

Research Article

A Modified Bat Algorithm with Conjugate Gradient Method for Global Optimization

Huda I. Ahmed ¹, Eman T. Hamed ¹ and Hamsa Th. Saeed Chilmeran²

¹Department of Operation Researches and Intelligent Techniques, College of Computers Sciences and Mathematics, University of Mosul, Mosul, Iraq

²Department of Mathematics, College of Computers Sciences and Mathematics, University of Mosul, Mosul, Iraq

Correspondence should be addressed to Eman T. Hamed; emantarik84@gmail.com

Received 4 March 2020; Revised 17 April 2020; Accepted 18 May 2020; Published 4 June 2020

Academic Editor: Birendra Nath Mandal

Copyright © 2020 Huda I. Ahmed et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Metaheuristic algorithms are used to solve many optimization problems. Firefly algorithm, particle swarm improvement, harmonic search, and bat algorithm are used as search algorithms to find the optimal solution to the problem field. In this paper, we have investigated and analyzed a new scaled conjugate gradient algorithm and its implementation, based on the exact Wolfe line search conditions and the restart Powell criterion. The new spectral conjugate gradient algorithm is a modification of the Birgin and Martínez method, a manner to overcome the lack of positive definiteness of the matrix defining the search direction. The preliminary computational results for a set of 30 unconstrained optimization test problems show that this new spectral conjugate gradient outperforms a standard conjugate gradient in this field and we have applied the newly proposed spectral conjugate gradient algorithm in bat algorithm to reach the lowest possible goal of bat algorithm. The newly proposed approach, namely, the directional bat algorithm (CG-BAT), has been then tested using several standard and nonstandard benchmarks from the CEC'2005 benchmark suite with five other algorithms and has been then tested using nonparametric statistical tests and the statistical test results show the superiority of the directional bat algorithm, and also we have adopted the performance profiles given by Dolan and More which show the superiority of the new algorithm (CG-BAT).

1. Introduction

In 2010, Yang proposed a new optimization algorithm, namely, bat algorithm (BA), based on swarm intelligence and the inspiration from observing the bats. Although the original BA presents superior results in the experiments than PSO, we notice that the performance and the accuracy of the original BA still have the capacity to present better. The algorithm exploits the so-called echolocation of bats.

Bats use sonar echoes to detect and avoid obstacles. It is generally known that sound pulses are transformed to frequency which reflects from obstacle. Bats can use time delay from emission to reflection and use it for navigation. They typically emit short loud, sound impulses. The pulse rate is usually defined as 10 to 20 times per second. After hitting and reflecting, the bats transform their own pulse

into useful information to gauge how far away the prey is. The bats are using wavelengths that vary from the range 0.7 to 17 mm or inbound frequencies of 20–500 kHz. To implement the algorithm, the pulse frequency and rate have to be defined. The pulse rate can be simply determined in the range from 0 to 1, where 0 means that there is no emission and 1 means that bats are emitting maximum [1].

The bat-inspired algorithm is a recent swarm-based intelligent system which mimics the echolocation system of microbats. In the bat-inspired algorithm, the bats randomly fly around the best bat locations found during the search so as to improve their hunting of prey. In practice, one bat location from a set of best bat locations is selected. Thereafter, that best bat location is used by local search with a random walk strategy to inform other bats about the prey location. This selection mechanism can be improved using

other natural selection mechanisms adopted from other advanced algorithms such as genetic algorithm. Therefore, six selection mechanisms are studied to choose the best bat location: global-best, tournament, proportional, linear rank, exponential rank, and random. Consequently, six versions of the bat-inspired algorithm are proposed and studied which are global-best bat-inspired algorithm (GBA), tournament bat-inspired algorithm (TBA), proportional bat-inspired algorithm (PBA), linear rank bat-inspired algorithm (LBA), exponential rank bat-inspired algorithm (EBA), and random bat-inspired algorithm (RBA). Using two sets of global optimization functions, the bat-inspired versions are evaluated and the sensitivity analyses of each version to its parameters studied [2]. A success of an algorithm always depends on well balanced of these components. The aim of this study is to improve the performance of the standard bat algorithm by increasing its exploration and exploitation abilities along the main line of the BA. In this paper, two improvement strategies are presented. The first improvement strategy is the development of a spectral conjugate gradient technique, which can be used to guide the research process, and the second improvement strategy is to improve the bat algorithm using the conjugate gradient method to arrive at the best solution for the current iteration, which can be used to enhance the ability to local search. The newly proposed optimally directional bat algorithm (CG-BAT) will be tested on several benchmark problems chosen from the well-known CEC'2005 benchmark set and compared with several other swarm and evolutionary algorithms. Therefore, this study is organized as follows. A new scalar in a spectral conjugate gradient is described in Section 2. A global convergence is described in Section 3, the standard bat algorithm is presented in Section 4. Then, the enhanced bat algorithm is presented in Section 5. Finally, the results of the numerical experiments are presented in Section 6, followed by the conclusions in Section 7. Too much exploration but too little exploitation may cause difficulties that algorithm converges towards optimal solutions. The conjugate gradient technique could be a helpful procedure to search out the minimum value of any nonlinear function to find optimal solutions:

$$\underset{x \in R^n}{\text{minimize}} f(x), \quad (1)$$

where $f: R^n \rightarrow R$ is a real-valued function. The numerical formula is given by

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

where α_k is a step length to be computed by a line search procedure [3]. The search direction $d_k^T = -g_k$ is outlined as follows:

$$d_{k+1} = \begin{cases} -g_{k+1}, & \text{for } k = 0, \\ -g_{k+1} + \beta_k d_k, & \text{for } k \geq 1, \end{cases} \quad (3)$$

and $g_k = \nabla f(x_k)$ is gradient and β_k is a parameter of conjugacy condition. Some famed formulas of this parameter are outlined as follows:

$$\begin{aligned} \beta_k^{\text{HS}} &= \frac{g_{k+1}^T y_k}{d_k^T y_k}, \\ \beta_k^{\text{PR}} &= \frac{g_{k+1}^T y_k}{g_k^T g_k}, \\ \beta_k^{\text{FR}} &= \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}, \\ \beta_k^{\text{DY}} &= \frac{g_{k+1}^T g_{k+1}}{d_k^T y_k}, \\ \beta_k^{\text{HS}} &= \frac{g_{k+1}^T y_k}{d_k^T y_k}, \\ \beta_k^{\text{PR}} &= \frac{g_{k+1}^T y_k}{g_k^T g_k}, \\ \beta_k^{\text{FR}} &= \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}, \\ \beta_k^{\text{DY}} &= \frac{g_{k+1}^T g_{k+1}}{d_k^T y_k}, \end{aligned} \quad (4)$$

where $y_k = g_{k+1} - g_k$, which are referred to as Hestenes–Stiefel (HS) [4], Polak–Ribière (PR) [5], Fletcher–Reeves (FR) [6], and Dai–Yuan (DY) [7] severally. Many authors have studied the convergence of the on top of formulas for years [8–11].

To prove the convergence analysis of the conjugate gradient technique, the following weak Wolfe conditions are used:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \widehat{\delta} \alpha_k \nabla f(x_k)^T d_k, \quad (5)$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \widehat{\sigma} \nabla f(x_k)^T d_k. \quad (6)$$

Used the strong Wolfe conditions consist of (5) and

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\widehat{\sigma} g_k^T d_k. \quad (7)$$

The constants are within the period $0 < \widehat{\delta} < \widehat{\sigma} < 1$, and additional details are found in [3]. Well, the sufficient descent property is defined as follows:

$$d_{k+1}^T g_{k+1} \leq -\widehat{c} \|g_{k+1}\|^2, \quad (8)$$

where $\|\cdot\|$ denotes the Euclidean norm, provided that c is any positive constant [7].

2. A New Scalar in Spectral CG Method (ξ_{k+1})

Birgin and Martínez (SS) [12] instructed a spectral conjugate gradient technique outlined by

$$d_{k+1} = -g_{k+1} + \beta_k s_k, \quad (9)$$

where $s_k = \alpha_k d_k$.

The parameter β_k has the following form:

$$\beta_k = \frac{(\xi_{k+1}y_k - s_k)^T g_{k+1}}{s_k^T y_k}, \tag{10}$$

where the spectral ξ_{k+1} in [13] is determined by using the following equation:

$$\xi_{k+1} = \frac{s_k^T s_k}{s_k^T y_k}. \tag{11}$$

In this section, we will derive a new spectral CG method as follows:

$$d_{k+1} = -\tilde{\rho}_{k+1} H_{k+1} g_{k+1}. \tag{12}$$

The matrix H_{k+1} is asymmetrical and positive definite and the scalar $\tilde{\rho}_{k+1}$ is defined by Al-Bayati and Salah [14] as $\tilde{\rho}_{k+1} = (y_k^T s_k / g_k^T H_k g_k)$.

By equating (9) and (12), we get

$$-\frac{y_k^T s_k}{g_k^T H_k g_k} H_{k+1} g_{k+1} = -g_{k+1} + \frac{(\xi_{k+1}y_k - s_k)^T g_{k+1}}{s_k^T y_k} s_k. \tag{13}$$

Multiplying both sides of (13) by y_k^T , we get

$$-\frac{y_k^T s_k}{g_k^T H_k g_k} y_k^T H_{k+1} g_{k+1} = -y_k^T g_{k+1} + \frac{(\xi_{k+1}y_k - s_k)^T g_{k+1}}{s_k^T y_k} y_k^T s_k,$$

$$\frac{y_k^T s_k}{g_k^T (-H_k g_k)} y_k^T H_{k+1} g_{k+1} = -y_k^T g_{k+1} + \xi_{k+1} y_k^T g_{k+1} - s_k^T g_{k+1}. \tag{14}$$

$$d_{k+1} = -g_{k+1} + \frac{((\alpha_k y_k^T s_k / g_k^T s_k y_k^T g_{k+1}) s_k^T g_{k+1} + 1 + (s_k^T g_{k+1} / y_k^T g_{k+1}) y_k - s_k)^T g_{k+1}}{s_k^T y_k} s_k. \tag{17}$$

Theorem 1. *Let the line search α_k in (2) satisfies the strong Wolfe condition, then the new search direction given by (17) is a sufficient descent direction.*

Proof. Under some algebraic operations, the direction of (17) can be written as follows:

$$d_{k+1} = -g_{k+1} + \left(\frac{\alpha_k s_k^T g_{k+1}}{g_k^T s_k} + \frac{y_k^T g_{k+1}}{s_k^T y_k} \right) s_k. \tag{18}$$

Now, multiplying both sides of (18) by $(g_{k+1} / \|g_{k+1}\|^2)$, then we get

$$\begin{aligned} \frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1 &= \left(\frac{\alpha_k s_k^T g_{k+1}}{g_k^T s_k} + \frac{y_k^T g_{k+1}}{s_k^T y_k} \right) \frac{s_k^T g_{k+1}}{\|g_{k+1}\|^2}, \\ \frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1 &= \frac{\alpha_k (s_k^T g_{k+1})}{g_k^T s_k} \left(\frac{s_k^T g_{k+1}}{\|g_{k+1}\|^2} \right) + \frac{y_k^T g_{k+1}}{s_k^T y_k} \left(\frac{s_k^T g_{k+1}}{\|g_{k+1}\|^2} \right), \\ \frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1 &= \frac{\alpha_k (s_k^T g_{k+1})^2}{\|g_{k+1}\|^2 g_k^T s_k} + \frac{y_k^T g_{k+1}}{s_k^T y_k} \frac{s_k^T g_{k+1}}{\|g_{k+1}\|^2}. \end{aligned} \tag{19}$$

Since $d_k = -H_k g_k$ and $y_k^T H_{k+1} = s_k^T$, we get

$$\frac{y_k^T s_k}{g_k^T d_k} s_k^T g_{k+1} = -y_k^T g_{k+1} + \xi_{k+1} y_k^T g_{k+1} - s_k^T g_{k+1}. \tag{15}$$

Since $s_k = \alpha_k d_k$,

$$\frac{\alpha_k y_k^T s_k}{g_k^T s_k} s_k^T g_{k+1} = -y_k^T g_{k+1} + \xi_{k+1} y_k^T g_{k+1} - s_k^T g_{k+1}, \tag{16}$$

$$\xi_{k+1} = \frac{\alpha_k y_k^T s_k}{g_k^T s_k y_k^T g_{k+1}} s_k^T g_{k+1} + 1 + \frac{s_k^T g_{k+1}}{y_k^T g_{k+1}}.$$

If we use exact line search, then the new scalar ξ_{k+1} is equal to one.

The new direction is defined by the following equation:

Since $s_k^T g_{k+1} \leq s_k^T y_k$ and $y_k^T g_{k+1} \leq \|y_k\| \cdot \|g_{k+1}\|$, we get

$$\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1 \leq \frac{\alpha_k (s_k^T y_k)^2}{\|g_{k+1}\|^2 g_k^T s_k} + \frac{\|y_k\| \|g_{k+1}\|}{s_k^T y_k} \frac{s_k^T y_k}{\|g_{k+1}\|^2} \tag{20}$$

$$\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1 \leq \frac{\alpha_k (s_k^T y_k)^2}{\|g_{k+1}\|^2 g_k^T s_k} + \frac{\|y_k\|}{\|g_{k+1}\|}.$$

Since $(1/s_k^T g_k) \leq (s_k^T y_k / (1 - \alpha_k))$,

$$\begin{aligned} \frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1 &\leq \frac{\alpha_k (s_k^T y_k)^2}{\|g_{k+1}\|^2 (s_k^T y_k / (1 - \alpha_k))} + \frac{\|y_k\|}{\|g_{k+1}\|}, \\ \frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1 &\leq -\frac{\alpha_k (s_k^T y_k) (1 - \alpha_k)}{\|g_{k+1}\|^2} + \frac{\|y_k\|}{\|g_{k+1}\|} \end{aligned} \tag{21}$$

$$\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1 \leq \frac{\|y_k\|}{\|g_{k+1}\|}.$$

Let $\bar{v} = (\|y_k\|/\|g_{k+1}\|)$ (where \bar{v} is the positive constant), then

$$\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1 \leq \bar{v}, \quad (22)$$

$$d_{k+1}^T g_{k+1} \leq -(1 - \bar{v})\|g_{k+1}\|^2. \quad \square$$

3. Global Convergence

In this section, the subsequent assumption is usually used in proving the global conjugate gradient methods.

Assumption 1 (see [15])

- (i) The level set $S = \{x : f(x) \leq f(x_0)\}$ is bounded, that is, there exists a constant $z > 0$, such as $\|x\| \leq z, \forall x \in S$.
- (ii) In neighborhood N of S , the function f is continuously differentiable and its gradient is Lipschitz continuous, i.e., there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in N. \quad (23)$$

Below the assumptions (i) and (ii) on f , we can deduce that there exists $\bar{\gamma} > 0$ such as the following equations:

$$\|\nabla f(x)\| \leq \bar{\gamma}. \quad (24)$$

$$(g(x) - g(y))(x - y) \geq \mu\|x - y\|^2, \quad \forall x, y \in S, \mu > 0. \quad (25)$$

Lemma 1 (see [16]). *Assume that Assumption 1 holds and suppose that, for any conjugate gradient method, d_{k+1} is a descent direction and the step size α_k satisfies conditions (5) and (7) if*

$$\sum_{k \geq 1} \frac{1}{\|d_{k+1}\|^2} = \infty. \quad (26)$$

Then

$$\lim_{k \rightarrow \infty} (\inf \|g_k\|) = 0. \quad (27)$$

Theorem 2. *Suppose that Assumption 1 holds, and the direction d_{k+1} defined by (17) is descent and α_k is computed using (5) and (7), then*

$$\lim_{k \rightarrow \infty} (\inf \|g_k\|) = 0. \quad (28)$$

Proof. By using some algebraic operations of (17) and taking the absolute value, we get

$$\|d_{k+1}\| \leq \|g_{k+1}\| + \left\| \frac{\alpha_k s_k^T g_{k+1}}{g_k^T s_k} + \frac{\gamma_k^T g_{k+1}}{s_k^T y_k} \right\| \|s_k\|, \quad (29)$$

$$\beta_k = \frac{\alpha_k s_k^T g_{k+1}}{g_k^T s_k} + \frac{\gamma_k^T g_{k+1}}{s_k^T y_k},$$

$$|\beta_k| \leq \left| \frac{\alpha_k s_k^T g_{k+1}}{g_k^T s_k} + \frac{\gamma_k^T g_{k+1}}{s_k^T y_k} \right|,$$

$$|\beta_k| \leq \left| \frac{\alpha_k s_k^T g_{k+1}}{g_k^T s_k} \right| + \left| \frac{\gamma_k^T g_{k+1}}{s_k^T y_k} \right|.$$

Since $s_k^T g_{k+1} \leq s_k^T y_k$ and $\gamma_k^T g_{k+1} \leq \|y_k\| \cdot \|g_{k+1}\|$, we get

$$|\beta_k| = \left| \frac{\alpha_k s_k^T y_k}{g_k^T s_k} \right| + \left| \frac{\|y_k\| \|g_{k+1}\|}{s_k^T y_k} \right|. \quad (30)$$

Since $d_k^T = -g_k$,

$$|\beta_k| \leq \frac{\alpha_k s_k^T y_k}{\alpha_k \|g_k\|} + \frac{\|y\| \|g_{k+1}\|}{s_k^T y_k}. \quad (31)$$

Since $\mu \|s_k\| \leq s_k^T y_k \leq L \|s_k\|$, we get

$$|\beta_k| \leq \frac{L \|s_k\|}{\|g_k\|} + \frac{\|y\| \|g_{k+1}\|}{\mu_k \|s_k\|} = \tilde{\vartheta}, \quad \|d_{k+1}\| \leq \|g_{k+1}\| + \tilde{\vartheta} \|s_k\| = \phi$$

$$\sum_{k \geq 1} \frac{1}{\|d_{k+1}\|^2} \geq \frac{1}{\phi} \sum_{k \geq 1} 1 = \infty, \quad (32)$$

that is, $\lim_{k \rightarrow \infty} \|g_k\| = 0$, the proof is complete. \square

4. Standard Bat Algorithm

The bat algorithm proposed by Yang [17] is an intelligent optimization algorithm inspired by the echolocation behavior of bats. When flying and hunting, bats emit some short, ultrasonic pulses to the environment and listen to their echoes. Studies show that the information from the echoes will enable bats to build a precise image of their surroundings and determine precisely the distance, shapes, and prey's location. The capability of such echolocation of microbats is fascinating, as these bats can find their prey and discriminate against different types of insects even in complete darkness [17]. The earlier studies showed that BA can solve unconstrained optimization problems with much more efficiency and robustness compared to GA and PSO [18, 19].

The used idealized rules in bat algorithm are as follows:

- (a) All bats use echolocation to sense distance and the location of a bat x_i is encoded as a solution to an optimization problem under consideration.
- (b) Bats fly randomly with velocity v_i at position x_i with a varying frequency (from a minimum f_{\min} to a maximum frequency f_{\max}) or a varying wavelength λ and loudness A to search for prey. They can automatically adjust the wavelengths (or frequencies) of

their emitted pulses and the rate of pulse emission r depending on the proximity of the target.

- (c) Loudness varies from a large positive value A_0 to a minimum constant value A_{\min} [17].

For each bat (i), its position (x_i) and velocity (v_i) in a d -dimensional search space should be defined. x_i and v_i should be subsequently updated during the iterations. The rules for updating the position and velocities of a virtual bat (i) are given as in [17]:

$$f_k = f_{\min} + (f_{\max} - f_{\min})\text{rand}, \quad (33)$$

$$v_k^{t+1} = v_k^t + (x^* - x_k^t)f_k, \quad (34)$$

$$x_k^{t+1} = x_k^t + v_k^{t+1}, \quad (35)$$

where $\text{rand} \in [0, 1]$ is a random vector drawn from a uniform distribution. Here, x^* is the current global best location (solution) which is located after comparing all solutions among all the n bats. A new solution for each bat is generated locally using random walk given by

$$X_{\text{new}} = x + \varepsilon \langle A^{k+1} \rangle, \quad (36)$$

where $\varepsilon \in [-1, 1]$ is a random number, while $\langle A_k^{t+1} \rangle$ is the average loudness of all the bats at this time step.

The loudness A_k and the rate of pulses emission r_k are updated as the iterations proceed. The loudness decreases and the pulse rate increases as the bat gets closer to its prey. The equation for updating the loudness and the pulse rate is given by

$$A_k^{t+1} = \omega A_k^t, \quad (37)$$

$$r^{t+1} = r_0 1 - \exp(-\eta), \quad (38)$$

where $0 < \omega < 1$ and $\eta > 0$ are constants. As $t \rightarrow \infty$, we have $A_k^t \rightarrow 0$ and $r_k^t \rightarrow r_k^0$.

The initial loudness A_0 can typically be $A_0 \in [1, 2]$, while the initial emission rate $r^0 \in [0, 1]$.

The basic steps of the standard bat algorithm are summarized in the pseudocode as shown in Algorithm 1.

5. Enhanced Bat Algorithm

This paper attempts to improve the bat algorithm from a different perspective from the previous improvements by hybridizing the bat method using optimization methods, by using the optimal size of the cubic step and the optimal search direction for the synchronous gradient feature of the optimal search direction for echo detection. First, local movements can be improved by controlling the optimum step sizes, while the second bat movement should be directed by other bats and the best local moves toward optimal movement. More specifically, two different adjustments will be made to improve the efficiency of the bat algorithm.

5.1. The First Modification (Optimal Step Size). The first modification concerns local search mechanisms: in standard bats, they are allowed to move from their current locations to

new random locations using local random walk. In modified bats, they are allowed to switch from their current locations to new locations optimally using local optimal walking, as we adjust this step to the optimal size using one of the optimization methods called optimal step size (α_k) when the step length is calculated by performing a line search [1].

5.2. The Second Modification Using New Spectral Conjugate Gradient Method. A bat emits two pulses in two different directions, one to the direction of the bat with the best position (the best solution is steepest descent) and the other to the direction of the new conjugate gradient bat. From the echoes, the bat can know if the food exists around these two bats or not. The best position is determined by the objective fitness, while, around the optimally selected bat, it depends on its fitness value. If it has a better fitness value as the actual bat, then the food is considered to exist; otherwise, there is not a food source in the neighborhood. If the food is confirmed to exist around the two bats (Choice 1), the current bat moves to a direction at the surrounding neighborhood of the two bats where the food is supposed to be plenty. If not (Choice 2), it moves toward the best bat.

The mathematical formulas of the bats' movements are thus given by

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k(-g_k), \quad \text{for } k = 0, \\ x_{k+1} &= x_k + \alpha_k(-g_{k+1} + \beta_k d_k), \quad \text{for } k \geq 1. \end{aligned} \quad (39)$$

The directions of the movement generated by equation (17) are directed towards the bat with the best position. This mechanism allows the BA to exploit more around the best position; however, if the best bat is not near the global optimality, there is a risk that the solutions generated by such moves could be trapped in local optima. The new proposed movement in equation (17) has the ability to diversify the movement directions which can enhance the exploration capability, especially at the different repetition stages especially at the initial stages of iterations, and can thus avoid premature convergence. Furthermore, when it approaches the end of the iteration process, the bats tend to get around the best bats with stronger exploitability which in turn can reduce the distances between them and thus enhance the speed of convergence which gives stability to the algorithm. The new algorithm CG-BAT is illustrated by presenting the algorithm and flowchart as follows.

5.2.1. CG-BAT Algorithm. In this section, we develop the movement of bat algorithm to reach the goal by using the new direction which is defined in equation (17).

- (1) objective function $f(x) = (x_1, x_2, \dots, x_d)^T$
- (2) initialize the bat population x_k and $v_k = d_k = -\nabla f(x) = -g_k$ for $k = 1, \dots, n$
- (3) Define pulse frequency f_k at x_k .
- (4) Initialize pulse rates r_k and the loudness A_k .
- (5) While ($t \leq t_{\max}$).
- (6) Adjust frequency equation (1)

- (1) define objective function $f(x) = (x_1, x_2, \dots, x_d)^T$
- (2) initialize the bat population x_k and for $k = 1, \dots, n$
- (3) Define pulse frequency f_k at x_k .
- (4) Initialize pulse rates r_k and the loudness A_k .
- (5) While ($t \leq t_{\max}$).
- (6) Adjust frequency equation (33)
- (7) Update velocities equation (34)
- (8) Update locations/solutions equation (35)
- (9) if ($\text{rand} > r_k$)
- (10) Select a solution among the best solutions
- (11) Generate a local solution around the selected best solution equation (36)
- (12) end if
- (13) Generate a new solution by flying randomly
- (14) if ($\text{rand} < A_k$ & $F(x_k) < F(x^*)$)
- (15) Accept the new solutions
- (16) Increase r_k equation (37)
- (17) Reduce A_k equation (38)
- (18) end if
- (19) Rank the bats and find the current best x^*
- (20) end while
- (21) Output results for post-processing

ALGORITHM 1: The standard bat algorithm.

- (7) Update locations/solutions equation (2)
- (8) Update velocities equation (3)
- (9) if ($d_k^T \nabla f(x_k) \leq 0$)
- (10) Generate new search movement using equation (17)
- (11) end if.
- (12) Generate new solution by a flying optimally step length using equation (39).
- (13) if ($f(x_k) < f(x^*)$).
- (14) Accept the new solutions
- (15) Increase r_k equation (37)
- (16) Reduce A_k equation (38)
- (17) end if
- (18) Rank the bats and find the current best x^*
- (19) end while
- (20) Output results for postprocessing

5.2.2. *CG-BAT Flowchart.* In this section, we describe the movement of CG-BAT algorithm to reach the goal by using the flowchart of new search direction which is defined in equation (17) (Figure 1).

6. Experimental Results and Comparisons

To prove the efficiency performance of all newly proposed algorithms, two comparison experiments have been conducted. The first is a comparison between the new spectral conjugate gradient and the standard algorithms in this field, and the second experiment is a comparison between the new bat algorithm (CG-BAT) and cuckoo search, firefly search, and practical swarm.

6.1. *Experimental Results and Comparisons in New CG.* In this section, we have reported some numerical experiments that are performed on a set of 30 unconstrained optimization test problems to analyze the efficiency of ξ_{k+1} . Detail of these test problems, with their given initial points. The termination criterion used in our experiments is $\|g_k\| \leq 10^{-6}$, where $\hat{\delta} = 0.001$ and $\hat{\sigma} = 0.1$. In our comparisons below, we employ the following algorithms:

- (i) SS: scalar in spectral in Birgin and Martínez algorithm with the Wolfe line search
- (ii) HS: Hestenes–Stiefel algorithm with the Wolfe line search
- (iii) New: new algorithm using equation (17) with the Wolfe line search

Table 1 shows the numerical computations of these newly proposed CG algorithms against other well-known CG-algorithms to check their performance and we have used the following well-known measures or tools used normally for this type of comparison of CG algorithms:

- NOI = the total number of iterations
- NOF = the total number of function evaluation
- TIME = the total CPU time required for the processor to execute the CG algorithm and reach the minimum value of the required function minimization

To evaluate the modified conjugate gradient technique, this technique is analyzed and tested in some numerical tests (see [20]) and to demonstrate the performance of those methods, we applied Dolan and Moré [21], a new tool to analyze the efficiency of algorithms.

They introduced the notion of a performance profile as means to evaluate and compare the performance of the set of

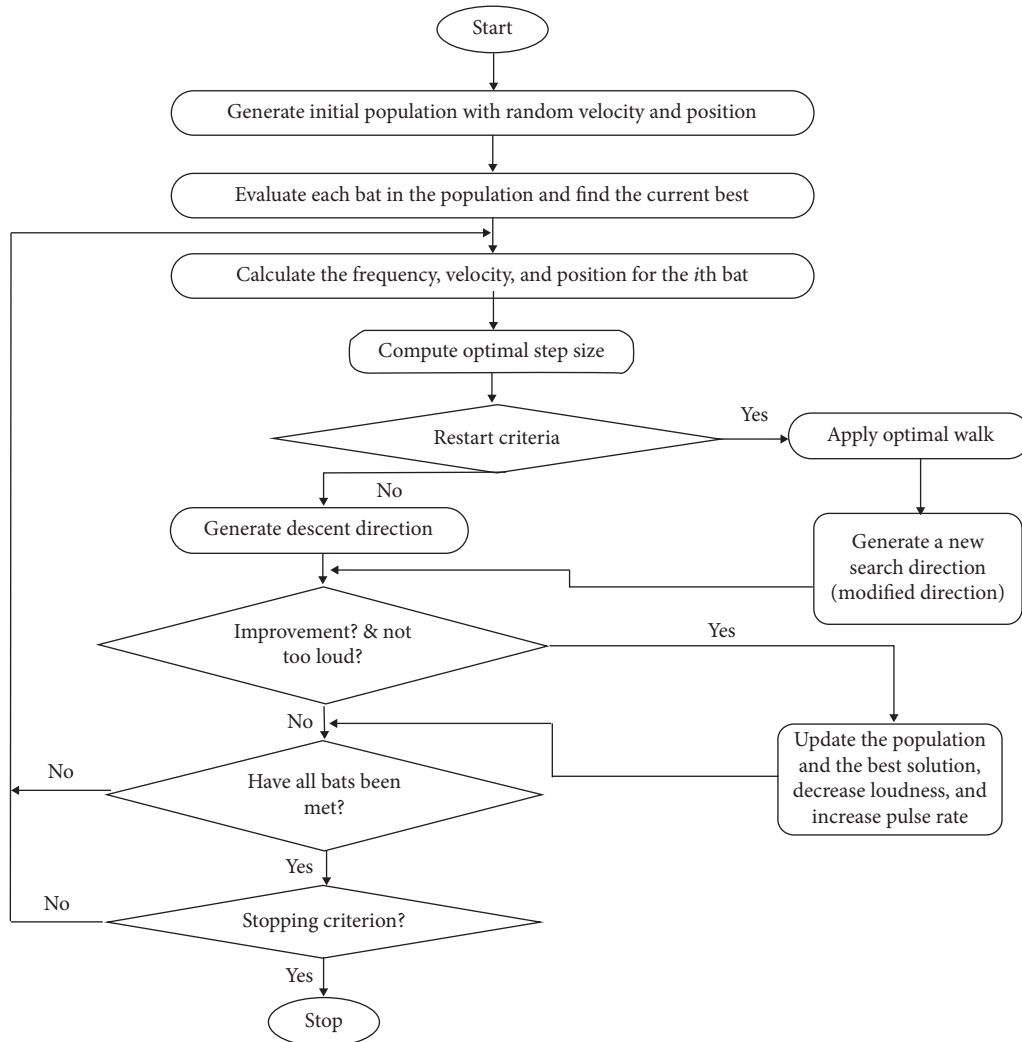


FIGURE 1: Flowchart of CG-BAT.

solvers S on a test set P . Assuming that there exist ns solvers and np problems, for each problem p and solvers, they defined

$t_{p,s}$ = computing time (the number of function evaluations or others) required to solve problem p by solver s

Requiring a baseline for comparisons, they compared the performance on problem p by solver s with the best performance by any solver on this problem, based on the performance ratio:

$$r_{p,s} = \frac{t_{p,s}}{\min(t_{p,s} : s \in S)} \quad (40)$$

Suppose that a parameter $r_M \geq r_p$ for all p, s is chosen, and $r_{p,s} = r_M$ if and only if solver s does not solve problem p (Figure 2).

Figure 3 shows the Dolan–More performance profile for these methods, which are subject to the frequency of a suitable performance compared to the basic methods.

Figure 4 shows us through the Dolan–More performance profile for these methods, which are measured by the CPU time, which makes us deduce from the three forms presented. The new method is very suitable for solving issues of many dimensions.

6.2. Experimental Results and Comparisons in CG-BAT.

To validate the performance of the proposed optimally directional bat algorithm, we have carried out various numerical experiments that have been then tested using several standard and nonstandard benchmarks from the CEC’2005 benchmark suite, which can be summarized as two comparison experiments. The first one is a comparison between the new directional bat algorithm and the standard algorithms including the bat algorithm on the classical benchmark functions, cuckoo search, firefly search, and practical swarm and the second one is a comparison has been performed against some advanced optimization algorithms

TABLE 1: Comparisons of the new algorithm against SS and HS algorithms for a total of 30 test problems.

Prob.	New algorithm	SS	HS
	NOI/NOF/TIME	NOI/NOF/TIME	NOI/NOF/TIME
1	100/253/0.14	324/6035/3.30	515/9706/2.69
2	406/886/2.03	413/890/1.99	399/876/2.15
3	823/2149/1.13	856/2120/1.07	826/2199/1.97
4	115/300/0.14	118/301/0.17	122/315/0.19
5	100/387/0.13	275/5485/2.14	168/1920/0.80
6	586/990/1.60	6577/18678/4.66	6858/20495/6.74
7	31/85/0.15	43/111/0.22	41/102/0.17
8	1029/2605/1.40	1044/2766/1.55	1028/2976/1.66
9	2377/4871/2.30	3588/7178/7.30	3674/7475/8.88
10	466/861/2.55	18785/503439/7.63	19344/517136/9.20
11	186/433/1.01	8659/270930/5.84	8250/256101/6.70
12	114/303/0.15	319/3882/1.70	280/2797/5.84
13	79/220/0.10	75/242/0.12	74/244/0.11
14	60/130/0.40	125/2139/7.13	161/2834/3.44
15	455/985/0.52	459/963/0.67	469/999/0.66
16	55/122/0.01	60/130/0.11	66/137/0.09
17	60/150/0.12	78/168/0.17	77/170/0.15
18	763/1587/0.89	699/1487/0.73	792/1697/0.92
19	64/148/0.22	122/1900/2.34	79/180/0.39
20	111/350/0.30	118/338/0.48	112/320/0.42
21	804/3213/1.55	430/1899/4.83	571/5294/5.79
22	62/265/0.41	74/356/0.77	2074/2223/9.59
23	4422/9553/1.23	20000/167265/5.95	19343/128760/5.22
24	61/211/0.32	564/15343/3.83	459/13555/3.39
25	449/1081/0.50	484/1111/0.75	541/1145/0.67
26	58/155/0.09	57/158/0.06	68/329/0.15
27	83/200/0.12	81/191/0.08	80/191/0.10
28	434/1129/0.65	495/1085/0.60	493/1077/0.63
29	530/1264/0.63	939/2239/1.09	953/2233/1.22
30	567/1430/0.90	19014/581435/2.76	18134/543276/2.13
Total	15420/35425/23.18	67795/66455/156.89	146559/894217/176.58

such as Dolan and Moré [19], a new tool to analyze the efficiency of algorithms.

6.2.1. *Benchmarking and Parameter Settings.* Thirty popular benchmark functions are shown in Tables 2–4. We have been used to verify the performance of the new bat algorithm (CG-BAT), compared with that of standard BA, FA, CS, and PSO. The description and the setting parameters of these algorithms are as follows:

- (1) CG-BAT: an extensive analysis was performed to carry out parameter settings of BA; for best practice, we recommend the following settings: $r_0=0.1$, $r_\infty=0.7$, $A_0=0.9$, $A_\infty=0.6$, $f_{\min}=0$ and $f_{\max}=2$, $\|g_k\| \leq 10^{-6}$, $\delta = 0.001$, $\hat{\sigma} = 0.1$, and $0 < \alpha_k \leq 1$.
- (2) BA: the standard bat algorithm was implemented as it is described in [17] with $r_0=0.1$, $A_0=0.9$, $\alpha = \gamma = 0.9$, $f_{\min}=0$, and $f_{\max}=2$.
- (3) FA: the firefly algorithm where $A_0=0.9$ is the intensity at the source point described in [15].

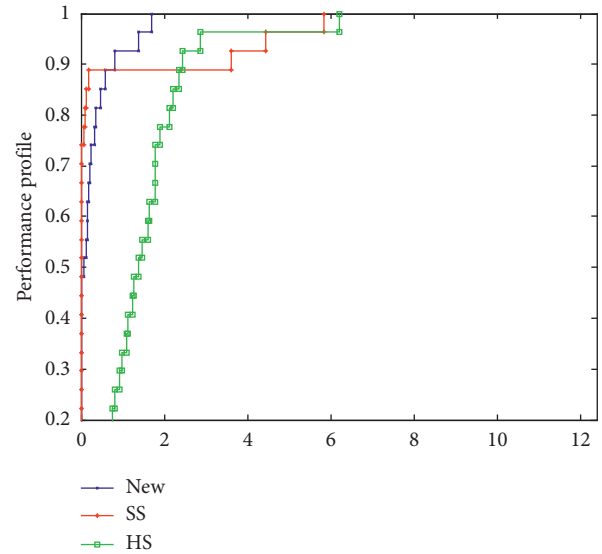


FIGURE 2: Performance profiles of the number of function evaluations.

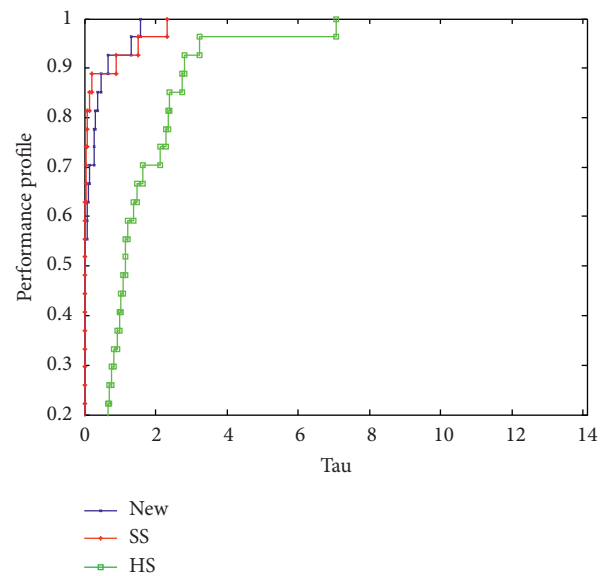


FIGURE 3: Performance profiles of the number of iterations.

- (4) CS: the cuckoo search via Lévy flights described in [22] is considered with the probability of the discovery of alien egg spa = 0.25.
- (5) PSO: a classical particle swarm optimization [23, 24] model has been considered. The parameter settings are $c_1 = 1.5$ and $c_2 = 1.2$ and the inertia coefficient w is a monotonically decreasing function from 0.9 to 0.4.

For a fair comparison, the common parameters are considered the same. The population size was set to $N = 50$, and the number of function evaluations is the same as 15000, without counting the initial evaluations, though all algorithms were initialized randomly in the similar manner. Therefore, we set $t_{\max} = 500$ except for CS. Due to the fact

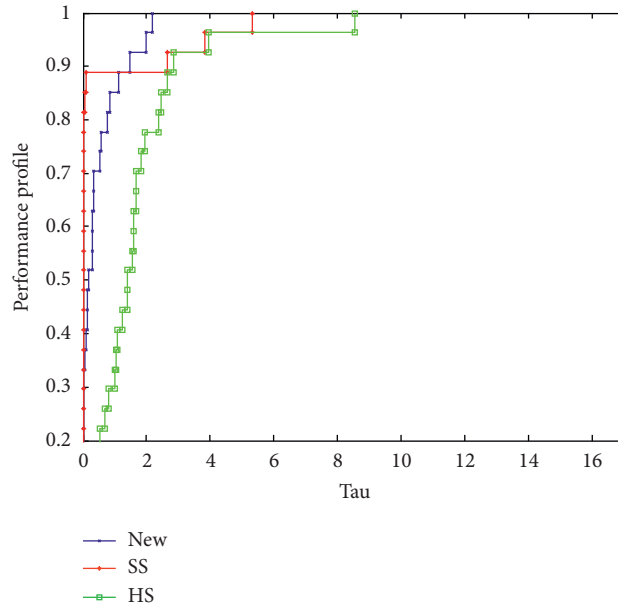


FIGURE 4: Performance profiles of the CPU.

TABLE 2: Comparison between CG-BAT and classical algorithm on benchmark functions ($F1 \sim F10$).

		CG-BAT	BA	FA	CS	PSO
$F1$	Best	1.927E-03	3.052E-01	2.481E+01	2.340E+02	2.340E+02
	Median	1.408E-02	5.480E+04	4.120E+01	4.357E+02	3.330E+02
	Worst	2.233E+00	6.569E+04	8.028E+01	6.119E+02	4.140E+02
	Mean	2.256E-01	4.920E+04	4.411E+01	4.153E+02	1.340E+02
	SD	4.869E-01	1.859E+04	1.259E+01	9.518E+01	2.000E+02
$F2$	Best	2.011E+06	3.313E+09	9.080E+08	3.229E+17	5.140E+12
	Median	1.171E+09	1.294E+15	1.177E+11	7.654E+19	6.3157E+12
	Worst	5.713E+10	5.893E+50	1.553E+12	2.433E+22	6.9119E+12
	Mean	2.363E+12	4.310E+49	3.051E+11	2.263E+21	4.153E+12
	SD	1.261E+12	1.461E+50	4.102E+11	5.976E+21	9.518E+11
$F3$	Best	1.634E+00	8.563E+00	9.877E+01	1.062E+03	3.229E+07
	Median	3.115E+01	2.996E+05	1.618E+02	1.996E+03	7.654E+09
	Worst	1.256E+02	4.370E+05	3.850E+02	3.409E+03	9.433E+12
	Mean	1.461E+01	2.612E+05	1.742E+02	2.138E+03	2.263E+01
	SD	7.456E+00	1.348E+05	6.173E+01	5.493E+02	5.976E+01
$F4$	Best	5.049E-03	3.210E+02	9.989E-03	3.026E+10	1.082E+03
	Median	8.544E-02	5.949E+02	8.997E-02	4.448E+10	1.886E+03
	Worst	5.630E-01	6.848E+02	2.136E+00	6.797E+00	3.809E+03
	Mean	1.405E-01	5.816E+02	2.303E-01	4.567E+00	2.138E+03
	SD	1.481E-01	7.884E+01	4.210E-01	9.934E-01	9.493E+02
$F5$	Best	2.685E+03	2.967E+06	-3.276E-03	2.831E+07	3.026E+10
	Median	2.553E+04	2.529E+06	3.007E-03	4.084E+07	4.248E+10
	Worst	3.907E+04	3.495E+06	2.215E-04	8.620E+07	6.497E+10
	Mean	2.423E+04	5.436E+06	4.901E-03	4.242E+04	4.567E+10
	SD	6.590E+04	6.360E+05	6.627E+03	1.118E+04	9.934E+10
$F6$	Best	1.812E+00	2.420E+02	2.998E+01	1.129E+02	2.931E+07
	Median	3.8097E+00	3.074E+02	1.575E+02	1.378E+02	4.984E+07
	Worst	8.1481E+00	3.670E+02	2.047E+02	1.644E+02	8.920E+07
	Mean	1.6193E+00	3.086E+02	1.551E+02	1.366E+02	4.242E+04
	SD	1.0623E+01	3.603E+01	3.368E+01	1.349E+01	1.228E+04
$F7$	Best	1.5168E+00	3.024E-01	1.053E+00	2.414E+00	1.339E+02
	Median	4.2901E+00	6.876E-01	1.928E+00	4.5175E+00	1.378E+02
	Worst	9.2997E+00	1.135E-02	3.388E+00	8.1813E+00	1.554E+02
	Mean	4.2716E+00	7.176E-01	2.017E+00	5.153E+00	1.366E+02
	SD	1.5826E+00	2.927E-01	5.223E-01	1.865E+00	1.349E+01

TABLE 2: Continued.

		CG-BAT	BA	FA	CS	PSO
F8	Best	0.214E+00	2.096E+01	2.302E+00	8.391E+00	2.414E+00
	Median	1.681E+00	2.196E+01	3.191E+00	1.400E+01	2.988E+01
	Worst	2.881E+00	5.996E+01	3.648E+00	1.750E+01	9.575E+02
	Mean	5.899E+00	6.996E+01	3.191E+00	1.209E+01	2.047E+02
	SD	1.730E+00	7.062E-04	2.904E-01	1.753E+00	1.451E+02
F9	Best	2.895E+03	7.685E+03	4.745E+03	4.522E-03	3.368E+01
	Median	4.492E+03	8.365E+03	5.370E+03	5.045E-03	1.753E+02
	Worst	5.646E+03	1.017E+04	6.006E+03	5.426E-03	1.928E+03
	Mean	4.357E+03	8.940E+03	5.407E+03	5.056E-03	3.398E+00
	SD	6.414E+02	1.242E+03	3.363E+02	1.747E-02	2.667E+00
F10	Best	2.911E-03	3.336E+01	4.637E+02	6.691E+02	5.223E+01
	Median	1.038E-02	2.473E+02	6.892E+02	9.105E+02	2.311E+02
	Worst	2.011E-01	2.944E+03	1.304E+03	2.290E+03	3.188E+03
	Mean	2.645E-02	4.916E+02	7.193E+02	1.073E+03	3.648E+00
	SD	2.926E-02	6.275E+02	2.121E+02	3.967E+02	3.191E+00

TABLE 3: Comparison between CG-BAT and classical algorithm on benchmark functions (F11~F20).

		CG-BAT	BA	FA	CS	PSO
F11	Best	7.536E+01	2.799E+00	1.414E+02	1.337E+02	2.904E-05
	Median	1.561E+02	2.103E+01	1.879E+02	2.190E+02	9.745E-03
	Worst	2.506E+02	0.334E+02	2.352E+02	3.009E+02	4.370E-03
	Mean	1.515E+02	3.629E+01	1.937E+02	2.214E+02	6.006E-03
	SD	4.105E+01	1.971E+01	2.433E+01	4.094E+01	5.407E-03
F12	Best	9.448E-03	1.323E+00	2.650E+01	1.059E+02	3.343E+02
	Median	1.528E-02	2.187E+01	6.164E+01	2.200E+02	4.437E+02
	Worst	3.044E-01	9.385E+02	1.438E+02	6.159E+02	6.492E+02
	Mean	1.911E-01	1.181E+02	6.790E+01	2.611E+02	1.304E+03
	SD	4.917E-01	2.293E+02	2.559E+01	1.384E+02	7.193E+02
F13	Best	2.094E-00	9.637E+01	1.259E+01	1.673E+02	2.121E+02
	Median	1.470E-03	8.037E+02	1.123E+02	1.444E+01	1.484E+02
	Worst	9.5017E-02	3.847E+02	1.011E+02	1.333E+02	1.889E+02
	Mean	1.495E-01	8.475E+00	1.121E+01	1.455E+01	2.452E+02
	SD	3.135E+00	6.847E-01	7.085E-01	7.923E-01	0.937E+02
F14	Best	1.344E+00	8.475E+00	2.073E+03	1.778E+02	1.433E+01
	Median	2.815E+01	4.501E+01	4.076E+03	3.536E+02	2.950E+01
	Worst	1.918E+02	7.523E+02	9.651E+03	7.081E+02	6.164E+01
	Mean	4.898E+01	1.695E+02	4.532E+03	3.554E+02	1.438E+02
	SD	5.028E+01	2.220E+02	1.861E+03	1.311E+02	6.790E+01
F15	Best	4.499E+01	1.382E+04	1.283E+05	1.542E+08	0.559E+08
	Median	3.283E+02	4.133E+05	2.388E+05	3.709E+08	0.259E+09
	Worst	2.518E+03	1.247E+07	3.528E+05	6.179E+08	1.123E+10
	Mean	4.926E+02	1.929E+06	2.392E+05	3.760E+08	1.011E+02
	SD	5.304E+02	3.115E+06	6.522E+04	1.192E+08	1.121E+01
F16	Best	3.462E-02	6.731E+00	1.271E+01	1.112E+01	3.085E+01
	Median	0.239E+01	1.505E+01	1.522E+01	1.404E+01	8.073E+03
	Worst	2.046E+01	2.832E+02	1.912E+02	1.834E+02	9.076E+03
	Mean	3.716E+00	1.647E+01	1.528E+01	1.452E+01	9.651E+03
	SD	4.409E+00	5.854E+00	1.682E+00	1.746E+00	4.532E+03
F17	Best	2.719E+02	3.067E+01	1.483E+01	2.157E+01	1.551E+03
	Median	3.085E+02	3.181E+01	2.232E+01	2.788E+01	1.743E+05
	Worst	3.320E+02	3.270E+01	2.805E+01	2.988E+01	2.388E+06
	Mean	3.053E+01	3.178E+01	2.276E+01	2.718E+01	3.528E+05
	SD	1.668E+00	4.720E-01	3.374E+00	2.463E+00	2.392E+05
F18	Best	1.131E+02	1.637E+02	2.948E+02	2.564E+02	6.522E+04
	Median	1.979E+02	2.550E+02	3.687E+02	3.163E+02	1.271E+11
	Worst	2.686E+02	4.179E+02	4.154E+02	3.596E+02	1.522E+11
	Mean	1.959E+02	2.651E+02	3.627E+02	3.168E+02	1.912E+02
	SD	3.767E+01	6.899E+01	3.236E+01	2.533E+01	1.528E+01

TABLE 3: Continued.

		CG-BAT	BA	FA	CS	PSO
F19	Best	5.054E-01	5.082E-01	2.426E+01	1.337E+02	1.682E+00
	Median	3.328E-00	5.697E+01	4.450E+01	2.190E+06	1.993E+01
	Worst	3.357E-00	7.542E+01	8.646E+01	3.377E+03	2.762E+01
	Mean	2.417E-00	5.172E+01	4.591E+01	5.082E+06	2.805E+01
	SD	4.826E-01	1.981E+03	1.265E+01	8.308E+04	7.542E+01
F20	Best	2.861E-00	5.453E+00	5.729E+00	5.193E+01	5.1842E+01
	Median	4.319E-00	5.964E+00	6.177E+00	1.241E+02	7.981E+03
	Worst	5.766E-00	6.693E+00	6.674E+00	2.769E+03	9.453E+00
	Mean	1.262E-00	6.019E+00	6.187E+00	3.347E+00	5.554E+00
	SD	7.905E-01	3.268E-01	2.475E-01	3.761E+00	6.693E+00

TABLE 4: Comparison between CG-BAT and classical algorithm on benchmark functions (F21~F30).

		CG-BAT	BA	FA	CS	PSO
F21	Best	1.118E+03	5.517E+00	5.919E+03	5.919E-03	6.019E+00
	Median	2.554E+03	6.560E+02	9.621E+03	9.621E-03	3.268E-01
	Worst	5.626E+03	7.964E+03	1.568E+04	1.568E-02	5.517E+00
	Mean	2.852E+03	1.678E+03	9.618E+01	9.618E-03	6.560E+02
	SD	1.105E+03	2.032E+03	2.226E+07	2.226E-03	7.964E+03
F22	Best	1.009E+01	1.488E+02	2.573E+05	2.573E+03	1.678E+03
	Median	1.585E+01	1.245E+03	7.580E+06	7.580E+03	2.032E+03
	Worst	1.824E+01	2.330E+04	8.664E+07	8.664E+04	1.488E+02
	Mean	1.046E+01	1.049E+02	3.533E+04	3.533E+04	1.245E+03
	SD	3.837E+05	4.971E+04	7.697E+04	1.697E+05	2.330E+04
F23	Best	5.828E+03	8.280E+01	4.124E-04	4.124E+04	1.049E+02
	Median	2.383E+04	5.373E+03	5.220E-04	5.220E+04	4.971E+04
	Worst	5.416E+04	3.294E+04	7.472E-04	7.472E+04	8.280E+01
	Mean	2.562E+04	8.130E+03	5.336E-04	5.336E+04	5.373E+03
	SD	7.676E+03	8.472E+03	8.132E-03	8.132E+03	3.294E+04
F24	Best	0.041E-03	1.080E-01	4.375E+01	4.375E+01	8.130E+03
	Median	1.258E-02	1.507E+01	8.306E+01	8.306E+01	8.472E+03
	Worst	2.684E-02	5.574E+01	1.201E+02	1.201E+02	1.080E-01
	Mean	8.481E-01	1.900E+01	8.040E+01	8.040E+01	1.507E+01
	SD	1.717E-01	1.828E+01	1.588E+01	1.588E+01	5.574E+01
F25	Best	7.078E-07	0.326E+03	5.169E+04	5.169E+03	1.900E+01
	Median	8.827E-06	2.920E+03	8.395E+05	8.395E+05	1.828E+01
	Worst	2.223E-05	7.001E+05	1.329E+06	1.329E+06	0.326E+03
	Mean	6.204E-05	3.194E+05	8.815E+05	8.815E+05	2.920E+03
	SD	2.312E-05	1.916E+05	1.932E+05	1.932E+05	7.001E+05
F26	Best	1.707E+02	2.994E+01	1.330E+02	1.330E+02	3.194E+15
	Median	2.517E+02	5.895E+01	1.625E+02	1.625E+02	1.916E+15
	Worst	3.456E+02	9.913E+01	1.845E+02	1.845E+02	2.994E+61
	Mean	2.599E+02	5.746E+01	1.580E+02	1.580E+02	5.885E+01
	SD	3.756E+01	1.825E+01	1.558E+01	1.558E+01	9.913E+01
F27	Best	2.126E+01	1.093E+00	1.366E+01	1.366E+01	5.746E+04
	Median	3.604E+01	4.073E+00	2.384E+01	2.384E+01	1.825E+05
	Worst	8.057E+01	9.562E+00	3.540E+01	3.540E+01	1.993E+06
	Mean	3.979E+01	5.675E+00	2.417E+01	2.417E+01	4.373E+00
	SD	1.681E+01	3.920E+00	5.004E+00	5.004E+00	9.772E+00
F28	Best	1.252E-03	2.595E+00	1.338E+01	1.338E+01	8.335E+05
	Median	1.462E-02	5.744E+00	1.559E+01	1.559E+01	1.329E+06
	Worst	1.737E-01	1.145E+01	1.640E+01	1.640E+01	8.995E+09
	Mean	1.474E+01	5.920E+00	1.540E+01	1.540E+01	1.932E+05
	SD	1.235E+00	2.453E+00	7.839E-01	7.839E-01	1.220E+02
F29	Best	7.293E+03	2.736E+04	2.281E+04	7.281E+03	1.895E+09
	Median	8.803E+03	4.228E+04	3.698E+04	8.698E+03	1.888E+09
	Worst	9.480E+03	5.993E+04	4.624E+04	9.624E+03	2.580E+09
	Mean	3.722E+03	4.208E+03	8.712E+03	8.712E+03	1.998E+01
	SD	5.060E+02	7.320E+02	5.463E+02	5.463E+02	1.336E+01

TABLE 4: Continued.

		CG-BAT	BA	FA	CS	PSO
F30	Best	8.437E+04	1.048E+02	8.566E+03	1.566E+01	1.384E+02
	Median	1.588E+05	2.756E+03	5.394E+04	2.394E+01	1.540E+03
	Worst	2.346E+05	4.793E+04	2.811E+05	3.811E+01	2.417E+01
	Mean	1.597E+05	5.961E+03	8.159E+04	8.159E+01	9.090E+03
	SD	4.048E+04	9.588E+03	6.481E+04	6.481E+04	4.237E+01

TABLE 5: Descriptive statistics for the five algorithms.

Algorithm	N	Mean	Std. deviation	Minimum	Maximum
CG-BAT	29	72929.28687	372879.0388	0.0000007080	2010000.000
BA	29	114242087.4	614631784.1	0.1080000000	3310000000
FA	29	31326970.45	168608177.7	-0.003280000	908000000.0
CS	29	1.11379E+16	5.99796E+16	0.0045200000	3.23000E+17
PSO	29	1.14457E+12	2.55784E+12	22.60000000	9.43000E+12

that the CS algorithm uses a number of $2N$ function evaluations at each iteration, we adjust t_{\max} for this case to 250. The dimensionality of all benchmark functions is $D = 30$.

6.2.2. The First Experiment. For meaningful statistical analysis, each algorithm was run 51 times using a different initial population at each turn. The global minimum obtained after each trial was recorded for further statistical analysis. Subsequently, the mean value of the global minimum, the standard deviation (SD), the best solution, the median, and the worst solution values have been computed and are presented in Tables 2–4. From the results presented in Tables 2–4, the new directional bat algorithm achieved better results for 20 functions ($F1, F2, F3, F4, F8, F10, F12, F13, F14, F15, F16, F18, F19, F20, F22, F23, F24, F25, F28,$ and $F29$), while the BA obtained better results for 3 functions ($F7, F26,$ and $F27$). The FA has better scores for 2 functions ($F5$ and $F17$). CS obtained best results for $F9, F21,$ and $F30$ and PSO for $F11$. We can show that the following monotonically decreasing function is more suitable and gives stability to the algorithm.

6.2.3. The Second Experiment (Nonparametric Statistical Tests). In this section, to evaluate CG-BAT performance, nonparametric statistical tests were carried out. We performed Friedman's test and pairwise comparisons. Table 5 shows the descriptive statistics for the five algorithms, which gives the number of values studied, the mean and the standard deviation, and the highest value of the values for each method and the lowest value for it. Note that CG-BAT has the least arithmetic mean of 72929.28687 and the standard deviation of 372879.0388, which is lower than the rest of the algorithms, with the lowest and greatest value for the method.

Table 6 presents the Friedman rank test. For this test, an algorithm is considered better if it has a low rank. From the results, CG-BAT has the lowest rank for the two tests, which means that it is the best performing algorithm from the comparison. In addition, the last two rows present the test

TABLE 6: Friedman rank test (Experiment 1).

Algorithm	Friedman
CG-BAT	1.86
BA	2.42
FA	3.00
CS	3.29
PSO	4.43
Statistic	44.139
p value	0.000

statistic and p value. The statistic is distributed according to the chi-square distribution with 4 degrees of freedom. The lower p value of the different tests suggests the existence of significant differences among the considered algorithms at $\alpha = 0.01$ level of significance.

To highlight the differences between CG-BAT with each of the other algorithms, Table 7 presents the pairwise comparison results using the Friedman test. The control method is CG-BAT. The analysis of the Friedman rank test shows significant differences between CG-BAT and four algorithms (BA, FA, CS, and PSO) according to the p values of chi-square statistic which all are less than $\alpha = 0.05$ level of significance, and since CG-BAT algorithm for all pairwise comparisons has a minimum rank (1.28 with BA, 1.24 with FA, 1.24 with CS, and 1.10 with PSO), the results reveal that CG-BAT algorithm is significantly superior to BA, FA, CS, and PSO algorithms.

6.2.4. The Third Experiment (Convergence Curve Analysis). The convergence curve is an important indicator for the performance of the algorithm, through which we can see the convergence speed and the ability of the new algorithm optimum. In order to evaluate the modified CG-BAT, this technique is analyzed and tested in some numerical tests, and to illustrate the performance of these methods, we applied Dolan and Moré [21] to analyze the efficiency of the algorithm. Performance profiles based on mean performance, standard deviation (SD), and best solution are shown in Figures 5–7.

TABLE 7: Pairwise comparison results over all algorithms with CG-BAT as a control method at $\alpha=0.05$ (Experiment 1).

Procedure	Algorithm	Rank	Chi-square statistic	p value
Friedman	BA	1.72	5.828	0.016
	FA	1.76	7.759	0.005
	CS	1.76	7.759	0.005
	PSO	1.90	18.241	0.000

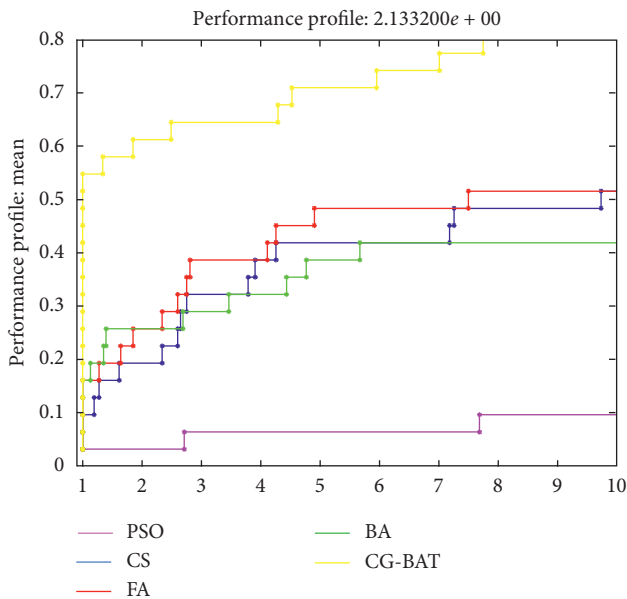


FIGURE 5: Performance profiles based on mean.

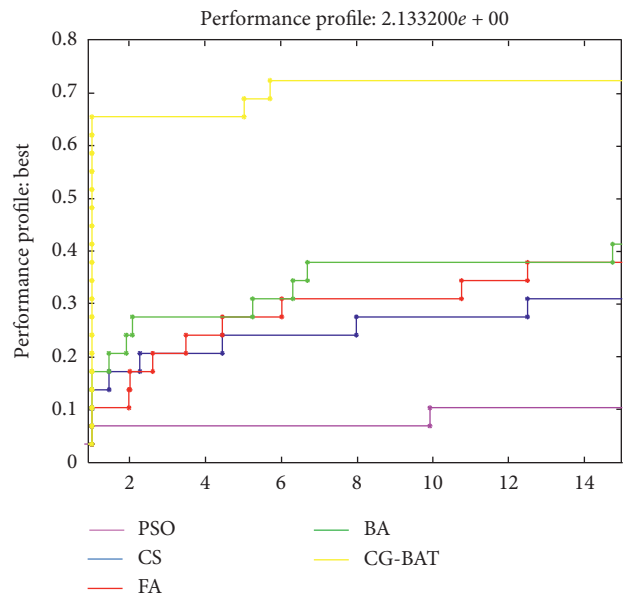


FIGURE 7: Performance profiles based on best solution.

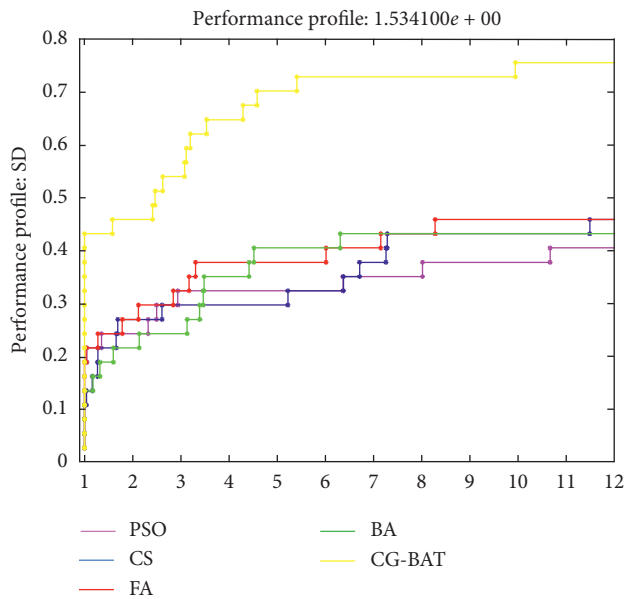


FIGURE 6: Performance profiles based on SD.

7. Conclusions

In this study, we have submitted new spectral CG methods. A crucial property of proposed CG methods is that it secures sufficient descent directions. Under mild conditions, we have demonstrated that the new algorithms are globally convergent for each uniformly convex and general functions using the strong Wolfe line search conditions. The preliminary numerical results show that new algorithms perform very well and also an improved version of the standard bat algorithm, called the new directional bat algorithm (CG-BAT), has been proposed and presented. Two modifications have been embedded to the BA to increase its exploitation and exploration capabilities and consequently have significantly enhanced the BA performance. Three sets of experiments have been carried out to prove the superiority of the proposed CG-BA). The performance is compared by using thirty test functions, under seven optimization algorithms (SS, HS, New, CG-BAT, BA, CS, FA, and PSO). The comparison results show that the enhanced algorithms (New and CG-BAT) are better than the original algorithms and have relatively stable performance in both the optimization ability and the convergence speed.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research was supported by College of Computer Sciences and Mathematics, University of Mosul, Republic of Iraq, under Project no. 4795793.

References

- [1] I. Fister Jr., D. Fister, and X.-S. Yang, "A hybrid bat algorithm," *Elektrote-Hniski Vestnik*, vol. 80, no. 1-2, pp. 1-7, 2013.
- [2] M. A. Al-Betar, M. A. Awadallah, H. Faris, X.-S. Yang, A. Tajudin Khader, and O. A. Alomari, "Bat-inspired algorithms with natural selection mechanisms for global optimization," *Neurocomputing*, vol. 273, pp. 448-465, 2018.
- [3] P. Wolfe, "Convergence conditions for ascent methods. II: some corrections," *SIAM Review*, vol. 13, no. 2, pp. 185-188, 1971.
- [4] M. R. Hestense and E. X. Stiefel, "Methods of conjugate gradients for solving linear system," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409-436, 1952.
- [5] E. Polak and G. Ribiere, "Note sur la convergence de méthodes de directions conjuguées," *ESAIM, Mathematical Modeling and Numerical Analysis*, vol. 3, no. 16, pp. 35-43, 1969.
- [6] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, no. 2, pp. 149-154, 1964.
- [7] 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.
- [8] L. Guanghai, H. Jiye, H. Jiye, and Y. Hongxia, "Global convergence of the Fletcher-Reeves algorithm with inexact linesearch," *Applied Mathematics—A Journal of Chinese Universities*, vol. 10, no. 1, pp. 75-82, 1995.
- [9] H. Liu, "A new conjugate gradient method for unconstrained optimization," *Far East Journal of Mathematical Sciences (FJMS)*, vol. 40, pp. 145-152, 2010.
- [10] S. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, Berlin, Germany, 1999.
- [11] L. Zhang, W. Zhou, and D. Li, "Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search," *Numerische Mathematik*, vol. 104, no. 4, pp. 561-572, 2006.
- [12] E. G. Birgin and J. M. Martínez, "A spectral conjugate gradient method for unconstrained optimization," *Applied Mathematics and Optimization*, vol. 43, no. 2, pp. 117-128, 2001.
- [13] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141-148, 1988.
- [14] A. Y. Al-Bayati and M. Salah, *New Variable Metric Method for Unconstrained Non-Linear Optimization*, Academic Press, London, UK, 1994.
- [15] H. Yabe and M. Takano, "Global convergence properties of nonlinear conjugate gradient methods with modified secant condition," *Computational Optimization and Applications*, vol. 28, no. 2, pp. 203-225, 2004.
- [16] E. T. Hamed, H. I. Ahmed, and A. Y. Al-Bayati, "A new hybrid algorithm for convex nonlinear unconstrained optimization," *Journal of Applied Mathematics*, vol. 2019, Article ID 8728196, 6 pages, 2019.
- [17] X.-S. Yang, "A new meta-heuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. González, D. Pelta, C. Cruz et al., Eds., vol. 284, pp. 65-74, Springer, Berlin-Heidelberg, Germany, 2010.
- [18] A. H. Gandomi, X.-S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Computing and Applications*, vol. 22, no. 6, pp. 1239-1255, 2013.
- [19] X. S. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464-483, 2012.
- [20] N. Andrei, "An unconstrained optimization test functions collection," *Advanced Modeling and Optimization*, vol. 10, pp. 147-161, 2008.
- [21] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201-213, 2002.
- [22] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Salamanca, Spain, October 2009.
- [23] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, October 1995.
- [24] R. C. Eberhart and S. Yuhui, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation Seoul*, Seoul, Korea, May 2001.