



Research Article

A Modified Bat Algorithm for Solving Large-Scale Bound Constrained Global Optimization Problems

Wali Khan Mashwani ¹, Ihsan Mehmood,¹ Maharani Abu Bakar ² and Ismail Koçcak³

¹Institute of Numerical Sciences, Kohat University of Science & Technology, Khyber Pakhtunkhwa (KPK), Kohat 26000, Pakistan

²Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Kaula Terengganu, Malaysia

³Faculty of Economics and Administrative Sciences, Kirikkale University, Ankara, Turkey

Correspondence should be addressed to Wali Khan Mashwani; mashwanigr8@gmail.com and Maharani Abu Bakar; maharani@umt.edu.my

Received 6 January 2021; Revised 30 January 2021; Accepted 12 February 2021; Published 28 February 2021

Academic Editor: Hassène Gritli

Copyright © 2021 Wali Khan Mashwani 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.

In the last two decades, the field of global optimization has become very active, and, in this regard, many deterministic and stochastic algorithms were developed for solving various optimization problems. Among them, swarm intelligence (SI) is a stochastic algorithm that is more flexible and robust and has had the ability to find an optimum solution for high-dimensional optimization and search problems. SI-based algorithms are mainly inspired by the social behavior of fish schooling or bird flocking. Among the SI-based algorithms, Bat algorithm (BA) is one of the recently developed evolutionary algorithms. It employs an echolocation behavior of microbats by varying pulse rates of emission and loudness to perform their search process. In this paper, a modified Bat algorithm (MBA) is developed. The main focus of the MBA is to further enhance the exploration and exploitation search abilities of the original Bat algorithm. The performance of the modified Bat algorithm (MBA) is examined over the benchmark functions designed for evolutionary algorithms competition in the special session of 2005 IEEE Congress on Evolutionary Computation. The used benchmark functions include the unimodal, multimodal, and hybrid benchmark functions with high dimensionality. Furthermore, the impact analysis with respect to different values of temperatures is conducted by executing the proposed algorithm twenty-five times independently by using each benchmark function with different random seeds.

1. Introduction

Global optimization has wide applications in many fields including physics, biology, engineering, economics, and business. Examples include minimizing the cost/consumption, while maximizing the profit/product is the need of the day in every time which leads human being towards the optimization process. It has been an essential part of research in applied mathematics since six or seven decades. The advancement of computers makes it possible to solve those optimization problems that were unsolvable before. The main objective of the optimization is to maximize the efficiency of the system while minimizing its cost. A scalable optimization problem can typically and generally be formulated as follows:

$$\begin{aligned} \text{Minimize } & f(x) = f_1(x), f_2(x), \dots, f_m(x), \\ & c_i(x) \leq 0, \quad i = 1, 2, \dots, p, \\ & c_j(x) = 0, \quad j = 1, 2, \dots, q, \\ & x_l^i \leq x^i \leq x_u^i, \quad i = 1, 2, \dots, N, \end{aligned} \quad (1)$$

where $x = (x_1, x_2, \dots, x_n)^T \in \Omega \subseteq R^n$ is the candidate solution with n decision variables, $f(x)$ comprises m objective function, $c_i(x) \leq 0$ and $c_j(x) = 0$ are $p + q$ constrained functions with p inequality, and q are equality constrained functions. Furthermore, a solution vector $x^* = [x_1, x_2, \dots, x_n]^T \in \Omega$ is said to be globally optimal if $f(x^*) \leq f(x), \forall x \in \Omega$, and the corresponding value of $f(x)$ is called global minimum. If $m = 1$ and $p + q = 0$, then

$f(x) = f_1(x)$ is single objective bound constrained function, while if $m \geq 2$, then the problem referred to as (1) is a multiobjective problem. The decision variables of the solution vector $x = [x_1, x_2, \dots, x_n]^T$ can be continuous, semicontinuous, discrete, integer, or as one of a finite set of explicit discrete categorical labels. Optimization problems can be classified based on the type of constraints, nature of design variables, physical structure of the problem, nature of the equations involved, deterministic nature of the variables, permissible value of the design variables, separability of the functions, and number of objective functions.

In the last few decades, different kinds of linear and nonlinear optimization techniques were developed to solve various linear and nonlinear optimization problems [1–5]. The optimization methods are further divided into two subclasses, local search methods and global search methods, as shown in Figure 1. The major difference between the two aforementioned subclasses is that local search methods work on single solution and provide a local optimum for the given problem, while global search methods operate on a set of solutions called population and usually provide a global optimum for given optimization problem.

Evolutionary computation (EC) is the subfield of soft computing [5, 7, 8] which studies a family of evolutionary algorithms (EAs) for global optimization. EAs are inspired by biological evolution and operate on a set of solutions which are generated uniformly and randomly undergo the crossover, mutation, selection, and survival of the fittest for next generation of evolutionary process. The four classical paradigms of EAs are the genetic algorithm (GA) [9], evolution strategy [10], evolution programming [11], and genetic programming [12]. Differential evolution [13, 14], particle swarm optimization [15], harmony search algorithm [16], gravitational search algorithm [17], biogeography-based optimization [18], ant lion optimizer [19], invasive weed optimization [20], shuffled frog leaping [21], ant colony optimization [22], artificial bee colony [23], firefly algorithm [24], Bat algorithm [25], cuckoo search [26], flower pollination algorithm [27], the whale optimization algorithm [28], grey wolf optimizer [29], salp swarm algorithm [30], ant colony optimization (ACO) [31, 32], firefly algorithms [33], and Bat algorithms [34] are all almost nature-inspired and swarm-intelligence-based algorithms.

Recently, in EC communities, hybrid swarm-intelligence-based algorithms and hybridization of nature-inspired algorithms [35] have got a great deal of attention to solve various test suites of single objective optimization and search problems [36–47]. However, proper settings of the control parameters include population size, selection rate, and operator probabilities, and other important intrinsic parameters in EAs play important role in their success while dealing with different optimization and search problems [48–50]. In this paper, a modified Bat algorithm (MBA) is developed, aiming at further improving the exploration versus exploitation issues of the baseline Bat algorithm in terms of convergence speed, convergence precision, and global searching ability. The performance of the proposed algorithm is evaluated over the benchmark functions, which are designed for the special session of evolutionary

algorithms competition of the IEEE Congress on Evolutionary Computation [51]. Furthermore, experiments are carried out with different values of temperature to show their impact on the performance of the proposed algorithm for dealing with each IEEE-CEC benchmark function [51].

The rest of the paper is organized as follows: Section 2 introduces the framework of the proposed modified Bat algorithm. Section 3 demonstrates the experimental results along with characteristics of benchmark functions used in the experiments we carried out. Section 4 finally concludes this paper.

2. Modified Bat Algorithm for Global Optimization

In the last few years, swarm intelligence- (SI-) based approaches have gained much attention in both academia and industrial applications [52]. Among them, ant colony optimization (ACO) [22], artificial bee colony (ABC) [3, 23, 52–56], cuckoo search (CS) [57, 58], firefly algorithm (FA) [24, 33, 34], and cuckoo search algorithm [57–59] are commonly used SI-based methods. They are inspired by collective behaviors of human beings, insects, birds flocking, the behavior of honeybees, the brooding behavior of cuckoo species, the flashing behavior of tropical fireflies, and the flocking behavior of birds. SI-based algorithms have tackled various optimization and search problems including traveling salesman problem (TSP) [58, 59], controlling unmanned vehicles, data mining, and many others.

Bat algorithm (BA) was first developed by Yang for global optimization problems [60–62]. It was inspired by the echolocation behavior of microbats, with varying pulse rates of emission and loudness. Bats are the only mammals with wings, and they have advanced capability of echolocation. Bats, dolphins, and other animals use the echolocation technique to determine the location of objects with reflected sound. This allows the animals to move around in pitch darkness, so they can navigate, hunt, identify friends and enemies, and avoid obstacles. The typical range of frequencies for most bat species is 25 kHz to 100 kHz. The bats emit a very loud sound pulse and listen for the echo that bounces back from objects. Bats emit 10 to 20 sound bursts every second. The pulse emission is going to increase up to 200 pulses per second while hunting prey. The wavelength λ of the ultrasonic sound bursts in the range from 2 mm to 14 mm. Recently, an improved version of BA with novel initialization techniques was introduced in [63]. The Bats harness three rules: (i) to sense distance, background barriers, and dissimilarity between prey/food, bats use echolocation; (ii) bats fly randomly with a certain velocity to a specific position with a fixed frequency with varying wavelengths to search for prey; and (iii) the loudness changes from a large to minimum constant value [64]. Three main traits of the microbat are employed to design the basic framework of BA as explained below:

- (i) Most of the species of microbats exercise echolocation to search for their prey.
- (ii) To search for prey, the microbats search with fixed frequency, Φ^{\min} , and varying wavelength, λ .

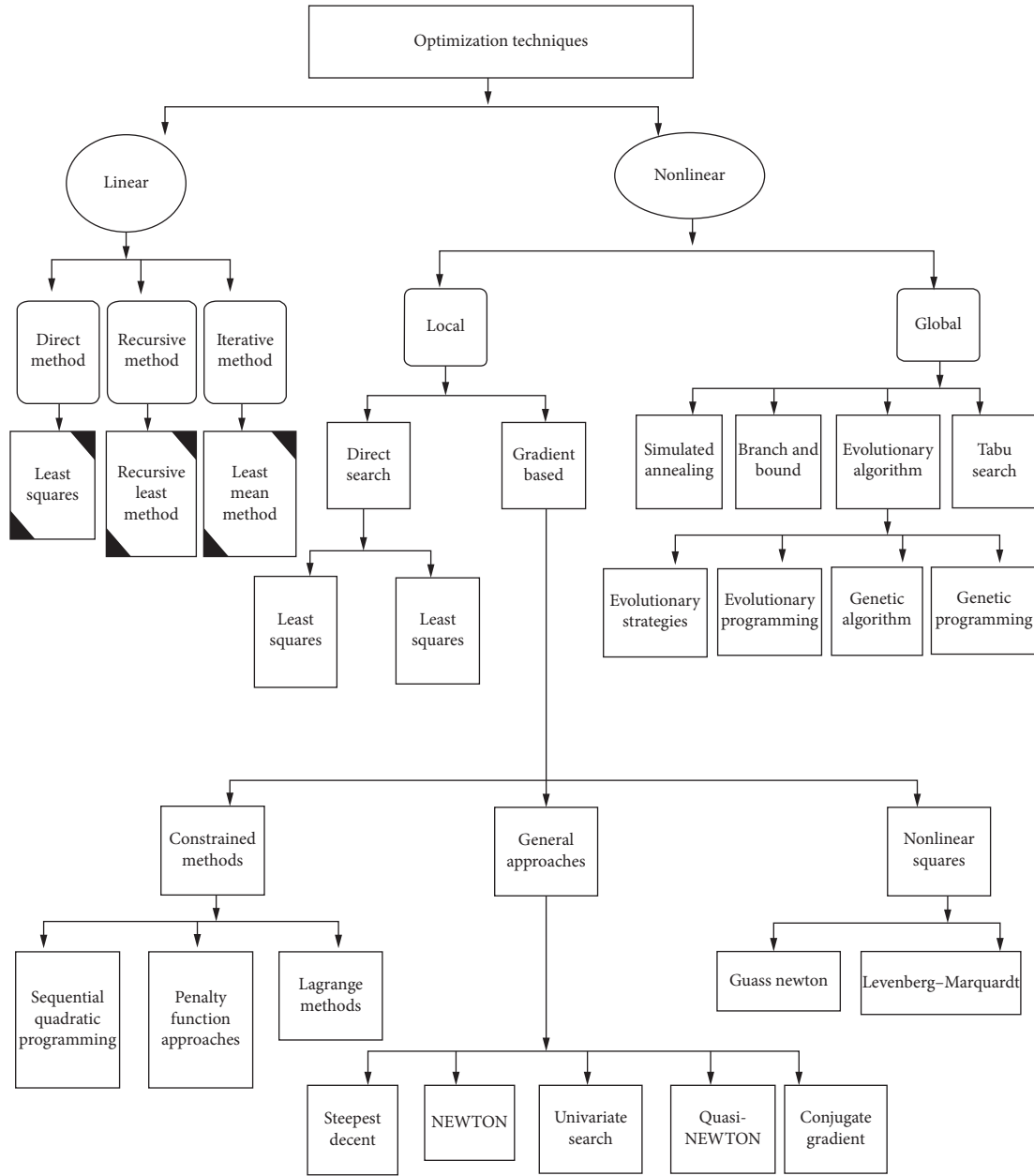


FIGURE 1: Flowchart of the optimization methods [6].

(iii) According to Yang’s method, the mathematical formulation of microbats is as follows:

$$\Phi^i = \Phi^{\min} + (\Phi^{\max} - \Phi^{\min}) \times \beta^i, \quad i = 1, 2, \dots, N, \quad (2)$$

$$v^i = v^{i-1} + (x^i - x^{\text{best}}) \times \Phi^i, \quad (3)$$

where Φ^{\min} is the minimum frequency and Φ^{\max} is the maximum frequency used by the i th Bat while searching for its prey, x^i denotes the location of the

i th Bat in the search space, v^i represents the velocity of the i th Bat, and $\beta \in [0, 1]$ is the parameter that controls the difference between the minimum and maximum frequencies of each Bat.

2.1. Exploration and Exploitation in the Proposed Algorithm. In this paper, equation (3) is used in the modified BA (MBA) and v^i as described in equation (4) along with δ^i as reported in equation (5), where δ^i is the distance between the sound wave source and the target in order to bound the wave back as defined by equation (5).

$$x^i = x^{i-1} + v^i, \quad i = 1, 2, \dots, N, \quad (4)$$

$$\delta^i = \frac{v^i \times \Delta T}{2}, \quad (5)$$

where $\Delta T \in [-1, 1]$. In the active sonar system, the distance between the sound wave source and the target that bounds the wave back as given in equation (5), where ΔT means the time difference between the sending sound wave and receiving echo and v represents the sound speed in the air, where $v^i = v_0^i + 0.61 \times TC^0$. The bats change their position according to equations (6) and (7) as per criteria given in flowchart 2.

$$x^i = x^{i-1} + \delta^i, \quad i = 1, 2, \dots, N, \quad (6)$$

$$x^i = x^{i-1} + (x^{\text{best}} - x^{i-1}) \times \beta^i, \quad i = 1, 2, \dots, N, \quad (7)$$

where $\beta \in [0, 1]$ move the bats around the current best solution. The impact analysis of temperature with sound speed in air was settled to be constant at 340 m/sec at 20°C [65]. In general, the speed of sound varies at different temperature levels in the air. The environmental effect of temperature with different values is also examined in the proposed MBA as outlined in Algorithm 1. The flowchart of the proposed modified Bat algorithm (MBA) is described in Figure 2. The main steps of the proposed MBA are summarized as follows:

Step 1. Initialization: spread bats uniformly and randomly within the search space of the problem.

Step 2. Generate new bats by using equations (4) to (7).

Step 3. Evaluate bats and update their global near best position (7).

Step 4. Check the termination condition to decide whether to continue or terminate the process.

2.2. Impact of Temperature with respect to the Speed of Sound in Air. The sound waves are longitudinal, which are propagating in air with speed described as follows:

$$v = \sqrt{\frac{\eta \times \rho \times T}{m}}. \quad (8)$$

Therefore,

$$v \propto \sqrt{T}. \quad (9)$$

The speed of sound in air is directly proportional to the square root of the absolute temperature of the air. If v and v_0 are the speeds of sound at temperatures T and T_0 , respectively, then

$$\frac{v}{v_0} = \sqrt{\frac{T}{T_0}}. \quad (10)$$

We have

$$T = (0^t + 273)K,$$

$$T_0 = (0^0 + 273)K,$$

$$\frac{v}{v_0} = \sqrt{\frac{T}{T_0}} = \sqrt{\frac{(0^t + 273)K}{(0^0 + 273)K}} \quad (11)$$

$$\frac{v}{v_0} = \sqrt{\frac{T}{T_0}} = \left[1 + \frac{0^t C}{273} K \right]^{1/2}.$$

Applying binomial theorem and neglecting higher-order terms, we get

$$\frac{v}{v_0} = \left[1 + \frac{0^t C}{2 \times 273} K \right] v = v_0 + \frac{v_0 0^t}{546}. \quad (12)$$

We have

$$0^0 C,$$

$$v_0 = 332 \text{ m/s},$$

$$\frac{v}{v_0} = \left[1 + \frac{0^t C}{2 \times 273} K \right], \quad (13)$$

$$v = v_0 + \frac{332 \times 0^t}{546} v = v_0 + 0.61 \times 0^t C.$$

3. Experimental Results and Discussion

Due largely to the nature of evolutionary algorithms (EAs), their behaviors and performance are mainly studied experimentally by using various test suites of optimization and search problems. For these purposes, several continuous test suites of benchmark functions have already been designed in the existing EC community over the last few years. These benchmark functions have played a crucial role in developing and scientifically studying the behavior of various existing EAs and modified enhanced versions. The benchmark functions which have recently been designed for the special session of the 2005 IEEE Congress on Evolutionary Computation are utilized in the study conducted in this paper in order to carefully assess the proposed modified Bat algorithm (MBA) and impact analysis of various temperatures used in its framework. For more details about these instances, the readers are referred to [51]. These benchmark functions that can be categorized as F_1 to F_5 are unimodal functions, F_6 to F_{12} are basic multimodal functions, F_{13} and F_{14} are both expanded multimodal benchmark functions, and F_{15} is a hybrid composition test problem. Optimization problems are classified on the basis of landscape as follows:

- (i) *Unimodal.* If the optimization problem has only one optimum that is global and local, it is called unimodal, for example, sphere function.

```

(1)  $N$ : population size;
(2)  $n$ : number of decision variables
(3)  $M_{it}$ : maximum Iteration.
(4)  $\beta \in [0, 1]$ : random number,  $M^t$ : maximum Iteration.
(5) Initialize bat the population  $x^i = [x_i + (x_u - x_l) \times \text{rand}(N, n)]$ ,  $i = 1, 2, \dots, N$ 
(6) Compute  $[f(x^i)]$ ,  $i = [1, 2, \dots, N]^T$ ,
(7) Find the current best  $x^{\text{best}}$ 
(8)  $it = 1\%$ 
(9) while  $it \leq M_{it}$  do
(10) Generate new solution by flying randomly.
(11)  $\Phi^i = \Phi^{\min} + (\Phi^{\max} - \Phi^{\min}) \times \beta_i$ 
(12)  $v^i = v^{i-1} + (x^i - x^g) \times \Phi^i$ 
(13)  $v^i = v_0 + 0.61 \times T0^C$ 
(14)  $\delta^i = ((v^i \times \Delta T)/2)$ 
(15)  $x^i = x^{i-1} + \delta^i$ 
(16) Compute the objective function value,  $f(x^i)$ 
(17) if  $f(x^i) < f(x^{i-1})$  then
(18) Accept  $x^i$ .
(19) else
(20) Retain  $x^{i-1}$ 
(21) end if
(22) Update current best solution denoted by  $x^{\text{best}}$ .
(23) Move the bats around current best solution randomly.
(24)  $x^i = x^{i-1} + (x^{\text{best}} - x^{i-1}) \times \beta^i$ 
(25) Repeating steps 11 to 15 and update the global best solution.
(26)  $it = it + 1$ ;
(27) end while

```

ALGORITHM 1: Framework of the proposed modified Bat algorithm.

(ii) *Multimodel*. If the optimization problems have more than one optimum, those may all be global, or some of them may be local, called unimodal, for example, Ackley function.

The second last column of Table 1 provides the known optimal solution of each of the benchmark functions.

3.1. PC Configuration and Programming Environment. All experiments were performed on a computer with Intel Core i5 – 6200 CPU, 2.40 GHz and 4 G RAM, under Windows 10 Pro, 64-bit OS. The proposed algorithm and all other existing algorithms used in the comparative analysis were implemented in MATLAB R2017b programming environment. The proposed algorithm was executed 25 times independently with different random seeds by using the Matlab command rand (“state,” sum (100*clock)).

3.2. Parameter Settings. All experiments were performed by solving the benchmark functions with parameters settings described as follows:

- (i) $N = 100$: population size.
- (ii) $n = 10, 30, 50$: the dimension of the search space.
- (iii) $M_{it} = n \times N$: Maximum iteration.
- (iv) $v_0 = 0.331$: velocity of sound at $0C^0$.
- (v) x_g : the global best.

- (vi) x_i : location of the i th Bat in the solution space.
- (vii) $nruns = 25$: the number of independent runs.
- (viii) $\Phi^{\min} = 0$: minimum frequency.
- (ix) $\Phi^{\max} = 2$: maximum frequency.
- (x) $T0^C = 0C^0, 10C^0, 20C^0, 30C^0, 40C^0$: % different temperature in Celsius scale.
- (xi) $\delta = [(0.331) + (6 \times t) \times (-1 + 2 \times \zeta) \times 0.5]\%$: distance between sound waves source and the target.
- (xii) $\beta \in [0, 1]$ and $\zeta \in [0, 1]$ are random numbers, respectively.
- (xiii) $\Delta T \in [-1, 1]$: time difference between the sent sound wave and echo.

Tables 2 and 3 include the numerical results provided by the proposed modified Bat algorithm in terms of minimum objective values, average objective values, standard deviation, and maximum objective function values after independent runs of twenty-five simulations with random seeds to solve each benchmark function whose characteristics are given in Table 1. All these numerical values were obtained with different values of temperatures like $t = 0, 10, 20, 30$, and 40 in the framework of the suggested algorithms, respectively. The last columns of Tables 2 and 3 include the average CPU time in seconds elapsed by the proposed modified Bat algorithm while solving each corresponding benchmark function.

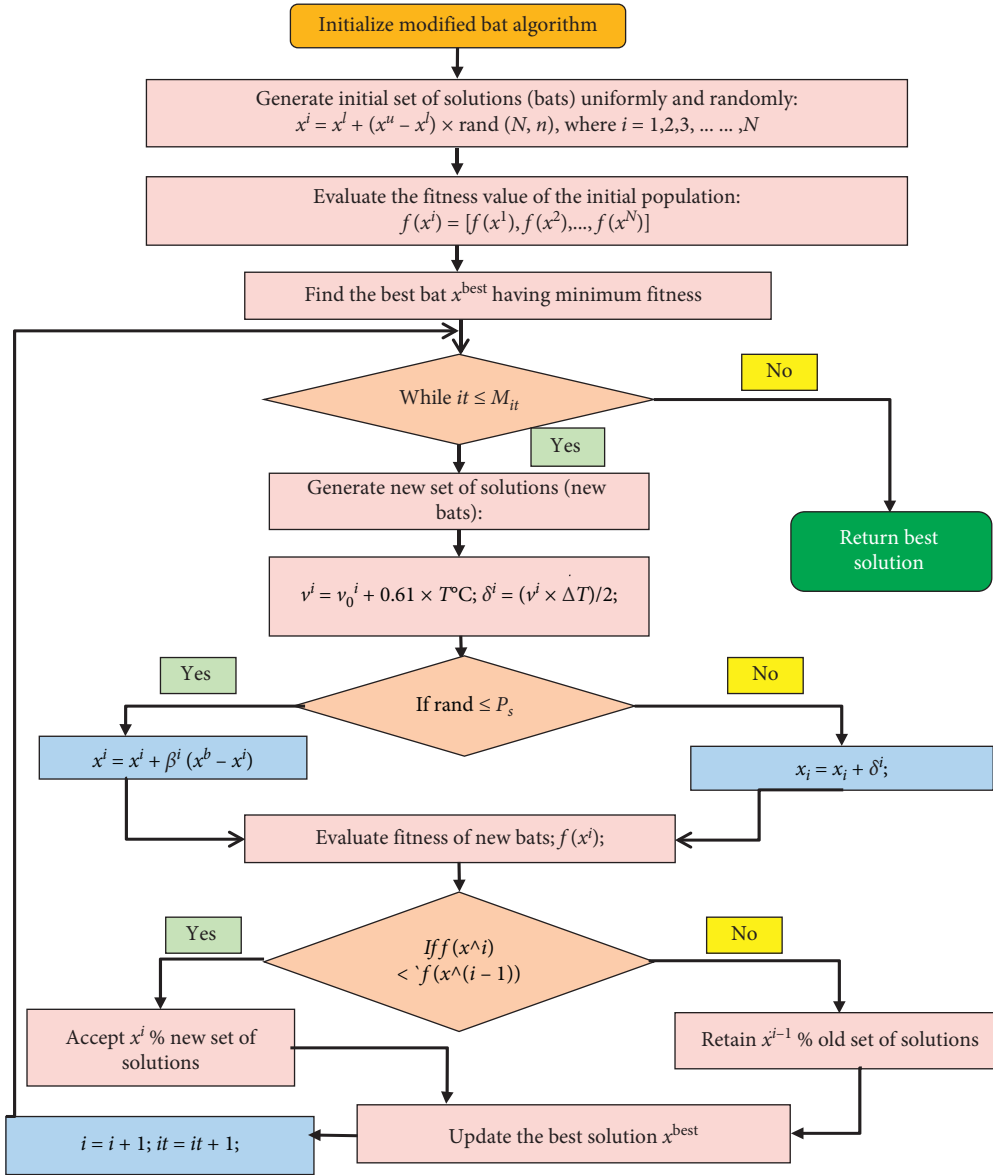


FIGURE 2: Flowchart of the modified Bat algorithm.

TABLE 1: Properties of the Congress of the IEEE-CEC'05 benchmark function [51].

Functions	Problem name	$[x_l, x_u]$	Dimension	Optimum
F_1	Shifted sphere function	$[-100, 100]$	2, 10	-450
F_2	Shifted Schwefel's problem 1.2	$[-100, 100]$	2, 10	-450
F_3	Shifted rotated high conditioned elliptic function	$[-100, 100]$	2, 10	-450
F_4	Shifted Schwefel's problem 1.2 with noise in fitness	$[-100, 100]$	2, 10	-450
F_5	Schwefel's problem 2.6 with global optimum on bounds	$[-100, 100]$	2, 10	-310
F_6	Shifted Rosenbrock's function	$[-100, 100]$	2, 10	390
F_7	Shifted rotated Griewank's function without bounds	$[0, 600]$	2, 10	-180
F_8	Shifted rotated Ackley's function with global optimum on bounds	$[-32, 32]$	2, 10	-140
F_9	Shifted Rastrigin's function	$[-5, 5]$	2, 10	-330
F_{10}	Shifted rotated Rastrigin's function	$[-5, 5]$	2, 10	-330
F_{11}	Shifted rotated Weierstrass function	$[-0.5, 0.5]$	2, 10	90
F_{12}	Schwefel's problem 2.13	$[-\pi, \pi]$	2, 10	-460
F_{13}	Expanded extended Griewank's plus Rosenbrock's function (F_8F_2)	$[-5, 5]$	2, 10	-130
F_{14}	Shifted rotated expanded Schaffer's F_6	$[-100, 100]$	2, 10	-300
F_{15}	Hybrid composition function	$[-5, 5]$	2, 10	120
F_{16}	Hybrid composition function	$[-5, 5]$	2, 10	120
F_{17}	Rotated hybrid composition function with noise in fitness	$[-5, 5]$	2, 10	120

TABLE 1: Continued.

Functions	Problem name	$[x_j, x_u]$	Dimension	Optimum
F_{18}	Rotated hybrid composition function	$[-5, 5]$	2, 10	10
F_{19}	Rotated hybrid composition function with a narrow basin for the global optimum	$[-5, 5]$	2, 10	10
F_{20}	Rotated hybrid composition function with the global optimum on the bounds	$[-5, 5]$	2, 10	10
F_{21}	Rotated hybrid composition function	$[-5, 5]$	2, 10	360
F_{22}	Rotated hybrid composition function with high condition number matrix	$[-5, 5]$	2, 10	360
F_{23}	Noncontinuous rotated hybrid composition function	$[-5, 5]$	2, 10	360
F_{24}	Rotated hybrid composition function	$[-5, 5]$	2, 10	260
F_{25}	Rotated hybrid composition function without bounds	$[2, 5]$	2, 10	260

TABLE 2: Experimental results provided by the modified Bat algorithm for IEEE-CEC'2005 benchmark functions [51].

Function	TC^0	Minimum	Maximum	Mean	St. dev.	Time/run (s)
F_1	0	$-4.499997e + 02$	$-4.499886e + 02$	$-4.499950e + 02$	$3.426879e - 03$	1.062845
	10	$-4.497749e + 02$	$-4.466384e + 02$	$-4.485960e + 02$	$9.442779e - 01$	1.114802
	20	$-4.490360e + 02$	$-4.391881e + 02$	$-4.449205e + 02$	$3.378278e + 00$	1.756442
	30	$-4.463289e + 02$	$-4.237131e + 02$	$-4.358022e + 02$	$6.604429e + 00$	1.110323
	40	$-4.444716e + 02$	$-3.921134e + 02$	$-4.255253e + 02$	$1.429518e + 01$	1.748652
F_2	0	$-4.499969e + 02$	$-4.487071e + 02$	$-4.499276e + 02$	$2.873045e - 01$	1.119200
	10	$-4.494949e + 02$	$-4.462499e + 02$	$-4.480276e + 02$	$1.060553e + 00$	1.182674
	20	$-4.482317e + 02$	$-4.327124e + 02$	$-4.425412e + 02$	$3.970974e + 00$	1.876191
	30	$-4.439913e + 02$	$-4.153382e + 02$	$-4.329552e + 02$	$9.600158e + 00$	1.189104
	40	$-4.481776e + 02$	$-3.986948e + 02$	$-4.270558e + 02$	$1.386966e + 01$	1.855762
F_3	0	$1.789746e + 04$	$2.182609e + 05$	$9.135860e + 04$	$5.720161e + 04$	2.896229
	10	$1.170948e + 04$	$3.945656e + 05$	$1.467675e + 05$	$1.014180e + 05$	2.019358
	20	$5.757183e + 04$	$6.137461e + 05$	$2.072769e + 05$	$1.368319e + 05$	2.520408
	30	$1.028405e + 05$	$5.121083e + 05$	$2.880849e + 05$	$1.221121e + 05$	2.012454
	40	$6.876147e + 04$	$5.377325e + 05$	$2.757455e + 05$	$1.494121e + 05$	2.547332
F_4	0	$2.349475e + 02$	$1.129217e + 04$	$4.673889e + 03$	$2.842841e + 03$	2.254365
	10	$-4.497415e + 02$	$-4.453140e + 02$	$-4.473806e + 02$	$1.388479e + 00$	2.280442
	20	$-4.472468e + 02$	$-4.324609e + 02$	$-4.424842e + 02$	$4.353951e + 00$	2.969045
	30	$-4.460352e + 02$	$-4.159675e + 02$	$-4.320192e + 02$	$9.286127e + 00$	2.277696
	40	$-4.471200e + 02$	$-3.635279e + 02$	$-4.180321e + 02$	$1.783432e + 01$	2.961883
F_5	0	$-3.069021e + 02$	$5.939871e + 03$	$2.846866e + 03$	$1.693554e + 03$	2.809083
	10	$-3.079270e + 02$	$-2.646887e + 02$	$-2.895821e + 02$	$1.108041e + 01$	2.834624
	20	$-3.046092e + 02$	$-2.173812e + 02$	$-2.748401e + 02$	$2.242043e + 01$	2.090643
	30	$-3.065533e + 02$	$-2.271017e + 02$	$-2.586472e + 02$	$2.139386e + 01$	2.795133
	40	$-2.981937e + 02$	$-1.764856e + 02$	$-2.401907e + 02$	$3.300340e + 01$	2.084235
F_6	0	$3.970419e + 02$	$6.154240e + 03$	$1.148499e + 03$	$1.533745e + 03$	2.095513
	10	$4.465394e + 02$	$8.376701e + 03$	$3.010844e + 03$	$2.745176e + 03$	2.150034
	20	$4.141301e + 02$	$1.292244e + 04$	$4.166460e + 03$	$3.215378e + 03$	2.819058
	30	$6.274048e + 02$	$8.757117e + 04$	$1.785373e + 04$	$2.040064e + 04$	2.077619
	40	$1.839170e + 03$	$3.971707e + 05$	$4.332860e + 04$	$8.926013e + 04$	2.786852
F_7	0	$1.491803e + 03$	$1.934009e + 03$	$1.746739e + 03$	$1.304394e + 02$	2.959575
	10	$1.087046e + 03$	$1.087261e + 03$	$1.087072e + 03$	$5.307234e - 02$	2.078052
	20	$1.087046e + 03$	$1.087098e + 03$	$1.087060e + 03$	$1.809096e - 02$	2.680421
	30	$1.087046e + 03$	$1.087226e + 03$	$1.087091e + 03$	$6.692838e - 02$	2.982949
	40	$1.087046e + 03$	$1.087226e + 03$	$1.087113e + 03$	$7.291012e - 02$	2.727031
F_8	0	$-1.198518e + 02$	$-1.195608e + 02$	$-1.196660e + 02$	$7.483006e - 02$	2.010502
	10	$-1.197559e + 02$	$-1.196069e + 02$	$-1.196791e + 02$	$4.099541e - 02$	2.071624
	20	$-1.198594e + 02$	$-1.196369e + 02$	$-1.197150e + 02$	$5.804527e - 02$	2.680243
	30	$-1.197914e + 02$	$-1.195242e + 02$	$-1.196708e + 02$	$7.431507e - 02$	2.991901
	40	$-1.198927e + 02$	$-1.195406e + 02$	$-1.196934e + 02$	$9.550049e - 02$	2.681372
F_9	0	$-3.213506e + 02$	$-3.097053e + 02$	$-3.161274e + 02$	$3.545332e + 00$	1.094128
	10	$-3.256990e + 02$	$-3.012993e + 02$	$-3.109745e + 02$	$7.290333e + 00$	1.178994
	20	$-3.207080e + 02$	$-2.998410e + 02$	$-3.092325e + 02$	$6.086134e + 00$	1.876494
	30	$-3.211287e + 02$	$-2.999921e + 02$	$-3.083039e + 02$	$5.086988e + 00$	1.108350
	40	$-3.206824e + 02$	$-2.975933e + 02$	$-3.089291e + 02$	$4.942759e + 00$	1.841847

TABLE 2: Continued.

Function	TC^0	Minimum	Maximum	Mean	St. dev.	Time/run (s)
F_{10}	0	$-3.197695e + 02$	$-2.896784e + 02$	$-3.083181e + 02$	$9.447741e + 00$	2.946432
	10	$-3.146875e + 02$	$-2.720312e + 02$	$-3.007771e + 02$	$1.112508e + 01$	2.024723
	20	$-3.152417e + 02$	$-2.721939e + 02$	$-2.974220e + 02$	$9.731745e + 00$	2.609384
	30	$-3.061357e + 02$	$-2.760622e + 02$	$-2.918401e + 02$	$9.292641e + 00$	2.920436
	40	$-3.119812e + 02$	$-2.611062e + 02$	$-2.951825e + 02$	$1.044157e + 01$	2.637320
F_{11}	0	$9.208964e + 01$	$9.767944e + 01$	$9.544874e + 01$	$1.240322e + 00$	10.972307
	10	$9.423624e + 01$	$9.874577e + 01$	$9.704723e + 01$	$1.193214e + 00$	11.500669
	20	$9.363069e + 01$	$9.917987e + 01$	$9.668650e + 01$	$1.510288e + 00$	11.392789
	30	$9.418018e + 01$	$9.897152e + 01$	$9.707386e + 01$	$1.050424e + 00$	11.080728
	40	$9.403592e + 01$	$9.890662e + 01$	$9.687178e + 01$	$1.279591e + 00$	11.438615

TABLE 3: Experimental results provided by the modified Bat algorithm for IEEE-CEC'2005 benchmark functions [51].

Function	TC^0	Minimum	Maximum	Mean	St. dev.	Time/run (s)
F_{12}	0	$-4.437762e + 02$	$1.019104e + 04$	$1.476383e + 03$	$2.927702e + 03$	6.177942
	10	$2.564232e + 03$	$3.557042e + 04$	$1.429018e + 04$	$8.658623e + 03$	6.406563
	20	$1.662075e + 03$	$3.180153e + 04$	$1.589825e + 04$	$8.896739e + 03$	6.766525
	30	$8.941822e + 02$	$5.057289e + 04$	$2.168591e + 04$	$1.372156e + 04$	6.303731
	40	$2.329014e + 03$	$4.750996e + 04$	$2.157708e + 04$	$1.488413e + 04$	6.595028
F_{13}	0	$-1.293552e + 02$	$-1.279225e + 02$	$-1.286647e + 02$	$3.905484e - 01$	2.252407
	10	$-1.292500e + 02$	$-1.270728e + 02$	$-1.282126e + 02$	$5.957121e - 01$	2.340792
	20	$-1.292576e + 02$	$-1.256479e + 02$	$-1.281832e + 02$	$8.822713e - 01$	2.129807
	30	$-1.290435e + 02$	$-1.273722e + 02$	$-1.282940e + 02$	$4.668497e - 01$	2.269008
	40	$-1.290267e + 02$	$-1.271944e + 02$	$-1.281464e + 02$	$5.781605e - 01$	2.096401
F_{14}	0	$-2.981243e + 02$	$-2.960070e + 02$	$-2.969594e + 02$	$4.898806e - 01$	3.153805
	10	$-2.973976e + 02$	$-2.964257e + 02$	$-2.969018e + 02$	$2.408217e - 01$	3.270247
	20	$-2.978127e + 02$	$-2.964904e + 02$	$-2.970541e + 02$	$2.741207e - 01$	3.028066
	30	$-2.982210e + 02$	$-2.964820e + 02$	$-2.969863e + 02$	$3.986971e - 01$	3.155276
	40	$-2.977934e + 02$	$-2.964774e + 02$	$-2.970809e + 02$	$3.690230e - 01$	3.052878
F_{15}	0	$2.545818e + 02$	$6.587341e + 02$	$4.421716e + 02$	$1.444719e + 02$	32.680917
	10	$2.494539e + 02$	$7.548692e + 02$	$4.982271e + 02$	$1.687343e + 02$	32.504363
	20	$2.889911e + 02$	$7.595381e + 02$	$5.319752e + 02$	$1.530076e + 02$	32.136422
	30	$3.345009e + 02$	$7.980487e + 02$	$5.815470e + 02$	$1.557751e + 02$	32.454858
	40	$2.696861e + 02$	$7.647580e + 02$	$5.475392e + 02$	$1.559211e + 02$	32.090621
F_{16}	0	$2.280200e + 02$	$4.038796e + 02$	$2.963230e + 02$	$4.101732e + 01$	32.183581
	10	$2.262273e + 02$	$3.178970e + 02$	$2.773537e + 02$	$2.372156e + 01$	32.404209
	20	$2.461453e + 02$	$4.040593e + 02$	$2.995398e + 02$	$3.940300e + 01$	32.505249
	30	$2.410189e + 02$	$3.691590e + 02$	$3.072550e + 02$	$3.636052e + 01$	31.637511
	40	$2.286962e + 02$	$3.578082e + 02$	$3.016452e + 02$	$3.370245e + 01$	33.545575
F_{17}	0	$2.344848e + 02$	$3.924305e + 02$	$2.870523e + 02$	$3.584210e + 01$	32.941267
	10	$2.517420e + 02$	$3.274312e + 02$	$2.943857e + 02$	$2.125994e + 01$	31.970876
	20	$2.579826e + 02$	$3.572980e + 02$	$3.026865e + 02$	$3.119593e + 01$	33.131067
	30	$2.847480e + 02$	$4.122628e + 02$	$3.189076e + 02$	$3.197554e + 01$	31.884657
	40	$2.607445e + 02$	$4.205725e + 02$	$3.109693e + 02$	$3.521222e + 01$	32.002564
F_{18}	0	$8.112274e + 02$	$1.140130e + 03$	$1.027758e + 03$	$7.544764e + 01$	32.280959
	10	$6.722426e + 02$	$1.037429e + 03$	$8.807608e + 02$	$1.152772e + 02$	32.473126
	20	$7.713579e + 02$	$1.016555e + 03$	$9.161868e + 02$	$7.896796e + 01$	32.098703
	30	$7.522107e + 02$	$1.054692e + 03$	$9.275311e + 02$	$9.516644e + 01$	32.495687
	40	$6.493147e + 02$	$1.057168e + 03$	$8.567678e + 02$	$1.220780e + 02$	32.245067
F_{19}	0	$8.105882e + 02$	$1.109775e + 03$	$1.016489e + 03$	$9.588698e + 01$	32.249585
	10	$5.697413e + 02$	$1.036857e + 03$	$9.038504e + 02$	$1.135260e + 02$	32.526338
	20	$6.455885e + 02$	$1.055063e + 03$	$9.246263e + 02$	$1.212405e + 02$	32.308435
	30	$7.347781e + 02$	$1.049692e + 03$	$8.926973e + 02$	$9.131554e + 01$	32.933979
	40	$7.875747e + 02$	$1.058803e + 03$	$9.551627e + 02$	$7.785110e + 01$	32.254910

Figure 3 presents the convergence graphs displayed by the proposed algorithm to solve the benchmark functions as denoted by F_1 to F_6 . Different values $T = 0, 10, 20, 30,$ and 40 along the x -axis represent the variation in temperature

values and the evolution in minimum objective function values is shown along the y -axis. Similarly, Figure 4 displays the convergence of the benchmark denoted by F_7 to F_{13} with different temperature settings like $T = 0, 10, 20$. Figure 5

Wali Khan Mashwani et al.

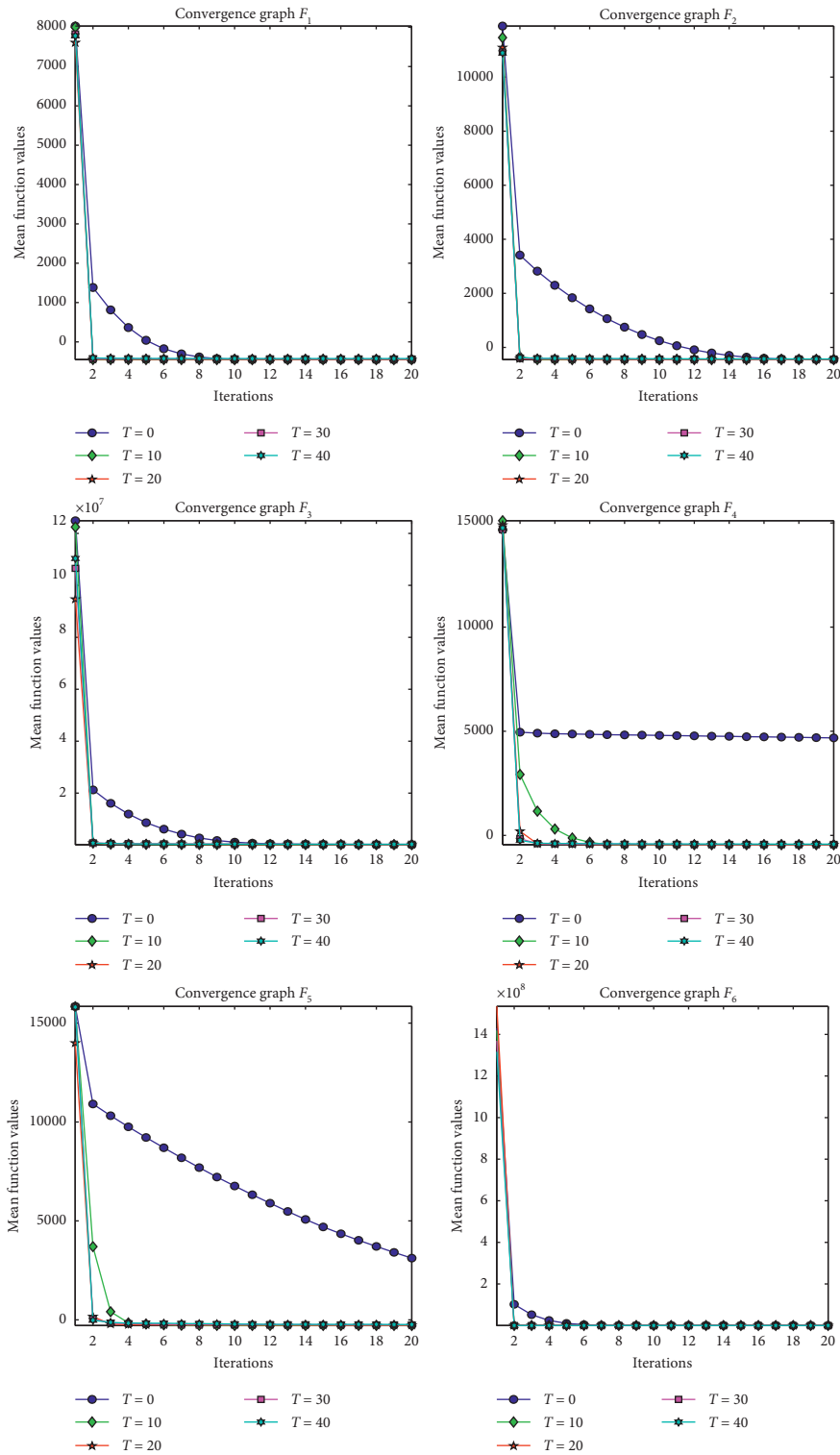


FIGURE 3: The convergence graph displayed by the proposed algorithm for IEEE-CEC'05 benchmark functions referred to as F_1 to F_6 [51].

Wali Khan Mashwani et al.

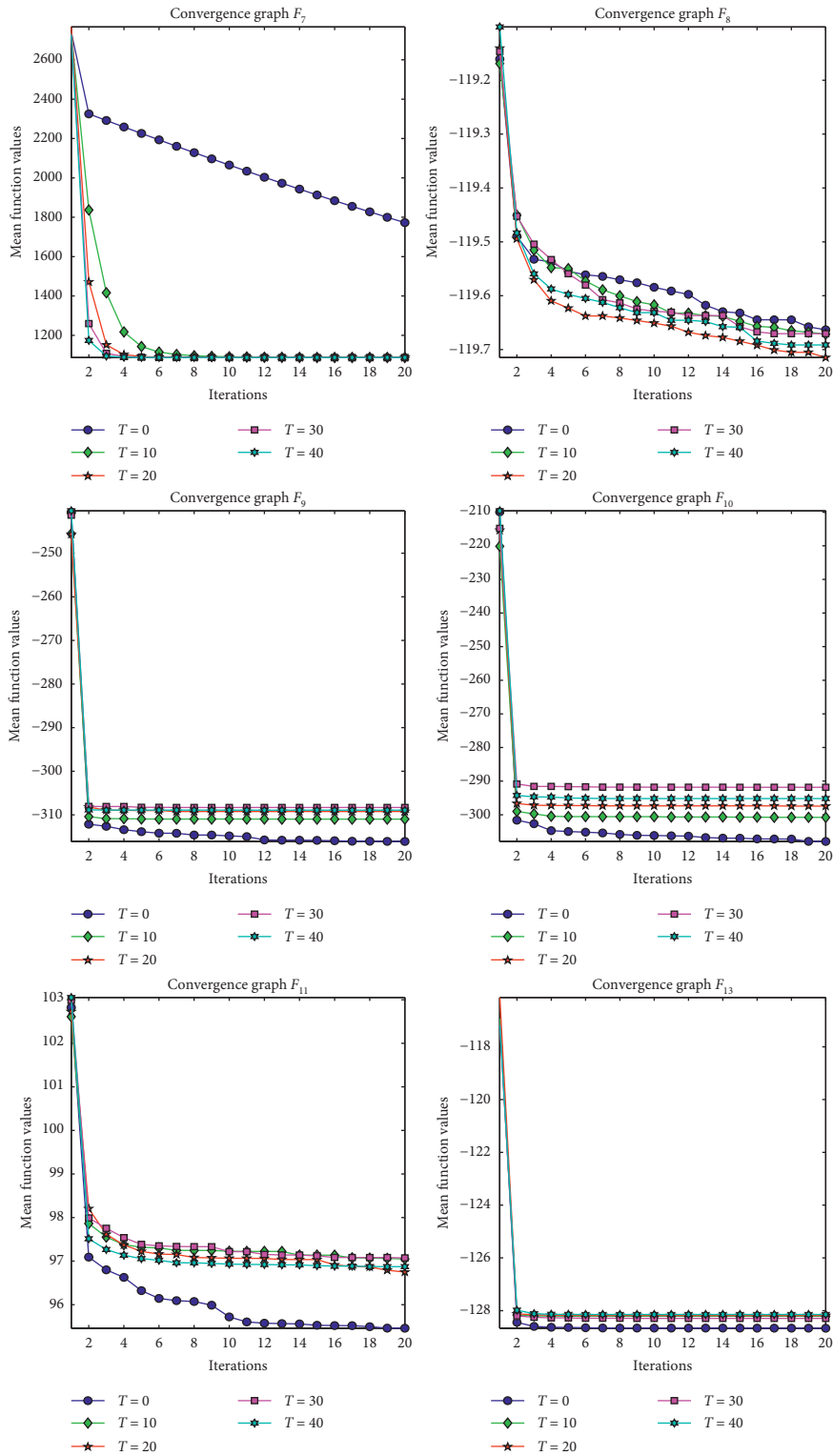


FIGURE 4: The convergence graph displayed by the proposed algorithm for IEEE-CEC'05 benchmark functions referred to as F_7 to F_{13} [51].

Wali Khan Mashwani et al.

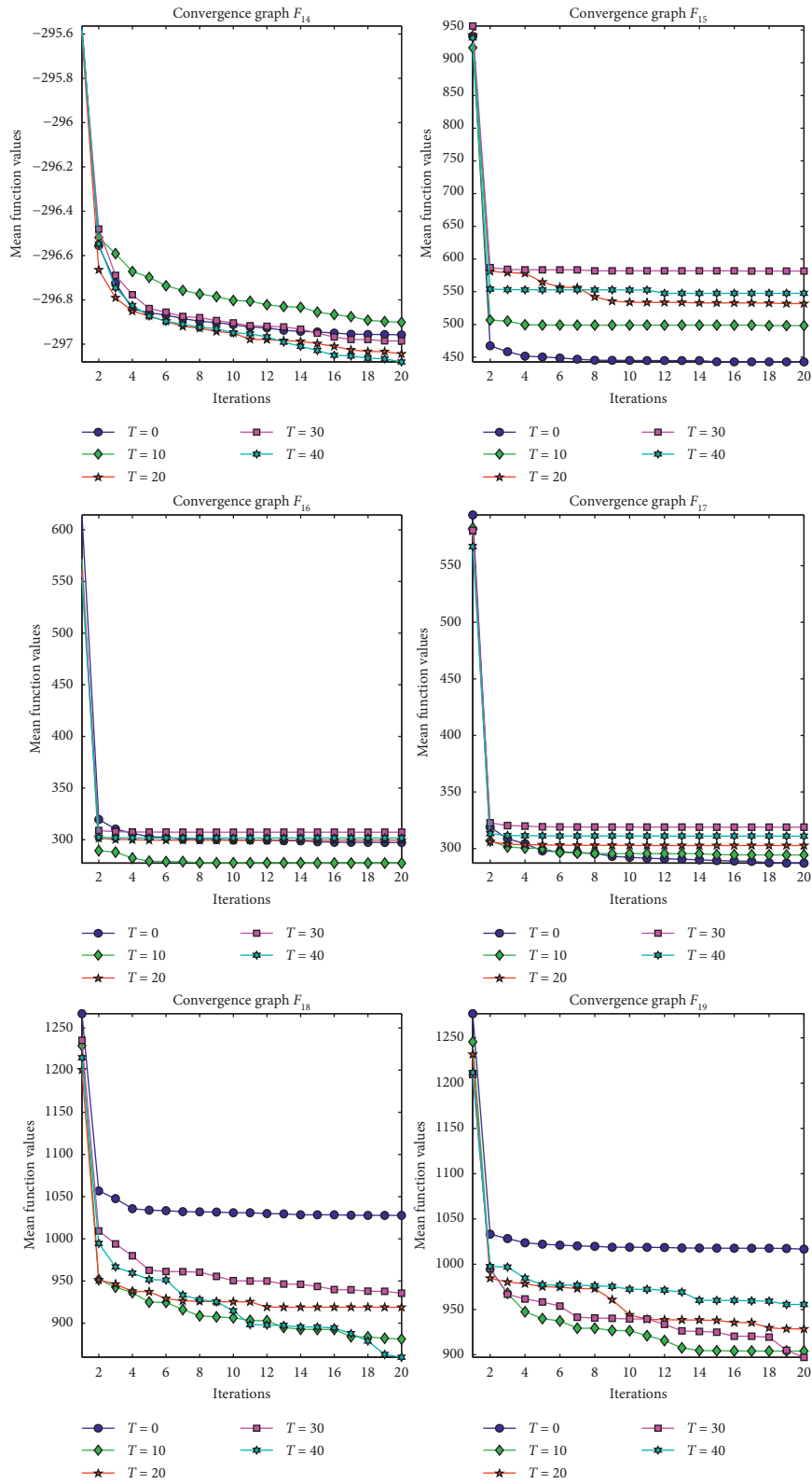


FIGURE 5: The convergence graph displayed by the proposed algorithm for IEEE-CEC'05 benchmark functions referred to as F_{14} to F_{19} [51].

displays the convergence for the test problems denoted by F_{14} to F_{19} in the twenty-five independent runs of executions.

4. Conclusion

Many real-life problems can be modeled as global optimization problems. Evolutionary algorithms (EAs) are population-based optimization methods that operate on a set of uniformly and randomly generated solutions in search space as specified for each optimization problem to be solved. EAs are more flexible and usually provide a high-quality set of solutions in a single simulation run, in contrast to traditional optimization methods. This paper presents a modified Bat algorithm to deal with benchmark functions that were designed for the special session of EAs competition of 2005 IEEE Congress on Evolutionary Computation (IEEE-CEC'05) [51]. The results provided by the modified Bat algorithm (MBA) seem to be more effective for dealing with most benchmark functions in terms of diversity and proximity:

- (i) In MBA, the basic framework of the Bat Algorithm is modified
- (ii) Based on the experiments that we carried out, the speed of sound varied with varying temperature
- (iii) The proposed algorithm has found promising results for most problems at $10C^0$
- (iv) The multimodel problems are tackled efficiently by the proposed algorithm at $0C^0$

In the future, we intend to conduct the following tasks:

- (i) To determine the efficiency, stability, and importance of the proposed MBA by using other existing unconstrained benchmark functions and some real-world problems
- (ii) To examine sensitivity of the different intrinsic parameters used in the framework of the proposed algorithm in order to ensure the correct use of parameters keeping in view the structure of different problems
- (iii) To further evaluate the robustness and efficiency of the proposed algorithm against some state-of-the-art evolutionary algorithms

Data Availability

Data are available upon request from the first author. Furthermore, the Matlab code of the benchmark functions used to conduct this study is available on <https://www3.ntu.edu.sg/home/epnsugan/>.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors thank and acknowledge the support of the Research Intensified Grant Scheme (RIGS) Fasa 1/2019, Vot Number 55192/7, Universiti Malaysia Terengganu.

References

- [1] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, New York, NY, USA, 1968.
- [2] R. E. Miller, *Optimization: Foundations and Applications*, John Wiley & Sons, Hoboken, NJ, USA, 1999.
- [3] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and Biological Systems: Towards a New Bionics*, P. Dario, G. Sandini, and P. Aebischer, Eds., pp. 703–712, Springer, Berlin, Germany, 1993.
- [4] A. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, Princeton, NJ, USA, 2006.
- [5] Z. Yoshida, *Nonlinear Science: The Challenge of Complex Systems*, Springer Science & Business Media, Berlin, Germany, 2010.
- [6] S. Wang and Z. Ma, "Supervisory and optimal control of building HVAC systems: a review," *HVAC&R Research*, vol. 14, no. 1, pp. 3–32, 2008.
- [7] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*, Springer Science & Business Media, Berlin, Germany, 2010.
- [8] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer Publishing Company, Berlin, Germany, 2nd edition, 2015.
- [9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing, Boston, MA, USA, 1st edition, 1989.
- [10] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, New York, NY, USA, 1996.
- [11] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*, Lulu Enterprises, UK Ltd, Morrisville, NC, USA, 2008.
- [12] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.
- [13] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [14] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*, Springer-Verlag, Berlin, Germany, 2005.
- [15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95—International Conference on Neural Networks*, vol. 4, Perth, Australia, 1995.
- [16] Z. W. Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*, Springer Publishing Company, Berlin, Germany, 1st edition, 2009.
- [17] S. Yazdani, H. Nezamabadi-pour, and S. Kamyab, "A gravitational search algorithm for multimodal optimization," *Swarm and Evolutionary Computation*, vol. 14, pp. 1–14, 2014.
- [18] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [19] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
- [20] B. Xing and W.-J. Gao, "Invasive weed optimization algorithm," in *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, pp. 177–181, Springer, Berlin, Germany, 2014.
- [21] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.

- [22] C. Blum, "Ant colony optimization: introduction and recent trends," *Physics of Life Reviews*, vol. 2, no. 4, pp. 353–373, 2005.
- [23] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, *The Bees Algorithm, Technical Note, Manufacturing Engineering Centre*, Cardiff University, Cardiff, UK, 2005.
- [24] N. F. Johari, A. Zain, N. Mustaffa, and A. Udin, "Firefly algorithm for optimization problem," *Applied Mechanics and Materials*, vol. 421, 2013.
- [25] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp. 65–74, Springer, Berlin, Germany, 2010.
- [26] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing*, IEEE, Coimbatore, India, 2009.
- [27] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proceedings of the 2012 International Conference on Unconventional Computing and Natural Computation*, Springer, Orleans, France, 2012.
- [28] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [29] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [30] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "SALP swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
- [31] J. Kim, T. Sharma, B. Kumar, G. Tomar, K. Berry, and W. Hyung, "Research article intercluster ant colony optimization algorithm for wireless sensor network in dense environment," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 457402, 10 pages, 2014.
- [32] W. Fang, X. Li, M. Zhang, and M. Hu, "Nature-inspired algorithms for real-world optimization problems," *Journal of Applied Mathematics*, vol. 2015, Article ID 359203, 2 pages, 2015.
- [33] X.-S. Yang, *Nature-Inspired Meta-Heuristic Algorithms*, Luniver Press, Frome, UK, 2010.
- [34] J. Xie, Y. Zhou, and H. Chen, "A novel bat algorithm based on differential operator and Lévy flights trajectory," *Computational Intelligence and Neuroscience*, vol. 2013, Article ID 453812, 13 pages, 2013.
- [35] X.-S. Yang, *Nature-Inspired Algorithms and Applied Optimization*, Vol. 744, Springer, Berlin, Germany, 2017.
- [36] S. Gupta and K. Deep, "A hybrid self-adaptive sine cosine algorithm with opposition based learning," *Expert Systems with Applications*, vol. 119, pp. 210–230, 2019.
- [37] S. Gupta and K. Deep, "Improved sine cosine algorithm with crossover scheme for global optimization," *Knowledge-Based Systems*, vol. 165, pp. 374–406, 2019.
- [38] S. Gupta and K. Deep, "Hybrid grey wolf optimizer with mutation operator," in *Soft Computing for Problem Solving*, pp. 961–968, Springer, Berlin, Germany, 2019.
- [39] S. Gupta, K. Deep, A. A. Heidari, H. Moayedi, and H. Chen, "Harmonized SALP chain-built optimization," *Engineering with Computers*, pp. 1–31, 2019.
- [40] S. Gupta, K. Deep, and A. P. Engelbrecht, "A memory guided sine cosine algorithm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 93, Article ID 103718, 2020.
- [41] W. K. Mashwani, A. Hamdi, M. A. Jan, and F. Khan, "Large-scale global optimization based on hybrid swarm intelligence algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 1, pp. 1257–1275, 2021.
- [42] W. K. Mashwani, S. N. A. Shah, S. B. Belhaouari, and A. Hamdi, "Ameliorated ensemble strategy-based evolutionary algorithm with dynamic resources allocations," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 412–437, 2021.
- [43] W. K. Mashwani and A. Salhi, "A decomposition-based hybrid multiobjective evolutionary algorithm with dynamic resource allocation," *Applied Soft Computing*, vol. 12, no. 9, pp. 2765–2780, 2012.
- [44] W. K. Mashwani and A. Salhi, "Multiobjective memetic algorithm based on decomposition," *Applied Soft Computing*, vol. 21, pp. 221–243, 2014.
- [45] W. K. Mashwani, "MOEA/D with DE and PSO: MOEA/D-DE + PSO," in *Proceedings of the 31st SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 217–221, Cambridge, UK, December 2011.
- [46] W. K. Mashwani, "Comprehensive survey of the hybrid evolutionary algorithms," *International Journal of Applied Evolutionary Computation (IJAEC)*, vol. 4, no. 2, pp. 1–19, 2013.
- [47] W. K. Mashwani, "Hybrid multiobjective evolutionary algorithms: a survey of the state-of-the-art," *International Journal of Computer Science Issues*, vol. 8, no. 6, pp. 374–392, 2011.
- [48] F. G. Lobo, C. F. Lima, and Z. Michalewicz, *Parameter Setting in Evolutionary Algorithms*, Springer Publishing Company, Berlin, Germany, 1st edition, 2007.
- [49] K. DeJong, "Parameter setting in EAS: a 30 year perspective," in *Parameter Setting in Evolutionary Algorithms, Ser. Studies in Computational Intelligence*, F. G. Lobo, C. F. Lima, and Z. Michalewicz, Eds., vol. 54, pp. 1–18, Springer, Berlin, Germany, 2007.
- [50] D. Thierens, "Adaptive strategies for operator allocation," in *Parameter Setting in Evolutionary Algorithms, Ser. Studies in Computational Intelligence*, F. G. Lobo, C. F. Lima, and Z. Michalewicz, Eds., vol. 54, pp. 77–90, Springer, Berlin, Germany, 2007.
- [51] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Technical report, Nanyang Technological University, Singapore, 2005.
- [52] H. Ahmed and J. Glasgow, "Swarm intelligence: concepts, models and applications," Technical report, School of Computing, Queens University, Kingston, Canada, 2012.
- [53] J. R. Rohan, C. N. Prasad, and P. Sadiq, "An introduction to the collective behaviour of swarm intelligence," in *Proceedings of the 2017 National Conference on Contemporary Research and Innovations in Computer Science*, St. Joseph's Evening College, Bengaluru, India, December 2017.
- [54] X.-S. Yang, "Swarm intelligence based algorithms: a critical analysis," *Evolutionary Intelligence*, vol. 7, no. 1, pp. 17–28, 2014.
- [55] B. Yuce, M. Packianather, E. Mastrocinque, D. Pham, and A. Lambiasi, "Honey bees inspired optimization method: the bees algorithm," *Insects*, vol. 4, no. 4, pp. 646–662, 2013.
- [56] L. Li and F. Liu, *Group Search Optimization for Applications in Structural Design*, Vol. 9, Springer, Berlin, Germany, 2011.
- [57] A. Joshi, O. Kulkarni, G. Kakandikar, and V. Nandedkar, "Cuckoo search optimization—a review," *Materials Today*, vol. 4, pp. 7262–7269, 2017.

- [58] M. Mareli and B. Twala, "An adaptive cuckoo search algorithm for optimisation," *Applied Computing and Informatics*, vol. 14, no. 2, pp. 107–115, 2018.
- [59] M. I. Solihin and M. F. Zanil, "Performance comparison of cuckoo search and differential evolution algorithm for constrained optimization," *IOP Conference Series: Materials Science and Engineering*, vol. 160, Article ID 012108, 2016.
- [60] X.-S. Yang, *A New Metaheuristic Bat-Inspired Algorithm*, Springer, Berlin, Germany, 2010.
- [61] X. S. Yang and X. He, "Bat algorithm: literature review and applications," *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, pp. 141–149, 2013.
- [62] M. Chawla and M. Duhan, "Bat algorithm: a survey of the state-of-the-art," *Applied Artificial Intelligence*, vol. 29, no. 6, pp. 617–634, 2015.
- [63] W. H. Bangyal, J. Ahmad, H. T. Rauf, and S. Pervaiz, "An improved bat algorithm based on novel initialization technique for global optimization problem," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 7, pp. 158–166.
- [64] N. Dey and A. S. Ashour, *Chapter 1—Computing in Medical Image Analysis*, Academic Press, Cambridge, MA, USA, 2018.
- [65] B. S. Finn, "Laplace and the speed of sound," *ISIS*, vol. 55, no. 1, pp. 7–19, 1964.