Research Article

A Novel Quantum-Behaved Bat Algorithm with Mean Best Position Directed for Numerical Optimization

Binglian Zhu,¹ Wenyong Zhu,¹ Zijuan Liu,¹ Qingyan Duan,¹ and Long Cao^{1,2}

¹Communications Engineering, Chongqing University, Chongqing 400030, China

²Jiuquan Satellite Launch Center, Jiuquan 732750, China

Correspondence should be addressed to Binglian Zhu; zhubl@cqu.edu.cn

Received 26 January 2016; Revised 11 April 2016; Accepted 26 April 2016

Academic Editor: Christian W. Dawson

Copyright © 2016 Binglian Zhu 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.

This paper proposes a novel quantum-behaved bat algorithm with the direction of mean best position (QMBA). In QMBA, the position of each bat is mainly updated by the current optimal solution in the early stage of searching and in the late search it also depends on the mean best position which can enhance the convergence speed of the algorithm. During the process of searching, quantum behavior of bats is introduced which is beneficial to jump out of local optimal solution and make the quantum-behaved bats not easily fall into local optimal solution, and it has better ability to adapt complex environment. Meanwhile, QMBA makes good use of statistical information of best position which bats had experienced to generate better quality solutions. This approach not only inherits the characteristic of quick convergence, simplicity, and easy implementation of original bat algorithm, but also increases the diversity of population and improves the accuracy of solution. Twenty-four benchmark test functions are tested and compared with other variant bat algorithms for numerical optimization the simulation results show that this approach is simple and efficient and can achieve a more accurate solution.

1. Introduction

In recent years, with the need of optimization problems in reality, all kinds of bioinspired optimization algorithms or swarm intelligence optimization algorithms have been proposed, such as the genetic algorithm (GA) [1], differential evolution (DE) [2], ant colony optimization (ACO) [3], firefly algorithm (FA) [4, 5], cuckoo search (CS) algorithm [6, 7], particle swarm optimization (PSO) [8], artificial bee colony (ABC) optimization [9], and bat algorithm (BA) [10, 11]. These bionic intelligent algorithms are random search methods which mimic natural biological systems [12]; entirely instinct depends on the organism itself, by the unconscious optimization behavior to optimize its survival to adapt to the environment. Compared with the traditional optimal algorithms, they do not depend on the characteristic of strict mathematical optimization problem itself and have the characteristic of strong parallelism. Each individual has self-organization and evolution; as a result, some high

dimensional optimization problems are superior to the traditional methods.

The bat algorithm is a new metaheuristic method which was proposed by Yang in 2010 [10, 11]. The capability of echolocation of microbats is fascinating as these bats can find their prey and discriminate different types of insects even in complete darkness [13]. Inspired by the echolocative behavior of bats, this algorithm carries the search process using artificial bats as search agents mimicking the natural pulse loudness and emission rate of real bats. When these bats are chasing prey, they tend to decrease the loudness and increase the rate of pulse emission. Bat algorithm has a good optimization performance in low dimensional case [14–16] and is widely used in engineering optimization [17, 18] and multiobjective optimization [19]; however, due to low population diversity, it severely suffers from premature convergence problem in the high dimensional case [20]. So many variant bat algorithms are proposed to enhance the population diversity and to avoid being trapped into local

optimum. In [21], Lin et al. put forward a chaotic Levy flight bat algorithm (CLBA) for parameter estimation in nonlinear dynamic biological system; in [13], Xie et al. introduced the difference operator and Levy flight trajectory into the bat algorithm (DLBA) to solve function optimization problems; in [22], Yilmaz and Kucuksille, inspired by standard particle swarm optimization (PSO) [8] algorithm and artificial bee colony (ABC) algorithm [9], proposed improved bat algorithm (IBA). In IBA, the velocity of each bat is updated with linear decreasing inertia weight factor and the frequency is self-adaptive to improve the exploration and exploitation. In [23], Wang and Guo put the harmony search method into the bat algorithm and developed a hybrid metaheuristic HSBA method for optimization problem, speeding up convergence of bat algorithm; in [24], Yilmaz et al. have studied the mechanism of updating loudness and pulse emission rate of BA and they found that loudness and pulse emission rate provide a balance between exploitation and exploration. So they modify the equations to improve the exploration capability of BA. Afrabandpey et al. [25] introduced the chaotic sequences into bat algorithm in different ways to avoid premature convergence. In [26], four parameters of BA are replaced by the chaotic maps separately; meanwhile, in [20], loudness and pulse emission rate are tuned via multiplying a linear decreasing or increasing function by a chaotic map function; in [25], chaotic map takes place of random number for parameter initialization. These approaches, to some extent, can avoid getting trapped into local minimum. However, in QMBA, bats adopt different search strategies in different times and have the mechanism of jumping out of local optimal solution; these strategies enhance the convergence speed of the algorithm and improve the accuracy of solutions.

The rest of paper is organized as follows. Section 2 describes the standard BA and Section 3 presents the quantum-behaved bat algorithm with the direction of mean best position. The simulation and comparison of this proposed algorithm are presented in Section 4. Finally, general conclusions are drawn in Section 5.

2. The Bat Algorithm

Echolocation is a very important character of bats; Yang proposed bat algorithm by mimics of bats' foraging behavior. Bats fly randomly in the air or in the process of searching for prey by using echolocation to catch food and to avoid obstacles. In order to transform these behaviors of bats to algorithm, there are some approximations and idealized rules [10].

- (i) All bats use echolocation to sense distance, and they also "know" the difference between food/prey and background barriers in some magical way.
- (ii) Bats fly randomly with velocity v_i at position x_i with a fixed frequency $f_{\min}$, varying wavelength λ , and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target.

- (iii) Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value $A_{\min}$.

In the BA, for the i th bats of swarm having position x_i (solution), velocity v_i , and frequency f_i , each bat will move toward the current best position (solution), and its position, velocity, and frequency are updated during the course of iteration as follows:

$$\begin{aligned} f_i &= f_{\min} + (f_{\max} - f_{\min})\beta, \\ v_i^t &= v_i^{t-1} + (x_i^{t-1} - x_g^{t-1})f_i, \\ x_i^t &= x_i^{t-1} + v_i^t, \end{aligned} \quad (1)$$

where β is a random number of a uniform distribution in $[0, 1]$ and x_g^{t-1} represents the current global best solution (position) after comparing all the solutions (positions) among all the n bats. These equations can guarantee the exploration ability of BA.

For the local search, when a solution is selected among the current best solutions, a new candidate solution can be generated as

$$x_{\text{new}} = x_{\text{old}} + \varepsilon \bar{A}^t, \quad (2)$$

where ε is a random number in $[-1, 1]$ and directs new solution apart from or close to the current best solution. Here, $\bar{A}^t$ is mean value of all bats of loudness.

When finding prey, bat will gradually decrease the loudness and increase the rate of pulse emission in order to track its prey to capture it. The loudness and pulse emission rate update accordingly as the iterations proceed as shown in

$$\begin{aligned} A_i^t &= \alpha A_i^{t-1}, \\ r_i^t &= r_i^0 (1 - \exp(-\gamma t)), \end{aligned} \quad (3)$$

where α and γ are constants. In fact, the α parameter controls the convergence of bat algorithm and therefore plays a similar role as the cooling factor in the simulated annealing algorithm [27]. For simplicity, we set $\alpha = \gamma = 0.9$ in our simulations.

The basic steps of BA can be summarized as the pseudocode shown in Algorithm 1.

3. Quantum-Behaved Bat Algorithm with the Direction of Mean Best Position

The standard bat algorithm has quick convergence and is easy to be implemented; therefore, it has been widely applied in practical engineering. However, BA is more easily to fall into local optimal point when optimizing the multimodal functions. There are many reasons for prematuration of BA. Firstly, through the analysis of the trajectory of bats, we found that many bats are trapped into local optimal point because of decreasing diversity of bats. Secondly, other bats are directed only by current optimal solution, if only the best bat falls into local point and it will misguide others.

- (1) Initialize the bat population $\mathbf{x}_i$ ($i = 1, 2, \dots, n$) and $\mathbf{v}_i$.
- (2) Define pulse frequency f_i , pulse rate r_i and the loudness A_i .
- (3) **While** ($t < \text{max_iteration}$)
- (4) **for** $i = 1$ **to** n
- (5) Generate new solutions by adjusting frequency, updating velocities and positions [equations (1)]
- (6) **if** ($\text{rand} > r_i$)
- (7) Select a solution among the best solutions randomly
- (8) Generate a local solution around the selected best solution
- (9) **end if**
- (10) Generate a new solution by flying randomly
- (11) **if** ($\text{rand} < A_i \ \&\& \ f(x_i) < f(x_g)$)
- (12) Accept the new solutions
- (13) Increase r_i and reduce A_i
- (14) **end if**
- (15) **end for**
- (16) Rank the bats and find the current best x_g .
- (17) **end while**
- (18) Output results and visualization

ALGORITHM 1: Bat algorithm (BA) pseudocode.

FIGURE 1: The curve of fitness value for F1.

FIGURE 2: The curve of fitness value for F2.

Thirdly, there are no mechanisms to jump out of local optima in BA. In order to solve the above problem, the behavior of quantum of bats is introduced into the algorithm to increase the diversity of population and it also contributes to avoiding prematuration. The current global optimal solution is used to guide other bats flying in the early stage of searching, while mean best position is used in later stage of searching. That can enhance the efficiency and convergence speed of BA.

Our proposed algorithm (QMBA) is based on the basics of framework of original bat algorithm. The parameters r and A control the exploitation and exploration, respectively,

by updating the two factors to guide BA into local search or global search; however, the new candidate solutions are generated by the following formulas which are different from original BA:

$$x_{i,j}^t = \begin{cases} x_{i,j}^{t-1} + (x_j^* - x_{i,j}^{t-1}) \text{rand}, & \delta_j > \text{TH}, \\ x_{i,j}^{t-1} + \varepsilon, & \delta_j \leq \text{TH}, \end{cases} \quad (4)$$

where

$$\delta_j = |x_j^* - x_{i,j}^{t-1}|, \quad (5)$$

FIGURE 3: The curve of fitness value for F3.

FIGURE 5: The curve of fitness value for F5.

FIGURE 4: The curve of fitness value for F4.

FIGURE 6: The curve of fitness value for F6.

and it indicates the distance between the position of j th dimension of i th bat and the j th dimension of current optimal position among all bats, and rand is a random number in $[0, 1]$. If δ_j is greater than the threshold TH , it means the distance between i th bat and current optimal position is far; hence, the current bat moves toward optimal position so far by random step. However, if δ_j is less than the threshold TH , it suggests that the current bat is nearby the current optimal position; therefore, the bat flies randomly. The diversity of bat population and the exploration ability are improved by self-adapting the step in terms of distance.

During the process of search, according to certain probability of mutation p_m , some bats will be mutated with quantum behavior [28, 29]; these bats are updated with the following formulas:

$$x_i^t = \begin{cases} x_{\text{best}}^t + \alpha |M_{\text{best}}^t - x_i^{t-1}| \ln\left(\frac{1}{\text{rand}}\right), & \text{rand} > 0.5, \\ x_{\text{best}}^t - \alpha |M_{\text{best}}^t - x_i^{t-1}| \ln\left(\frac{1}{\text{rand}}\right), & \text{rand} \leq 0.5, \end{cases} \quad (6)$$

FIGURE 7: The curve of fitness value for F7.

FIGURE 8: The curve of fitness value for F8.

where α is the contraction-expansion coefficient defined as

$$\alpha = \alpha_0 - \frac{\alpha_0 - \alpha_1}{T}t. \quad (7)$$

α_0 and α_1 are the initial and the final values of α , respectively, and T is the maximum number of cycles. We usually set $\alpha_0 = 1$ and $\alpha_1 = 0.5$ to obtain good performance in general [30].

M_{best}^t is the mean best position defined as the average of P_i^t positions of all bats. That is,

$$M_{\text{best}}^t = \frac{1}{M} \left(\sum_{i=1}^M P_{i,1}^t, \sum_{i=1}^M P_{i,2}^t, \sum_{i=1}^M P_{i,3}^t, \dots, \sum_{i=1}^M P_{i,D}^t \right), \quad (8)$$

FIGURE 9: The curve of fitness value for F9.

FIGURE 10: The curve of fitness value for F10.

where P_i^t indicate the best position of the i th bat experienced, M is the population size, and D is the dimension of problem. Bat with quantum behavior increases the diversity of population and contributes to jumping out of the local optima.

In the late state of searching, positions of bats are updated as follows:

$$x_i^t = x_i^{t-1} + (M_{\text{best}} - x_i^{t-1}) * \text{rand}. \quad (9)$$

The mean best position is used to guide other bats flying in the late stage of searching; it improves the accuracy of solutions and speeds up the convergence of the algorithm because of using the statistical information of better position of bats. The

```

(1) Initialize the bat population  $\mathbf{x}_i$  ( $i = 1, 2, \dots, n$ ) and  $\mathbf{v}_i$ 
(2) Define pulse frequency  $f_i$ , pulse rate  $r_i$  and the loudness  $A_i$ 
(3) While ( $t < \text{max\_iteration}$ )
(4)   for  $i = 1$  to  $n$ 
(5)     Generate new solutions by calculating the distance between the bat and current global position, updating positions
     as equations (4)-(5).
(6)     if ( $\text{rand} > r_i$ )
(7)       if ( $\text{rand} > p_m$ )
(8)         Bats fly with quantum behavior and positions updated as (6)~(8).
(9)       else
(10)        The mean best position is used to guide other bats and positions updated as (9).
(11)      end if
(12)    end if
(13)    if ( $\text{rand} < A_i$  &&  $f(x_i) < f(x_g)$ )
(14)      Accept the new solutions
(15)      Increase  $r_i$  and reduce  $A_i$ 
(16)    end if
(17)  Rank the bats and find the current best  $x_g$ .
(18) end while
(19) Output results and visualization

```

ALGORITHM 2: Bat algorithm with mean best position (QMBA) pseudocode.

FIGURE 11: The curve of fitness value for F11.

FIGURE 12: The curve of fitness value for F12.

pseudocode of quantum-behaved bat algorithm with mean best position is shown in Algorithm 2.

4. The Simulations

In order to verify the efficiency of the proposed algorithm, we select twenty-four standard benchmark functions [24, 31, 32] to test the ability to search QMBA. The results are compared with BA and other variants of BA to show the performance of global numerical optimization. For this purpose, we have

used BA, IBA, and HSBA to carry out numerical experiments for 24 standard test benchmarks, and the results will be discussed in this section.

4.1. Benchmark Functions. There are many different functions used to test the performance of algorithms. These benchmark functions can be divided into four categories. Category I is unimodal, category II is multimodal, category III is shifted or rotated, and category IV is composite function. Table 1 lists these functions, respectively, where D indicates the dimension of the function, range is the

TABLE 1: Benchmark function.

Category	Number	Function	D	Range	$f_{\min}$
I	F1	$f(x) = \sum_{i=1}^n x_i^2$	30	$[-5.12, 5.12]$	0
	F2	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0
	F3	$f(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$	0
	F4	$f(x) = \max_i \{ x_i \}, 1 \leq i \leq n$	30	$[-100, 100]$	0
	F5	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
	F6	$f(x) = \sum_{i=1}^n ix_i^4 + \text{random}(0, 1)$	30	$[-1.28, 1.28]$	0
II	F7	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
	F8	$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
	F9	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
	F10	$f(x) = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	30	$[-50, 50]$	0
	F11	$f(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0
	F12	$f(x) = \left\{ \left[\sum_{i=1}^n \sin^2(x_i) \right] - \exp\left(-\sum_{i=1}^n x_i^2\right) \right\} * \exp\left[-\sum_{i=1}^n \sin^2 \sqrt{ x_i }\right]$	30	$[-10, 10]$	-1
III	F13	$f(x) = \sum_{i=1}^n z_i$	30	$[-100, 100]$	0
	F14	$f(x) = \sum_{i=1}^n [100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2]$	30	$[-100, 100]$	0
	F15	$f(x) = \sum_{i=1}^n [z_i^2 - 10 \cos(2\pi z_i) + 10]$	30	$[-5, 5]$	0
	F16	$f(x) = \sum_{i=1}^n (10^6)^{(i-1)/(D-1)} z_i^2$	30	$[-100, 100]$	0
	F17	$f(x) = \frac{1}{4000} \sum_{i=1}^n z_i^2 - \prod_{i=1}^n \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1$	30	$[0, 600]$	0
	F18	$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n z_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi z_i)\right) + 20 + e$	30	$[-32, 32]$	0

TABLE 1: Continued.

Category	Number	Function	D	Range	$f_{\min}$
IV	F19 (CF1)	Hybrid Composition Function	30	[-5, 5]	0
		f_1, f_2 = Rastrigin's Function			
		f_3, f_4 = Weierstrass Function			
	F20 (CF2)	f_5, f_6 = Griewank's Function	30	[-5, 5]	0
		f_7, f_8 = Ackley's Function			
		f_9, f_{10} = Sphere Function			
IV	F21 (CF3)	$[\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_{10}] = [1, 1, 1, \dots, 1]$	30	[-5, 5]	0
		$[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [1, 1, 10, 10, 5/60, 5/60, 5/32, 5/32, 5/100, 5/100]$			
		Rotated version of Hybrid Composition Function F19			
	F22 (CF4)	Except M_i which are different linear transformation matrixes with condition number of 2, all other settings are the same as F19	30	[-5, 5]	0
		F20 with noise in fitness			
		Let F20 be $G(x)$; then $F_{21}(x) = G(x) * (1 + 0.2 N(0, 1))$			
IV	F23 (CF5)	All settings are the same as F20	30	[-5, 5]	0
		Rotated Hybrid Composition Function			
		f_1, f_2 = Ackley's Function			
	F24 (CF6)	f_3, f_4 = Rastrigin's Function	30	[-5, 5]	0
		f_5, f_6 = Sphere Function			
		f_7, f_8 = Weierstrass Function			
IV	F25 (CF7)	$a = 0.5, b = 0.3, k_{\max} = 20$	30	[-5, 5]	0
		f_9, f_{10} = Griewank's Function			
		$[\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_{10}] = [1, 2, 1.5, 1.5, 1, 1, 1.5, 1.5, 2, 2]$			
	F26 (CF8)	$\lambda = [2 * 5/32, 2 * 5/32, 2 * 1, 1, 2 * 5/100, 5/100, 2 * 10, 10, 2 * 5/60, 5/60]$	30	[-5, 5]	0
		M_i are all rotation matrixes. Condition numbers are $[2 \ 3 \ 2 \ 3 \ 2 \ 3 \ 20 \ 200 \ 300]$			
		$\mathbf{o}_{10} = [0, 0, \dots, 0]$			
IV	F27 (CF9)	Rotated Hybrid Composition Function with narrow basin global optimum	30	[-5, 5]	0
		All settings are the same as F22 except			
		$[\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_{10}] = [0.1, 2, 1.5, 1, 1, 1.5, 1.5, 2, 2]$			
	F28 (CF10)	$\lambda = [2 * 5/32, 2 * 5/32, 2 * 1, 1, 2 * 5/100, 5/100, 2 * 10, 10, 2 * 5/60, 5/60]$	30	[-5, 5]	0
		$\mathbf{o}_{10} = [1, 0, \dots, 0]$			
		Rotated Hybrid Composition Function with global optimum on the bounds			
IV	F29 (CF11)	All settings are the same as F23 except, after loading the data file,	30	[-5, 5]	0
		set $\sigma_{1(2,j)} = 5$, for $j = 1, 2, \dots, D/2$			
	F30 (CF12)		30	[-5, 5]	0

FIGURE 13: The curve of fitness value for F13.

FIGURE 14: The curve of fitness value for F14.

boundary of the function's search space, and $f_{\min}$ is the minimum value of the function. Unimodal function (F1~F6) has only one extreme point and these algorithms can easily find the point; however, in multimodal function (F7~F12) there will be many local minimums; it is relatively difficult to find the global minimum. As the global minimum of most benchmark functions are zero for all dimensions, therefore, we select some shifted and rotated functions to test the algorithm's robustness. For shifted and rotated functions (F13~F18) which have different parameter values for different

FIGURE 15: The curve of fitness value for F15.

FIGURE 16: The curve of fitness value for F16.

dimensions, as can be seen in Table 1, F13, F14, and F15 are shifted functions whose global optimum is shifted to random positions to make different parameter values for different dimensions, where $\mathbf{z}_i = \mathbf{x}_i - \mathbf{o}_i$, $\mathbf{o}_i$ define the new shifted optimum position for primitive test function. The other way is rotating the functions by using $F(x) = f(R * x)$ formula, where R is an orthogonal rotation matrix, such as F16, F17, and F18. Finally, composite test function (F19~F24) combines different other test functions by stretching and manipulating their coverage range. It should be noted that $\mathbf{M}_i$ define the linear transformation matrix for each $f_i(x)$, σ_i is used to

FIGURE 17: The curve of fitness value for F17.

FIGURE 19: The curve of fitness value for F19.

FIGURE 18: The curve of fitness value for F18.

FIGURE 20: The curve of fitness value for F20.

control each $f_i(x)$'s coverage range, and λ_i indicate the stretch or compression level of the primitive test functions (see [32] for the detailed description of the composite functions). These composite functions provide challenge to find the global optimum, but that can verify the searching capability of the algorithm effectively.

4.2. Parameter Setting. In order to determine whether QMBA algorithm can be as effective as BA and other variants of BA, we compared its performance on numerical optimization with BA and other variants of BA, which include BA with inertia weight (IBA) [22], modified bat algorithm (MBA)

[24], BA with harmony search (HSBA) [23], and bat algorithm based on chaotic map (CBA) [25].

All these algorithms are tested with 30 independent runs, the number of bats in population is fixed to 50, the dimension of problem is 30, and maximum number of iterations is set to 900 except HSBA which is 300, so these algorithms have reached 45000 FEs. The other parameter settings of these algorithms are given in Table 2.

4.3. Comparison of Experiment Results. The comparisons of test results of BA and other variants of BA are shown in

TABLE 2: The parameter settings of these algorithms.

Algorithms	Parameter design
BA	$f_{\min} = 0, f_{\max} = 2, A = 0.5, r = 0.5, \alpha = 0.95, \gamma = 0.05$
IBA	$f_{\min} = 0, f_{\max} = 1, \alpha = \gamma = 0.9, \text{limit} = 200, w_{\min} = 0.2, w_{\max} = 0.9$
MBA	$A = 1.8, r = 0.8, \alpha = \gamma = 0.9$
HSBA	$A = 0.95, r = 0.6, \varepsilon = 0.1, \text{PAR} = 0.1, \text{HCMR} = 0.95, \text{bw} = 0.5, \text{Keep} = 2$
CBA	A and r are initialized by chaotic maps; $\alpha = \gamma = 0.9$
QMBA	$A = 0.25, r = 0.5, \alpha = \gamma = 0.9, \text{TH} = 0.005$

TABLE 3: Experimental results of unimodal benchmark functions by algorithms (best results in bold).

Algorithms		F1	F2	F3	F4	F5	F6
BA	Mean	1.0520e + 02	2.2066e + 10	6.9423e + 04	8.7571e + 01	2.1222e + 08	1.7977e + 01
	Min	8.4269e + 01	1.0492e + 02	4.0504e + 04	8.7084e + 01	1.8225e + 08	3.4554e + 00
	Max	1.6958e + 02	2.6524e + 11	8.3920e + 04	8.9196e + 01	2.4983e + 08	8.6022e + 01
	Std	2.6294e + 01	6.1313e + 10	1.2163e + 04	4.8092e - 01	1.6327e + 07	2.8488e + 01
IBA	Mean	1.3493e - 03	1.2324e + 01	1.9501e + 03	2.3259e + 01	1.0208e + 02	3.7488e - 02
	Min	1.1048e - 03	2.2161e - 01	2.0172e + 02	1.5919e + 01	2.2386e + 01	2.0379e - 02
	Max	1.7220e - 03	1.0659e + 02	5.8598e + 03	3.3558e + 01	5.8034e + 02	6.1803e - 02
	Std	1.6313e - 04	2.5659e + 01	1.4319e + 03	4.5757e + 00	1.4067e + 02	1.1157e - 02
MBA	Mean	2.2972e + 01	4.0940e + 01	2.4648e + 04	4.7065e + 01	5.5687e + 06	4.6924e + 00
	Min	1.1901e + 01	2.9348e + 01	1.6771e + 04	3.3576e + 01	1.1137e + 06	2.2631e + 00
	Max	3.4470e + 01	5.0545e + 01	3.2960e + 04	5.6452e + 01	1.5316e + 07	9.3272e + 00
	Std	5.9513e + 00	5.3719e + 00	3.9012e + 03	5.5703e + 00	3.1130e + 06	1.7066e + 00
HSBA	Mean	1.4556e - 01	1.6079e + 00	4.0125e + 00	1.1026e + 00	6.4326e + 01	1.3798e - 01
	Min	8.6816e - 02	1.0869e + 00	9.7708e - 01	1.6676e - 01	3.8138e + 01	5.0585e - 02
	Max	1.9280e - 01	1.9813e + 00	6.9179e + 00	6.3851e + 00	1.5983e + 02	2.4781e - 01
	Std	2.9025e - 02	2.0790e - 01	1.2881e + 00	1.5226e + 00	3.4549e + 01	5.0364e - 02
CBA	Mean	1.3400e - 03	1.1039e + 05	7.4202e + 03	3.2386e + 01	1.1515e + 02	3.5127e - 02
	Min	9.5933e - 04	7.8449e + 00	1.6956e + 03	2.1754e + 01	2.3036e + 01	1.2743e - 02
	Max	1.6616e - 03	2.7800e + 06	1.8489e + 04	4.9108e + 01	4.2422e + 02	6.8828e - 02
	Std	1.6009e - 04	4.9947e + 05	3.6881e + 03	7.6207e + 00	1.3531e + 02	1.3576e - 02
QMBA	Mean	7.6898e - 70	6.1368e - 36	1.7587e - 21	5.1883e - 35	2.3768e + 01	4.2512e - 04
	Min	3.3739e - 84	7.3443e - 41	2.2692e - 39	1.7424e - 39	0.0000e + 00	4.2318e - 05
	Max	2.1833e - 68	1.4449e - 34	4.3799e - 20	1.1658e - 33	2.7561e + 01	1.1393e - 03
	Std	3.9168e - 69	2.5906e - 35	7.8785e - 21	2.1000e - 34	7.9530e + 00	3.1788e - 04

Tables 3, 5, 7, and 9. As can be seen from these tables, the mean fitness values, maximum values, minimum values, and standard deviations are obtained in thirty trials.

According to Derrac et al. [33], to improve the evaluation of evolutionary algorithms' performance, statistical tests should be conducted. In other words, it is not enough to compare algorithms based on the mean and standard deviation values. A statistical test is necessary to prove that a proposed new algorithm presents a significant improvement over other existing methods for a particular problem.

In order to judge whether the results of the QMBA differ from the best results of the other algorithms in a statistically

significant way, a nonparametric statistical test, t -test [32, 33], was carried out at a 5% significance level. The p values calculated in t -test comparing QMBA and other algorithms over all the benchmark functions are given in Tables 4, 6, 8, and 10. In these tables, according to Derrac et al. [33], those p values that are less than 0.05 could be considered as strong evidence against the null hypothesis.

In the following subsections, the details of the results and discussion at each group of benchmark functions are provided.

Unimodal functions can evaluate the capability of exploitation of an optimization algorithm because they have

TABLE 4: p values calculated for t -test on unimodal benchmark functions.

Algorithms	F1	F2	F3	F4	F5	F6
BA	$2.317e-29$	$5.748e-02$	$1.367e-37$	$4.476e-124$	$9.919e-58$	$1.232e-03$
IBA	$1.489e-46$	$1.223e-03$	$8.049e-10$	$7.398e-35$	$4.053e-03$	$2.840e-25$
MBA	$1.465e-28$	$1.485e-44$	$5.116e-40$	$4.473e-47$	$1.217e-13$	$2.319e-21$
HSBA	$1.532e-34$	$6.517e-45$	$6.411e-24$	$2.527e-04$	$7.394e-08$	$3.149e-21$
CBA	$7.616e-47$	$2.388e-03$	$1.463e-15$	$1.006e-30$	$5.995e-04$	$6.406e-20$

TABLE 5: Experimental results of multimodal benchmark functions by algorithms (best results in bold).

Algorithms		F7	F8	F9	F10	F11	F12
BA	Mean	$2.2196e+02$	$1.9930e+01$	$5.3680e+02$	$4.2534e+08$	$9.6509e+08$	$2.2439e-03$
	Min	$9.8765e+01$	$1.9905e+01$	$4.2285e+02$	$3.9948e+08$	$8.4923e+08$	$1.5573e-03$
	Max	$3.9100e+02$	$1.9946e+01$	$5.5131e+02$	$4.9814e+08$	$1.0976e+09$	$3.3311e-03$
	Std	$8.8763e+01$	$8.6821e-03$	$3.1930e+01$	$2.6578e+07$	$5.3834e+07$	$4.2665e-04$
IBA	Mean	$1.4073e+02$	$1.4867e+01$	$6.3576e+01$	$1.7971e+01$	$8.9507e+01$	$1.3564e-03$
	Min	$9.7746e+01$	$1.2459e+01$	$3.2813e+01$	$8.0753e+00$	$6.5495e+01$	$1.0861e-03$
	Max	$1.9825e+02$	$1.7726e+01$	$9.7849e+01$	$3.0759e+01$	$1.1203e+02$	$1.6601e-03$
	Std	$3.8512e+01$	$1.3673e+00$	$1.6637e+01$	$5.4157e+00$	$1.2749e+01$	$1.4210e-04$
MBA	Mean	$1.5356e+02$	$1.4910e+01$	$8.0691e+01$	$1.7537e+06$	$8.5116e+06$	$2.2243e+00$
	Min	$1.2267e+02$	$1.1881e+01$	$4.8878e+01$	$4.6576e+03$	$9.4329e+05$	$1.3556e+00$
	Max	$1.8516e+02$	$1.6365e+01$	$1.3017e+02$	$7.9333e+06$	$2.4844e+07$	$3.2322e+00$
	Std	$1.6806e+01$	$1.1256e+00$	$1.8766e+01$	$1.6657e+06$	$5.6320e+06$	$4.9722e-01$
HSBA	Mean	$3.6746e+01$	$6.3046e-01$	$9.1292e-03$	$1.7592e-03$	$2.7202e-02$	$1.0172e-01$
	Min	$1.8996e+01$	$3.7880e-01$	$5.2827e-03$	$9.4920e-04$	$1.2030e-02$	$-6.0711e-01$
	Max	$6.2000e+01$	$3.8051e+00$	$1.3725e-02$	$6.9317e-03$	$3.9820e-02$	$1.6138e-01$
	Std	$1.0779e+01$	$5.9347e-01$	$2.1471e-03$	$1.0197e-03$	$6.1571e-03$	$1.3415e-01$
CBA	Mean	$1.6923e+02$	$1.7117e+01$	$1.2186e+02$	$2.2994e+01$	$9.2521e+01$	$1.3414e-03$
	Min	$6.3918e+01$	$1.3699e+01$	$5.8054e+01$	$8.3428e+00$	$7.3413e+01$	$9.4095e-04$
	Max	$1.9831e+02$	$1.9960e+01$	$2.0184e+02$	$3.6968e+01$	$1.1491e+02$	$1.6389e-03$
	Std	$4.4037e+01$	$1.6093e+00$	$3.8238e+01$	$7.2960e+00$	$1.0007e+01$	$1.7965e-04$
QMBA	Mean	$0.0000e+00$	$8.8818e-16$	$0.0000e+00$	$1.5705e-32$	$1.3498e-32$	$-8.3333e-01$
	Min	$0.0000e+00$	$8.8818e-16$	$0.0000e+00$	$1.5705e-32$	$1.3498e-32$	$-1.0000e+00$
	Max	$0.0000e+00$	$8.8818e-16$	$0.0000e+00$	$1.5705e-32$	$1.3498e-32$	$1.7997e-30$
	Std	$0.0000e+00$	$0.0000e+00$	$0.0000e+00$	$5.4738e-48$	$5.4738e-48$	$2.2439e-03$

TABLE 6: p values calculated for t -test on multimodal benchmark functions.

Algorithms	F7	F8	F9	F10	F11	F12
BA	$1.679e-19$	$6.569e-188$	$3.765e-64$	$6.420e-63$	$9.280e-66$	$1.828e-17$
IBA	$2.390e-27$	$2.701e-53$	$2.451e-28$	$2.936e-25$	$1.462e-42$	$1.910e-17$
MBA	$5.346e-49$	$3.351e-58$	$5.435e-31$	$4.746e-07$	$3.590e-11$	$4.241e-34$
HSBA	$7.736e-26$	$3.915e-07$	$9.787e-31$	$4.412e-13$	$1.311e-31$	$2.058e-18$
CBA	$1.841e-28$	$9.488e-53$	$2.122e-24$	$3.641e-24$	$2.745e-49$	$1.912e-17$

TABLE 7: Experimental results of shifted and rotated benchmark functions by algorithms (best results in bold).

Algorithms		F13	F14	F15	F16	F17	F18
BA	Mean	8.0602e + 04	6.0696e + 10	2.5512e + 02	1.0210e + 09	2.5458e + 03	2.0988e + 01
	Min	6.1153e + 04	3.0545e + 10	1.5947e + 02	6.1482e + 08	1.1580e + 03	2.0797e + 01
	Max	8.7793e + 04	7.1715e + 10	5.3011e + 02	1.5343e + 09	3.6974e + 03	2.1122e + 01
	Std	7.9571e + 03	1.2295e + 10	8.5965e + 01	2.1564e + 08	6.2572e + 02	7.3264e - 02
IBA	Mean	2.4479e + 04	3.0626e + 09	2.5975e + 02	9.6466e + 07	2.7969e + 03	2.0763e + 01
	Min	1.9079e + 04	1.8150e + 09	1.8629e + 02	4.2226e + 07	2.3800e + 03	2.0563e + 01
	Max	3.4241e + 04	5.0137e + 09	3.3371e + 02	1.4886e + 08	3.3734e + 03	2.0916e + 01
	Std	3.4561e + 03	7.8361e + 08	3.9198e + 01	2.9739e + 07	2.1945e + 02	1.0013e - 01
MBA	Mean	9.4981e + 03	9.0058e + 08	1.8004e + 02	1.8433e + 08	4.5504e + 02	2.0980e + 01
	Min	4.3322e + 03	2.4050e + 08	1.2816e + 02	5.0917e + 07	2.2488e + 02	2.0828e + 01
	Max	1.5707e + 04	3.0371e + 09	2.2829e + 02	3.2034e + 08	7.3538e + 02	2.1086e + 01
	Std	3.0561e + 03	5.2168e + 08	2.6217e + 01	7.1234e + 07	1.3193e + 02	6.3286e - 02
HSBA	Mean	1.8143e - 01	3.9316e + 03	2.5801e + 02	1.4158e + 07	1.2396e - 01	2.1028e + 01
	Min	1.1470e - 01	1.4240e + 02	1.8963e + 02	6.2258e + 06	6.1688e - 02	2.0814e + 01
	Max	2.4444e - 01	7.7746e + 03	3.3713e + 02	4.6223e + 07	2.5443e - 01	2.1106e + 01
	Std	2.9407e - 02	2.1534e + 03	3.4819e + 01	8.9814e + 06	3.8659e - 02	6.3454e - 02
CBA	Mean	2.9796e + 04	5.1498e + 09	2.7382e + 02	2.1605e + 08	3.2119e + 03	2.0789e + 01
	Min	2.1297e + 04	2.1656e + 09	1.9228e + 02	8.7480e + 07	2.4267e + 03	2.0628e + 01
	Max	3.7477e + 04	1.0308e + 10	3.3493e + 02	4.2612e + 08	3.8657e + 03	2.0942e + 01
	Std	4.2828e + 03	2.0642e + 09	3.8427e + 01	8.9164e + 07	3.8695e + 02	7.9736e - 02
QMBA	Mean	3.5374e - 01	3.6205e + 03	1.9473e + 02	8.5901e + 06	2.1283e + 00	2.1008e + 01
	Min	1.1148e - 01	2.6836e + 02	7.8661e + 01	3.6992e + 06	1.0633e + 00	2.0885e + 01
	Max	1.0003e + 00	1.1367e + 04	2.5432e + 02	1.3569e + 07	4.6080e + 00	2.1097e + 01
	Std	2.3888e - 01	3.2243e + 03	3.6201e + 01	2.6158e + 06	9.5384e - 01	5.1029e - 02

TABLE 8: p values calculated for t -test on shifted and rotated benchmark functions.

Algorithms	F13	F14	F15	F16	F17	F18
BA	1.532e - 51	3.562e - 34	9.394e - 04	5.246e - 33	1.016e - 29	2.304e - 01
IBA	8.960e - 43	7.742e - 29	1.588e - 08	9.590e - 23	3.212e - 57	5.906e - 17
MBA	7.156e - 24	4.317e - 13	8.200e - 02	3.125e - 19	5.487e - 26	7.426e - 02
HSBA	2.918e - 02	6.673e - 01	6.727e - 09	2.194e - 03	2.696e - 16	1.816e - 01
CBA	2.432e - 42	1.857e - 19	4.712e - 11	3.896e - 18	1.265e - 46	5.559e - 18

only one global optimum. For function F1, Sphere Function, only QMBA could get a solution near the global optimal solution, and other algorithms are trapped into local minimum. F2 is Schwefel's problem 2.22. For this function, HSBA and QMBA outperform all the other algorithms; however, QMBA offered the highest accuracy of solutions. For function F3, Schwefel's problem 1.2, this function is difficult to optimize because there are too many terms for high dimensional ones, but the mean fitness value of QMBA was the best among all the algorithms. In the results for F4, QMBA performed significantly better than any other algorithm. The fifth function F5 is known as Rosenbrock function which has a very narrow valley from local optimum to global optimum. So the algorithms are easily trapped into local optimum. But QMBA obtained the best solution among all the algorithms and its advantages over the competitors were statistically

significant. In results obtained for F6, QMBA offered the highest accuracy. As can be seen in Table 6, the p values indicate that QMBA achieves significant improvement in all the unimodal benchmark functions compared to other algorithms.

The results of the multimodal functions (F7-F12) are provided in Table 5. It should be noted that these benchmark functions have many local optimums with the number increasing exponentially with dimension, so they are useful for evaluating the exploration ability of an optimization algorithm. As observed in Tables 5 and 6, QMBA performs better than other algorithms in all benchmark multimodal functions. That means QMBA has also a very good exploration ability.

The results of shifted and rotated functions are provided in Tables 7 and 8. These functions can be used to test the

TABLE 9: Experimental results of hybrid composite benchmark functions by algorithms (best results in bold).

Algorithms		F19	F20	F21	F22	F23	F24
BA	Mean	9.5618e + 02	6.6067e + 02	8.0640e + 02	1.1242e + 03	1.0997e + 03	1.1241e + 03
	Min	6.1358e + 02	3.7942e + 02	5.5352e + 02	1.0306e + 03	9.8772e + 02	1.0554e + 03
	Max	1.1205e + 03	8.9018e + 02	1.0230e + 03	1.2059e + 03	1.2254e + 03	1.2457e + 03
	Std	1.0641e + 02	1.1969e + 02	1.4405e + 02	5.6971e + 01	5.3599e + 01	5.3205e + 01
IBA	Mean	6.4540e + 02	7.5060e + 02	9.6044e + 02	1.3335e + 03	1.3238e + 03	1.3419e + 03
	Min	4.0066e + 02	7.0373e + 02	6.2563e + 02	1.2690e + 03	1.2388e + 03	1.2700e + 03
	Max	1.1696e + 03	7.9351e + 02	1.2650e + 03	1.3964e + 03	1.3828e + 03	1.3951e + 03
	Std	3.3213e + 02	2.2954e + 01	1.6472e + 02	3.5586e + 01	3.7108e + 01	3.3787e + 01
MBA	Mean	6.1633e + 02	3.8661e + 02	4.7736e + 02	1.0240e + 03	1.0072e + 03	1.0218e + 03
	Min	5.3347e + 02	3.1116e + 02	3.1559e + 02	9.8563e + 02	9.6854e + 02	9.7411e + 02
	Max	8.0680e + 02	4.6281e + 02	6.9084e + 02	1.0607e + 03	1.0587e + 03	1.0681e + 03
	Std	7.5831e + 01	3.3274e + 01	1.1163e + 02	2.1509e + 01	2.4038e + 01	2.4225e + 01
HSBA	Mean	5.4040e + 02	3.7639e + 02	4.0158e + 02	9.4292e + 02	9.4360e + 02	9.3351e + 02
	Min	4.4617e + 02	2.3980e + 02	2.4471e + 02	9.1859e + 02	9.1968e + 02	9.1525e + 02
	Max	1.0346e + 03	5.1007e + 02	5.6114e + 02	9.8567e + 02	1.0001e + 03	9.8476e + 02
	Std	1.7834e + 02	6.3846e + 01	7.5353e + 01	2.0442e + 01	1.8972e + 01	1.6070e + 01
CBA	Mean	9.6917e + 02	7.2755e + 02	1.1000e + 03	1.4076e + 03	1.4183e + 03	1.3862e + 03
	Min	4.0076e + 02	6.3728e + 02	9.8843e + 02	1.3586e + 03	1.3625e + 03	1.3639e + 03
	Max	1.3061e + 03	7.7493e + 02	1.3919e + 03	1.4419e + 03	1.4468e + 03	1.4186e + 03
	Std	3.7487e + 02	3.5249e + 01	1.0808e + 02	1.8473e + 01	2.0678e + 01	1.3209e + 01
QMBA	Mean	4.7921e + 02	5.4202e + 02	6.7557e + 02	9.0000e + 02	9.0000e + 02	9.0000e + 02
	Min	4.0309e + 02	3.8418e + 02	4.3168e + 02	9.0000e + 02	9.0000e + 02	9.0000e + 02
	Max	7.9571e + 02	6.4515e + 02	8.2616e + 02	9.0000e + 02	9.0000e + 02	9.0000e + 02
	Std	1.1200e + 02	6.9503e + 01	1.0464e + 02	0.0000e + 00	0.0000e + 00	0.0000e + 00

TABLE 10: p values calculated for t -test on hybrid benchmark functions.

Algorithms	F19	F20	F21	F22	F23	F24
BA	4.911e - 16	6.116e - 04	2.751e - 03	1.735e - 19	1.086e - 18	1.738e - 20
IBA	4.562e - 02	5.961e - 15	1.812e - 07	2.907e - 37	3.253e - 36	2.028e - 38
MBA	7.994e - 05	1.078e - 10	1.737e - 06	2.807e - 25	2.453e - 21	3.648e - 23
HSBA	2.130e - 01	3.303e - 09	2.753e - 11	3.778e - 11	3.247e - 12	4.543e - 11
CBA	3.144e - 06	1.207e - 12	8.094e - 15	1.336e - 50	4.355e - 49	2.048e - 55

algorithms' robustness. As the results of mean, standard deviation, and minimum and maximum show, QMBA performs better than the other algorithms in two of the shifted and rotated (F14, F16) functions. The HSBA has the best results in two, and IBA and MBA have only one in category III, separately. It should be noted that HSBA and MBA are not significantly better than QMBA in functions F13 and F15, respectively. So it can be claimed that the results of QMBA in these functions are slightly better than the other algorithms.

The results of hybrid composite benchmark functions are provided in Tables 9 and 10. For function F19, QMBA has the best results, but the p values of t -test in Table 10 show that the results of QMBA are not significantly better than HSBA. In the remaining functions (F22, F23, and F24), the results for QMBA are significantly better than the other algorithms except for functions F20 and F21. Therefore, the results can

strongly show that the QMBA has high performance in dealing with complex problem as well.

Furthermore, the convergence graphs of BA and other variants of BA are shown in Figures 1–24 which present the process of optimization. The fitness values shown in Figures 1–24 are the mean objective function optimum obtained from 30 independent runs for each algorithm. Figure 1 shows the results achieved from the six methods when Sphere Function is applied. From Figures 1, 2, 3, 4, 5, and 6, clearly, we can draw the conclusion that QMBA is significantly superior to all the other algorithms during the process of optimization, and it also has the fastest convergence rate compared with other algorithms.

Figures 7, 8, 9, 10, 11, and 12 are the curves of fitness value for multimodal functions. As can be seen from these figures, BA is more easily trapped into local minimum, especially

FIGURE 21: The curve of fitness value for F21.

FIGURE 23: The curve of fitness value for F23.

FIGURE 22: The curve of fitness value for F22.

FIGURE 24: The curve of fitness value for F24.

in functions F8 and F9, and it has the slowest convergence rate; however, QMBA has also the best results and the fastest convergence rate. HSBA is showed to have the second best overall performance.

Figures 13, 14, 15, 16, 17, and 18 are the curves of fitness value for shifted rotated functions. As can be seen in these figures, CBA has faster convergence rate than other algorithms; however, it cannot obtain the better results except F15 and F18. This is probably due to CBA suffering from premature.

Figures 19, 20, 21, 22, 23, and 24 are the curves of fitness value hybrid composite functions. From these figures, we can observe that QMBA has the fastest convergence rate compared to other algorithms in four of six functions, while

IBA and CBA severely suffer from premature convergence problem in the complex cases, especially in functions F22, F23, and F24.

5. Conclusions

In this paper, a variant of novel bat algorithm, namely, QMBA, is proposed by introducing quantum-behaved bat with the direction of mean best position during the searching process. In QMBA, the position of each bat not only depends on the current optimal solution, but also is determined by the mean best position with the iteration proceeding. In the process of searching, quantum behavior of bats is introduced to increase the diversity of population and avoid all bats

getting trapped into local minimum. In other words, the current optimal solution leads global search to guarantee that all the bats converge, while the quantum-behaved bat and mean best position lead local search to jump out of local positions.

This new method can also speed up the global convergence rate without losing the strong robustness of the basic BA. Twenty-four benchmark test functions are tested, and compared with other variant bat algorithms for numerical optimization, from the analysis of the simulation results, we observed that the proposed QMBA makes good use of the information in past solutions more effectively to generate better quality solutions frequently and when compared to the other variants of BA showed that this approach is simple and efficient and can achieve a more accurate solution.

Our future work will focus on applying the QMBA algorithm to engineering optimization problems and developing new metahybrid approach to solve more complex optimization problem.

Competing Interests

All authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work is supported by National Science Foundation of China under Grant no. 61201177.

References

- [1] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.
- [2] 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.
- [3] M. Dorigo, V. Maniezzo, and A. Colnari, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
- [4] I. Fister Jr., X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation*, vol. 13, no. 1, pp. 34–46, 2013.
- [5] X. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms Foundations & Applications*, vol. 5792, pp. 169–178, Springer, Berlin, Germany, 2010.
- [6] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC '09)*, IEEE, Coimbatore, India, 2009.
- [7] X.-S. Yang and S. Deb, "Cuckoo search: recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1, pp. 169–174, 2014.
- [8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, November–December 1995.
- [9] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [10] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, pp. 65–74, Springer, 2010.
- [11] X. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2010.
- [12] E. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 2009.
- [13] 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.
- [14] Y. Zhou, J. Xie, and H. Zheng, "A hybrid bat algorithm with path relinking for capacitated vehicle routing problem," *Mathematical Problems in Engineering*, vol. 2013, Article ID 392789, 10 pages, 2013.
- [15] J. Xie, Y. Zhou, and H. Zheng, "A hybrid metaheuristic for multiple runways aircraft landing problem based on bat algorithm," *Journal of Applied Mathematics*, vol. 2013, Article ID 742653, 8 pages, 2013.
- [16] 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.
- [17] 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.
- [18] A. Kaveh and P. Zakian, "Enhanced bat algorithm for optimal design of skeletal structures," *Asian Journal of Civil Engineering*, vol. 15, no. 2, pp. 179–212, 2014.
- [19] X.-S. Yang, "Bat algorithm for multi-objective optimisation," *International Journal of Bio-Inspired Computation*, vol. 3, no. 5, pp. 267–274, 2011.
- [20] A. Rezaee Jordehi, "Chaotic bat swarm optimisation (CBSO)," *Applied Soft Computing*, vol. 26, pp. 523–530, 2015.
- [21] J. Lin, C. Chou, C. Yang, and H. Tsai, "A chaotic Levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems," *Computer and Information Technology*, vol. 2, no. 2, pp. 56–63, 2012.
- [22] S. Yilmaz and E. U. Kucuksille, "Improved Bat Algorithm (IBA) on continuous optimization problems," *Lecture Notes on Software Engineering*, vol. 1, no. 3, pp. 279–283, 2013.
- [23] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, Article ID 696491, 21 pages, 2013.
- [24] S. Yilmaz, E. U. Kucuksille, and Y. Cengiz, "Modified bat algorithm," *Elektronika Ir Elektrotehnika*, vol. 20, no. 2, pp. 71–78, 2014.
- [25] H. Afrabandpey, M. Ghaffari, A. Mirzaei, and M. Safayani, "A novel Bat Algorithm based on chaos for optimization tasks," in *Proceedings of the Iranian Conference on Intelligent Systems (ICIS '14)*, pp. 1–6, IEEE, Bam, Iran, February 2014.
- [26] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, no. 2, pp. 224–232, 2014.
- [27] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

- [28] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 1, pp. 325–331, June 2004.
- [29] J. Sun, W. Fang, V. Palade, X. Wu, and W. Xu, "Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point," *Applied Mathematics and Computation*, vol. 218, no. 7, pp. 3763–3775, 2011.
- [30] J. Sun, W. Fang, X. Wu, V. Palade, and W. Xu, "Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection," *Evolutionary Computation*, vol. 20, no. 3, pp. 349–393, 2012.
- [31] S. Mirjalili, S. M. Mirjalili, and X. Yang, "Binary bat algorithm," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 663–681, 2014.
- [32] 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," KanGAL Report, 2005.
- [33] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

Hindawi

Submit your manuscripts at
<http://www.hindawi.com>