

Research Article

Modified Bat Algorithm Based on Lévy Flight and Opposition Based Learning

Xian Shan,¹ Kang Liu,² and Pei-Liang Sun³

¹School of Science, China University of Petroleum, Qingdao 266580, China

²College of Mechanical and Electronic Engineering, China University of Petroleum, Qingdao 266580, China

³School of Economics and Management, China University of Petroleum, Qingdao 266580, China

Correspondence should be addressed to Kang Liu; lkzsw@163.com

Received 14 July 2016; Accepted 25 October 2016

Academic Editor: Xiang Li

Copyright © 2016 Xian Shan 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.

Bat Algorithm (BA) is a swarm intelligence algorithm which has been intensively applied to solve academic and real life optimization problems. However, due to the lack of good balance between exploration and exploitation, BA sometimes fails at finding global optimum and is easily trapped into local optima. In order to overcome the premature problem and improve the local searching ability of Bat Algorithm for optimization problems, we propose an improved BA called OBMLBA. In the proposed algorithm, a modified search equation with more useful information from the search experiences is introduced to generate a candidate solution, and Lévy Flight random walk is incorporated with BA in order to avoid being trapped into local optima. Furthermore, the concept of opposition based learning (OBL) is embedded to BA to enhance the diversity and convergence capability. To evaluate the performance of the proposed approach, 16 benchmark functions have been employed. The results obtained by the experiments demonstrate the effectiveness and efficiency of OBMLBA for global optimization problems. Comparisons with some other BA variants and other state-of-the-art algorithms have shown the proposed approach significantly improves the performance of BA. Performances of the proposed algorithm on large scale optimization problems and real world optimization problems are not discussed in the paper, and it will be studied in the future work.

1. Introduction

With the development of natural science and technology, a large number of complicated optimization problems arise in a variety of fields, such as engineering, manufacturing, information technology, and economic management. These real world problems usually have an objective function with lots of decision variables and constrains, appearing with discontinuous, nonconvex features [1]. Optimization methods are always needed to obtain the optimal solution of these problems.

Optimization algorithms proposed by researchers can be categorized into two groups: deterministic algorithms and stochastic algorithms. Deterministic algorithms usually need the information of gradient. They are effective for problems with one global optimum, while they might be invalid for problems with several local optima or problems with gradient

information unavailable. Compared with deterministic algorithms, stochastic algorithms require only the information of the objective function [2]. They have been successfully used in many optimization problems characterized as nondifferentiable and nonconvex.

Swarm intelligence algorithms, regarded as a subset of stochastic algorithms, have been widely used to solve optimization problems during the past decades. They are inspired by the collective behavior of swarms in nature. Researchers have observed such behaviors of animals, plants, or humans, analyzed the driving force behind the phenomena, and then proposed various types of algorithms. For example, Genetic Algorithm (GA) [3] and Differential Evolution (DE) [4] were generalized by inspiration of the biological evolution and natural selection. Particle Swarm Optimization (PSO) [5] was proposed by simulating the social and cognitive behavior of fish or bird swarms. Ant Colony Algorithm (ACO) [6]

and Artificial Bee Colony (ABC) [7] were introduced by the foraging behavior of ants and bees. Cat Swarm Algorithm (CSO) [8], Cuckoo Search Algorithm (CS) [9], and Bat Algorithm (BA) [10] were also proposed by inspiration from the intelligence features in behaviors or organisms of swarms.

Bat Algorithm, proposed by Yang in 2010, is a swarm intelligence algorithm inspired by the echolocation behavior of bats. Echolocation is a type of sonar which is used by bats to detect prey, hunt, and avoid obstacles. With the help of echolocation, not only can the bats be able to detect the distance of the prey, but also they can identify the shape, position, and angle of the prey [10]. Compared with other existing swarm intelligence algorithms, BA has many advantages such as less control parameters, excellent global optimization ability, and implementation simplicity, which have shown excellent performances in some classes of optimization problems. Due to its simplicity, convergence speed, and population feature, it has been intensively used to solve academic and real life problems, like multiobjective optimization [11], engineering optimization [12], cluster analysis [13], scheduling problems [14], structure optimization [15], image processing [16], manufacturing designing [17], and various other problems.

It has been shown that BA is an excellent and powerful algorithm for global optimization problems, discrete optimization problems, and constrained optimization problems. However, due to the lack of good balance between exploration and exploitation in basic BA, the algorithm sometimes fails at finding global optimum and is easily trapped into local optima. Much efforts have been made to improve the performance of BA. These improvements or modifications can be divided into two categories.

Introduction of New Search Mechanisms for Generating Candidate Solutions. Inspired by DE, Xie et al. [18] presented a modified approach DLBA with a new solution update mechanism combining the concept of differential operator and Lévy Flight trajectory. Li and Zhou [19] embedded the complex value encoding mechanism into Bat Algorithm to improve the diversity and convergence performance. Gandomi and Yang [20] introduced chaotic maps into the population initialization and position update equation and proposed a modified algorithm. Bahmani-Firouzi and Azizpanah-Abarghoee [21] proposed four improved velocity update equations in order to achieve a better balance between the exploration and exploitation. Jaddi et al. [22] divided the swarm population into two subpopulation groups with each group designed to generate solutions according to different equations, respectively. Yilmaz and Küçükşille [23] defined two modification structures of the velocity equation inspired by PSO and then hybridized the modified BA with Invasive Weed Optimization algorithm. The velocity equations used information of the best-so-far solution and the neighborhood solutions, which can effectively enhance the search ability of BA.

Hybridization of BA with Other Operators. For example, Khan and Sahai [24] proposed a hybrid approach involving PSO, HS, and SA for generating new solutions. Wang and Guo [25]

introduced a mutation operator into BA using SA. He et al. [26] developed a hybrid BA which combines SA and Gauss distribution. Sadeghi et al. [27] developed a hybrid BA with local search based on PSO. Lin et al. [28] developed a chaotic BA with Lévy Flights and parameters estimated by chaotic maps. Meng et al. [29] incorporated the bats' habitat selection and their self-adaptive compensation for Doppler effect in echoes into the basic BA and proposed a new self-adaptive modified algorithm NBA.

The study is not limited to the above two aspects and more work can be seen in [30–34] and so on.

As technology and science develop dramatically fast, more and more complicated optimization problems need to be solved by advanced numerical methods. As an important heuristic optimization algorithm, researches on how to improve the convergence accuracy and efficiency of BA are of great significance to improve the heuristic optimization theory and solve the real world optimization problems.

Exploration and exploitation are two critical characteristics in the updating process of an algorithm. Exploration is the ability of the algorithm to find the global optimum solutions in different areas of the search space, while exploitation refers to the capability of the algorithm to find better solutions with previous experience. Researches in the literature show that, for a swarm intelligence algorithm, the exploration capability should be employed first so as to search in the whole space, while the exploitation capability should be considered later by improving the quantity of the solution in the local search process [23].

To achieve a better balance between exploration and exploitation behavior of BA, it is highly required to modify the global search approach to explore the search region and develop a local search method to exploit in the nonvisited regions. Accordingly, modifications of the search equation and local search strategies for BA are studied in the research. A modified algorithm called OBMLBA is proposed in this study. The main difference between OBMLBA and other BAs is the interaction behavior between bats. In the proposed approach, the bat explores the search space with the help of the best-so-far solution and the neighborhood solutions by using echolocation, which can well balance the exploration and exploitation. The new population P is generated by using a new modified position update equation with frequency f defined by sinusoidal function. Lévy Flight random walk is used to exploit the local search space whereas opposition based learning is used to introduce opposition based solutions in population OP. Individuals of P and OP are merged together, and the best N individuals are used as a new population for the next generation. The proposed method has been tested on 16 benchmark functions. Comparison with variants of BA and other state-of-the-art algorithms demonstrates the effectiveness and superiority of OBMLBA in solving optimization problems.

The paper is organized as follows. After the introduction, the basic concept of BA is introduced in Section 2. The proposed OBMLBA is presented in Section 3. Section 4 evaluates the performance of the proposed algorithm by comparing results with other algorithms. Finally, conclusions and future work are discussed in Section 5.

- (1) Input the parameters.
- (2) Set iter = 1. Initialize the position and velocity of each individual in the population.
- (3) Evaluate the objective function and related parameter values of each individual.
- (4) **While** iter ≤ iter_max **do**
- (5) Generate new solutions using Eq. (2)~Eq. (4).
- (6) **if** rand(0, 1) > r_i
- (7) Generate a new local solution around the selected solution x^* using Eq. (5).
- (8) **end if**
- (9) **if** rand(0, 1) < A_i and $f(x_{new}^i) < f(x_i)$
- (10) update the solution, loudness and pulse emission rate by Eq. (6).
- (11) **end if**
- (12) **end while**
- (13) Output the individual with the smallest objective function value in the population.

ALGORITHM 1: Basic Bat Algorithm.

2. Basic Bat Algorithm

The optimization problems considered in this paper are single-objective, unconstrained continuous problems to be minimized.

Bat Algorithm is a heuristic algorithm proposed by Yang in 2010. It is based on the echolocation capability of micro bats guiding them on their foraging behavior. In BA, the position of a bat represents a possible solution of the given optimization problem. The position of the food source found by the i th bat can be expressed as $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$. The fitness of x_i corresponds to the quality of the position the bat locates in.

2.1. Initiation. At the beginning, bats do not know the location of food sources. They generate a randomly distributed population P of N solutions, where N denotes the number of bats in the population. Each solution can be produced within the search space as follows:

$$x_{ij} = x_{\min} + \text{rand}(0, 1)(x_{\max} - x_{\min}), \quad (1)$$

where $i = 1, 2, \dots, N$, $j = 1, 2, \dots, D$. x_{\max} and x_{\min} denote the upper and lower bounds of the solution in dimension j , respectively. rand(0, 1) is a uniformly distributed value generated in the range of [0, 1].

2.2. Generation of New Solutions. Bats modify the solution based on the information of the current location and the best source food location. Bats navigate by adjusting their flying directions using their own and other swarm members' best experience to find the optimum of the problem.

At this stage, for each food source position x_i , a new position was formulated as follows:

$$f_i = f_{\min} + \beta(f_{\max} - f_{\min}), \quad (2)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*)f_i, \quad (3)$$

$$x_i^t = x_i^{t-1} + v_i^t, \quad (4)$$

where i represents each bat in the population, $i = 1, 2, \dots, N$, and t represents the t th iteration. x_i^t and v_i^t are the position

and velocity components of the i th bat in the population at the t th iteration. f_i denotes the pulse frequency that affects the velocity of the i th bat. f_{\max} and f_{\min} represent the maximum and minimum of f_i . β is a random number between [0, 1]. x^* is the best position found by the whole population.

2.3. Local Search. Once the new solutions are generated, the local search is invoked by bats' random walk. If the pulse emission rate r of the i th bat is smaller than a random number, select x_{old}^i from the population and generate a new position x_{new}^i as follows:

$$x_{new}^i = x_{old}^i + \varepsilon A^t, \quad (5)$$

where x_{old}^i represents a solution chosen in current population by some mechanism and ε is a random vector drawn from a uniform distribution. A^t is the average loudness of all bats at the time step t .

2.4. Solutions, Loudness, and Pulse Emission Rate Updating. If a random number is bigger than the loudness A^t and $f(x_{new}^i) < f(x_i)$, accept the new solution x_{new}^i . At the same time, the loudness A^t is decreased while its pulse emission r_i is increased as follows:

$$\begin{aligned} A_i^{t+1} &= \alpha A_i^t, \\ r_i^{t+1} &= r_i^0 (1 - e^{-\gamma t}), \end{aligned} \quad (6)$$

where α and γ are constants. The initial loudness A^0 and initial pulse emission rate r^0 are randomly generated numbers in the range of [1, 2] and [0, 1], respectively.

The pseudocode of Bat Algorithm is presented in Algorithm 1.

3. Enhanced Bat Algorithm

3.1. Location Update Equation Based on DE. Yilmaz and Küçükşille [23] pointed that the update equations of velocity in basic BA provide local search only with the guidance of the best-so-far solution, which makes the BA easily entrapped

into a local minimum. In order to improve the search ability, they proposed a modified equation:

$$v_i^t = \omega v_i^{t-1} + (x_i^t - x^*) f_i \zeta_1 + (x_i^t - x_k^t) f_i \zeta_2, \quad (7)$$

$$\zeta_1 + \zeta_2 = 1, \quad (8)$$

$$\zeta_1 = 1 + (\zeta_{\text{init}} - 1) \frac{(\text{iter}_{\text{max}} - \text{iter})^n}{(\text{iter}_{\text{max}})^n}, \quad (9)$$

where x_k^t is a solution randomly chosen from the population and ζ_1 and ζ_2 are learning factors ranging from 0 to 1. ζ_{init} is the initial value of ζ_1 , iter_{max} is the maximal number of iterations, iter is the number of iterations, and n is a nonlinear index.

The second term in the modified search equation is a factor affecting the velocity of the solution with guidance of the best solution x^* , which emphasizes exploitation. The third term is another factor that affects the velocity with help of randomly selected solution x_k^t which emphasizes exploration. Effects of the best solution and the k th solution are adjusted by changing the value of ζ_1 . Experiments results indicate that the modified algorithm can outperform BA in most test functions.

Xie et al. [18] proposed an improved BA based on Differential Evolution operator. The position updating equations were defined as follows:

$$x_i^{t+1} = x_{\text{best}}^t + f_{1i} (x_{r1}^t - x_{r2}^t) + f_{2i} (x_{r3}^t - x_{r4}^t), \quad (10)$$

where x_{best}^t is the global best location in current population. x_{ri}^t ($i = 1, 2, 3, 4$) is a randomly chosen solution in the population. f_{1i} and f_{2i} are defined as follows:

$$f_{1i}^t = \left((f_{1,\text{min}} - f_{1,\text{max}}) \frac{t}{n_t} + f_{1,\text{max}} \right) \beta_1, \quad (11)$$

$$f_{2i}^t = \left((f_{2,\text{min}} - f_{2,\text{max}}) \frac{t}{n_t} + f_{2,\text{max}} \right) \beta_2, \quad (12)$$

where $f_{1,\text{max}}$ and $f_{1,\text{min}}$ are minimum and maximum frequency values of f_{1i} , while $f_{2,\text{max}}$ and $f_{2,\text{min}}$ are minimum and maximum frequency values of f_{2i} . n_t is a fixed parameter. t is the number of iterations. $\beta \in [0, 1]$ is a random vector drawing a uniform distribution. The modified algorithm has shown a better performance than classical BA.

Inspired by Yilmaz and Zhou's methods, we propose a modified location updating equation:

$$x_i^{t+1} = x_i^t + f_{1i} (x_{\text{best}}^t - x_{r1}^t) + f_{2i} (x_{r2}^t - x_{r3}^t), \quad (13)$$

where x_{ri}^t ($i = 1, 2, 3$) are individuals randomly selected from the population. f_{1i} and f_{2i} are frequencies.

As in mutation operation of DE, the frequency f is an important parameter that affects the convergence performance of the algorithm. The value of f changes according to the predefined formula. In basic BA and DLBA, f is defined in (2), (11), and (12), respectively. Here we use a sinusoidal formulation [35] permitting certain flexibility in changing

the direction of f . The frequencies f_{1i} and f_{2i} are defined as follows:

$$f_{1i} = \frac{1}{2} \left(\sin(2\pi * \omega * t) * \frac{\text{iter}}{\text{iter}_{\text{max}}} + 1 \right), \quad (14)$$

$$f_{2i} = \frac{1}{2} \left(\sin(2\pi * \omega * t) * \frac{\text{iter}}{\text{iter}_{\text{max}}} + 1 \right), \quad (15)$$

where ω represents frequency of the sinusoidal function.

3.2. Local Search Based on Lévy Flight. Lévy Flight, which is based on the flight behavior of animals and insects, has been variously studied in literatures [36]. It is defined as a kind of non-Gauss stochastic process with a random walk whose step sizes are based on Lévy stable distribution.

There are lots of versions of Lévy distribution. Mantegna Algorithm [37] has been tested to be the most efficient method to generate Lévy distribution values. It has been adopted in many evolution algorithms in order to escape from local minimum.

When generating a new solution x_i' for the i th solution x_i by performing Lévy Flight, the new candidate x_i' is defined as follows:

$$x_i' = x_i + \alpha \oplus \text{Levy}(\lambda), \quad (16)$$

where α is a random step size parameter, λ is a Lévy Flight distribution parameter, and \oplus denotes entry wise multiplication.

In Mantegna Algorithm, the step size s can be defined as follows:

$$s = \alpha \oplus \text{Levy}(\lambda) \sim 0.01 \frac{u}{|v|^{1/\beta}} (x_j - x_{\text{best}}), \quad (17)$$

where u and v are obtained from normal distribution; that is,

$$\begin{aligned} u &\sim N(0, \sigma_u^2), \\ v &\sim N(0, \sigma_v^2) \end{aligned} \quad (18)$$

with

$$\sigma_u = \left\{ \frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\beta\Gamma[(1 + \beta)/2] 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad (19)$$

$$\sigma_v = 1,$$

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-z} dz. \quad (20)$$

The pseudocode of Lévy Flight Search is shown in Algorithm 2.

3.3. Opposition Based Learning. Opposition based learning was proposed by Rahnamayan et al. [38] in 2005. It has been successfully used in machine learning and evolutionary computing [39]. For an optimization problem, if the candidate solutions are near the global optima, it is easy to find the ideal solutions with fast convergence speed. But if the candidate

- (1) Input the optimization function $f(x)$ and β .
- (2) Select the individual x_i which is chosen to be modified by Lévy Flight.
- (3) Compute σ_u using Eq. (19).
- (4) Compute step size s using Eq. (17) and Eq. (18).
- (5) Generate a new candidate solution x'_i using Eq. (16).
- (6) Calculate $f(x'_i)$.
- (7) **if** $f(x'_i) < f(x_i)$
- (8) $x_i = x'_i$.
- (9) **end if**

ALGORITHM 2: Pseudocode of Lévy Flight Search.

- (1) Input original population P , population size SN , the dimension of solution D , optimization function $f(x)$.
- (2) Generate opposite population OP as follows:
- (3) **for** $i = 1 : SN$
- (4) **for** $j = 1 : D$
- (5) $x_{ij} = a_i + b_i - x_{ij}$;
- (6) **end for**
- (7) **end for**
- (8) Merge the original population P and opposite population OP together.
- (9) Calculate the fitness of population $\{P, OP\}$.
- (10) Choose the top half of population $\{P, OP\}$ as the current population.

ALGORITHM 3: Opposition based optimization.

solutions are far away from the global optima, it will take more computational efforts to get the required solutions. The concept of OBL is to consider the candidate solutions and their opposite counterpart solutions simultaneously in an optimization algorithm. Then the greedy selection is applied between them according to their objective function values. With the help of OBL, the probability of finding global optima is increased.

In basic OBL, let $x = (x_1, x_2, \dots, x_D)$ be a D -dimensional solution in the search space, with component variables $x_i \in [a_i, b_i]$. Then the opposite point $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_D)$ is defined by its coordinates $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_D$, where

$$\tilde{x}_i = a_i + b_i - x_i, \quad i = 1, 2, \dots, D. \quad (21)$$

OBL can be used not only to initialize the population initialization, but also to improve the population diversity during the evolutionary process. Once the opposite population generated, it is combined with the original population for selection. Individuals with fitness ranking in the top half of all candidate solutions will be selected as the current population. The pseudocode of OBL is given in Algorithm 3.

3.4. Proposed Modified Bat Algorithm. Based on the basic BA and the above considerations, we propose a modified method, called OBMLBA. It is clear that local search with Lévy Flight can improve the ability of exploitation. Incorporating with OBL strategy can enhance the population diversity. Inspired by DLBA, the new location update equation takes advantage of the information of both the global best solution

and randomly chosen solution near the candidate solution. So the modified BA can get a good balance between the exploration and exploitation. The main steps of the algorithm are summarized in Algorithm 4.

4. Experimental Results and Discussion

In order to evaluate the performance of the proposed algorithm, we select 16 standard benchmark functions to investigate the efficiency of the proposed approach. These functions are presented in Table 1. The function set comprises unimodal functions and multimodal functions, which can be used to test the convergence speed and local premature convergence problem, respectively. $F_1 \sim F_5$ are continuous unimodal functions. F_7 is a discontinuous step function. F_9 is a noisy quartic function. $F_{10} \sim F_{16}$ are multimodal functions with high dimension. These functions are categorized into low-dimension, middle-dimension, and high-dimension functions according to $D = 15, 30, \text{ and } 60$, respectively.

Related parameters of the basic BA, DLBA, and OBMLBA are presented in Table 2. The maximum number of iterations is set to be 500, 1000, and 2000 in case of $D = 15, 30, \text{ and } 60$.

Experiments have been carried out on the test functions with different dimensions. The computational results are summarized in Tables 3–5 in terms of the mean error and standard deviation of the function error values. In addition, some convergence characteristics graphs are shown in Figure 1.

```

(1) Input the parameters.
(2) Set iter = 1. Initialize the position and velocity of each individual in the population.
(3) Evaluate the objective function and related parameter values of each individual.
(4) While iter ≤ iter_max do
(5)   Apply Lévy Flight Local Search strategy using Algorithm 2.
(6)   Generate new solutions using Eq. (13).
(7)   if rand(0, 1) > ri
(8)     Apply the Opposition based learning using Algorithm 3.
(9)     Generate a new local solution around the selected solution x* using Eq. (5).
(10)  end if
(11)  if rand(0, 1) < Ai and f(xnewi) < f(xi)
(12)    update the solution, loudness and pulse emission rate.
(13)  end if
(14) end while
(15) Output the individual with the smallest objective function value in the population.

```

ALGORITHM 4: Pseudocode of OBMLBA.

TABLE 1: Benchmark problems.

Fun	Name	Scope	f_{\min}
F_1	Sphere	[-100, 100]	0
F_2	Elliptic	[-100, 100]	0
F_3	SumSquare	[-10, 10]	0
F_4	SumPower	[-1, 1]	0
F_5	Schwefel2.22	[-10, 10]	0
F_6	Schwefel2.21	[-100, 100]	0
F_7	Step	[-100, 100]	0
F_8	Exponential	[-1.28, 1.28]	0
F_9	Quartic	[-1.28, 1.28]	0
F_{10}	Rosenbrock	[-5, 10]	0
F_{11}	Rastrigin	[-5.12, 5.12]	0
F_{12}	Griewank	[-600, 600]	0
F_{13}	Ackley	[-32, 32]	0
F_{14}	Alpine	[-10, 10]	0
F_{15}	Levy	[-10, 10]	0
F_{16}	Weierstrass	[-0.5, 0.5]	0

TABLE 2: Parameters setting.

Parameters	BA	DLBA	OBMLBA
Population size N		40	
Run time		20	
Loudness		0.95	
Pulse emission		0.85	
Factors updating A and γ		0.9	
Minimum of frequency f_{\min}		0	
Maximum of frequency f_{\max}		1	
Factors updating frequency	—	—	0.3

As seen from Tables 3–5, OBMLBA can get the global optimal on $f_1 \sim f_4$, f_7 , f_8 , and $f_{11} \sim f_{13}$ with $D = 15, 30$, and 60. On f_6 and f_{14} , the precision of solutions obtained by OBMLBA is smaller than the order of $1E - 90$ and the order for $1E - 100$, respectively. Results indicate that OBMLBA can get solutions extremely close to the global optimal values on most of the functions.

Solutions obtained by BAs are compared with results obtained by MLBA and OBMLBA. As shown in Tables 3–5, OBMLBA performs significantly better than basic BA, LBA, and DLBA on $f_1 \sim f_6$, f_{14} , and f_{15} . All algorithms except for basic BA and LBA perform well on f_7 , f_8 , $f_{11} \sim f_{13}$, and f_{16} . OBMLBA is comparable to DLBA and MLBA on functions f_9 and f_{10} .

Compared with MLBA, OBMLBA provides better performance on functions $f_1 \sim f_6$, f_{14} , and f_{15} . For f_9 , f_{10} , and

f_{15} , results obtained by MLBA and OBMLBA are quite close to each other. Both MLBA and OBMLBA get the global optimum of f_7 , f_8 , $f_{11} \sim f_{13}$, and f_{16} .

From the convergence characteristic graphs, it can be obviously observed that OBMLBA is the fastest algorithm in all the considered BAs. MLBA is slower than OBMLBA but faster than DLBA, LBA, and basic BA. OBMLBA exhibits the best accuracy and achieves the fastest convergence speed. The results indeed indicate the advantage of the modified position update equation and OBL strategy.

On the whole, algorithms proposed in this study such as MLBA and OBMLBA work better than other BAs considered. And OBMLBA is more effective than MLBA.

In Table 6, OBMLBA is compared with some state-of-the-art algorithms, such as ABC, PSO, OCABC, CLPSO, and OLPSO-G. Results of these algorithms are all derived directly from their corresponding literatures [40]. Results show that OBMLBA has outperformed other methods on 6 of 7 functions. OBMLBA performs worse than ABC and OCABC only on Rosenbrock function.

According to the analyses above, it can be clearly observed that OBMLBA provides outstanding performance with fast convergence speed and high convergence accuracy on most of the test functions.

TABLE 3: Results of comparisons of BAs on 15-dimension benchmark functions.

Fun	BA		DLBA		MLBA		OBMLBA			
	Mean	SD	Mean	SD	Mean	SD	Mean	SD		
F_1	1.19E+04	1.84E+03	2.60E+03	2.53E+03	4.45E-54	1.39E-53	2.30E-69	7.07E-69	3.6601E-204	0
F_2	7.05E+07	3.04E+07	1.89E+07	1.93E+07	1.02E-42	4.54E-42	3.96E-67	1.25E-66	3.4713E-202	0
F_3	8.10E+02	1.39E+02	1.32E+02	2.25E+02	3.77E-52	1.69E-51	8.10E-66	2.56E-65	6.2179E-202	0
F_4	2.59E-02	1.50E-02	1.64E-03	2.35E-03	5.22E-90	2.27E-89	4.57E-126	1.2069E-125	1.12E-305	0
F_5	4.08E+01	7.24E+00	1.08E+01	1.43E+01	3.73E-28	1.45E-27	1.75E-33	5.52E-33	1.17E-104	5.2441E-104
F_6	5.01E+01	3.16E+00	1.47E+01	1.46E+01	3.04E-27	1.00E-26	2.50E-35	7.85E-35	6.68E-95	2.98692E-94
F_7	1.11E+04	2.13E+03	1.36E+03	2.16E+03	0	0	0	0	0	0
F_8	1.52E+00	4.43E-01	1.26E-01	2.62E-01	0	0	0	0	0	0
F_9	2.65E+00	1.13E+00	1.34E-01	2.70E-01	4.79E-04	4.31E-04	3.99E-04	2.82E-04	1.19E-04	1.00E-04
F_{10}	5.89E+04	2.02E+04	6.03E+02	1.34E+03	1.35E+01	2.10E-01	1.29E+01	3.71E-01	1.30E+01	3.40E-01
F_{11}	1.31E+02	8.16E+00	7.81E+01	4.48E+01	0	0	0	0	0	0
F_{12}	1.06E+02	2.24E+01	1.66E+01	2.27E+01	0	0	0	0	0	0
F_{13}	1.83E+01	4.88E-01	1.07E+01	5.65E+00	0	0	0	0	0	0
F_{14}	1.42E+01	1.04E+00	5.29E+00	5.26E+00	1.82E-27	7.92E-27	5.67E-35	1.79E-34	8.91E-106	3.6823E-105
F_{15}	1.52E+02	1.13E+01	2.73E+01	1.76E+01	7.47E-01	3.00E-01	2.78E-01	2.74E-01	4.83E-01	5.26E-01
F_{16}	1.68E+01	1.60E+00	9.87E+00	5.23E+00	0	0	0	0	0	0

TABLE 4: Results of comparisons of BAs on 30-dimension benchmark functions.

Fun	BA			LBA			DLBA			MLBA			OBMLBA		
	Mean	SD		Mean	SD		Mean	SD		Mean	SD		Mean	SD	
F_1	4.05E+04	2.80E+03		3.80E+03	6.80E+03		3.73E-49	1.67E-48		1.80E-117	5.68E-117		0	0	
F_2	5.30E+08	1.15E+08		9.54E+07	8.35E+07		2.96E-50	1.25E-49		3.20E-117	1.01E-116		0	0	
F_3	5.15E+03	7.97E+02		7.29E+02	1.21E+03		9.80E-48	4.38E-47		2.72E-123	6.51E-123		0	0	
F_4	7.53E-02	2.97E-02		5.80E-03	1.39E-02		1.74E-85	7.78E-85		1.76E-198	0		0	0	
F_5	2.33E+05	4.28E+05		7.58E+01	1.78E+02		9.72E-26	4.33E-25		5.12E-60	1.62E-59		6.80E-210	0	
F_6	6.83E+01	3.50E+00		1.76E+01	1.96E+01		5.42E-30	1.11E-29		2.63E-55	8.33E-55		1.40E-205	0	
F_7	3.71E+04	4.26E+03		5.69E+03	8.14E+03		0	0		0	0		0	0	
F_8	2.39E+01	6.19E+00		1.23E+00	2.62E+00		0	0		0	0		0	0	
F_9	4.02E+01	6.91E+00		2.25E+00	5.19E+00		2.85E-04	2.83E-04		2.30E-04	1.20E-04		6.25E-05	4.21E-05	
F_{10}	3.85E+05	6.51E+04		2.59E+04	6.15E+04		2.87E+01	6.22E-02		2.83E+01	2.16E-01		2.82E+01	2.48E-01	
F_{11}	3.37E+02	1.57E+01		2.13E+02	7.34E+01		0	0		0	0		0	0	
F_{12}	3.61E+02	3.39E+01		3.74E+01	5.97E+01		0	0		0	0		0	0	
F_{13}	1.96E+01	2.71E-01		6.56E+00	5.25E+00		0	0		0	0		0	0	
F_{14}	4.26E+01	2.74E+00		1.55E+01	1.42E+01		1.26E-27	5.56E-27		2.91E-60	9.10E-60		3.30E-215	0	
F_{15}	5.00E+02	8.97E+01		6.96E+01	6.04E+01		9.01E+00	3.07E+00		5.78E+00	2.82E+00		5.17E+00	2.40E+00	
F_{16}	4.13E+01	1.17E+00		1.61E+01	1.38E+01		0	0		0	0		0	0	

TABLE 5: Results of comparisons of BAs on 60-dimension benchmark functions.

Fun	BA			LBA			DLBA			MLBA			OBMLBA		
	Mean	SD		Mean	SD		Mean	SD		Mean	SD		Mean	SD	
F_1	1.03E+05	9.37E+03		3.53E+03	8.71E+03		4.54E-49	2.02E-48		1.21E-219	0.00E+00		0	0	
F_2	2.38E+09	2.51E+08		6.07E+08	8.73E+08		9.65E-43	4.32E-42		1.61E-217	0.00E+00		0	0	
F_3	2.95E+04	3.02E+03		1.24E+03	2.03E+03		2.13E-54	5.66E-54		8.19E-224	0.00E+00		0	0	
F_4	1.47E-01	4.25E-02		2.79E-03	6.28E-03		2.05E-88	9.18E-88		0.00E+00	0		0	0	
F_5	1.00E+10	0.00E+00		2.76E+01	3.04E+01		4.78E-24	2.14E-23		4.78E-112	1.51E-111		0	0	
F_6	8.21E+01	2.42E+00		1.15E+01	1.53E+01		3.21E-28	7.05E-28		1.95E-107	6.13E-107		0	0	
F_7	9.83E+04	8.72E+03		1.17E+04	2.18E+04		0	0		0	0		0	0	
F_8	5.46E+03	4.31E+03		2.74E+01	8.12E+01		0	0		0	0		0	0	
F_9	2.76E+02	4.78E+01		2.55E+00	7.94E+00		2.70E-04	2.54E-04		1.56E-04	7.75E-05		2.82E-05	2.77E-05	
F_{10}	1.77E+06	3.90E+05		1.80E+04	3.47E+04		5.86E+01	7.21E-02		5.83E+01	1.52E-01		1.01E+01	8.32E-01	
F_{11}	7.73E+02	2.79E+01		3.09E+02	2.70E+02		0	0		0	0		0	0	
F_{12}	9.03E+02	1.21E+02		3.72E+01	6.01E+01		0	0		0	0		0	0	
F_{13}	2.03E+01	6.71E-02		6.36E+00	5.72E+00		0	0		0	0		0	0	
F_{14}	1.08E+02	4.42E+00		5.06E+01	3.91E+01		5.26E-29	2.20E-28		6.46E-104	2.04E-103		0	0	
F_{15}	1.49E+03	1.92E+02		2.74E+02	2.97E+02		4.14E+01	9.79E+00		2.42E+01	6.61E+00		4.58E-02	6.53E-02	
F_{16}	9.30E+01	1.52E+00		3.35E+01	2.68E+01		0	0		0	0		0	0	

TABLE 6: Comparisons of the state-of-the-art algorithms on 30-dimension benchmark functions.

Fun	ABC		PSO		OCABC		CLPSO		OLPSO-G		OBMLBA	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Sphere	6.80E-14	8.74E-14	3.34E-14	5.39E-14	4.05E-41	6.97E-41	1.58E-12	7.70E-13	4.12E-54	6.34E-54	0	0
Schweel222	4.58E-08	1.98E-08	1.70E-10	1.39E-10	9.13E-22	7.98E-22	2.51E-08	5.84E-09	9.85E-30	1.01E-29	6.80E-210	0
Step	0	0	0	0	0	0	0	0	0	0	0	0
Rosenbrock	1.68E-01	2.04E-01	2.80E+01	2.17E+01	4.39E-01	7.26E-01	1.13E+01	8.50E+01	2.15E+01	2.99E+01	2.82E+01	2.48E-01
Rastrigin	2.90E-07	8.64E-07	3.57E+01	6.89E+00	0	0	9.09E-05	1.25E-04	1.07E+00	9.92E-01	0	0
Griewank	1.69E-05	4.88E-05	1.53E-03	4.32E-03	0	0	9.02E-09	8.57E-09	4.83E-03	8.63E-03	0	0
Ackley	1.09E-06	5.96E-07	8.20E-08	6.73E-08	3.83E-15	9.83E-15	3.66E-07	7.57E-08	7.98E-15	2.03E-15	0	0

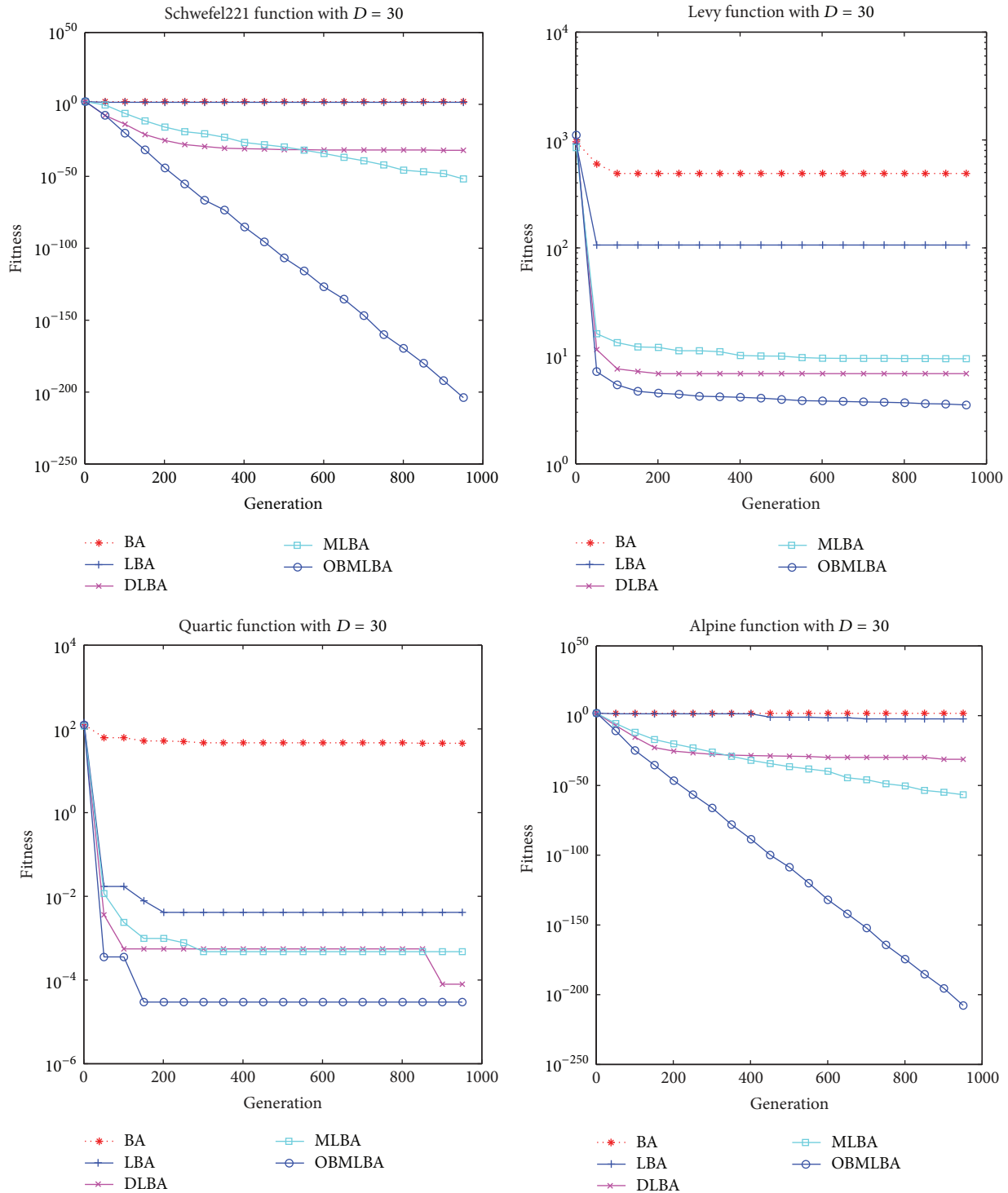


FIGURE 1: Convergence performance of BAs on the test functions.

5. Conclusions

Bat Algorithm, a metaheuristic algorithm proposed recently, has been successfully used to solve different types of optimization problems. However, it also has some inefficiency on exploration and exploitation capabilities.

In this study, a new variant of Bat Algorithm called OBMLBA is presented. The exploration and exploitation abilities of BA are balanced and enhanced in the search process by combining the information of the best-so-far solution and the neighborhood solutions into the search equations. At the same time, the proposed equation uses sinusoidal function

to define the bat pulse frequency f , allowing flexibility in changing the direction of f . Furthermore, Lévy Flight random walk is taken into consideration to escape from local optima. The concept of opposition based learning is introduced to the algorithm to improve the diversity and convergence capability. The proposed approach has been successfully used on 16 benchmark functions. Results and comparisons with the basic BA, DLBA, and other state-of-the-art algorithms show that OBMLBA has outperformed considered approaches in most of the functions. However, these experiments are just small scale optimization problems. The performance of the proposed algorithm on large scale optimization problems and real world optimization problems should be deeply investigated. The future study of this method may focus on investigating the mathematical foundations and applications of the proposed algorithm.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

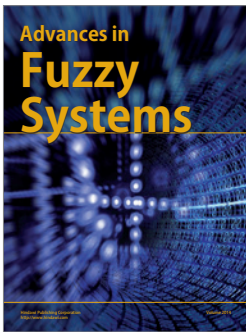
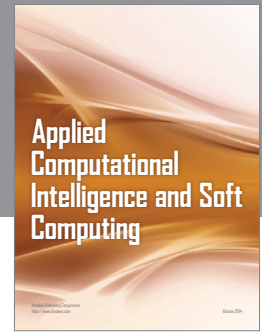
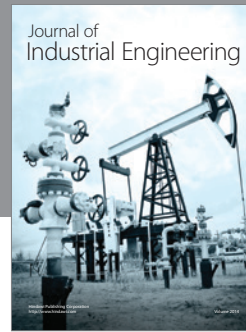
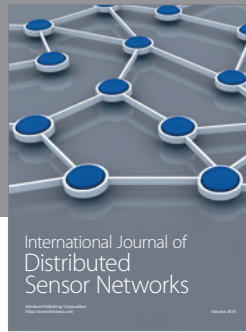
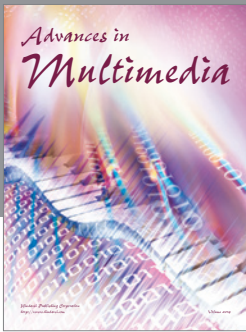
Acknowledgments

This work was supported in part by the National Nature Science Foundation of China under Grant 61402534, by the Shandong Provincial Natural Science Foundation, China, under Grant ZR2014FQ002, and by the Fundamental Research Funds for the Central Universities under Grant 16CX02010A.

References

- [1] S. Rao, *Engineering Optimization: Theory and Practice*, New Age International, 1996.
- [2] X. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2nd edition, 2010.
- [3] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 22–37, 1996.
- [4] 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.
- [5] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, December 1995.
- [6] M. Dorigo and T. Stutzle, *Ant Colony Optimization*, MIT Press, Cambridge, Mass, USA, 2004.
- [7] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [8] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in *PRICAI 2006: Trends in Artificial Intelligence*, Q. Yang and G. Webb, Eds., vol. 4099 of *Lecture Notes in Computer Science*, pp. 854–858, Springer, Berlin, Germany, 2006.
- [9] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, Coimbatore, India, December 2009.
- [10] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization*, pp. 65–74, Springer, 2010.
- [11] T. C. Bora, L. D. S. Coelho, and L. Lebensztajn, "Bat-inspired optimization approach for the brushless DC wheel motor problem," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 947–950, 2012.
- [12] M. R. Sathya and M. M. T. Ansari, "Load frequency control using Bat inspired algorithm based dual mode gain scheduling of PI controllers for interconnected power system," *International Journal of Electrical Power & Energy Systems*, vol. 64, pp. 365–374, 2015.
- [13] S. Mishra, K. Shaw, and D. Mishra, "A new meta-heuristic bat inspired classification approach for microarray data," *Procedia Technology*, vol. 4, pp. 802–806, 2012.
- [14] P. Musikapun and P. Pongcharoen, "Solving multi-stage multi-machine multi-product scheduling problem using bat algorithm," in *Proceedings of the 2nd International Conference on Management and Artificial Intelligence (IPEDR '12)*, vol. 35, pp. 98–102, 2012.
- [15] O. Hasançebi, T. Teke, and O. Pekcan, "A bat-inspired algorithm for structural optimization," *Computers and Structures*, vol. 128, pp. 77–90, 2013.
- [16] J. Zhang and G. Wang, "Image matching using a bat algorithm with mutation," *Applied Mechanics and Materials*, vol. 203, pp. 88–93, 2012.
- [17] E. S. Ali, "Optimization of Power System Stabilizers using BAT search algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 61, pp. 683–690, 2014.
- [18] 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.
- [19] L. L. Li and Y. Q. Zhou, "A novel complex-valued bat algorithm," *Neural Computing and Applications*, vol. 25, no. 6, pp. 1369–1381, 2014.
- [20] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, no. 2, pp. 224–232, 2014.
- [21] B. Bahmani-Firouzi and R. Azizipanah-Abarghooee, "Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 56, pp. 42–54, 2014.
- [22] N. S. Jaddi, S. Abdullah, and A. R. Hamdan, "Multi-population cooperative bat algorithm-based optimization of artificial neural network model," *Information Sciences*, vol. 294, pp. 628–644, 2015.
- [23] S. Yilmaz and E. U. Küçükşille, "A new modification approach on bat algorithm for solving optimization problems," *Applied Soft Computing*, vol. 28, pp. 259–275, 2015.
- [24] K. Khan and A. Sahai, "A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context," *International Journal of Intelligent Systems and Applications*, vol. 4, no. 7, pp. 23–29, 2012.
- [25] 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.

- [26] X.-S. He, W.-J. Ding, and X.-S. Yang, "Bat algorithm based on simulated annealing and gaussian perturbations," *Neural Computing and Applications*, vol. 25, no. 2, pp. 459–468, 2014.
- [27] J. Sadeghi, S. M. Mousavi, S. T. A. Niaki, and S. Sadeghi, "Optimizing a bi-objective inventory model of a three-echelon supply chain using a tuned hybrid bat algorithm," *Transportation Research Part E: Logistics and Transportation Review*, vol. 70, no. 1, pp. 274–292, 2014.
- [28] J.-H. Lin, C.-W. Chou, C.-H. Yang, and H.-L. Tsai, "A chaotic levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems," *Journal of Computing and Information Technology*, vol. 2, no. 2, pp. 56–63, 2012.
- [29] X.-B. Meng, X. Z. Gao, Y. Liu, and H. Zhang, "A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6350–6364, 2015.
- [30] X. Cai, W. Li, L. Wang, Q. Kang, Q. Wu, and X. Huang, "Bat algorithm with Gaussian walk for directing orbits of chaotic systems," *International Journal of Computing Science and Mathematics*, vol. 5, no. 2, pp. 198–208, 2014.
- [31] 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.
- [32] A. M. Taha and A. Y. C. Tang, "Bat algorithm for rough set attribute reduction," *Journal of Theoretical and Applied Information Technology*, vol. 51, no. 1, pp. 1–8, 2013.
- [33] P.-W. Tsai, J.-S. Pan, B.-Y. Liao, M.-J. Tsai, and V. Istanda, "Bat algorithm inspired algorithm for solving numerical optimization problems," *Applied Mechanics and Materials*, vol. 148-149, pp. 134–137, 2012.
- [34] I. Pavlyukevich, "Lévy flights, non-local search and simulated annealing," *Journal of Computational Physics*, vol. 226, no. 2, pp. 1830–1844, 2007.
- [35] A. Draa, S. Bouzoubia, and I. Boukhalifa, "A sinusoidal differential evolution algorithm for numerical optimisation," *Applied Soft Computing Journal*, vol. 27, pp. 99–126, 2015.
- [36] X.-S. Yang and S. Deb, "Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization," *Studies in Computational Intelligence*, vol. 284, pp. 101–111, 2010.
- [37] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes," *Physical Review E*, vol. 49, no. 5, pp. 4677–4683, 1994.
- [38] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution for optimization of noisy problems," in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 1865–1872, Vancouver, Canada, July 2006.
- [39] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, pp. 695–701, 2005.
- [40] W.-F. Gao, S.-Y. Liu, and L.-L. Huang, "A novel artificial bee colony algorithm based on modified search equation and orthogonal learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1011–1024, 2013.



Hindawi

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

