Global Convergence Analysis of Cuckoo Search Using Markov Theory

Xing-Shi He, Fan Wang, Yan Wang and Xin-She Yang

Abstract The cuckoo search (CS) algorithm is a powerful metaheuristic algorithm for solving nonlinear global optimization problems. In this book chapter, we prove the global convergence of this algorithm using a Markov chain framework. By analyzing the state transition process of a population of cuckoos and the homogeneity of the constructed Markov chains, we can show that the constructed stochastic sequences can converge to the optimal state set. We also show that the algorithm structure of cuckoo search satisfies two convergence conditions and thus its global convergence is guaranteed. We then use numerical experiments to demonstrate that cuckoo search can indeed achieve global optimality efficiently.

Keywords Cuckoo search · Convergence rate · Global convergence · Markov chain theory · Optimization · Swarm intelligence

1 Introduction

Nature-inspired algorithms have become widely used for optimization and computational intelligence [11, 12, 26–28, 30]. Many new optimization algorithms are based on the so-called swarm intelligence with diverse characteristics in mimicking natural systems. However, there is a significant gap between theory and practice. Most metaheuristic algorithms have very successful applications in practice, but their mathematical analysis lags far behind. In fact, apart from a few limited results about the convergence and stability concerning particle swarm optimization, genetic algorithms, simulated annealing and others [4, 10, 16], many algorithms do not have any

X.-S. He · F. Wang · Y. Wang
College of Science, Xi'an Polytechnic University, Xi'an, People's Republic of China

X.-S. Yang (✉)
School of Science and Technology, Middlesex University, London NW4 4BT, UK
e-mail: x.yang@mdx.ac.uk; xy227@cam.ac.uk

X.-S. Yang (ed.), *Nature-Inspired Algorithms and Applied Optimization*,
Studies in Computational Intelligence 744, https://doi.org/10.1007/978-3-319-67669-2_3

theoretical analysis. Therefore, we may know they can work well in practice, but we rarely understand why they work and how to improve them with a good understanding of their working mechanisms.

In this work, we will try to prove the convergence of the cuckoo search (CS) so as to gain insight into its search mechanisms. The rest of this paper is organized as follows: we will introduce the details of the cuckoo search algorithm in Sect. 2, followed by the introduction of the convergence criteria in Sect. 3 and the detailed convergence analysis in Sect. 4. Then, we validate the cuckoo search algorithm by numerical experiments and observe its convergence behaviour in Sect. 5. Finally, we conclude by summarizing the main results in Sect. 6.

2 Cuckoo Search

Cuckoo search (CS) is one of the recent nature-inspired metaheuristic algorithms, developed in 2009 by Xin-She Yang and Suash Deb [23]. CS is based on the brood parasitism of some cuckoo species. In addition, this algorithm is enhanced by the so-called Lévy flights [15], rather than by simple isotropic random walks. Recent studies show that CS is potentially far more efficient than PSO and genetic algorithms [8, 24]. A relatively comprehensive review of the studies up to 2014 was carried out by Yang and Deb [25].

2.1 Standard Cuckoo Search

Cuckoo behaviour is intriguing because of the so-called brood parasitism reproduction strategy. Some species such as the *ani* and *Guira* cuckoos lay their eggs in communal nests, though they may remove others' eggs to increase the hatching probability of their own eggs. Quite a number of species engage obligate brood parasitism by laying their eggs in the nests of other host birds (often other species such as warblers). In addition, the eggs laid by cuckoos may be discovered and thus abandoned with a probability, around 1/4 to 1/3, depending on species and the average number of eggs in a nest.

For simplicity in describing the cuckoo search, we now use the following three idealized rules [23]:

1. Each cuckoo lays one egg at a time, and dumps it in a randomly chosen host nest.
2. The best nests with high-quality eggs will be carried over to the next generations.
3. The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$. In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new nest at a new location.

As a further approximation, this last assumption can be approximated by a fraction p_a of the n host nests are replaced by new nests (with new random solutions). For a maximization problem, the quality or fitness of a solution can simply be proportional to the value of the objective function.

From the implementation point of view, we can use the following simple representation, that each egg in a nest represents a solution, and each cuckoo can lay only one egg (thus representing one solution), the aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests. Obviously, this algorithm can be extended to the more complicated case where each nest has multiple eggs representing a set of solutions. For this present work, we will use the simplest approach where each nest has only a single egg. In this case, there is no distinction between an egg, a nest, or a cuckoo, as each nest corresponds to one egg which also represents one cuckoo, corresponding to a single solution vector.

This algorithm uses a balanced combination of a local random walk and the global explorative random walk, controlled by a switching parameter p_a. The local random walk can be written as

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \beta s \otimes H(p_a - \epsilon) \otimes (\mathbf{x}_j^t - \mathbf{x}_k^t), \tag{1}$$

where $\mathbf{x}_j^t$ and $\mathbf{x}_k^t$ are two different solutions selected randomly by random permutation, $H(u)$ is a Heaviside function, ϵ is a random number drawn from a uniform distribution, and s is the step size. Here β is the small scaling factor. On the other hand, the global random walk is carried out by using Lévy flights

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \alpha \otimes L(s, \lambda), \tag{2}$$

where

$$L(s, \lambda) \sim \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (s \gg 0). \tag{3}$$

Here $\alpha > 0$ is the step size scaling factor, which should be related to the scales of the problem of interest. Here '$\sim$' denotes that the fact that the random numbers $L(s, \lambda)$ should be drawn from the Lévy distribution on the right-hand side, which is approximated by a power-law distribution with an exponent λ. In addition, $\otimes$ is an entry-wise operation.

The above equation is essentially the stochastic equation for a random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above equation) and the transition probability (the second term). However, a substantial fraction of the new solutions should be generated by far field randomization and their locations should be far enough from the current best solution; this will make sure that the system will not be trapped in a local optimum [23, 25].

2.2 Cuckoo Search in Applications

Cuckoo search has been applied in many areas of optimization, engineering design, data ming and computational intelligence with promising efficiency. For example, in the engineering design applications, cuckoo search has superior performance over other algorithms for a range of continuous optimization problems such as spring design and welded beam design problems [8, 24, 25].

In addition, a modified cuckoo search by Walton et al. [21] has demonstrated to be very efficient for solving nonlinear problems such as mesh generation. Vazquez [20] used cuckoo search to train spiking neural network models. Yildiz [32] has used cuckoo search to select optimal machine parameters in milling operation with enhanced results. Then Durgun and Yildiz [7] used CS for the optimization of vehicle components, while Zheng and Zhou [33] provided a variant of cuckoo search using Gaussian process. In the context of data fusion and wireless sensor network, cuckoo search has been shown to be very efficient [5, 6].

Among the diverse applications, an interesting performance enhancement has been obtained by using cuckoo search to train neural networks as shown by Valian et al. [18] and reliability optimization problems [19].

For complex phase equilibrium applications, Bhargava et al. [2] have shown that cuckoo search offers a reliable method for solving thermodynamic calculations. Furthermore, Moravej and Akhlaghi [13] have solved DG allocation problem in distribution networks with good convergence rate and performance. Taweewat and Wutiwiwatchi have combined cuckoo search and supervised neural network to estimate musical pitch with reduced size and higher accuracy [17].

As a further extension, Yang and Deb [31] developed a multiobjective cuckoo search (MOCS) for design engineering applications. For multiobjective scheduling problems, another progress was made by Chandrasekaran and Simon [3] using cuckoo search algorithm, which demonstrated the superiority of their proposed methodology. Recent studies have demonstrated that cuckoo search can performance significantly better than other algorithms in many applications [8, 14, 29, 32, 33].

2.3 Simplified Cuckoo Search

In the cuckoo search algorithm, a set of two updating equations are used. One equation is mainly for global moves, while the other is mainly for local exploitation. Whether it is global or local is largely determined by the step sizes of the moves of new solutions from the existing solutions in the population. However, since Lévy flights can have both small steps and occasionally large steps, it can carry out both local and global moves simultaneously. Thus, it is difficult to put into a fixed category. However, in order to simplify the analysis and also to emphasize the global search capability, we now use a simplified version of cuckoo search. That is, we use only the global branch with a random number $r \in [0, 1]$, compared with a discovery/switching probability p_a. Now we have

$$\begin{cases} \mathbf{x}_i^{(t+1)} \leftarrow \mathbf{x}_i^{(t)} & \text{if } r < p_a, \\ \mathbf{x}_i^{(t+1)} \leftarrow \mathbf{x}_i^{(t)} + \alpha \otimes L(s, \lambda) & \text{if } r > p_a. \end{cases} \quad (4)$$

Obviously, due to the stochastic and iterative nature of the cuckoo search algorithm, we have to focus on the key steps. Therefore, we use the following steps to represent the simplified cuckoo search [22]:

- Step 1. Generate randomly an initial population of n nests at the positions, $\mathbf{X} = \{\mathbf{x}_1^0, \mathbf{x}_2^0, ..., \mathbf{x}_n^0\}$, then evaluate their objective values and record the initial best $\mathbf{g}_t^0$.
- Step 2. Generate new solutions/moves by

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha \otimes L(s, \lambda). \quad (5)$$

- Step 3. Draw a uniformly distributed random number r from $[0, 1]$. Update $\mathbf{x}_i^{(t+1)}$ if $r > p_a$. Then, evaluate the new solutions and update the new global best $\mathbf{g}_t^*$ at iteration t.
- Step 4. Stop if the stopping criterion is satisfied and output the global best $\mathbf{g}_t^*$. Otherwise, go to step (2).

Though this is a simplified version of cuckoo search, it captures all the main characteristics of the standard cuckoo search. Thus, the proof of its global convergence will be equivalent to the proof of the global convergence of the original algorithm.

3 Markov Chains and Convergence Criteria

For the ease of analysis and notations, let us first use $<\Omega_s, f>$ to denote the optimization problem with an objective f in the search space Ω_s. This problem is to be solved by a stochastic search algorithm A. The solution obtained at the t-th iteration can be written as

$$\mathbf{x}_{t+1} = A(\mathbf{x}_t, \xi), \quad (6)$$

where Ω_s is the feasible solution space. ξ denotes the set of the visited solutions of algorithm A during the iterative process.

Loosely speaking, the infimum of the search in the Lebesgue measure space can be defined as

$$\phi = \inf \Big(t : v(x \in \Omega_s \Big| f(x) < t] > 0 \Big), \quad (7)$$

where $v[X]$ denotes the Lebesque measure on the set X. In essence, Eq. (7) represents the non-empty set in the search space, and the region or regions for optimal solutions can be defined as

$$R_{\epsilon,M} = \begin{cases} \{x \in \Omega_s | f(x) < \phi + \epsilon\} & \text{if } \phi \text{ is finite,} \\ \{x \in \Omega_s | f(x) < -C\} & \text{if } \phi = -\infty, \end{cases} \tag{8}$$

where $\epsilon > 0$ and $C \gg 1$ is a sufficiently large positive number. Loosely speaking, the set $R_{\epsilon,M}$ is a set that can be belong to different regions in the search space, depending on the objective landscapes. As long as this set is accessible, for any solution or a point falling into $R_{\epsilon,M}$ during the iteration, we can say that algorithm A has reached the optimal set and thus found the globally optimal solution or its best approximation.

The two conditions for convergence are as follows [9, 10]:

- 1 If $f(A(x, \xi)) \leq f(x)$ and $\xi \in \Omega_s$, we have

$$f(A(x, \xi)) \leq f(\xi). \tag{9}$$

  Here we focus on minimization problems. For maximization problems, the inequality is reversed, but the rest are the same.
- 2 For any set $S \in \Omega_s$ with $\nu(S) > 0$, we have

$$\prod_{k=0}^{\infty}(1 - u_k(S)) = 0, \tag{10}$$

  where $u_k(S)$ corresponds to the probability measure on S at the kth iteration of the algorithm A.

Before we proceed, let us use the results about the global convergence of an algorithm, based on existing studies without repeating the proofs [9, 10]:

Theorem 1 *If the objective f is measurable and its feasible solution space Ω_s forms a measurable subset in $\Re^n$, then algorithm A can indeed satisfy the above two conditions with the search sequence $\{x_k\}_{k=0}^{\infty}$, which will lead to*

$$\lim_{k\to\infty} P(x_k \in R_{\epsilon,M}) = 1. \tag{11}$$

That means that algorithm A will converge globally with a probability one. Here $P(x_k \in R_{\epsilon,M})$ is the probability measure of the kth solution on $R_{\epsilon,M}$ at the kth iteration.

This same methodology has been used by He et al. to prove the global convergence of the flower pollination algorithm [9]. In this book chapter, we will use essentially the same procedure to prove the global convergence of cuckoo search by first proving the constructed Markov chains are proper and the conditions of convergence are satisfied.

4 Global Convergence Analysis

In order to simplify the presentations and analysis, let us first introduce some formal definitions and some preliminary results.

4.1 Preliminaries

Now we start to define the state and state space to be used later for proving the global convergence of the cuckoo search. For simplicity of notations, we use the standard non-bold case symbols for vectors and variables in the rest of this chapter.

Definition 1 The positions of a cuckoo/nest and its corresponding global best solution g in the search history forms the states of cuckoos: $y = (x, g)$ where $x, g \in \Omega_s$ and $f(g) \le f(x)$ (for minimization). The set of all the possible states forms the state space, denoted by

$$Y = \{y = (x, g)|x, g \in \Omega_s, f(g) \le f(x)\}. \tag{12}$$

The state and state space of the cuckoo population or group can be defined as follows:

Definition 2 The states of all n cuckoos/nests form the states of the group, denoted by $q = (y_1, y_2, ..., y_n)$. All the states of all the cuckoos form a state space for the group, denoted by

$$Q = \{q = (y_1, ..., y_i, ..., y_n), y_i \in Y, 1 \le i \le n\}. \tag{13}$$

As Q contains all the states found during the iterations, it also contains the historical global best solution g^* for the whole population as well as all individual best solutions $g_i (1 \le i \le n)$ in history. Obviously, the global best solution of the whole population is the best among all g_i, so that $f(g^*) = \min(f(g_i)),\ 1 \le i \le n$.

Furthermore, the state transition for the positions of cuckoos representing solutions can be defined as follows. For $\forall y_1 = (x_1, g_1) \in Y$ and $\forall y_2 = (x_2, g_2) \in Y$, the state transition from y_1 to y_2 can be denoted by

$$T_y(y_1) = y_2. \tag{14}$$

4.2 Markov Chain Model for Cuckoo Search

One of the main tasks here is that we have to build a Markov chain model for cuckoo search algorithm, and the first step is to prove a theorem to be used later.

Theorem 2 *The transition probability from state y_1 to y_2 in the cuckoo search is*

$$P(T_y(y_1) = y_2) = P(x_1 \to x_1')P(g_1 \to g_1')P(x_1' \to x_2)P(g_1' \to g_2), \tag{15}$$

where $P(x_1 \to x_1')$ is the transition probability at Step 2 in cuckoo search, and $P(g_1 \to g_1')$ is the transition probability for the historical global best at this step. $P(x_1' \to x_2)$ is the transition probability at Step 3, while $P(g_1' \to g_2)$ is the transition probability of the historical global best.

Proof In the simplified cuckoo search, the state transition from y_1 to y_2 only has one middle transition state (x_1', g_1'), which means that $x_1 \to x_1', g_1 \to g_1', x_1' \to x_2$ and $g_1' \to g_2$ are valid simultaneously. Then, the probability for $P(T_y(y_1) = y_2)$ is

$$P(T_y(y_1) = y_2) = P(x_1 \to x_1')P(g_1 \to g_1')P(x_1' \to x_2)P(g_1' \to g_2). \tag{16}$$

From Eq. (5), the transition probability for $x_1 \to x_1'$ is

$$P(x_1 \to x_1') = \begin{cases} \frac{1}{|g-x_1|} & \text{if } x_1' \in [x_1, x_1 + (x_1 - g)], \\ 0 & \text{if } x_1' \notin [x_1, x_1 + (x_1 - g)]. \end{cases} \tag{17}$$

Since x and g are higher-dimensional vectors, the mathematical operations here should be interpreted as vector operations, while the $|\cdot|$ means the volume of the hypercube.

The transition probability of the historical best solution is

$$P(g_1 \to g_1') = \begin{cases} 1 & f(x_1') \leq f(g_1), \\ 0 & f(x_1') > f(g_1). \end{cases} \tag{18}$$

From Step 3 in the simplified cuckoo search algorithm, we know that a random number $r \in [0, 1]$ is compared with the discovery probability $p_a = 0.25 = 1/4$. If $r > p_a$, then the position/solution of a cuckoo can be changed randomly; otherwise, it remains unchanged. Therefore, the transition probability for $x_1' \to x_2$ is

$$P(x_1' \to x_2) = \begin{cases} 1 - p_a & \text{if } r > p_a, \\ p_a & \text{if } r \leq p_a \end{cases} = \begin{cases} \frac{3}{4} & \text{if } r > p_a, \\ \frac{1}{4} & \text{if } r \leq p_a. \end{cases} \tag{19}$$

The transition probability for the historical best solution is

$$P(g_1' \to g_2) = \begin{cases} 1 & f(x_2) \leq f(g_1), \\ 0 & f(x_2) > f(g_1). \end{cases} \tag{20}$$

Furthermore, the group transition probability in the cuckoo search can be defined as $T_q(q_i) = q_j$ for $\forall q_i = (y_{i1}, y_{i2}, ..., y_{in}) \in \Omega_s$ and $\forall q_j = (y_{j1}, y_{j2}, ..., y_{jn}) \in \Omega_s$.

Theorem 3 *In the simplified cuckoo search, the group transition probability from q_i to q_j in one step is*

$$P(T_q(q_i) = q_j) = \prod_{k=1}^{n} P(T_y(y_{ik}) = y_{jk}). \tag{21}$$

Proof If the group states can be transferred from q_i to q_j in one step, then all the states will be transferred simultaneously. That is, $T_y(y_{i1} = y_{j1}, T_y(y_{i2}) = y_{j2}$, ..., $T_y(y_{in}) = y_{jn}$, and the group transition probability can be written as the joint probability

$$P(T_q(q_i) = q_j) = P(T_y(y_{i1}) = y_{j1})P(T_y(y_{i2}) = y_{j2}) \cdots P(T_y(y_{in}) = y_{jn})$$

$$= \prod_{k=1}^{n} P(T_y(y_{ik}) = y_{jk}). \tag{22}$$

Theorem 4 *The state sequence $\{q(t); t \geq 0\}$ in the cuckoo search is a finite homogeneous Markov chain.*

Proof First, let us assume that all search spaces for a stochastic algorithm are finite. Then, x and g in any cuckoo/nest state $y = (x, g)$ are also finite, so that the state space for cuckoos/nests are finite. Since the group state $q = (y_1, y_2, ..., y_n)$ consists of n positions of the n cuckoos/nests where n is positive and finite, so group states q are also finite.

From the previous theorems, we know that the group transition probability

$$P(T_q(q(t-1)) = q(t), \tag{23}$$

for $\forall q(t-1) \in Q$ and $\forall q(t) \in Q$ is the group transition probability $P(T_y(y_i(t-1)) = y_i(t))$ for $1 \leq i \leq n$. From Eq. (16), we have the transition probability for any cuckoo is

$$P(T_y(y(t-1)) = y(t)) = P(x(t-1) \to x'(t-1))P(g(t-1) \to g'(t-1))$$

$$\times P(x'(t-1) \to x(t))P(g'(t-1) \to g(t)), \tag{24}$$

where $P(x(t-1) \to x'(t-1))$, $P(g(t-1) \to g'(t-1))$, $P(x'(t-1) \to x(t))$ and $P(g'(t-1) \to g(t))$ are all only depend on x and g at $t-1$. Therefore, $P(T_q(q(t-1)) = q(t))$ also only depends on the states $y_i(t-1), 1 \leq i \leq n$ at time $t-1$. Consequently, the group state sequence $\{q(t); t \geq 0\}$ has the property of a Markov chain.

Finally, $P(x(t-1) \to x'(t-1))$, $P(g(t-1) \to g'(t-1))$, $P(x'(t-1) \to x(t))$ and $P(g'(t-1) \to g(t))$ are all independent of t, so is $P(T_y(y(t-1)) = y(t))$. Thus, $P(T_q(q(t-1)) = q(t)$ is also independent of t, which implies that this state sequence is also homogeneous.

In summary, the group state sequence $\{q(t); t \geq 0\}$ is a finite, homogeneous Markov chain.

4.3 Global Convergence of Cuckoo Search

For the globally optimal solution g_b for an optimization problem $< \Omega_s, f >$, the optimal state set is defined as $R = \{y = (x, g) | f(g) = f(g_b), y \in Y\}$. In addition, for the globally optimal solution g_b to an optimization problem $< \Omega_s, f >$, the optimal group state set can be defined as

$$H = \{q = (y_1, y_2, ..., y_n) | \exists y_i \in R, 1 \leq i \leq n\}. \tag{25}$$

With the above results and definitions, we are now ready to prove the following theorems:

Theorem 5 *Given the position state sequence* $\{y(t); t \geq 0\}$ *in cuckoo search, the state set* R *of the optimal solutions corresponding to optimal nests/cuckoos form a closed set on* Y.

Proof For $\forall y_i \in R, \forall y_j \notin R$, the probability for $T_y(y_j) = y_i$ is $P(T_y(y_j) = y_i) = P(x_j \to x_i')P(g_j \to g_j')P(x_j' \to x_i)P(g_j' \to g_j)$. Since for $\forall y_i \in R$ and $\forall y_j \notin R$, it holds that $f(g_i) \geq f(g_j) = f(g_b) = \inf(f(a)), a \in \Omega_s$.

From Eqs. (18–20), we have $P(g_j \to g_j')P(g_j' \to g_i) = 0$, which leads to $P(T_y(y_j) = y_i) = 0$. This condition implies that R is closed on Y.

Theorem 6 *Given the group state sequence* $\{q(t); t \geq 0\}$ *in cuckoo search, the optimal group state set* H *is closed on the group state space* Q.

Proof From Eq. (21), the probability

$$P(T_q(q_j) = q_i) = \prod_{k=1}^{n} P(T_y(y_{jk}) = y_{ik}), \tag{26}$$

for $\forall q_i \in H, \forall q_j \in H$ and $T_q(q_j) = q_i$. Since $\forall q_i \in H$ and $\forall q_j \notin H$, in order to satisfy $T_q(q_j) = q_i$, there exists at least one cuckoo whose position will transfer from the inside of R to the outside of R. That is, $\exists T_y(y_{jk}) = y_{ik}, y_{jk} \in R, y_{ik} \notin R, 1 \leq k \leq n$. From the previous theorem, we know that R is closed on Y, which means that $P(T_y(y_{jk}) = y_{ik}) = 0$. Therefore,

$$P(T_q(q_j) = q_i) = \prod_{k=1}^{n} P(T_y(y_{jk}) = y_{ik}) = 0.$$

From the definition of a closed set, we can conclude that the optimal set H is also closed on Q.

Theorem 7 *In the group state space Q for cuckoos/nests, there does not exist a non-empty closed set B so that $B \cap H = \emptyset$.*

Proof Reductio ad absurdum. Assuming that there exists a close set B so that $B \cap H = \emptyset$ and that $f(g_j) > f(g_b)$ for $q_i = (g_b, g_b, ..., g_b) \in H$ and $\forall q_j = (y_{j1}, y_{j2}, ..., y_{jn}) \in B$, then Eq. (21) implies that

$$P(T_q(q_j) = q_i) = \prod_{k=1}^{n} P(T_y(y_{jk}) = y_{ik}). \tag{27}$$

For each $P(T_y(y_j) = y_i)$, it holds that $P(T_y(y_j) = y_i) = P(x_j \to x'_j)P(g_j \to g'_j)P(x'_j \to x_i)P(g'_j \to g_i)$. Since $P(g'_j \to g_i) = 1, P(g_j \to g'_j), P(x_j \to x'_j)P(x'_j \to x_i) > 0$, then $P(T_y(y_j) = y_i) \neq 0$, implying that B is not closed, which contradicts with the assumption. Therefore, there exists no non-empty closed set outside H in Q.

Using the above definitions and results, it is straightforward to arrive another theorem:

Theorem 8 *Assuming that a Markov chain has a non-empty set C and there does not exist a non-empty closed set D so that $C \cap D = \emptyset$, then*

$$\lim_{n \to \infty} P(x_n = j) = \pi_j,$$

only if $j \in C$, and $\lim_{n \to \infty} P(x_n = j) = 0$ *only if $j \notin C$.*

Now using the above three theorems, it is straightforward to show

Theorem 9 *When the number of iteration approaches infinity, the group state sequence will converge to the optimal state/solution set H.*

This is the foundation for proving the global convergence theorem, which states

Theorem 10 *The cuckoo search with the Markov chain model outlined earlier has guaranteed global convergence.*

Proof Since the iteration process in cuckoo search always keeps/updates the current global best solution for the whole population, which ensures that it satisfies the first convergence condition. In addition, the previous theorem means that the group state sequence will converge towards the optimal set after a sufficiently large number of iterations or infinity. Thus, the probability of not finding the globally optimal solution is asymptotically 0, which satisfies the second convergence condition. Consequently, from Theorem 1, we can conclude that cuckoo search has guaranteed global convergence towards its global optimality.

5 Validation by Numerical Experiments

All new algorithms should be validated using various benchmarks to test their basic performance, rate of convergence and other properties. However, since the cuckoo search has been tested in the literature with a diverse range of benchmarks and design case studies, the numerical experiments we have done here are mainly to see if the global convergence can be reached easily and the rate of convergence. For this purpose, we have selected five benchmark functions with different modalities and objective landscapes:

The first function is the Ackley function [1]

$$f(\mathbf{x}) = -20 \exp\Big[-\frac{1}{5}\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\Big] - \exp\Big[\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\Big] + 20 + e, \tag{28}$$

which has a global minimum $f_* = 0$ at $(0, 0, ..., 0)$. This function is highly nonlinear and multimodal.

De Jong's functions is unimodal and convex, which can be written as

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2, \quad -5.12 \le x_i \le 5.12, \tag{29}$$

whose global minimum is obviously $f_* = 0$ at $(0, 0, ..., 0)$. It is also commonly referred to as the sphere function.

Rosenbrock's function

$$f(\mathbf{x}) = \sum_{i=1}^{d-1}\Big[(x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2\Big], \tag{30}$$

has a narrow valley where lies its global minimum $f_* = 0$ at $\mathbf{x}_* = (1, 1, ..., 1)$ in the domain $-5 \le x_i \le 5$ where $i = 1, 2, ..., d$.

Xin-She Yang's forest-like function

$$f(\mathbf{x}) = \Big(\sum_{i=1}^{d}|x_i|\Big)\exp\Big[-\sum_{i=1}^{d}\sin(x_i^2)\Big], \quad -2\pi \le x_i \le 2\pi, \tag{31}$$

has a global minimum $f_* = 0$ at $(0, 0, ..., 0)$. This function is highly nonlinear and multimodal, and its first derivatives do not exist at the optimal point due to the modulus $|.|$ factor.

Zakharov's function

$$f(\mathbf{x}) = \sum_{i=1}^{d} x_i^2 + \Big(\sum_{i=1}^{d}\frac{ix_i}{2}\Big)^2 + \Big(\sum_{i=1}^{d}\frac{ix_i}{2}\Big)^4, \tag{32}$$

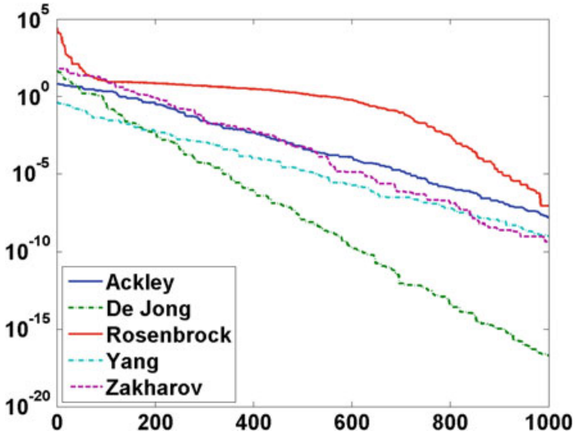

Fig. 1 Convergence of 5 test functions using cuckoo search

is nonlinear and has its global minimum $f(\mathbf{x}_*) = 0$ at $\mathbf{x}_* = (0, 0, ..., 0)$ in the domain $-5 \leq x_i \leq 5$.

All these functions have the global minimum $f_{\min} = 0$, and such simplicity allows to test the accuracy of an algorithm with various dimensions. For this reason, we set $d = 8$ for all these five functions.

For the implementation of cuckoo search algorithm, we have used $n = 25$, $\lambda = 1.5$, $p_a = 0.25$ and a fixed number of iterations $t = 1000$. The convergence graphs for all these functions are summarized and shown in Fig. 1 where the vertical axis is plotted using the logarithm scale. From the figure, it is clearly seen that the cuckoo search can converge quickly and the best objective values decrease in an almost exponential manner, except for Rosenbrock's function which has a narrow valley. However, as the search has gone through some part of the valley during iterations, its objective values once again decrease almost exponentially with a higher slope.

6 Conclusions

Cuckoo search is an efficient optimization algorithm with a wide range of applications. We have used the Markov chain theory and proved the global convergence of the simplified version of cuckoo search. Then, we have used a few benchmark functions with diverse properties to show that CS can indeed converge very quickly. In fact, cuckoo search has been used in many applications and the rate of convergence is usually very good in practice.

The current results are mainly for a simplified variant, derived from the standard cuckoo search. It can be expected that this methodology can be used to prove both standard cuckoo search algorithm and its variants. Therefore, it will be useful if further research can focus on the extension of the proposed methodology to analyze the convergence of other variants of the cuckoo search algorithm and other metaheuristic algorithms.

In addition, though we can show the cuckoo search will converge in the probabilistic sense, there is no information about how quickly it can convergence. Therefore, further research can also try to figure out the rate of convergence and its link to the algorithmic structure, parameter setting and even the modal shapes of the objective landscapes. After all, the rate of convergence is crucially important from the implementation point of view.

Furthermore, as the setting of parameters in an algorithm can affect the performance of the algorithm significantly, and consequently affect the rate of convergence. It would be useful to find the relationship between parameter values and the convergence rate, and then to control the rate of convergence by fine tuning the algorithm-dependent parameters.

Acknowledgements The authors would like to thank the financial support by Shaanxi Provincial Education Grant (12JK0744) and Shaanxi Provincial Soft Science Foundation (2012KRM58).

References

1. Ackley, D.H.: A Connectionist Machine For Genetic Hillclimbing. Kluwer Academic Publishers (1987)
2. Bhargava, V., Fateen, S.E.K., Bonilla-Petriciolet, A.: Cuckoo search: a new nature-inspired optimization method for phase equilibrium calculations. Fluid Phase. Equilib. **337**, 191–200 (2013)
3. Chandrasekaran, K., Simon, S.P.: Multi-objective scheduling problem: hybrid appraoch using fuzzy assisted cuckoo search algorithm. Swarm Evol. Comput. **5**(1), 1–16 (2012)
4. Clerc, M., Kennedy, J.: The particle swarm–explosion stability and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
5. Dhivya, M., Sundarambal, M., Anand, L.N.: Energy efficient computation of data fusion in wireless sensor networks using cuckoo based particle approach (CBPA). Int. J. Commun. Netw. Syst. Sci. **4**(4), 249–255 (2011)
6. Dhivya, M., Sundarambal, M.: Cuckoo search for data gathering in wireless sensor networks. Int. J. Mob. Commun. **9**(4), 642–656 (2011)
7. Durgun, I., Yildiz, A.R.: Structural design optimization of vehicle components using cuckoo search algorithm. Mater. Test. **3**(3), 185–188 (2012)
8. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng. Comput. **29**(1), 17–35 (2013)
9. He, X.S., Yang, X.S., Karamanoglu, M., Zhao, Y.X.: Global convergence analysis of the flower pollination algorithm: a discrete-time Markov chain approach. Proced. Comput. Sci. **108**(1), 1354–1363 (2017)
10. Jiang, M., Luo, Y.P., Yang, S.Y.: Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Inf. Process. Lett. **102**(1), 8–16 (2007)
11. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. Piscataway, NJ, pp. 1942–1948 (1995)
12. Koziel, S., Yang, X.S.: Computational Optimization Methods And Algorithms. Springer, Berlin (2011)
13. Moravej, Z., Akhlaghi, A.: A novel approach based on cuckoo search for DG allocation in distribution network. Electr. Power Energy Syst. **44**(1), 672–679 (2013)
14. Noghrehabadi, A., Ghalambaz, M., Vosough, A.: A hybrid power series–cuckoo search optimization algorithm to electrostatic deflection of micro fixed-fixed actuators. Int. J. Multidiscip. Sci. Eng. **2**(4), 22–26 (2011)

15. Pavlyukevich, I.: Lévy flights, non-local search and simulated annealing. J. Comput. Phys. **226**(2), 1830–1844 (2007)
16. Ren, Z.H., Wang, J., Gao, Y.L.: The global convergence analysis of particle swarm optimization algorithm based on Markov chain. Control Theory Appl. **28**(4), 462–466 (2011). (in Chinese)
17. Taweewat, P., Wutiwiwatchai, C.: Musical pitch estimation using a supervised single hidden layer feed-forward neural network. Expert Syst. Appl. **40**(2), 575–589 (2013)
18. Valian, E., Mohanna, S., Tavakoli, S.: Improved cuckoo search algorithm for feedforward neural network training. Int. J. Artif. Intell. Appl. **2**(3), 36–43 (2011)
19. Valian, E., Tavakoli, S., Mohanna, S., Haghi, A.: Improved cuckoo search for reliability optimization problems. Comput. Ind. Eng. **64**(1), 459–468 (2013)
20. Vazquez, R.A.: Training spiking neural models using cuckoo search algorithm. In: IEEE Congress on Eovlutionary Computation (CEC'11), pp. 679–686 (2011)
21. Walton, S., Hassan, O., Morgan, K., Brown, M.R.: Modified cuckoo search: a new gradient free optimization algorithm. Chaos Solitons Fractals **44**(9), 710–718 (2011)
22. Wang, F., He, X.S., Wang, Y., Yang, S.M.: Markov model and convergence analysis of cuckoo search algorithm. Comput. Eng. **38**(11), 180–185 (2012) (in Chinese)
23. Yang, X.S., Deb, S.: Cuckoo search via Lévy flights. In: Proceeding of World Congress on Nature and Biologically Inspired Computing (NaBic), pp. 210–214. IEEE Publications, Coimbatore, India, USA (2009)
24. Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. Int. J. Math. Model. Numer. Optim. **1**(4), 330–343 (2010)
25. Yang, X.S., Deb, S.: Cuckoo search: recent advances and applications. Neural Comput. Appl. **24**(1), 169–174 (2014)
26. Yang, X.S.: Engineering Optimization: An Introduction With Metaheuristic Applications. Wiley (2010)
27. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature-Inspired Cooperative Strategies for Optimization (NICSO), vol. 284, pp. 65–74. SCI, Springer (2010)
28. Yang, X.S.: Bat algorithm for multi-objective optimisation. Int. J. Bio-Inspir. Comput. **3**(5), 267–274 (2011)
29. Yang, X.S.: CUCKOO search and firefly algorithm: theory and applications. In: Studies in Computational Intelligence, vol. 516 Springer (2014)
30. Yang, X.S.: Nature-Inspired Optimization Algorithms. Elsevier Insight, London (2014)
31. Yang, X.S., Deb, S.: Multiobjective cuckoo search for design optimization. Comput. Oper. Res. **40**(6), 1616–1624 (2013)
32. Yildiz, A.R.: Cuckoo search algorithm for the selection of optimal machine parameters in milling operations. Int. J. Adv. Manuf. Technol. **64**(1), 55–61 (2013)
33. Zheng, H.Q., Zhou, Y.: A novel cuckoo search optimization algorithm based on Gauss distribution. J. Comput. Inf. Syst. **8**(10), 4193–4200 (2012)