconstrained optimization problems. This method meets the descent condition and is able to achieve global convergence, provided the Wolfe condition and other necessary assumptions are satisfied. Numerous studies have been conducted in prior research, which has mostly resulted in noteworthy enhancements. Therefore, the concept of scaled memoryless Broyden-Fletcher-Goldfarb-Shanno (BFGS) is applied. Consequently, the implementation of the newly developed three-term algorithm is expected to significantly improve numerical outcomes, including reduced iteration count and computational time. The following section presents the derivation of the hybrid method and its convergence. In section 3, numerical experimental results are presented.

2 ALGORITHM AND THEORETICAL RESULTS

In this section, the three-term CG method determines the search direction in the following manner:

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1} + \gamma_k g_k, & \text{if } k \geq 1, \end{cases} \tag{9}$$

where β_k, γ_k are parameters.

The direction in the memoryless BFGS method [17, 18] is determined by

$$d_k^{\text{BFGS}} = -H_k g_k,$$

where the value of H_k is derived using the memoryless BFGS formula.

$$H_k = \left(I - \frac{s_{k-1} y_{k-1}^T}{s_{k-1} y_{k-1}^T}\right)\left(I - \frac{y_{k-1}^T s_{k-1}}{s_{k-1} y_{k-1}^T}\right) + \frac{s_{k-1} s_{k-1}^T}{s_{k-1} y_{k-1}^T},$$

where I stands for the identity matrix. It is worth noting that the direction d_k of the memoryless BFGS technique can be expressed as a linear combination of $g_k.d_{k-1}$ and y_{k-1}. Given that $s_{k-1} = x_k - x_{k-1} = \alpha_{k-1} d_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. The direction d_k^{BFGS} can therefore be expressed as

$$d_k^{\text{BFGS}} = -g_k + \left(\beta_k^{\text{HS}} - \frac{||y_{k-1}||^2 g_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}\right) d_{k-1} + \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} (y_{k-1} - s_{k-1}). \tag{10}$$

The replacement of β_k^{HS} is carried out using β_k^{BNC} and β_k^{BTC}, respectively. Besides, $\frac{||y_{k-1}||^2 g_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}$ is also replace with $\frac{||g_k||^2 g_k^T d_{k-1}}{\left((f_k - f_{k-1})/\alpha_k + 3/2 d_{k-1}^T g_{k-1}\right)^2}$ and $\frac{||g_k||^2 g_k^T d_{k-1}}{\left((f_k - f_{k-1})/\alpha_k + 3/2 d_{k-1}^T y_{k-1}\right)^2}$, respectively. By replacing these, the following is the definition of a three-term search direction:

$$\begin{aligned} d_k^{\text{TTBNC}} = &- g_k + \left(\beta_k^{\text{BNC}} - \frac{||g_k||^2 g_k^T d_{k-1}}{\left((f_k - f_{k-1})/\alpha_k - 3/2 d_{k-1}^T g_{k-1}\right)^2}\right) d_{k-1} \\ &+ \left[t_k \frac{g_k^T d_{k-1}}{\left((f_k - f_{k-1})/\alpha_k - 3/2 d_{k-1}^T g_{k-1}\right)^2}\right] g_k, \end{aligned} \tag{11}$$

$$\begin{aligned} d_k^{\text{TTBTC}} = &- g_k + \left(\beta_k^{\text{BTC}} - \frac{||g_k||^2 g_k^T d_{k-1}}{\left((f_k - f_{k-1})/\alpha_k + 3/2 d_{k-1}^T y_{k-1}\right)^2}\right) d_{k-1} \\ &+ \left[t_k \frac{g_k^T d_{k-1}}{\left((f_k - f_{k-1})/\alpha_k + 3/2 d_{k-1}^T y_{k-1}\right)^2}\right] g_k. \end{aligned} \tag{12}$$

To ensure sufficient descent in the search direction, a restriction is imposed on the range of t_k such that $0 \leq t_k \leq \bar{t} < 1$ where $\bar{t}$ is a predefined upper bound that limits t_k to ensure stability and convergence. Therefore, t_k has been chosen as

$$t_k = \min\Big\{\bar{t}, \max\Big\{0, \frac{g_k^T(y_{k-1} - s_{k-1})}{||g_k||^2}\Big\}\Big\}, \tag{13}$$

which is the solution to the given univariate problem:

$$\min||(y_{k-1} - s_{k-1}) - t_k g_k||^2, \quad t \in \mathbb{R}.$$

Hence, inspired by the three-term CG directions established in (11) and (12), a hybrid three-term CG-based method is presented for solving equation (1), wherein the search direction is specified as

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{\text{TTBNTC}} d_{k-1} + \gamma_k g_k, & \text{if } k \geq 1, \end{cases} \tag{14}$$

where

$$\beta_k^{\text{TTBNTC}} = \frac{||g_k||^2}{w_k} - \frac{||g_k||^2 g_k^T d_{k-1}}{w_k^2}, \gamma_k = -t_k \left[\frac{g_k^T d_{k-1}}{w_k}\right], \tag{15}$$

and

$$w_k = \max\Big\{\mu||d_{k-1}||||g_k||, \frac{(f_k - f_{k-1})}{\alpha_k} - \frac{3}{2} d_{k-1}^T g_{k-1}, \frac{(f_k - f_{k-1})}{\alpha_k} + \frac{3}{2} d_{k-1}^T y_{k-1}\Big\}. \tag{16}$$

At this juncture, a novel algorithm, denoted as TTBNTC is being delineated.

Algorithm 1 An algorithm for TTBNTC

Step 1 Given constants $0 < \rho < \sigma < 1, \mu \geq 0, \varepsilon > 0$. Choose an initial point $x_0 \in \mathbb{R}^n$. Let $k := 0$.

Step 2 If $||g_k|| \leq \varepsilon$, where $\varepsilon = 10^{-6}$ then the algorithm stops. Otherwise, continue to Step 3

Step 3 Calculate the search direction d_k:

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1} + \gamma_k g_k, & \text{if } k \geq 1. \end{cases} \tag{17}$$

Step 4 Determine a steplength $\alpha_k > 0$ such that the Strong Wolfe line search conditions hold

$$\begin{cases} f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k, \\ |g(x_k + \alpha_k d_k)^T d_k| \geq \sigma |g_k^T d_k|. \end{cases}$$

Step 5 Set $x_{k+1} := x_k + \alpha_k d_k$, and $k := k + 1$. Return to step 2.

Remark 2.1. It can be observed that the parameter β_k in equations (11) and (12) is a hybrid of the $\beta_k^{\text{BNC}} - \frac{||g_k||^2 g_k^T d_{k-1}}{\left((f_k - f_{k-1})/\alpha_k - 3/2 d_{k-1}^T g_{k-1}\right)^2}$ and $\beta_k^{\text{BTC}} - \frac{||g_k||^2 g_k^T d_{k-1}}{\left((f_k - f_{k-1})/\alpha_k + 3/2 d_{k-1}^T y_{k-1}\right)^2}$ methods. Additionally, the path indicated by equation (14) closely aligns with the direction of the memoryless BFGS algorithm as $t_k = \frac{g_k^T (y_{k-1} - s_{k-1})}{||g_k||^2}$.

Remark 2.2. It should be noted that the selection of the first component of Equation (16) was done with great consideration in order to ensure that the direction exhibits a descending property, as discussed in Lemma 2.7. This property remains irrespective of the line search procedure.

Remark 2.3. The structure of d_k in equation (17) is inspired by the work of [19], as evidenced by the inclusion of g_k in both the first and third terms. To prove the sufficient descent properties and establish an upper bound, it is necessary to separate the third term, as demonstrated in Lemmas 2.4 and 2.8. Specifically, in Lemma 2.4, the second term will undergo a 'completing the square' process to effectively cancel out the third term, thereby ensuring the sufficient descent properties. By designating the third term as g_k, the algorithm can satisfy these properties. However, using other terms such as g_{k-1}, y_{k-1}, s_{k-1}, d_k, or d_{k-1} would prevent the algorithm from meeting the necessary conditions.

Lemma 2.4. *The search direction d_k, as described by equation (14), satisfies equation (4) with the constant $c = \frac{3}{4}$.*

Proof. By performing the operation of multiplication on both sides of the equation (14) with g_k^T, it can be deduced that

$$
\begin{aligned}
g_k^T d_k &= -||g_k||^2 + \beta_k^{\text{TTBNTC}} g_k^T d_{k-1} + \gamma_k g_k^T g_k \\
&= -||g_k||^2 + \frac{||g_k||^2}{w_k} g_k^T d_{k-1} - \frac{||g_k||^2}{w_k^2} (g_k^T d_{k-1})^2 - t_k \left(\frac{||g_k||^2}{w_k} \right) g_k^T d_{k-1} \\
&\leq -||g_k||^2 + (1 - t_k) \frac{||g_k||^2}{w_k} g_k^T d_{k-1} - \frac{||g_k||^2}{w_k^2} (g_k^T d_{k-1})^2 \\
&= -||g_k||^2 + 2 \left(\frac{(1 - t_k)}{2} g_k^T \right) \frac{g_k}{w_k^2} g_k^T d_{k-1} - \frac{||g_k||^2}{w_k^2} (g_k^T d_{k-1})^2 \\
&\leq -||g_k||^2 + \frac{(1 - t_k)^2}{4} ||g_k||^2 + \frac{||g_k||^2}{w_k^2} (g_k^T d_{k-1})^2 - \frac{||g_k||^2}{w_k^2} (g_k^T d_{k-1})^2 \\
&= -||g_k||^2 + \frac{(1 - t_k)^2}{4} ||g_k||^2 \\
&= -\left(1 - \frac{(1 - t_k)^2}{4} \right) ||g_k||^2 \\
&\leq -\frac{3}{4} ||g_k||^2.
\end{aligned}
$$

$\square$

2.1 Convergence analysis

In the subsequent analysis, the focus will be directed towards verifying the convergence of the suggested scheme. This will be accomplished by initially examining the strong Wolfe line search conditions.

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k, \tag{18}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \geq \sigma |g_k^T d_k|, \tag{19}$$

where $0 < \rho < \sigma < 1$.Furthermore, it is assumed that

Assumption 2.5. The level set

$$M = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$$

is bounded.

Assumption 2.6. In a given neighbourhood N of M, the function f is continuously differentiable and its gradient exhibits Lipschitz continuity. This implies the existence of a positive constant $L > 0$ such that the following condition holds:

$$||g(x) - g(\overline{x})|| \leq L||x - \overline{x}||, \quad \forall \overline{x} \in N. \tag{20}$$

Based on Assumptions 2.5 and 2.6, it can be concluded that for all $x \in M$ there are positive constants B_1 and B_2 that satisfy the given condition

$$||x|| \leq l_1,$$

and

$$||g(x)|| \leq l_2,$$

Moreover, it can be concluded that the sequence x_k in M is due to the fact that the sequence $f(x_k)$ exhibits a decreasing pattern. Moving forward, it will be considered that Assumptions 2.5 and 2.6 are valid and that the objective function has a lower bound. In this section, the convergence result will be demonstrated.

Theorem 2.7. *Assume that (18) and (19) are true. If*

$$\sum_{k=0}^{\infty} \frac{1}{||d_k||^2} = +\infty, \tag{21}$$

then

$$\lim_{k \to \infty} \inf ||g_k|| = 0. \tag{22}$$

Proof. For the purpose of contradiction, let's say that Equation (22) is not met. In this case, there is a scalar ε that is nonnegative, such that

$$||g_k|| \geq \varepsilon, \forall k \in \mathbb{N} \tag{23}$$

From Lemma 2.4 and (18),

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k g_k^T d_k \leq f(x_k) - \rho\alpha_k g_k^T d_k \leq f(x_k) \leq f(x_{k-1}) \leq ... \leq f(x_0).$$

□

Similarly, according to Lemma 2.4, condition (19), and Assumption 2.6, it may be inferred that

$$-(1-\sigma)g_k^T d_k \leq (g_{k+1} - g_k)^T d_k \leq ||g_{k+1} - g_k||||d_k|| \leq \alpha L||d_k||^2.$$

By using the aforementioned inequality with equation (18), The derivation is obtained.

$$\frac{\rho(1-\sigma)}{L}\frac{(g_k^T d_k)^2}{||d_k||^2} \leq f(x_k) - f(x_{k+1})$$

and

$$\frac{\rho(1-\sigma)}{L}\sum_{k=0}^{\infty}\frac{(g_k^T d_k)^2}{||d_k||^2} \leq (f(x_0) - f(x_1)) + (f(x_1) - f(x_2)) + \ldots \leq f(x_0) < +\infty.$$

Given that the sequence $f(x_k)$ is bounded. The statement above suggests that

$$\sum_{k=0}^{\infty}\frac{(g_k^T d_k)^2}{||d_k||^2} < +\infty. \tag{24}$$

From (23) and (4), it can be deduced that

$$\begin{aligned} g_k^T d_k &\leq -\frac{3}{4}||g_k||^2 \\ &\leq -\frac{3}{4}\varepsilon^2. \end{aligned} \tag{25}$$

Next,

$$\sum_{k=0}^{\infty}\frac{(g_k^T d_k)^2}{||d_k||^2} \geq \frac{9}{16}\sum_{k=0}^{\infty}\frac{\varepsilon^2}{||d_k||^2} = +\infty. \tag{26}$$

This statement is in contradiction with the information presented in (24). Hence, it may be inferred that the conclusion of the theorem is valid.

In the following section, we will examine the convergence of the proposed approach using the Armijo-type backtracking line search procedure. This procedure is defined by the parameters ϕ and ρ, both of which lie within the open interval (0, 1). The step size α_k is determined as ϕ^i, where i s the smallest nonnegative integer that satisfies the line search criteria.

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k^2 ||d_k||^2 \tag{27}$$

satisfied. Based on equation (27) and the observed decreasing character of the sequence $f(x_k)$, it may be inferred that

$$\sum_{k=0}^{\infty} \alpha_k^2 ||d_k||^2 < +\infty.$$

This finding suggests that

$$\lim_{k\to\infty} \alpha_k ||d_k|| = 0. \tag{28}$$

Lemma 2.8. *If the sequence d_k is defined by equation (14), then there exists a positive value G_1 such that $||d_k|| \leq K_1$.*

Proof. From (15),

$$\begin{aligned} |\beta_k^{\text{TTBNTC}}| &= \left| \frac{||g_k||^2}{w_k} - \frac{||g_k||^2 g_k^T d_{k-1}}{w_k^2} \right| \\ &\leq \frac{||g_k||^2}{\mu ||d_{k-1}|| ||g_k||} - \frac{||g_k||^3 ||d_{k-1}||}{(\mu ||d_{k-1}|| ||g_k||)^2} \\ &= \left(\frac{1}{\mu} + \frac{1}{\mu^2} \right) \frac{||g_k||}{||d_{k-1}||}. \end{aligned} \tag{29}$$

Also,

$$\begin{aligned} |\gamma_k| &= \left| -t_k \frac{g_k^T d_{k-1}}{w_k} \right| \\ &= t_k \left| \frac{g_k^T d_{k-1}}{w_k} \right| \\ &\leq \bar{t} \frac{||g_k|| ||d_{k-1}||}{w_k} \\ &\leq \bar{t} \frac{||g_k|| ||d_{k-1}||}{\mu ||d_{k-1}|| ||g_k||} \\ &= \frac{\bar{t}}{\mu}. \end{aligned} \tag{30}$$

Hence, based on equations (14), (29), and (30), it can be inferred that

$$\begin{aligned}||d_k|| &= ||-g_k + \beta_k^{\text{TTBNTC}} d_{k-1} + \gamma_k g_k|| \\ &\leq ||g_k|| + |\beta_k^{\text{TTBNTC}}|||d_{k-1}|| + ||\gamma_k||||g_k|| \\ &\leq ||g_k|| + \left(\frac{1}{\mu} + \frac{1}{\mu^2}\right)\frac{||g_k||}{||d_{k-1}||}||d_{k-1}|| + \frac{\bar{t}}{\mu}||g_k|| \\ &= \left(1 + \frac{1+\bar{t}}{\mu} + \frac{1}{\mu^2}\right)||g_k|| \qquad (31) \\ &= \left(1 + \frac{1+\bar{t}}{\mu} + \frac{1}{\mu^2}\right)l_2. \qquad (32)\end{aligned}$$

By letting $K_1 = (1 + \frac{1+\bar{t}}{\mu} + \frac{1}{\mu^2})l_2$, it can be observed that

$$||d_k|| \leq K_1. \qquad (33)$$

□

Theorem 2.9. *If the value of the step size α_k is such that equation (27) is satisfied,*

$$\lim_{k\to\infty} \inf ||g_k|| = 0. \qquad (34)$$

Proof. Let it be assumed by the method of contradiction, that equation (34) is false. For any given value of k, it is possible to identify a positive value of q such that

$$||g_k|| \geq q. \qquad (35)$$

By defining $\alpha_k = \phi^{-1}\alpha_k$, it becomes evident that α_k does not satisfy equation (27). This statement signifies

$$f(x_k + \phi^{-1}\alpha_k d_k) > f(x_k) - \rho\phi^{-2}\alpha_k^2||d_k||^2. \qquad (36)$$

By utilising the mean value theorem in conjunction with Lemma 2.8, equations (4) and (20), it can be concluded that there exists a value z_k within the interval (0, 1).

$$\begin{aligned}f(x_k + \phi^{-1}\alpha_k d_k) - f(x_k) &= \phi^{-1}\alpha_k g(x_k + z_k\phi^{-1}\alpha_k d_k) \\ &= \phi^{-1}\alpha_k g_k^T d_k + \phi^{-1}\alpha_k(g(x_k + z_k\phi^{-1}\alpha_k d_k) - g_k)^T d_k \\ &\leq \phi^{-1}\alpha_k g_k^T d_k + L\phi^{-2}\alpha_k^2||d_k||^2.\end{aligned}$$

By substituting the aforementioned relation in equation (36), along with equations (33) and (35), it can be inferred that

$$\alpha_k \geq \frac{\phi||g_k||^2}{(L+\rho)||d_k||^2} \geq \frac{\phi z^2}{(L+\rho)K_1^2} > 0.$$

Together, this and (28) results in

$$\lim_{k\to\infty} ||d_k|| = 0. \qquad (37)$$

However, by utilising the backward Cauchy-Schwartz inequality on equation (4), it can be concluded that

$$||d_k|| \geq \frac{3}{4}||g_k||.$$

As a result, $\lim_{k\to\infty} ||g_k|| = 0$. This contradicts the assumption in (35), and thus, confirming that indeed $\lim_{k\to\infty} \inf ||g_k|| = 0$. □

3 NUMERICAL RESULTS

In this section, the computational efficiency of the proposed TTBNTC method is examined. A common approach for evaluating the efficiency of a method is to employ the test function. The test function plays a crucial role in validating and comparing newly developed optimization methods. The comparison is made between TTBNTC with BNC and BTC by [16], FR [6], and TTRMIL [20] methods. The FR method is a classical and widely used conjugate gradient approach, making it a benchmark for assessing the performance of new algorithms. On the other hand, the TTRMIL method represents a more recent advancement in optimization techniques, providing a modern reference point. Comparing TTBNTC with both traditional and contemporary methods ensures a robust analysis of its effectiveness and efficiency. The methods were coded in Matlab R2023a and run on a personal computer (PC) with AMD Ryzen 3 processor, 12 GB RAM, 64 bit Windows 10 operating system.

The 48 test functions, encompassing a total of 119 problems are derived from the works of [21], [22], and [23]. These test functions are widely recognized in the field of optimization and provide a comprehensive benchmark for evaluating the performance of the proposed method. The full list of test problems is presented along with the results in Tables 1 and 2. In order to provide evidence of the efficiency of the method, test problems were carried out utilizing low, medium, and high dimensions ranging from 2 to 1,000,000 respectively. All parameters of the BNC, BTC, FR, and TTRMIL approaches are kept the same to enable objective comparison between them. Particularly for the TTBNTC, the parameters were set at $\theta = 0.0001, \sigma = 0.009, \bar{t} = 0.3$ and $\mu = 0.01$.

The comparison of the numerical results is done based on the number of iterations (NOI), and the amount of time spent by the CPU in seconds (CPU). In the conducted experiment, the stopping condition $||g_k|| \leq 10^{-6}$ is considered.

Tables 1 and 2 present the numerical results of all the methods. If the NOI exceeds 10,000 or if this approach never reaches the ideal value, the algorithm will be considered as failed which is denoted by "-" in its place.

Table 1 : The numerical results for TTBNTC, BNC & BTC

No	Functions	TTBNTC		BNC		BTC	
		NOI	CPU	NOI	CPU	NOI	CPU
1	Extended White & Holst	16	1.9919	99	9.4229	30	3.0931
2	Extended White & Holst	16	4.0182	108	18.0254	29	5.9673
3	Extended White & Holst	17	42.3818	-	-	34	60.5813
4	Extended Rosenbrock	31	1.6438	-	-	46	1.894
5	Extended Rosenbrock	31	3.2097	-	-	46	3.6231
6	Extended Rosenbrock	35	36.7358	-	-	46	32.1314
7	Extended Beale	95	0.3251	456	1.1144	47	0.1827

(continued on next page)

Table 1 – *(continued)*

No	Functions	TTBNTC		BNC		BTC	
		NOI	CPU	NOI	CPU	NOI	CPU
8	Extended Beale	106	9.9221	548	55.6499	52	4.9762
9	Extended Beale	107	20.3603	566	91.7813	52	10.0978
10	Raydan 1	19	0.0102	19	0.0032	19	0.0157
11	Raydan 1	47	0.0347	47	0.0095	47	0.0088
12	Raydan 1	68	0.0414	68	0.0332	69	0.0219
13	Extended Tridiagonal 1	14	0.0158	49	0.0058	41	0.0188
14	Extended Tridiagonal 1	22	0.0098	64	0.013	55	0.0292
15	Extended Tridiagonal 1	22	0.0235	79	0.0317	70	0.039
16	Diagonal 4	2	0.0345	2	0.0052	2	0.0203
17	Diagonal 4	2	0.0205	2	0.0189	2	0.1123
18	Diagonal 4	2	0.0952	2	0.0917	2	0.0901
19	Extended Himmelblau	10	0.0444	-	-	21	0.078
20	Extended Himmelblau	11	0.5157	-	-	22	1.1808
21	Extended Himmelblau	11	0.996	-	-	22	2.5221
22	FLETCHCR	39	0.024	-	-	42	0.0467
23	FLETCHCR	37	0.2894	-	-	41	0.3072
24	FLETCHCR	33	1.7382	-	-	43	1.9986
25	NONSCOMP	1027	0.0557	-	-	186	0.0187
26	Extended Penalty Function U52	13	0.0156	58	0.005	18	0.0194
27	Extended Penalty Function U52	15	0.006	18	0.0026	62	0.0186
28	Hager	9	0.0113	9	0.0004561	9	0.0195
29	Hager	11	0.0006607	11	0.0012	11	0.0027
30	Hager	20	0.0095	20	0.0032	20	0.0059
31	Cube	24	0.0037	36	0.0042	21	0.0026
32	Extended Maratos	46	0.0212	-	-	64	0.0505
33	Extended Maratos	46	0.0274	-	-	-	-
34	Extended Maratos	46	0.0311	-	-	-	-
35	Six Hump Camel	6	0.011	21	0.0027	12	0.0019
36	Six Hump Camel	10	0.001	20	0.0011	10	0.0007152
37	Three Hump Camel	10	0.0036	-	-	-	-
38	Three Hump Camel	15	0.0034	-	-	-	-
39	Booth	2	0.0001403	2	0.0001595	2	0.0002708
40	Booth	2	0.0069	2	0.0073	2	0.0003551
41	Trecanni	1	0.0058	1	0.0013	1	0.0045
42	Trecanni	9	0.0007807	29	0.0016	17	0.0009675
43	Zettl	1	0.0001616	2	0.0001674	2	0.0001657
44	Zettl	11	0.0007833	-	-	22	0.007
45	Shallow	3	0.0098	3	0.0067	3	0.0223
46	Shallow	4	0.2162	4	0.2146	4	0.1994
47	Shallow	4	0.3914	4	0.3856	4	0.3451
48	Generalized Quartic	6827	6.2308	-	-	-	-
49	Generalized Quartic	1956	32.0205	-	-	-	-
50	Quadratic QF2	173	0.0499	-	-	98	0.0264
51	Quadratic QF2	453	0.5062	-	-	284	0.4948
52	Leon	15	0.0008656	-	-	39	0.0015
53	Leon	477	0.0393	-	-	-	-
54	Generalized Tridiagonal 1	33	0.0193	50	0.0038	26	0.0188
55	Generalized Tridiagonal 1	32	0.0056	47	0.0044	33	0.0059
56	Generalized Tridiagonal 1	38	0.0293	-	-	34	0.0228
57	Generalized Tridiagonal 2	106	0.0219	-	-	-	-
58	Generalized Tridiagonal 2	104	0.0194	-	-	-	-
59	POWER	10	0.0064	10	0.0005597	10	0.008
60	POWER	66	0.0188	65	0.0096	65	0.014
61	Quadratic QF1	56	0.0331	56	0.0145	56	0.0323
62	Quadratic QF1	187	0.2332	187	0.2576	187	0.229
63	Quadratic QF1	606	4.6706	606	4.6666	606	4.1022
64	Extended Quadratic Penalty QP2	302	0.1713	-	-	88	0.0289
65	Extended Quadratic Penalty QP2	728	1.3224	-	-	-	-
66	Extended Quadratic Penalty QP1	14	0.0077	-	-	33	0.0118
67	Extended Quadratic Penalty QP1	15	0.0024	19	0.001	18	0.0025
68	Quartic	45	0.0224	-	-	134	0.0339
69	Quartic	11	0.0008424	52	0.0054	48	0.0397
70	Matyas	1	0.0002056	1	0.0001583	1	0.0235
71	Matyas	1	0.0002417	1	0.0002029	1	0.000226
72	Colville	17	0.0012	57	0.0063	35	0.02
73	Colville	60	0.0162	-	-	47	0.0021
74	Dixon and Price	93	0.1871	171	0.2559	86	0.1244
75	Dixon and Price	93	0.8054	171	1.6125	86	0.6616
76	Dixon and Price	93	6.7615	171	10.8141	86	5.2843
77	Sphere	1	0.0038	1	0.0033	1	0.013
78	Sphere	1	0.017	1	0.0187	1	0.0167
79	Sphere	1	0.0919	1	0.1066	1	0.0857
80	Sum Squares	178	0.2448	178	0.2613	178	0.2434
81	Sum Squares	578	4.3451	578	4.1716	578	4.6117
82	Sum Squares	1310	44.5184	1310	45.5226	1310	44.8238
83	DENSCHNA	14	0.0639	11	0.0326	15	0.0735
84	DENSCHNA	16	0.2859	13	0.2344	15	0.2604
85	DENSCHNA	16	2.4835	13	1.8789	17	2.3966
86	DENSCHNB	18	0.024	19	0.0062	11	0.0244
87	DENSCHNB	19	0.1201	21	0.1224	12	0.1151
88	DENSCHNB	20	0.8964	10	0.9297	12	0.5461
89	DENSCHNC	22	0.0358	33	0.0143	27	0.0415
90	DENSCHNC	23	0.2729	-	-	28	0.3495
91	DENSCHNC	25	2.4985	-	-	-	-
92	Extended Block-Diagonal BD1	18	0.0209	-	-	61	0.0458

(continued on next page)

Table 1 – *(continued)*

No	Functions	TTBNTC		BNC		BTC	
		NOI	CPU	NOI	CPU	NOI	CPU
93	Extended Block-Diagonal BD1	20	0.1485	-	-	68	0.4008
94	Extended Block-Diagonal BD1	21	0.9693	-	-	70	3.3075
95	HIMMELBG	1	0.0177	1	0.0009967	1	0.0195
96	HIMMELBG	1	0.0036	1	0.0041	1	0.0056
97	HIMMELBG	1	0.0106	1	0.0063	1	0.0094
98	HIMMELBH	6	0.0112	6	0.0005468	6	0.0142
99	HIMMELBH	7	0.0025	7	0.0025	7	0.0023
100	HIMMELBH	7	0.0066	7	0.0024	7	0.003
101	Extended Hiebert	36	0.1851	-	-	-	-
102	Extended Hiebert	37	0.665	-	-	-	-
103	Extended Hiebert	37	5.3553	-	-	-	-
104	Linear Perturbed	54	0.0377	54	0.021	54	0.0412
105	Linear Perturbed	406	1.8376	406	1.9088	406	1.548
106	QUARTICM	32	0.2869	35	0.2777	33	0.2767
107	Zirilli or Aluffi-Pentini's	4	0.0074	4	0.0001531	4	0.0086
108	Zirilli or Aluffi-Pentini's	4	0.0002128	4	0.0001511	4	0.000218
109	Extended Quadratic Penalty QP3	16	0.0181	21	0.0015	26	0.0226
110	Extended Quadratic Penalty QP3	21	0.0097	22	0.0018	-	-
111	Extended Quadratic Penalty QP3	23	0.0296	28	0.0093	5	0.0083
112	DIAG-AUP1	4	0.0036	16	0.0019	18	0.014
113	DIAG-AUP1	4	0.0199	18	0.0361	20	0.0545
114	DIAG-AUP1	4	0.0947	20	0.2287	22	0.3398
115	Strait	17	0.0451	52	0.0981	36	0.059
116	Strait	16	1.6526	18	1.7959	39	3.0996
117	Strait	15	15.1503	18	17.1426	36	28.6742
118	Perturbed Quadratic	2	0.0002076	2	0.0002208	2	0.0029
119	Perturbed Quadratic	2	0.000195	2	0.0001587	2	0.0003016

Table 2 : The numerical results for TTBNTC, FR & TTRMIL

No	Functions	TTBNTC		FR		TTRMIL	
		NOI	CPU	NOI	CPU	NOI	CPU
1	Extended White & Holst	16	1.9919	59	4.801	15	1.5989
2	Extended White & Holst	16	4.0182	58	9.5997	15	2.8469
3	Extended White & Holst	17	42.3818	40	68.9642	15	28.4067
4	Extended Rosenbrock	31	1.6438	73	2.6959	24	1.1387
5	Extended Rosenbrock	31	3.2097	142	9.7789	24	1.8668
6	Extended Rosenbrock	35	36.7358	-	-	24	19.15
7	Extended Beale	95	0.3251	104	0.2689	35	0.1158
8	Extended Beale	106	9.9221	119	9.2643	35	2.9919
9	Extended Beale	107	20.3603	120	19.0151	35	5.8235
10	Raydan 1	19	0.0102	19	0.0113	19	0.0027
11	Raydan 1	47	0.0347	49	0.0099	58	0.008
12	Raydan 1	68	0.0414	68	0.0591	101	0.0785
13	Extended Tridiagonal 1	14	0.0158	50	0.0185	6	0.0014
14	Extended Tridiagonal 1	22	0.0098	79	0.0224	16	0.0057
15	Extended Tridiagonal 1	22	0.0235	95	0.043	19	0.0356
16	Diagonal 4	2	0.0345	2	0.0153	2	0.0158
17	Diagonal 4	2	0.0205	2	0.2785	2	0.0227
18	Diagonal 4	2	0.0952	2	0.0803	2	0.0925
19	Extended Himmelblau	10	0.0444	154	0.2702	16	0.0566
20	Extended Himmelblau	11	0.5157	164	6.4544	17	0.6928
21	Extended Himmelblau	11	0.996	166	13.2781	17	1.4242
22	FLETCHCR	39	0.024	37	0.036	47	0.0368
23	FLETCHCR	37	0.2894	61	0.3206	43	0.2824
24	FLETCHCR	33	1.7382	55	2.0232	42	1.5731
25	NONSCOMP	1027	0.0557	9184	0.3438	1697	0.0512
26	Extended Penalty Function U52	13	0.0156	17	0.0163	18	0.0124
27	Extended Penalty Function U52	15	0.006	271	0.0867	26	0.0041
28	Hager	9	0.0113	9	0.0095	9	0.0075
29	Hager	11	0.0006607	11	0.0008157	12	0.0009293
30	Hager	20	0.0095	20	0.0048	20	0.0041
31	Cube	24	0.0037	1094	0.0554	21	0.0032
32	Extended Maratos	46	0.0212	125	0.0257	103	0.0195
33	Extended Maratos	46	0.0274	-	-	110	0.5922
34	Extended Maratos	46	0.0311	-	-	102	0.0762
35	Six Hump Camel	6	0.011	16	0.01	8	0.0073
36	Six Hump Camel	10	0.001	35	0.0015	10	0.0007031
37	Three Hump Camel	10	0.0036	-	-	-	-
38	Three Hump Camel	15	0.0034	-	-	21	0.001
39	Booth	2	0.0001403	2	0.00017	2	0.0001457
40	Booth	2	0.0069	2	0.014	2	0.0078
41	Trecanni	1	0.0058	1	0.007	1	0.0395
42	Trecanni	9	0.0007807	181	0.0083	13	0.0072
43	Zettl	1	0.0001616	2	0.0001769	2	0.0018
44	Zettl	11	0.0007833	21	0.0191	19	0.0017
45	Shallow	3	0.0098	3	0.0193	3	0.0229
46	Shallow	4	0.2162	4	0.2117	3	0.1409

(continued on next page)

Table 2 – *(continued)*

No	Functions	TTBNTC		FR		TTRMIL	
		NOI	CPU	NOI	CPU	NOI	CPU
47	Shallow	4	0.3914	4	0.3626	3	0.2741
48	Generalized Quartic	6827	6.2308	-	-	-	-
49	Generalized Quartic	1956	32.0205	-	-	-	-
50	Quadratic QF2	173	0.0499	231	0.0707	136	0.1043
51	Quadratic QF2	453	0.5062	759	0.7131	1051	1.2613
52	Leon	15	0.0008656	67	0.0093	60	0.0132
53	Leon	477	0.0393	870	0.1052	156	0.0093
54	Generalized Tridiagonal 1	33	0.0193	21	0.0177	24	0.0154
55	Generalized Tridiagonal 1	32	0.0056	45	0.0054	31	0.003
56	Generalized Tridiagonal 1	38	0.0293	50	0.0297	34	0.0494
57	Generalized Tridiagonal 2	106	0.0219	-	-	-	-
58	Generalized Tridiagonal 2	104	0.0194	-	-	-	-
59	POWER	10	0.0064	10	0.0094	130	0.0176
60	POWER	66	0.0188	66	0.0128	2583	1.0392
61	Quadratic QF1	56	0.0331	56	0.0264	123	0.0605
62	Quadratic QF1	187	0.2332	187	0.2122	940	1.0322
63	Quadratic QF1	606	4.6706	606	3.7859	9407	70.9183
64	Extended Quadratic Penalty QP2	302	0.1713	-	-	341	0.0721
65	Extended Quadratic Penalty QP2	728	1.3224	5648	9.4561	68	0.2453
66	Extended Quadratic Penalty QP1	14	0.0077	18	0.0145	9	0.012
67	Extended Quadratic Penalty QP1	15	0.0024	20	0.0027	17	0.0012
68	Quartic	45	0.0224	583	0.0423	758	0.0566
69	Quartic	11	0.0008424	46	0.0025	137	0.008
70	Matyas	1	0.0002056	1	0.0099	1	0.0081
71	Matyas	1	0.0002417	1	0.0002097	1	0.000239
72	Colville	17	0.0012	183	0.0083	199	0.0154
73	Colville	60	0.0162	168	0.0209	386	0.0259
74	Dixon and Price	93	0.1871	98	0.103	4832	5.8391
75	Dixon and Price	93	0.8054	98	0.6169	-	-
76	Dixon and Price	93	6.7615	98	5.7976	-	-
77	Sphere	1	0.0038	1	0.0121	1	0.0103
78	Sphere	1	0.017	1	0.0158	1	0.0205
79	Sphere	1	0.0919	1	0.0906	1	0.078
80	Sum Squares	178	0.2448	178	0.2064	834	0.9395
81	Sum Squares	578	4.3451	578	3.4987	8507	62.9604
82	Sum Squares	1310	44.5184	1310	35.9132	-	-
83	DENSCHNA	14	0.0639	19	0.0699	9	0.052
84	DENSCHNA	16	0.2859	19	0.2911	9	0.1601
85	DENSCHNA	16	2.4835	21	2.6787	10	1.3422
86	DENSCHNB	18	0.024	132	0.063	11	0.0258
87	DENSCHNB	19	0.1201	145	0.7759	11	0.0659
88	DENSCHNB	20	0.8964	155	5.7595	11	0.4505
89	DENSCHNC	22	0.0358	12	0.029	19	0.0403
90	DENSCHNC	23	0.2729	75	6.9901	-	-
91	DENSCHNC	25	2.4985	-	-	-	-
92	Extended Block-Diagonal BD1	18	0.0209	25	0.0288	16	0.0268
93	Extended Block-Diagonal BD1	20	0.1485	28	0.1556	18	0.1281
94	Extended Block-Diagonal BD1	21	0.9693	29	1.2047	19	0.9095
95	HIMMELBG	1	0.0177	1	0.0177	1	0.0176
96	HIMMELBG	1	0.0036	1	0.0037	1	0.0362
97	HIMMELBG	1	0.0106	1	0.0065	1	0.0211
98	HIMMELBH	6	0.0112	7	0.014	5	0.0122
99	HIMMELBH	7	0.0025	7	0.002	5	0.0211
100	HIMMELBH	7	0.0066	7	0.0043	5	0.0074
101	Extended Hiebert	36	0.1851	8531	9.7725	38	0.0983
102	Extended Hiebert	37	0.665	6894	47.7664	39	0.458
103	Extended Hiebert	37	5.3553	160	11.0264	39	3.7912
104	Linear Perturbed	54	0.0377	54	0.0308	111	0.0724
105	Linear Perturbed	406	1.8376	406	1.4496	4259	15.0115
106	QUARTICM	32	0.2869	35	0.2633	31	0.282
107	Zirilli or Aluffi-Pentini's	4	0.0074	4	0.012	4	0.0111
108	Zirilli or Aluffi-Pentini's	4	0.0002128	4	0.000114	4	0.0001648
109	Extended Quadratic Penalty QP3	16	0.0181	14	0.0193	15	0.0154
110	Extended Quadratic Penalty QP3	21	0.0097	-	-	19	0.002
111	Extended Quadratic Penalty QP3	23	0.0296	-	-	36	0.0103
112	DIAG-AUP1	4	0.0036	16	0.0134	22	0.0155
113	DIAG-AUP1	4	0.0199	19	0.0501	25	0.0668
114	DIAG-AUP1	4	0.0947	20	0.2365	26	0.2717
115	Strait	17	0.0451	25	0.0688	31	0.0961
116	Strait	16	1.6526	32	2.5726	49	3.6429
117	Strait	15	15.1503	27	22.3355	49	37.0309
118	Perturbed Quadratic	2	0.0002076	2	0.0002082	2	0.0008204
119	Perturbed Quadratic	2	0.000195	2	0.0002325	2	0.0002161

In accordance with Tables 1 and 2, the performance profile of each approach can be constructed. Figures 1 and 2 provide the findings, with each figure representing a performance profile based on NOI, and CPU times. In the performance profile plot, the uppermost curve represents the method that successfully resolved the most number of problems within a specified time factor of the optimal time. The left side of the plot illustrates the proportion of test problems in which a particular approach demonstrates the fastest convergence. Conversely, the right side of the plot displays the percentage of test problems that are effectively resolved by each of the methods.

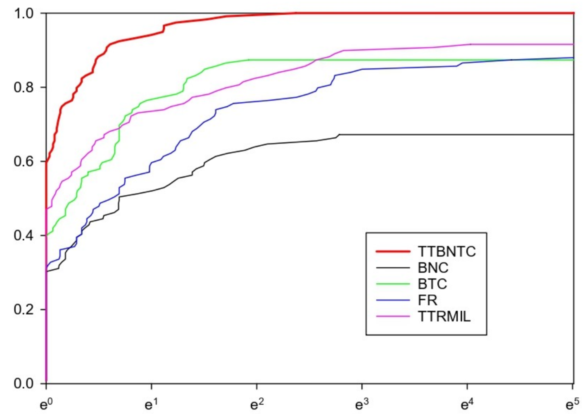

Figure 1 : Performance profile in terms of the number of iterations

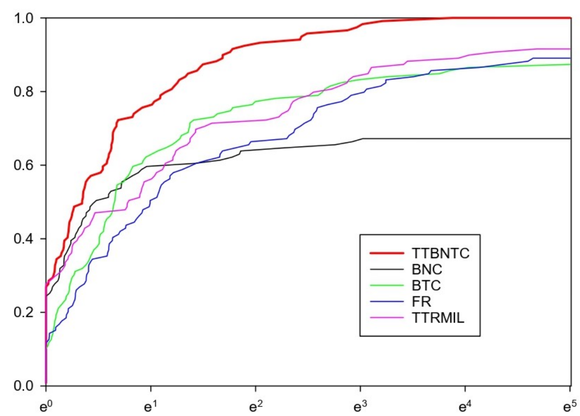

Figure 2 : Performance profile in terms of CPU time

Upon comparing all approaches with respect to the number of iterations, it is evident from Figures 1 and 2 that TTBNTC outperformed the other ways in 119 problems, as it consistently attained the fewest number of iterations in these instances. TTBNTC is able to solve 100% (119 out of 119) of the problems. TTRMIL, on the other hand, solves 92% of the problems (109 out of 119), whereas FR solves 89% (106 out of 119), BTC solves 84% (104 out of 119), and BNC solves 67% (80 out of 119).

Figure 1 and 2 show the TTBNTC curves exhibit rapid convergence, whereas the BNC, BTC, FR, and TTRMIL methods demonstrate a gradual approach towards a value of 1.0. When evaluating the TTBNTC method in comparison to the BNC, BTC, FR, and TTRMIL methods using the CPU time measure, it becomes evident that TTBNTC outperforms the other approaches in terms of success rate and robustness. It can be observed from Figure 2 that the BTC, BNC, FR and TTRMIL curves exhibit a high degree of proximity, indicating inferior performance compared to the TTBNTC curve.

There exists a significant level of competition between TTBNTC and the other methods, however, it has been observed that the other approaches demonstrate comparatively slower performance in certain instances. Hence, TTBNTC demonstrates superior performance, effectively resolving approximately 100% of the 119 test problems with the lowest CPU time.

Last but not least, it is evident from the findings shown in Tables 1 and 2 as well as Figures 1 and 2 that TTBNTC achieves the best performance in terms of generating the optimal search direction and achieving the most favourable steplength when considering the average outcomes.

4 CONCLUSIONS

In this study, an effective conjugate gradient approach was introduced. Furthermore, aside from satisfying the sufficient descent condition, the suggested technique exhibits global convergence. The key contributions of this paper include the integration of BNC and BTC parameters within a hybrid framework, inspired by the memoryless BFGS quasi-Newton method. This combination ensures both the descent requirement and the potential for global convergence. Based on numerical results, the proposed method demonstrates outstanding performance and exhibits superiority over existing methods such as BNC, BTC, FR, and TTRMIL. These results highlight the practical applicability and efficiency of the TTBNTC method in solving unconstrained optimization problems, thereby advancing the field of optimization techniques.

ACKNOWLEDGEMENT

The authors would like to extend their heartfelt gratitude to the reviewers and editors for their intelligent comments and suggestions, which greatly improved the quality of this work.

REFERENCES

[1] M. R. Hestenes, E. Stiefel *et al.*, *Methods of conjugate gradients for solving linear systems.* NBS Washington, DC, 1952, vol. 49, no. 1.

[2] E. Polak and G. Ribiere, "Note sur la convergence de méthodes de directions conjuguées," *Revue française d'informatique et de recherche opérationnelle. Série rouge*, vol. 3, no. 16, pp. 35–43, 1969.

[3] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.

[4] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, part 1: theory," *Journal of Optimization Theory and Applications*, vol. 69, pp. 129–137, 1991.

[5] Y.-H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.

[6] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The computer journal*, vol. 7, no. 2, pp. 149–154, 1964.

[7] R. Fletcher, *Practical methods of optimization.* John Wiley & Sons, 2000.

[8] N. Andrei, "A simple three-term conjugate gradient algorithm for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 241, pp. 19–29, 2013.

[9] J. K. Liu and S. J. Li, "New hybrid conjugate gradient method for unconstrained optimization," *Applied Mathematics and Computation*, vol. 245, pp. 36–43, 2014.

[10] J. Jian, L. Han, and X. Jiang, "A hybrid conjugate gradient method with descent property for unconstrained optimization," *Applied Mathematical Modelling*, vol. 39, no. 3-4, pp. 1281–1290, 2015.

[11] M. Li, "A three term Polak-Ribiére-Polyak conjugate gradient method close to the memoryless BFGS quasi-Newton method," *Journal of Industrial & Management Optimization*, vol. 16, no. 1, pp. 245–260, 2018.

[12] ——, "A modified Hestense–Stiefel conjugate gradient method close to the memoryless BFGS quasi-Newton method," *Optimization Methods and Software*, vol. 33, no. 2, pp. 336–353, 2018.

[13] Z. Liu, H. Liu, and Y.-H. Dai, "An improved Dai–Kou conjugate gradient algorithm for unconstrained optimization," *Computational Optimization and Applications*, vol. 75, no. 1, pp. 145–167, 2020.

[14] H. Iiduka and Y. Narushima, "Conjugate gradient methods using value of objective function for unconstrained optimization," *Optimization Letters*, vol. 6, pp. 941–955, 2012.

[15] B. A. Hassan, "A new formula for conjugate parameter computation based on the quadratic model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 3, no. 3, pp. 954–961, 2019.

[16] B. A. Hassan and H. Alashoor, "A new type coefficient conjugate on the gradient methods for impulse noise removal in images," *European Journal of Pure and Applied Mathematics*, vol. 15, no. 4, pp. 2043–2053, 2022.

[17] J. Nocedal, "Updating quasi-Newton matrices with limited storage," *Mathematics of Computation*, vol. 35, no. 151, pp. 773–782, 1980.

[18] D. F. Shanno, "Conjugate gradient methods with inexact searches," *Mathematics of Opera-*

tions Research, vol. 3, no. 3, pp. 244–256, 1978.

[19] J. Deepho, A. B. Abubakar, M. Malik, and I. K. Argyros, "Solving unconstrained optimization problems via hybrid CD-DY conjugate gradient methods with applications," *Journal of Computational and Applied Mathematics*, vol. 405, p. 113823, 2022.

[20] J. Liu, Y. Feng, and L. Zou, "Some three-term conjugate gradient methods with the inexact line search condition," *Calcolo*, vol. 55, pp. 1–16, 2018.

[21] N. Andrei *et al.*, *Nonlinear conjugate gradient methods for unconstrained optimization*. Springer International Publishing, 2020.

[22] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.

[23] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "Testing unconstrained optimization software," *ACM Transactions on Mathematical Software (TOMS)*, vol. 7, no. 1, pp. 17–41, 1981.