Global Convergence Analysis of the Bat Algorithm Using a Markovian Framework and Dynamical System Theory

Si Chen, Guo-Hua Peng

College of Natural and Applied Sciences, Northwestern Polytechnical University,
127 West Youyi Road, Beilin District, Xi'an, Shaanxi, 710072, P. R. China.
(emails: chen1991si@126.com, penggh@nwpu.edu.cn)

Xing-Shi He

College of Science, Xi'an Polytechnic University,
No. 19 Jinhua South Road, Xi'an, Shaanxi, 710048, P. R. China.
(email: xsh1002@126.com)

Xin-She Yang*

School of Science and Technology, Middlesex University,
The Burroughs, London NW4 4BT, United Kingdom
(email: x.yang@mdx.ac.uk)

Abstract

The bat algorithm (BA) has been shown to be effective to solve a wider range of optimization problems. However, there is not much theoretical analysis concerning its convergence and stability. In order to prove the convergence of the bat algorithm, we have built a Markov model for the algorithm and proved that the state sequence of the bat population forms a finite homogeneous Markov chain, satisfying the global convergence criteria. Then, we prove that the bat algorithm can have global convergence. In addition, in order to enhance the convergence performance of the algorithm and to identify the possible effect of parameter settings on convergence, we have designed an updated model in terms of a dynamic matrix. Subsequently, we have used the stability theory of discrete-time dynamical systems to obtain the stable parameter ranges for the algorithm. Furthermore, we use some benchmark functions to demonstrate that BA can

---

*Corresponding author

indeed achieve global optimality efficiently for these functions.

---

1. Introduction

The developments of computational intelligence and swarm intelligence have shown that nature-inspired algorithms can be effective (Engelbrecht, 2007; Yang, 2014; Jain et al., 2008; Akerkar and Sajja, 2009), and thus become widely used for various optimization problems (Kennedy and Eberharht, 1995; Yang, 2014; Deb et al. 2002; Leung and Wang 2001; Koziel and Yang, 2011). However, there is still a significant gap between theory and practice. Though the applications of algorithms are very successful, the relevant fundamental theory lacks behind. For example, the bat algorithm (BA), developed by Xin-She Yang in 2010 (Yang, 2010; Yang, 2011), has been shown to be very efficient in practice, but there is no mathematical theory for analyzing this algorithm. In fact, it lacks a general framework for systematical analysis of all swarm intelligence based algorithms. Though we know these algorithms can work well in practice, we rarely understand why they work so well and under what conditions or parameter ranges. These key challenges require further in-depth theoretical studies.

An encouraging trend is that researchers have realized the importance of theoretical analysis of algorithms and new studies start to emerge in recent years. Some analyses of nature-inspired algorithms used a dynamical system approach. For example, the stability analysis of particle swarm optimization (PSO) by Clerc and Kennedy (2002) generalized the parameters of the standard PSO, developed by Kennedy and Eberhardt (1995). In this work, Clerc and Kennedy (2002) analyzed the trajectory of a particle in the discrete-time space and then discussed a set of coefficients controlling the system's convergence tendencies. They also identified a critical bifurcation point, leading to different stable and oscillatory characteristics. In addition, Jiang et al. (2007) considered each particle's position in PSO as a stochastic vector using the standard stochastic process theory, and they derived the stochastic convergent conditions for PSO. On the other hand, Ren et al. (2012) used a Markov chain approach to study the convergence of PSO by first identifying the convergence conditions and then constructing a suitable Markov chain.

For the analysis of the bat algorithm, some recent preliminary results have been obtained. For example, Sheng et al. (2013) analyzed the convergence of a simplified variant of the BA according to the global convergence criterion of stochastic optimization algorithms, but there is no parametric analysis in their study. In addition, Li et al. defined the two modes of speeds and positions updating for the bat algorithm, followed by the analysis of the two modes defined by the characteristic equation (Li et al., 2013), but they have not considered the frequency variation in the bat algorithm. Huang et al. (2013) constructed a class of globally convergent BA variants to prove their global convergence, while applying it to solve large-scale optimization problems. But the version by Huang et al. was a modified variant, which is no longer the standard bat algorithm. To some extent, all the above studies have focused on the modified variants of the bat algorithm with further assumptions so as to simplify their mathematical

analyses. Therefore, there still lacks rigorous results about the standard bat algorithm, concerning both its convergence and stability.

Therefore, the main purpose of this paper to prove the convergence of the standard bat algorithm with new insight on parameter ranges and stability. The contribution of this paper is twofold: the convergence of the BA is analyzed using a Markovian framework in terms of finite-state, discrete-time Markov chains, while the stability and parameter ranges are obtained by the theory of dynamical systems. The paper is thus organized as follows. We will first introduce the basics of the BA in Section 2, followed by a brief review of the related work on its application and current theoretical analyses. Then, we introduce the global convergence criteria of random search algorithms in Section 3, and we will subsequently build a proper Markov model for the BA, followed by the main steps of proof of convergence in the same section. In Section 4, we use a dynamical system approach to study parameter variations and stability conditions for the bat algorithm. After theoretical analyses, we carry out some numerical experiments in Section 5 using a subset of 15 benchmark functions with diverse modality and properties to study the convergence behaviour of the algorithm during iterations. Finally, we draw some brief conclusions in Section 6.

2. Bat Algorithm and its Applications

2.1. Standard Bat Algorithm

The standard bat algorithm was developed by Yang in 2010 to solve continuous optimization problems (Yang, 2010). It has been extended to multiobjective optimization with many different applications (Yang, 2011; Gandomi et al., 2013; Natarajan et al., 2012). The BA, inspired by the echolocation behavior of microbat species, is a population-based algorithm using the frequency tuning with varying pulse emission rates and loudness so as to mimic the main nature of bats' echolocation when hunting for prey. The intention of BA is to act as a global optimizer using sufficient randomization and auto-switching between local and global moves, controlled by the actual emission rates and loudness of individuals (Yang, 2014).

Based on the original bat algorithm (Yang, 2010), each bat has a position vector x_i^t and a flying velocity v_i^t at iteration t in a D-dimensional search space. Their main algorithmic equations can be written as follows:

$$w_i = w_{min} + (w_{max} - w_{min})\beta, \tag{1}$$

$$v_i^{t+1} = \omega v_i^t + (p - x_i^t)w_i, \tag{2}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{3}$$

where w_i is the acoustic frequency of the i-th bat in the range of $[w_{\min}, w_{\max}]$. Here, ω indicates the inertia weight in the update of velocity, and in the standard bat algorithm $\omega = 1$ was used. For generality, we can use $\omega \in (0, 1]$. In addition, $\beta \in [0, 1]$ is a uniformly distributed random variable and p corresponds to the current best solution found by all the bats.

In a local search move, a new solution will be generated randomly around the old solution, often the current best solution. That is

$$X_{new} = X_{\text{old}} + \varepsilon A^t, \tag{4}$$

3

where X_{old} is a solution chosen from the current best solution set, A^t is the mean of the bats' loudness at t, and ε is a random number in $[-1, 1]$. The loudness A_i and the velocity r_i of the bats can be updated as follows:

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)], \tag{5}$$

where α and γ are constants ($0 < \alpha < 1, 0 < \gamma$).

For the convenience of the discussions below, we can rewrite the above equations [Eqs. (1)–(3)] in the following form:

$$v_i^{t+1} = \omega v_i^t + (p - x_i^t)w_i, \tag{6}$$

which is valid for each individual bat. The position vectors can be updated iteratively as

$$x_i^{t+1} = x_i^t + v_i^{t+1} = (1 - w_i)x_i^t + \omega v_i^t + pw_i \tag{7}$$

The stopping condition is usually the maximum number of iterations, or when the optimal value obtained by the population is sufficiently close to the known minimum fitness value.

2.2. Related Work

Since the appearance of the bat algorithm in 2010, it has been applied to solve a wide range of optimization problems (Yang and Gandomi 2012; Akhtar et al. 2012; Khan et al. 2011; Tsai et al. 2011; Bora et al. 2012; Ramesh et al. 2013). In addition, Mishra et al. (2012) used the bat algorithm for micro-array data classification, while Bazier et al. (2014) used the bat algorithm to solve economic load dispatch problems. The diverse applications were initially reviewed by Yang and He (2013), then a survey of the state-of-the-art developments was carried out by Chawla and Duhan (2015). There are more recent studies in this active area of research. Here, we will briefly highlight some of most recent developments.

Some new variants have been also been developed. For example, Kang et al. (2015) used a binary bat algorithm for fault diagnosis, while Jaddi et al. (2015) applied a modified bat algorithm to neural networks. Niknam et al. (2014) used the self-adaptive bat algorithm to solve unit commitment problems. Gandomi and Yang (2014) developed a chaotic bat algorithm to solve global numerical optimization problems, while Adarsh et al. (2016) used the chaotic bat algorithm to solve economic dispatch problems. More recently, Osaba et al. (2017) developed an improved discrete bat algorithm to solve symmetric and asymmetric travelling salesman problems. Banati and Chaudhary (2017) developed a multi-modal bat algorithm with improved search. Chakri et al. (2017) developed a new directional bat algorithm by incorporating the echolocation directions so as to enhance the search efficiency for continuous optimization problems. Alsalibi et al. (2017) developed a membrane-inspired bat algorithm to recognize faces in unconstrained scenarios. Furthermore, Gan et al. (2018) developed a new variant of the bat algorithm by using stochastic inertia weight and iterative local search. Al-Betar and Awadlallah (2018) developed an island bat algorithm for solving function optimization and economic load dispatch problems.

4

Another area of the state-of-the-art developments is the hardware implementation and specialization. For example, the hardware implementation of specialized robotic swarm has been carried out by Suárez et al. (2018). Ameur and Sakly (2017) provided an FPGA-based hardware implementation of the bat algorithm to make it suitable for solving large scale computing capabilities.

Even with the most recent developments concerning new variants and new applications, theoretical analysis still lags behind. Rigourous mathematical analysis is highly needed. This paper is such an attempt to analyze the bat algorithm using the Markov chain theory and dynamical system theory.

3. Global Convergence Analysis

3.1. Recent Studies using Markov chains

The analysis of algorithms can be carried out from different perspectives. Different methods can describe different properties and thus can gain different insight into the studied algorithms (Yang 2014). For the convergence analysis, the theory of Markov chains has been applied to the standard particle swarm optimization (Pan et al. 2013; Xu and Yu 2018). Pan et al. (2013) used a Markov chain approach to analyze PSO by defining the state sequence of a particle and swarm state sequences, and then proved that the standard PSO algorithm could converge with a certain probability. They also explain the effect of the inertia weight and acceleration factor on premature convergence. Xu and Yu (2018) used a supermartingale approach to analyze the evolutionary sequences of a particle swarm with the best fitness values, and they concluded that the standard PSO could reach the global optimum in probability.

Furthermore, a Markovian framework has been applied to study the convergence behaviour of simulated annealing and its variants. Meise (1998) considered a parallel simulated annealing algorithm as multiple independent Markov chains using different selection strategies, and concluded that the selection strategies by best-wins did not in general converge to the set of global optima. Gerber and Bornn (2018) provided a set of conditions which could ensure the almost sure convergence of a class of simulated annealing algorithms based on a time-varying Markov kernel. For the classical differential evolution (DE) algorithm, Hu et al. (2014) showed that DE could not convergence to the global optimal set with probability one. Then, they modified the algorithm by introducing two additional operators to increase the diversity of the population, and then proved its convergence. For the cuckoo search (CS) algorithm, He et al. (2018) used a discrete-time Markov chain approach and introduced a simplified CS algorithm by focusing on the main updating equation in CS. They then showed that the simplified CS algorithm could converge with probability one.

However, there is no analysis of the convergence of the bat algorithm (BA). Therefore, this work is new in the sense that we analyze the convergence of the BA rigorously for the first time. In addition, the ways of constructing Markov chains are different, and consequently the transition probability for the bat algorithm is very different. A key difference of our present work is that we use a full version of the bat algorithm (BA) and construct Markov chains based on the updating equations of the BA.

In the rest of this section, we will first state the conditions for global convergence, followed by the construction of proper Markov chains from the bat algorithm. Then, we will prove the convergence of the bat algorithm.

3.2. Global Convergence Criteria

Depending on the actual algorithms and the framework of theoretical analysis, the convergence of an algorithm can be tested by certain criteria. One of the commonly used criteria is based on the two conditions outlined by Solis and Wets (1981).

Let $\langle H, f \rangle$ be the optimization problem with a fitness function and a feasible solution space H. A stochastic optimization S iterates for t iterations and the new solution x^{t+1} can be obtained from solution x^t by

$$x^{t+1} = S(x^t, \zeta), \tag{8}$$

where ζ is the solution set found by the algorithm S during the iterative process.

Let us define the bounds of the search on the Lebesgue metric space as the infinum

$$\theta = \inf\{k : v[x \in H | f(x) < k] > 0\}, \tag{9}$$

where $v[X]$ is the measure on set X, which means that there are non-empty subsets in the search space and the fitness value corresponding to the element in the non-empty subset can be infinitely close to θ. Thus, the neighbourhood or region of optimal solutions can be defined as

$$R_{\varepsilon,M} = \begin{cases} \{x \in H \, | f(x) < \theta + \varepsilon\}, & -\infty < \theta < \infty, \\ \{x \in H \, | f(x) < M\}, & \theta = -\infty, \end{cases} \tag{10}$$

where $\varepsilon > 0$ and $M < 0$. If a stochastic algorithm finds a point in $R_{\varepsilon,M}$, then we can consider that the algorithm finds the global optimal solution or an approximation to the global optimal solution.

In general, two conditions are necessary to guarantee the global optimality is achievable during iterations:

Condition 1: An optimization algorithm S should guarantee that the sequence $\{f(x^t)\}_{t=0}^{\infty}$ is decreasing. Also, there is $\zeta \in H$ such that

$$f(S(x, \zeta)) \leq f(\zeta). \tag{11}$$

Condition 2: For all subsets $\forall B \in H$ subject to $v(B) > 0$, we have

$$\prod_{t=0}^{\infty} (1 - u_t(B)) = 0, \tag{12}$$

where $u_t(B)$ represents the probability measure of the t-th iterative result of the random algorithm S on B.

Mathematically speaking, a stochastic optimization algorithm that can have a guaranteed global convergence is based on the following lemma or criterion (Solis and Wets, 1981; Jiang et al., 2007; Villobos-Arias, 2005).

Criterion 1. *For f is measurable and the feasible solution space H is a measurable subset on R^n, if the stochastic algorithm S satisfies both Condition 1 and Condition 2, sequence $\{x^t\}_{t=0}^{\infty}$ is generated by the algorithm S will lead to*

$$\lim_{t \to \infty} P(x^t \in R_{\varepsilon,M}) = 1, \tag{13}$$

where $P(x^t \in R_{\varepsilon,M})$ represents the probability that the best solution obtained by algorithm S after t iterations belongs to $R_{\varepsilon,M}$.

In other words, the above criterion means that the algorithm will converge with a probability one as the number of iterations is sufficiently large, which equivalently means that the algorithm can have almost guaranteed global convergence.

3.3. Global Convergence Analysis of the Bat Algorithm

In order to prove the convergence of the bat algorithm, we will introduce some preliminaries. If the position of each bat individual in the BA is considered as a state x, then the process of states x^t with pseudotime or iteration counter t can be considered as a random process. For such a stochastic process, the Markov chain can be an effective tool to analyze its convergence in a probability sense.

3.3.1. Preliminaries

Let us first define the state, state space and other relevant concepts that will be used later for proving the global convergence of the BA.

The states of bats and the state space can be defined as follows:

Definition 1. *The position of a bat individual x with velocity v and historical best position p forms its state or status, denoted by $a = (x, v, p)$, where $x, p \in H$.*

In addition, we have $f(p) \leq f(x)$ and $v \in [v_{min}, v_{max}]$. All possible states of all bats form a state space for bats, denoted by

$$A = \Big\{ a = (x, v, p) | x, p \in H, \ f(p) \leq f(x), \ v \in [v_{min}, v_{max}] \Big\}. \tag{14}$$

Furthermore, the states and state space of the bats population or group can be defined as follows:

Definition 2. *The set of all N bat individuals is called the bat group, and the states of this bat group can be denoted by $b = (a_1, a_2, \ldots, a_N)$. The collection of all possible bat group status or states forms the bat group status space, denoted by*

$$B = \Big\{ b = (a_1, a_2, \ldots, a_N), a_i \in A \ (1 \leq i \leq N) \Big\}. \tag{15}$$

From the above definitions, it is obvious that the definition of bat group status B already contains the best position (or the best solution vector) in the group history. Furthermore, the state transition for the positions of bats representing solutions can be defined as follows:

7

For $\forall a_1 = (x_1, v_1, p_1) \in A$ and $\forall a_2 = (x_2, v_2, p_2) \in A$ during the iterations of the BA, the state transition from a_1 to a_2 can be denoted by

$$F_A(a_1) = a_2, \tag{16}$$

where F_A is the transition function from a_1 to a_2 in the state space A.

Similarly, for $\forall b_i = (a_{i,1}, a_{i,2}, \ldots, a_{i,N}) \in H$ and $\forall b_j = (a_{j,1}, a_{j,2}, \ldots, a_{j,N}) \in H$, the iterative process of the BA in essence transfers the bat group states from b_i to b_j. That is

$$F_b(b_i) = b_j. \tag{17}$$

3.3.2. Markov Chain Model for BA

In order to prove the convergence using a Markov chain framework, we have to build a Markov chain model for the bat algorithm. Let us first start with a theorem:

Theorem 1. *In the BA, the bat status a_1 is essentially shifted in one step to the status a_2, and its transition probability is the joint probability*

$$P(F_A(a_1) = a_2) = P(x_1 \rightarrow x_2)P(v_1 \rightarrow v_2)P(p_1 \rightarrow p_2), \tag{18}$$

where $P(x_1 \rightarrow x_2)$ is the transition probability of the bat position from x_1 to x_2, $P(v_1 \rightarrow v_2)$ is the transition probability of the bat velocity from v_1 to v_2, and $P(p_1 \rightarrow p_2)$ is the transition probability of the best position (in the whole history) from p_1 to p_2.

Proof. The status of a bat is transferred via $a_1(x_1, v_1, p_1) \rightarrow a_2(x_2, v_2, p_2)$. That is, $x_1 \rightarrow x_2$, $v_1 \rightarrow v_2$, and $p_1 \rightarrow p_2$ are carried out simultaneously. The joint probability of $F_A(a_1) \rightarrow a_2$ is

$$P(F_A(a_1) = a_2) = P(x_1 \rightarrow x_2)P(v_1 \rightarrow v_2)P(p_1 \rightarrow p_2). \tag{19}$$

From the updating equations for velocities and positions (see Eqs.(2) and (3)), it is easy to see that the transition probability of the positions of bats can be calculated by

$$P(x_1 \rightarrow x_2) = \begin{cases} \frac{1}{|f(p_g - x_1)|}, & v_2 \in [x_1 + \omega v_1, \omega v_1 + w_i(p_g - x_1)], \\ 0, & v_2 \notin [x_1 + \omega v_1, \omega v_1 + w_i(p_g - x_1)]. \end{cases} \tag{20}$$

Similarly, the transition probability concerning the velocities of bats can be calculated by

$$P(v_1 \rightarrow v_2) = \begin{cases} \frac{1}{|f(p_g - x_1)|}, & v_2 \in [\omega v_1, \omega v_1 + w_i(p_g - x_1)], \\ 0, & v_2 \notin [\omega v_1, \omega v_1 + w_i(p_g - x_1)]. \end{cases} \tag{21}$$

In addition, the transition probability of the best position p of all bats is

$$P(p_1 \rightarrow p_2) = \begin{cases} 1, & f(p_2) < f(p_1), \\ 0, & f(p_2) \geq f(p_1). \end{cases} \tag{22}$$

It is worth pointing out that we treat the optimization problem as a minimization problem. Thus, p_2 is better than p_1 if $f(p_2) < f(p_1)$.

$\square$

With these results, we can now prove the following theorem:

Theorem 2. *In the iterative process of the BA, the transition probability of the bat group status from b_i to b_j is given by*

$$P(F_b(b_i) = b_j) = \prod_{t=1}^{t_N} P(F_A(a_{it}) = a_{jt}), \tag{23}$$

where t_N is the total number of iterations so far.

Proof. As $F_b(b_i) = b_j$ indicates that each state in the bat group state b_i is simultaneously transferred to group state b_j; that is

$$F_A(a_{i1}) = a_{j1}, F_A(a_{i2}) = a_{j2}, \ldots, F_A(a_{it_N}) = a_{jt_N}.$$

Then, the transition probability of a group transition of the bat group should be the joint probability of each iteration step. Thus, we have

$$\begin{aligned} P(F_b(b_i) = b_j) &= P(F_A(a_{it}) = a_{jt})P(F_A(a_{it}) = a_{jt})\ldots P(F_A(a_{it_N}) = a_{jt_N}) \\ &= \prod_{t=1}^{t_N} P(F_A(a_{it}) = a_{jt}), \end{aligned} \tag{24}$$

which concludes the proof. $\square$

Now we have to show that the state sequence a is a finite, homogeneous Markov chain.

Theorem 3. *In the BA, the bat group state sequence a is indeed a finite homogeneous Markov chain.*

Proof. For any optimization algorithm, its search space during the whole iterative process is finite because both the population size and the number of iterations are finite, so each of the bat state $a = (x, v, p)$ among the x, v, p are finite, which leads to the fact that the bat state space is finite.

From the algorithmic equations outlined in Section 2, the position updates of each bat individual is an iterative equation, so the random process of positions of the BA changes with time, which is not the Markov process. However, if we can group the position, velocity and global optimal values together as one state B, then state $B(t + 1)$ is only related to state $B(t)$, not its history. Then, sequence B has proper Markov chain properties.

From Eqs.(1)-(3), $\beta \in [0, 1]$ is a random vector that is uniformly distributed, and the algorithmic equations [i.e., (1) to (3)] form a stochastic system. It is straightforward to show that the state $B(t - 1)$ of the system at time t transferring to the new state $B(t)$ is completely determined by its state at time t. In addition, the factor γ and ω as well as the pseudotime t in the iterative formulas are independent of the state of the system before time t.

From $B(t - 1)$ to $B(t)$ of bats group state sequence $\{B(t); t \geq 0\}$, the transition probability $P(F_B(B(t - 1)) = B(t))$ of the two states is determined by the transition

9

probability of all individuals in the bats group, and the probability of transition can be calculated by the joint probability of $P(x(t-1) \rightarrow x(t))$, $P(v(t-1) \rightarrow v(t))$, and $P(p(t-1) \rightarrow p(t))$, according to Theorem 1.

In addition, $P(x(t-1) \rightarrow x(t))$ and $P(v(t-1) \rightarrow v(t))$ are only related to x, v, p at time $t-1$. Thus, $P(F_B(B(t-1)) = B(t))$ is only related to the state $a_i(t-1), 1 \leq i \leq N$ of all bats at time $t-1$. Therefore, the Markov chains are finite.

Furthermore, from Theorem 1, $P(F_A(a(t-1)) = a(t))$ is independent of time $t-1$. A similar argument also indicates that $P(F_B(B(t-1)) = B(t))$ is also independent of $t-1$. Therefore, the finite Markov chains are homogeneous. $\quad\square$

3.3.3. Global Convergence of the BA

With the above definitions and theorems, let us proceed to prove the convergence of the bat algorithm.

For the true optimal solution g for an optimization problem $\langle H, f \rangle$ with an objective function $f(x)$ where x is a vector, the optimal state set can be defined as

$$L = \{a = (x, v, p) | f(p) = f(g), a \in A\}. \tag{25}$$

Obviously we have $L \subseteq A$ as L should be a subset of A. If in any case $L = A$, any solution in A is equally optimal, which means that the objective landscape is flat (thus it is equivalent to a feasibility problem in which the objective does not exert any selection pressure on different solutions). This is just a special case and the optimal solution is already achieved, and thus we will not discuss this case any further.

In addition, for the optimal solution g to an optimal problem $\langle H, f \rangle$, the optimal bat group state set can be defined as

$$U = \{B = (a_1, a_2, \ldots, a_N) | \exists a_i \in L, 1 \leq i \leq N\}, \tag{26}$$

which means that the optimal bat groups state set U is the set of all bat groups such that at least one bat individual in the population with its state belong to L.

Using the same methodology as outlined in He et al. (2018) (Theorems 7 and 8 in their paper) and the results in Zhang and Li (2003), we can prove the following three theorems:

Theorem 4. *When $U \subset B$, there is no closed set I other than B such that $I \bigcap U = \emptyset$.*

Theorem 5. *If a Markov chain has a non-empty set Z with no closed set D satisfying $Z \bigcap D = \emptyset$, then $\lim_{t \to \infty} P(x^t = j) = \pi_j$, only if $j \in Z$, and $\lim_{t \to \infty} P(x^t = j) = 0$ only if $j \notin Z$.*

Theorem 6. *If the number of iteration approaches infinity or sufficiently large, the group state sequence will converge to the optimal state set U.*

From the above four theorems, it is straightforward to prove the following global convergence theorem:

Theorem 7. *The bat algorithm with the Markov chain model defined in Section 3.3.2 has guaranteed global convergence.*

Proof. From the convergence criterion (Criterion 1), we know that if a stochastic optimization algorithm can satisfy both Condition 1 and Condition 2, it will converge to the global optimality. In essence, the first condition (Condition 1) can guarantee that the fitness value $f(x)$ of the stochastic optimization algorithm is decreasing. From the above discussions about the iterative process of the BA, we can conclude that it is indeed $f(p) \leq f(x^t)$.

Furthermore, the previous theorem means that the group state sequence will converge towards the optimal set after a sufficiently large number of iterations, which means that the probability of not reaching the globally optimal solution is asymptotically zero. This means that the second convergence condition is also satisfied. As a result, a conclusion can be drawn that the BA has guaranteed global convergence towards its global optimality with a probability one. □

This proof is based on the Markov chain framework, and thus the convergence is in a probabilistic sense. It is an important result because it shows that the bat algorithm can indeed converge. However, there is no information about the rate of convergence and how the parameters may affect the convergence behaviour of this algorithm.

It is worth pointing out that the above proof has been based on a simplified Markov model for the bat algorithm. The standard bat algorithm also includes the variations of pulse emission rate and loudness, and effects of such variations have not been considered here. However, the overall convergence behaviour can be very similar.

In order to gain further insight into the parameter values and their effect on the convergence of the bat algorithm, we now use a completely different approach to analyze the algorithm in terms of dynamic matrix theory.

4. Convergence Analysis Based on Dynamic Matrix Theory

4.1. Stability of Algorithms and Dynamical Systems

The advantage of the above algorithm analysis in terms of the Markovian framework is that it provides important insights into the convergence of an algorithm in the probability sense, and the convergence properties are less sensitive to the exact settings of parameters. However, this approach does not give enough information about how quickly the convergence can be achieved and how stable the algorithm can be under given parameter settings. Dynamical system theories, on the other hand, provide an alternative approach to analyze algorithms rigorously from a different perspective where parameter settings are important. In this case, dynamic matrices can be constructed from the updating equations of an algorithm, and the insight can be gained about the possible parameter ranges and sufficient conditions for the algorithm to converge.

One of the earliest studies on the stability of the particle swarm optimizer is by Clerc and Kennedy (2002) in which they showed that the stability and convergence tendencies could largely depend on the coefficients of various terms in PSO such as the inertia weight and learning rates. Later, Kadirkamanathan et al. (2006) analyzed the particle swarm optimizer using the Lyapunov stability theory and derived sufficient conditions for its stability. For differential evolution, Dasgupta et al. (2009) used the dynamical system theory to analyze a one-dimensional DE population and they analyzed its stability in the light of Lyapunov's stability theorems. However, for the

bat algorithm, there are no existing studies on its stability. Therefore, in the rest of this section, we will analyze the stability of the bat algorithm by using the theory of dynamical systems.

4.2. Bat Algorithm as a Dynamical System

For this purpose and for simplicity of calculations without losing generality, it is assumed that the current optimal solution in the bat algorithm population is a constant vector p (even though it is updated at each iteration). It is assumed that the frequency w_i is also a constant $m \geq 0$. Within this framework, the velocities and positions of bats during the iterations can be written as

$$v^{k+1} = lv^k + (p - x^k)m, \tag{27}$$

$$x^{k+1} = cx^k + uv^{k+1}, \tag{28}$$

where coefficient m is essentially the average of the frequencies, while l, c and u are the weight coefficients so that we can analyze the algorithm in general. The attraction point p in the D-dimensional space is the current optimal position. The algorithm represented by the system of (27) and (28) now has four parameters to be tuned. They are l, m, c, u. We will show that two of these parameters are key parameters.

4.3. Dynamic Matrix Model for the Bat Algorithm

From the algorithmic equations (27) and (28), we can rewrite (27) equivalently using the previous iteration as

$$x^k = cx^{k-1} + uv^k.$$

Then multiplying its both sides by l and re-arranging it slightly, we have

$$luv^k = lx^k - clx^{k-1}. \tag{29}$$

Combining (27) and (28), we have

$$x^{k+1} = cx^k + uv^{k+1} = cx^k + u[lv^k + (p - x^k)m]$$

$$= cx^k + ulv^k + ump - umx^k = cx^k + [lx^k - clx^{k-1}] + ump - umx^k, \tag{30}$$

where we have used Eq. (29).

By re-arranging the above equation, we have a recursive relationship for x^k

$$mup = x^{k+1} + (mu - c - l)x^k + lcx^{k-1}. \tag{31}$$

It is obvious that m and u always appear as a single factor mu, not individually. This means that only their product matters. Thus, for simplicity (without loss of generality), we can set

$$u \equiv 1. \tag{32}$$

12

Therefore, we have a reduced system of algorithmic equations for the bat algorithm as

$$v^{k+1} = lv^k + m(p - x^k), \tag{33}$$

$$x^{k+1} = cx^k + v^{k+1}. \tag{34}$$

In addition, as the number of iterations k increases, it can be expected that the series should converge to p (the current best solution found by all the bats). That is

$$\lim_{k\to\infty} x^k = p, \quad \lim_{k\to\infty} x^{k+1} = p, \quad \lim_{k\to\infty} x^{k-1} = p. \tag{35}$$

Taking the limit of (31) and using the above results, we have

$$mp = p + (m - c - l)p + lcp, \tag{36}$$

which gives that

$$p(l - 1)(c - 1) = 0. \tag{37}$$

Thus, it has three solutions: either $p = 0$, or $l = 1$, or $c = 1$. The case of $p = 0$ is either a special case when the actual global minimization value is $f_{min} = p = 0$ for some functions, or it is not possible to satisfy. Therefore, we have either $l = 1$ or $c = 1$.

Considering the role of c in the algorithm, it acts as an equivalent inertia weighting factor so we can set $c = 1$ for the moment, which does not affect the update of the position vectors.

Now we have finally obtained the reduced dynamic system for the bat algorithm

$$v^{k+1} = lv^k + m(p - x^k) = -mx^k + lv^k + mp, \tag{38}$$

$$x^{k+1} = x^k + v^{k+1} = x^k + [-mx^k + lv^k + mp], \tag{39}$$

which leads to

$$v^{k+1} = -mx^k + lv^k + mp, \tag{40}$$

$$x^{k+1} = x^k + lv^k + mp - mx^k. \tag{41}$$

We can rewrite the above dynamic system in a matrix form as

$$Y^{k+1} = CY^k + Mp, \tag{42}$$

where

$$Y^k = \begin{bmatrix} x^k \\ v^k \end{bmatrix}, \qquad C = \begin{bmatrix} 1 - m & l \\ -m & l \end{bmatrix}, \qquad M = \begin{bmatrix} m \\ m \end{bmatrix}. \tag{43}$$

Here, the Y^k column vector corresponds to the states of positions and velocities of the bats at iteration k. Matrix C is the dynamic matrix that governs the main properties of this dynamic system. M is the input of the frequencies and p is the current best solution in the system.

As the iterations continue and the bat population move towards p, the velocity of the bat population will approach zero. That is

$$\lim_{k\to\infty} v^k = 0. \tag{44}$$

13

Therefore, the final fixed point or point of convergence in the state space Y_k is

$$Y* = \begin{bmatrix} p \\ 0 \end{bmatrix}. \tag{45}$$

The final state of convergence is that $\lim_{k\to\infty} x^k = p$ and $\lim_{k\to\infty} v^k = 0$ if there is no perturbation.

4.4. Algorithm Convergence and Parameter Selection

The main properties of the dynamic system is now determined by the eigenvalues of the dynamic matrix C. That is

$$\det \begin{vmatrix} 1 - m - \lambda & l \\ -m & l - \lambda \end{vmatrix} = 0, \tag{46}$$

which gives

$$(1 - m - \lambda)(l - \lambda) + ml = 0, \tag{47}$$

or simply

$$\lambda^2 + \lambda(m - l - 1) + l = 0. \tag{48}$$

Thus, their solutions are

$$\lambda = \frac{-(m - l - 1) \pm \sqrt{(m - l - 1)^2 - 4l}}{2}, \tag{49}$$

which gives two eigenvalues λ_1 and λ_2. For the dynamic system to be stable (Bhatia and Szegö 2002), the theory of discrete-time dynamical systems requires that the modulus of the eigenvalues must be smaller than one; that is $|\lambda| \leq 1$. From Vieta's formulas for polynomials, we know that $\lambda_1 \cdot \lambda_2 = l$ whose modulus should also be less than one, so we have $|l| \leq 1$ or $-1 \leq 1 \leq 1$.

In addition, Vieta's formulas also indicates that

$$\lambda_1 + \lambda_2 = -(m - l - 1) = l - m + 1 \leq 2, \tag{50}$$

which must be less than 2 (i.e., $\lambda_1 + \lambda_2 \leq 2$) so that each eigenvalue is potentially less than 1. We have

$$l \leq m + 1, \tag{51}$$

For the conditions that the modulus of the biggest eigenvalue must be smaller than one $|\lambda| \leq 1$, we have

$$\frac{(l - m + 1) \pm \sqrt{(m - l - 1)^2 - 4l}}{2} \leq +1, \tag{52}$$

or

$$-1 \leq \frac{(l - m + 1) \pm \sqrt{(m - l - 1)^2 - 4l}}{2}. \tag{53}$$

14

The first equation (52) becomes

$$(l - m - 1) \leq \mp \sqrt{(m - l - 1)^2 - 4l}, \tag{54}$$

or

$$(l - m - 1)^2 \geq (m - l - 1)^2 - 4l, \tag{55}$$

which gives

$$(l - m - 1)^2 - [(m - 1 - 1)^2 - 4l] = 4m \geq 0, \tag{56}$$

or simply $m \geq 0$. Here, we have used the fact $l \leq m + 1$ (or $l - m - 1 \leq 0$), thus the inequality should be reversed when taking the square.

Similarly, the other condition becomes

$$-(l - m + 3) \leq \pm \sqrt{(m - l - 1)^2 - 4l}, \tag{57}$$

or

$$(l - m + 3)^2 \geq (m - l - 1)^2 - 4l, \tag{58}$$

which gives

$$(l - m + 3)^2 - [(m - l - 1)^2 - 4l] = 4(2l + 2 - m) \geq 0, \quad \text{or} \quad 2l + 2 \geq m. \tag{59}$$

Therefore, the conditions for stability and convergence are

$$\begin{cases} -1 \leq l \leq +1, \\ m \geq 0, \\ 2l - m + 2 \geq 0. \end{cases} \tag{60}$$

which form a triangular region in the parameter space of l and m, as shown in Fig. 1. The above analysis shows that within the parameter ranges of m and l, the bat algorithm will not only converge towards the optimality but also converge stably. In this case, the algorithm is stable and can converge quickly in practice. However, it should be emphasized that the dynamic model presented in this paper has not considered the variation of pulse emission rate and loudness, thus the actual parameter ranges may be different from the above results. Even so, this simplified model has enabled us to understand the influence of parameter values for the bat algorithm.

In the rest of the paper, we will use some selected benchmarks to show the convergence properties of the bat algorithm under different parameter settings. This will allow us to validate that the above theoretical results are consistent with practical observations for functions with a diverse range of properties and modality.

5. Validation by Numerical Experiments

In order to verify the bat algorithm and show the convergence characteristics discussed above in this paper, we have conducted some numerical experiments using a few selected benchmark functions with very different properties and modalities. These
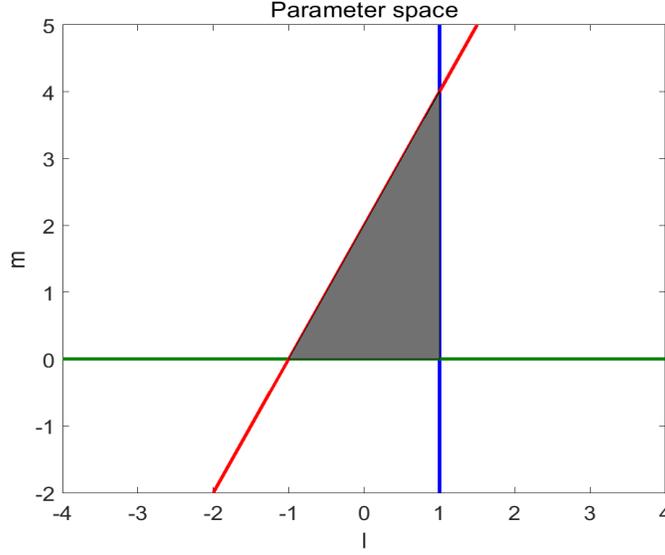
Figure 1: Parameter ranges for the bat algorithm to be stable.

Table 1: simulation benchmarks

Function name	Function	Range	f_{min}
Sphere	$f_1(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2$	[-5.12,5.12]	0
Griewank	$f_2(\boldsymbol{x}) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0
Schwefel	$f_3(\boldsymbol{x}) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|$	[-10,10]	0
Quartic	$f_4(\boldsymbol{x}) = \sum_{i=1}^{D} ix_i^4 + \text{rand}[0,1]$	[-100,100]	0
Rosenbrock	$f_5(\boldsymbol{x}) = \sum_{i=1}^{D-1} [(x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2]$	[-5,5]	0
Yang	$f_6(\boldsymbol{x}) = \sum_{i=1}^{D} |x_i| \exp[-\sum_{i=1}^{D} \sin(x_i^2)]$	[-2\pi, 2\pi]	0
Zakharov	$f_7(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} \frac{ix_i}{2}\right)^2 + \left(\sum_{i=1}^{D} \frac{ix_i}{2}\right)^4$	[-5,5]	0
Step Function	$f_8(\boldsymbol{x}) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	[-100,100]	0
Rastrigin	$f_9(\boldsymbol{x}) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	0

functions are listed in Table 1 where the dimensionality is chosen as $D = 30$ for all functions.

For each function, the bat algorithm has been executed with a maximum number of iterations $t_{max} = 500$ with a population size $n = 12$, $m = 2$ and $l = 0.5$. The dimensions for all functions are $D = 30$.

The convergence curves for all the functions are shown in Fig. 2 where we can clearly see that all functions can converge quickly, especially at the early stage of the iterations. However, if the parameter ranges lie outside the stable domain, the rate of convergence can be significantly lower, and the very slow convergence or even premature convergence can occur as can be seen in Fig. 3 where $m = -3$ and $l = 4$ are used,
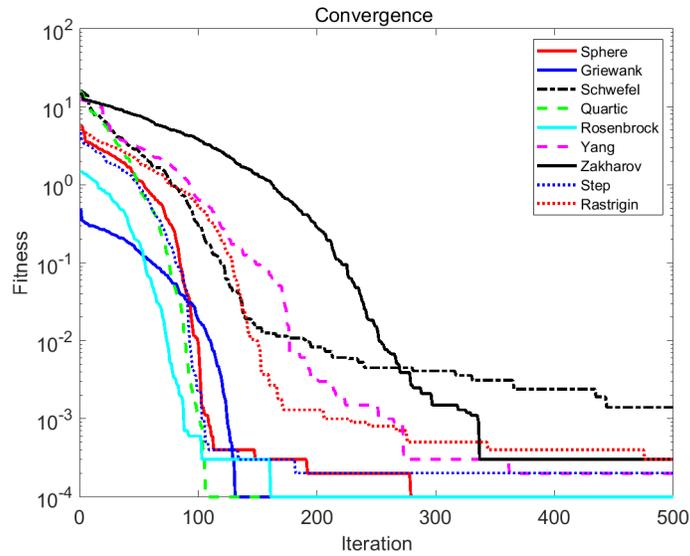
Figure 2: Plots of convergence for various functions.

even though all the other parameters remain the same.
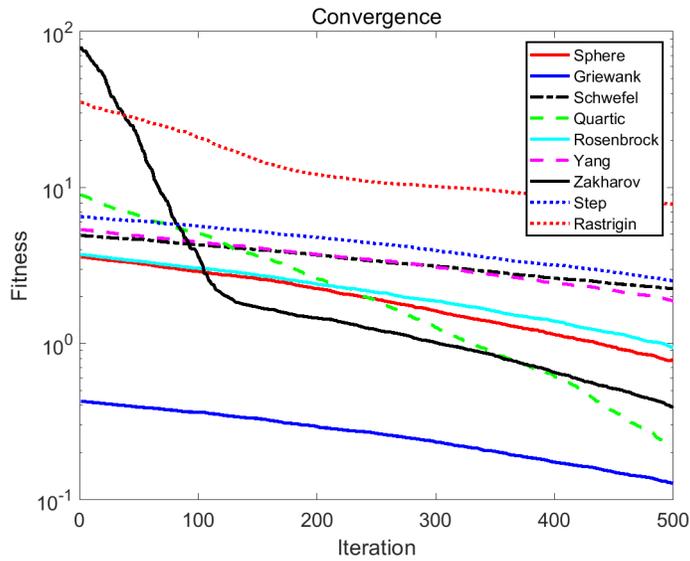
Figure 3: Plot of convergence when the parameters lie outside the triangular domain.

To further test the convergence properties of the bat algorithm for more complex

benchmarks, we have also used a subset of functions from the standard CEC2015 benchmark suite (Chen et al. 2014; Qu et al. 2016). We have selected six more functions

- Rotated high conditioned elliptic function with $F_{1,min} = 100$.

- Rotate cigar function with $F_{2,min} = 200$.

- Rotated discus function with $F_{3,min} = 200$.

- Shifted and rotated Ackley's function with $F_{4,min} = 300$.

- Shifted and rotated Rastrigin's function with $F_{5,min} = 400$.

- Shifted and rotated Schwefel's function with $F_{6,min} = 500$.

As the main purpose of these tests is to see the convergence behaviour, we have used three different settings of parameters for each function. One setting is within the stable region, one setting is on the boundary (solid line) and one setting is outside of the normal parameter ranges. The results are shown in Fig. 4. As we can clearly see, good convergence can be achieved for parameters inside the stable region and even on its boundary. However, if the settings are outside the stable region, the convergence is usually far worse than those with settings inside the right parameter region. Such results confirm our theoretical results, and also highlight the importance of parameter settings.

It is worth pointing out that the CEC2015 functions are more challenging to solve, and the number of iterations needed to reach the optimality is much higher, at least 10000 (at least 20 times more than we used here). To some extent, the extensive simulations of the bat algorithm have been carried out in the literature (Al-Betar et al. 2018; Banati and Chaudhary 2017; Chakri et al. 2017; Chawla and Duhan 2015). However, the main purpose of this part of tests is to compare the convergence behaviour with different parameter settings, which clearly shows such effects.

All the above analyses and simulations have demonstrated that the algorithm can converge both quickly and robustly. Thus, the algorithm can be suitable for difficult optimization problems where optimal or nearly optimal solutions are needed quickly.

6. Conclusions

The bat algorithm has been shown to be effective in practice, but there is not much theoretical analysis in the literature. This paper provides some theoretical analysis of the standard bat algorithm using both a Markov chain model and a dynamic matrix model. The Markov model shows that the algorithm can converge to the global optimality with probability one as the number of iterations becomes sufficiently large. The dynamic model looks at the algorithm from a different perspective. By extending the models with more parameters, we have then gained some insight and explained why some parameters are not important, while others can be tuned. As a result, the parameter ranges of some key parameters have been identified.

Following the theoretical analyses, we have used some benchmark test functions to validate the bat algorithm using the appropriate parameter settings. Good convergence has been observed for all functions, which is consistent with the theoretical results. Then we have used a set of benchmark functions to validate the theoretical results so as to see how the convergence characteristics may be affected by the settings of algorithm-dependent parameters. Simulation results confirm that good convergence and stability can be obtained if the parameters are within the stable region in the parameter space. However, convergence will be slowed down or premature convergence may appear if the parameters are outside the stable region. These are clearly consistent with the theoretical analysis carried out earlier in this paper.

It is worth pointing out that the models used in this paper are simplified models without considering the variation of pulse emission rate and loudness. It will be useful to investigate the effect of such factors in the convergence properties and stability of the bat algorithm. In addition, even we now understand why the bat algorithm converge with a clear parameter region, it still lacks the information on the rate of convergence and how the parameter values will affect the rate of convergence. Future work will also investigate this issue further with more rigorous analyses.

References

Adarsh, B. R., Raghunathan, T., Jayabarathi, T., Yang, X. S. (2016). Economic Dispatch Using Chaotic Bat Algorithm. Energy. 96, 666-675.

Akerkar, R. and P. S. Sajja, P. S., (2009). Bio-Inspired Computing: Constituents and Challenges. International Journal of Bio-Inspired Computation. 1(3): 135-150.

Akhtar, S., Ahmad, A. R., Abdel-Rahmanm, E. M. (2012). A Metaheuristic Bat-Inspired Algorithm for Full Body Human Pose Estimation. Proceedings of 9th Conference on Computer and Robot Vision (CRV2012), pp. 369-375.

Al-Betar, M. A., Awadallah, M. A. (2018). Island Bat Algorithm for Optimization. Expert Systems with Applications. 107(1), 126-145.

Alsalibi, B., Venkat, I., Al-Betar, M. A. (2017). A Membrame-Inspired Bat Algorithm to Recognize Faces in Unconstrained Scenario. 64, 242-260.

Ameur, M. S. B., Sakly, A. (2017). FPGA Based Harward Implementation of Bat Algorithm. Applied Software Computing. 58(2), 378-387.

Banati, H., Chaudhary, R. (2017). Multi-Modal Bat Algorithm with Improved Search (MMBAIS). Journal of Computational Science. 23(1), 130-144.

Baziar, A., Rostamim, M., Akbari-Zadeh, M. (2014). An Intelligent Approach Based On Bat Algorithm for Solving Economic Dispatch with Practical Constraints. Journal of Intelligent Fuzzy Systems. 27(3): 1601-1607.

Bhatia, N. P., Szegö, G. P. (2002). Stability Theory of Dynamical Systems. Springer.

Bora, T. C., Coelho, L. D. S., Lebensztajn, L. (2012). Bat-Inspired Optimization Approach for the Brushless DC Wheel Motor Problem. IEEE Transactions on Magnetics. 48(2), 947-950.

Chakri, A., Khelif, R., Benouret, M., Yang, X. S. (2017). New Directional Bat ALgoirthm for Continuous Optimization Problems. Expert Systems with Applications. 69(1), 159-175.

Chawla, M., Duhan, M. (2015). Bat Algorithm: A Survey of the State-of-the-Art. Applied Artificial Intelligence. 29(6), 617-634.

Chen, Q., Liu, B., Zhang, Q., Suganthan, P. N., Qu, B. Y.,(2014). Problem Defintion and Evoluation Criteria for CEC2015 Special Seesion and Competition on Bound Constrained Single Objective Computationally Expensive Numerical Optimization. Technical Report. Computational Intelligence Laboratory, Zhengzhou University, China and Technical Report, Nanyang Technical University, Singapore.

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evolutionary Computation. 6(1), 58-73.

Dasgupta, S., Das, S., Biswas, A., Abraham, A., (2009). On stability and convergence of the population-dynamics in differential evolution, AI Communication. 22(1), 1-20.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 6(2), 182-197.

Engelbrecht, A. P. (2007). Computational Intelligence: An Introduction. John Wiley Sons, pp. 5-24.

Gerber, M. and Bornn, L. (2018). Convergence Results for a Class of Time-Varying Simulated Annealing Algorithms. Stochastic Processes and their Applications. 128(4), 1073-1094.

Gan, C., Cao, W. H., Wu, M., Chen, X. (2018). A New Bat Algorithm Based on Iterative Local Search and Stochastic Inertia Weight. Expert Systems with Applications. 104, 202-212.

Gandomi, A. H., Yang, X. S., Alavi, A. H., Talatahari, S. (2013). Bat Algorithm for Constrained Optimization Tasks. Neural Computing Applications. 22(6), 1239-1255.

Gandomi, A. H., Yang, X. S. (2014). Chaotic Bat Algorithm. Journal of Computational Science. 5(2), 224-232.

He, X. S., Wang, F., Wang, Y., Yang, X. S. (2018). Global convergence analysis of cuckoo search using Markov theory, in: Nature-Inspired Algorithms and Applied Optimization (Eds. X.S. Yang), Springer Nature, Cham, Switzerland, pp. 53-67.

Hu, Z. B., Xiong, S. W., Su, Q. H., Fang, Z. X. (2014). Finite Markov Chain Analysis of Classical Differential Evolution Algorithm. J. Computational and Applied Mathematics. 268 (1), 121-134.

Huang, G. Q., Zhao, W. J., Lu, Q. Q. (2013). Bat algorithm with global convergence for solving large-scale optimization problem, Application Research of Computers. 30 (5), 1323-1328.

Jaddi, B. S., Abdullah, S., Hamdan, A. R. (2015). Optimization of Neural Network Model Using Modified Bat-Inspired Algorithm. Applied Soft Computing. 2015, 37: 71-86.

Jain, L. C., Tan, S. C., Lim, C. P. (2008). An Introduction to Computational Intelligence Paradigms. Springer, pp. 36-75.

Jiang, M., Luo, Y. P., Yang, S. Y. (2007). Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm, Information Processing Letters. 102(1), 8-16.

Kadirkamanathan, V., Selvarajah, K., Fleming, P. J., (2006). Stability analysis of the particle dynamics in particle swarm optimizer, IEEE Transactions on Evolutionary Computation. 10(3), 245-255.

Kang, M., Kim, J. Kim, J. (2015). Reliable Fault Diagnosis for Incipient Low-Speed Bearings Using Feature Analysis Based On a Binary Bat Algorithm. Information Sciences. 294(3), 423-438.

Kennedy, J., Eberhart, R. C., (1995). Particle swarm optimization, in: Proc. Of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948.

Khan, K., Nikov, A., Sahai, A. (2011). A fuzzy bat clusterig method for ergonomic screening of office workplaces, S3T 2011, Advances in Intelligent and Soft Computing. pp. 59-66.

Koziel, S., Yang, X. S. (2011). Computational Optimization, Methods and Algorithms, Springer, Berlin.

Leung, Y., Wang, Y. (2001). An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization. IEEE Transactions on Evolutionary Computation. 5(1), 41-53.

Li, Z. Y., Ma, L., Zhang, H. Z. (2013). Convergence Analysis of Bat Algorithm. Mathematical in Practice and Theory. 43 (12): 182-190.

Meise, C. (1998). On the Convergence of Parallel Simulated Annealing. Stochastic Processes and their Applications. 76(1), 99-115.

Mishra, S., Shaw, K., Mishra, D. (2012). A New Meta-Heuristic Bat Inspired Classification Approach for Microarray Data. Procedia Technology. pp. 802-806.

Natarajan, A., Subramanian, S., Premalatha, K. (2012). A Comparative Study of Cuckoo Search and Bat Algorithm for Bloom Filter Optimisation in Spam Filtering. International Journal of Bio-Inspired Computation. 4(2), 89-99.

Niknam, T., Bavafa, F., Azizipanah-Abarghooee, R. (2014). New Self-Adaptive Bat-Inspired Algorithm for Unit Commitment Problem. IET Science Measurement Technology. 8(6), 505-517.

Osaba, E., Yang, X. S., Diaz, F., Lopez-Garcia, P., Carballedo, R. (2017). An Improved Discrete Bat Algorithm for Symmetric and Asymmetric Travelling Salesman Problems. Engineering Applications of Artificial Intelligence. 48(1), 59-71.

Pan, F., Li, X.-T., Zhou, Q., Li, W.-X. (2013). Analysis of Standard Particle Swarm Optimization Algorithm Based on Markov Chain. Acta Automatica Sinica. 39(4), 381-389.

Qu, B. Y., Liang, J. J., Wang, Z. Y., Chen, Q., Suganthan, P. N. (2016). Novel Benchmark Functions for Continous Multimodal Optimization with Comparative Results. Swarm and Evolutionary Computation. 26(1), 23-34.

Ramesh, B., Mohan, V. C. J., Reddy, V. V., (2013). Application of Bat Algorithm for Combined Economic Load and Emission Dispatch. International Journal of Electricl Engineering and Telecommunications. 2013, 2(1): 1-9.

Ren, Z. H., Wang, J., Gao, Y. L., (2012). The global convergence analysis of particle swarm optimization algorithm based on Markov chain, Control Theory and Applications (in Chinese), 2012,8(4): 462-466.

Sheng, M. L., He, X. S., Ding, W. J. (2013). Analysis of bat algorithm's global convergence, Basic Sciences Journal of Textiles Universities. 26(4), 543-547.

Solis, F., Wets, R. (1981). Minimization by random search techniques. Mathematics of Operations Research, 6(1), 19-30.

Suárez, P., Iglesias, A., Gálvez, A. (2018). Make Robots be Bats: Specializing Robotic Swarms to the Bat Algorithm. Swarm and Evolutionary Computation. (In Press). Online 14 Feb 2018. https://doi.org/10.1016/j.swevo.2018.01.005

Tsai, P. W., Pan, J. S., Liao, B. Y., Tsai, M. J., Istanda, V. (2011). Bat Algorithm Inspired Algorithm for Solving Numerical Optimization Problems. Applied Mechanics and Materials. 148(1), 134-137.

Villalobos-Arias, M., Coello Coello, C. A., Hernández-Lerma, O. (2005). Asymptotic convergence of metaheuristics for multiobjective optimization problems, Soft Computing, 2005, 10(8): 1001-1005.

Xu, G, and Yu, G. S. (2018). On Convergence Analysis of Particle Swarm Optimization Algorithm. J. Computational and Applied Mathmatics. 333(1), 65-73.

Yang, X. S. (2014). Nature-Inspired Optimization Algorithms, Elsevier Insight, London.

Yang, X. S. (2010). A New Met Heuristic Bat-Inspired Algorithm, Nature Inspired Cooperative Strategies for Optimization. pp. 65-74.

Yang, X. S. (2011). Bat Algorithm for Multi-Objective Optimisation. International Journal of Bio-Inspired Computation. 3(5), 267-274.

Yang, X. S., Gandomi, A. H. (2012). Bat Algorithm: A Novel Approach for Global Engineering Optimization. Engineering Computations. 29(5), 464-483.

Yang, X. S., He, X.-S. (2014). Bat Algorithm: Literature Review and Applications. Int. Journal of Bio-inspired Computation. 5(3), 141-149.

Zhang, W. X., Yi, L. (2003). The Mathematical Basis of Genetic Algorithm (in Chinese), Xi'an Jiaotong University Press.
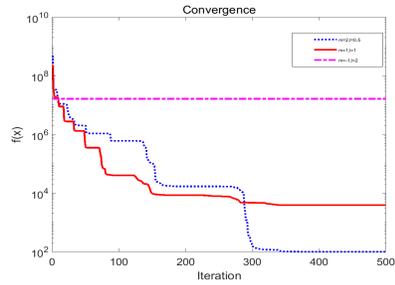
(a) CEC2015 f_1
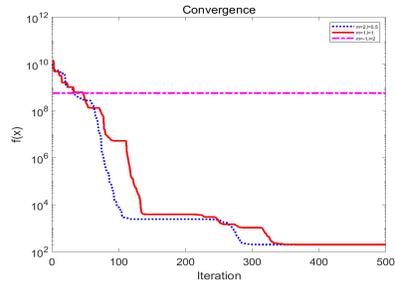
(b) CEC2015 f_2
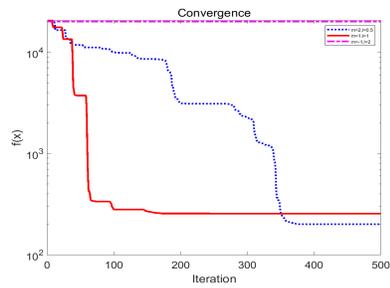
(c) CEC2015 f_3
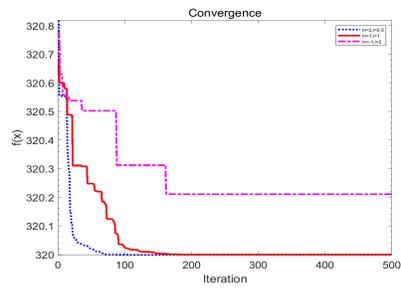
(d) CEC2015 f_4
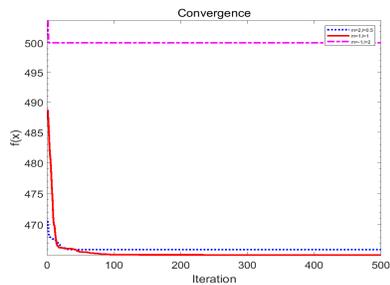
(c) CEC2015 f_5
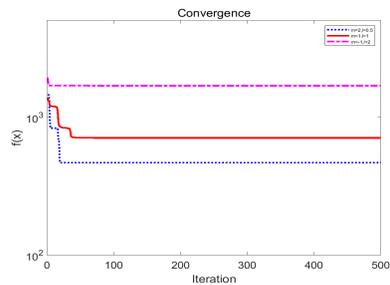
(d) CEC2015 f_6

Figure 4: Convergence behaviour of different parameter settings using CEC2015 benchmarks.