Convergence Analysis of Brain Storm Optimization Algorithm

Ziwei Zhou
School of Automation Science and Electrical Engineering
Beihang University, Beijing 100191, China
E-mail: 574691389@qq.com

Haibin Duan, *Senior Member, IEEE*,
School of Automation Science and Electrical Engineering
Beihang University, Beijing 100191, China
E-mail: hbduan@buaa.edu.cn

Yuhui Shi, *Fellow, IEEE*
Xi'an Jiaotong-Liverpool University
Suzhou 215123, China
E-mail: yuhui.shi@xjtlu.edu.cn

Abstract—**Brain storm optimization (BSO) algorithm is a new kind of swarm intelligence algorithm, which is inspired by collective behavior of human beings. In this paper, a Markov model for brain storm optimization algorithm is derived. The model gives the theoretical probability of the occurrence of each possible population as the number of generation count goes to infinity. Using the Markov model, the convergence of the brain storm optimization is analyzed.**

Keywords—brain storm optimization; markov chain; style; convergence analysis

I. INTRODUCTION

Swarm Intelligence (SI) optimization algorithms have been widely used in solving optimization problems in recent years [1-2], including Particle Swarm Optimization (PSO) [3-4], Artificial Bee Colony (ABC) [5], and Pigeon-Inspired Optimization (PIO) [6-7] etc. Brain Storm Optimization (BSO) is a novel swarm intelligence optimization algorithm, which was first proposed in [8-9].

Since the introduction of the BSO, some novel developments based on BSO have been achieved [10-17], but the analysis of the convergence of BSO is lacking. Different methods for proving the convergence of Evolution Algorithm (EA) algorithms [18] have been proposed, such as the Markov chain models method for simple genetic algorithms (GAs) [19-20] and Ant Colony Optimization (ACO) [21], and the spectral radius method for PSO [22]. In this paper, Markov models will be derived for BSOs with grouping, replacing, creating, crossing, selecting, and chaotic operators. With the Markov models, a novel approach to the convergence proof that applies directly to the BSO will be presented.

A Markov chain is a random process that has a discrete set of possible state values $s_i (i = 1,...,T)$ [23-24]. The probability that the system transitions from state s_i to s_j is given by the probability p_{ij}, which is called a transition probability. The $T \times T$ matrix $P = [p_{ij}]$ is called the transition matrix. A Markov chain is called regular if it is possible to go from any state to any other state (not necessarily in one step). The fundamental limit theorem for regular Markov chains states that if P is regular, then

$$\lim_{n\to\infty} P^n = P_{ss} \tag{1}$$

where each row p_{ss} of P_{ss} is the same. The ith element of p_{ss} denotes the probability that the Markov chain is in state s_i after an infinite number of transitions. p_{ss} is independent of the initial state.

The rest part of the paper is organized as follows. Section II describes the principles of BSO. Then in Section III, the Markov models of BSO are proposed. Subsequently, we analyze the convergences of the BSO with Markov models in Section IV, followed by our concluding remarks in Section V.

II. BRAIN STORM OPTIMIZATION ALGORITHM (BSO)

As in Refs [8] and [9], the process of BSO can be described as follows.

First, the parameters are initialized. A set of Nc ideas within the searching space are randomly generated. We denote a BSO population as $X = \{x_i = [x_{i1}, x_{i2}, ..., x_{iD}] \mid x_i \in A, 1 \le i \le Nc\}$, where x_i represents the ith idea of the BSO population, $A = R^D$ denotes the feasible solution space of the idea, Nc is the population size, and D is the problem dimension [8-9]. Let $X(0)$ be the initial population and $X(n)$ be the nth iteration population. Each dimension signifies one design variable [25]. Then each idea is evaluated and its fitness value $f(x_i)$ is obtained. The process of iteration begins afterwards. During the evolutionary process, BSO generally uses grouping, replacing, creating, crossing, and selecting operators to create new ideas based on the current ideas, so as to improve the ideas generation over generation to approach the optimal problem solution [26]. Here an idea is similar to an individual in other SI algorithms, and represents a potential solution to

This work was partially supported by National Natural Science Foundation of China (NSFC) under grant # 61333004 and 61273367, and Aeronautical Foundation of China under grant #2015ZA51013.

the problem being solved. In the grouping operator, BSO uses a *k*-mean clustering method to group the *N* ideas into *M* clusters.

In the creating operator [26], BSO creates Nc new ideas one by one based on the current ideas. To create a new idea $y_i = [y_{i1}, y_{i2}, ..., y_{iD}]$, $(1 \le i \le Nc)$, BSO first determines whether to create the new idea y_i based on one selected cluster or based on two selected clusters. After the cluster(s) have been randomly selected, BSO then determines whether to create the new idea y_i base on the cluster center(s) or randomly selected idea(s) inside the cluster(s). No matter to use the cluster center or to use randomly selected idea inside the cluster, we can regard the selected idea as x , then the new idea y_i is created as:

$$y_{i,d} = x_d + \xi_d \times N(\mu, \sigma)_d \tag{2}$$

$$x_d = \begin{cases} x_{i,d} & one \quad cluster \\ \omega_1 x_{i1,d} + \omega_2 x_{i2,d} & two \quad clusters \end{cases} \tag{3}$$

where d is the dimension index, $N(\mu, \sigma)$ is the Gaussian random value with mean μ and variance σ , and ξ is a coefficient that weights the contribution of the Gaussian random value, which is calculated as:

$$\xi = \log sig(\frac{0.5 \times Iter_{\max} - i}{K}) \times random(0,1) \tag{4}$$

where $\log sig()$ is a logarithmic sigmoid transfer function whose values are within the range $(0,1)$, $Iter_{\max}$ and i denote the maximum number of iteration and current number of iteration respectively. K is for adjusting $\log sig()$ function's slope, $random(0,1)$ is a random value between 0 and 1.

After the new idea y_i has been created, a crossover between the new one and the old one is conducted. The two ideas $x_i^{'}$ and $y_i^{'}$ generated by crossover, together with the old one and the created one, are evaluated and the old one is replaced by the best one of the four [9]. This new idea creating process repeats for *Nc* times to complete a generation. If the termination criteria met, BSO terminates and reports the best idea of the population as the solution [8]. Otherwise, BSO goes to the next generation to repeat the grouping, replacing, creating, crossing, and selecting processes. The implementation procedure of our proposed BSO approach can be described as follows:

Step 1: Initialize the parameters of BSO, such as the number of ideas Nc , the maximum number of iterations $Iter_{\max}$, solution space dimension *D*, and the number of clusters *M*. The *N* ideas are randomly generated, and evaluated;

Step 2: With the *k*-means clustering algorithm, cluster *Nc* ideas into *M* clusters; then rank ideas in each cluster and record the best idea as cluster center in each cluster;

Step 3: Compare with the given probability p_{5a} , if a random value between 0 and 1 is smaller, then randomly select a cluster center to be replaced by an randomly generated idea; otherwise, do nothing;

Step 4: With the probability p_{6b} , select one cluster and go to Step 5; otherwise, select two clusters and go to Step 6;

Step 5: With the probability p_{6biii} , select cluster center and go to Step 7; otherwise, randomly select an idea from this cluster and go to Step 8;

Step 6: With the probability p_{6c} , select two cluster centers and go to Step 6; otherwise, select two ideas from each selected cluster go to Step 7; otherwise, select other ideas and go to Step 7;

Step 7: According to (2), (3) update the cluster center(s), then go to Step 9;

Step 8: According to (2), (3), update the idea(s)

Step 9: The newly generated idea crossovers with the existing idea with the same idea index to generate two more ideas. Compare the four ideas, the best one is kept and recorded as the new individual [9];

Step 10: If *Nc* ideas have been updated, go to Step 11. Otherwise go back to Step 4;

Step 11: Evaluate the Nc ideas, and then update the cluster center;

Step 12: If the current number of iterations is less than $Iter_{\max}$, go back to Step 2. Otherwise the algorithm is terminated and the best idea is outputted as the solution to the optimization problem.

III. MARKOV MODELS FOR BSO

A. Markov Chain

A Markov chain, named after Andrey Markov, is a mathematical system that undergoes transitions from one state to another, among a finite or countable number of possible states [27]. It is a random process usually characterized as memoryless: the next state depends only on the current state and not on the sequence of events that preceded it. This specific kind of "memory lessness" is called the Markov property. Markov chains have many applications as statistical models of real-world processes.

B. Finite Element Models of Continuous Optimizers

In [28-29], a method was proposed to build discrete Markov chain models of continuous stochastic optimizers which can approximate them on arbitrary continuous problems to any precision in principle. The idea is using a finite element method grid to discretize the objective function which produces corresponding distinct states in the search algorithm.

First, partition a continuous *N*-dimensional search space Ω into a finite number *n* of compact non-overlapping sub-domains Ω_i [28-29]. Next take the fitness of each sub-domain

Ω_i as the fitness at its center (i.e. $f_i = f(x_{c_i})$ where x_{c_i} is the centroid of cell *i*) and adopt the convention of ordering sub-domains by fitness so that $f_i \le f_j$ for $i < j$. Then regard Ω as an *N*-dimensional cube, which has partitioned using a regular grid of hypercubic cells:

$$\Omega_i = [x_{c_{i1}} - r, x_{c_{i1}} + r] \times [x_{c_{i2}} - r, x_{c_{i2}} + r] \times \ldots \times [x_{c_{iN}} - r, x_{c_{iN}} + r]$$

where *r* is the cell "radius" and $x_{c_{ij}}$ is the *j*-th component of a lattice point x_{c_i}. So the algorithm can be discretized by only allowing it to be in discrete states, effectively disallowing all points in the search space except the centroids, x_{c_i} of the domains.

With the technique above, we model the BSO. Naturally, at any given time the best idea will be located in some sub-domain Ω_i. In a discretized BSO the best idea x_d can only take one of a discrete set of values, namely $x_d = x_{c_k}$ for some *j* in $\{0, \ldots, n-1\}$. So, the discretized algorithm can only be in one of a finite set of states. However, we don't need to represent explicitly the best idea, since the information is implicit in the fitness values f_i of each centroid. So, if *P* is the population size, there are n^P such states and we can represent states of the whole algorithm as *P* dimensional vectors with integer elements, i.e. $s = (s_1, s_2, \ldots, s_P)$.

C. Markov Models for BSO

1) Grouping operator

In the grouping operator, BSO uses a *k*-mean clustering method to group the *N* ideas into *M* clusters. After *grouping* process, the population can be depicted as:

$$\begin{aligned} &Population \\ &= \{C_1, C_2, \ldots, C_M\} \\ &= \{c_{1,1}, c_{2,1}, \ldots, c_{v_1,1}, c_{1,2}, \ldots, c_{v_2,2}, \ldots, c_{1,M}, \ldots, c_{v_M,M}\} \end{aligned} \tag{5}$$

where $C_k (k = 1, 2, \ldots, M)$ denotes the *k*th cluster, and $c_{i,k}$ denote the *i*th individual of the *k*th cluster, further, we suppose $c_{1,k}$ represents the cluster center, and $v_k (k = 1, 2, \ldots, M)$ represents the numbers of the individuals in the *k*th cluster, which satisfied $\sum_{k=1}^{M} v_k = Nc$.

2) Replacing operator

Replacing operator of BSO is given as in Step 3. After many iterations, all solutions may be clustered into a small region [8]. A probability value p_{5a} is utilized to control the probability of replacing a cluster center by a randomly generated solution. This could avoid the premature convergence, and help ideas "jump out" of the local optima. The replacing operator can be regarded as random mapping, i.e., $T_r : \Omega \to \Omega$. The probability distribution can be written as

$$P_{rk}(T_r(C_k) = C_k') = \begin{cases} \frac{1}{M} & rand < p_{5a} \\ 0 & rand \ge p_{5a} \end{cases} \tag{6}$$

where $P_{rk}(k = 1, 2, \ldots, M)$ denotes the probability of the *k*th cluster center which is replaced. C_k is the cluster center before the replacement, and C_k' is the cluster center after replacement, *rand* represent a random value between 0 and 1.

3) Creating operator

In the creating operator, BSO creates Nc new ideas one by one based on the current ideas. BSO first determines whether to create the new idea y_i based on one selected cluster or based on two selected clusters. After the cluster(s) have been selected, BSO then determines whether to create the new idea y_i based on the cluster center(s) or randomly selected idea(s) inside the cluster(s). No matter to use the cluster center or to use randomly selected idea inside the cluster, we can regard the selected idea as x, then the new idea y_i is created according to (2). It also can be regarded as random mapping, i.e. $T_m : \Omega^{Nc} \to \Omega$. The probability distribution can be written as

$$P(T_m(X) = y_i) = \begin{cases} \begin{cases} \frac{1}{M} & rand_2 < p_{6biii} \\ \frac{1}{v_k - 1} & rand_2 \ge p_{6biii} \end{cases} & rand_1 < p_{6b} \\ \begin{cases} \frac{1}{M(M-1)} & rand_3 < p_{6c} \\ \frac{1}{v_k - 1}\frac{1}{v_j - 1} & rand_3 \ge p_{6c} \end{cases} & rand_1 \ge p_{6b} \end{cases} \tag{7}$$

where *M* is the number of the cluster centers, v_j, v_k denote the numbers of individuals in the *j*th and the *k*th clusters $(1 \le j \ne k \le M)$, respectively, $rand_1$, $rand_2$, $rand_3$ represent the random value between 0 and 1.

4) Crossing operator

Crossing operator of BSO is given as the first half of Step 9. It's a one-point crossover. The newly generated idea crossovers with the existing idea with the same idea index to generate two more ideas (offspring), therefore it also can be seen as random mapping, i.e., $T_c : \Omega^2 \to \Omega^2$. The probability distribution can be written as

$$P(T_c(x_i, y_i) = (x_i', y_i')) = \frac{1}{D} \tag{8}$$

where *D* is the problem dimension.

5) Selecting operator

Selecting operator of BSO is given as the last half of Step 9. The four ideas created in the previous section are compared, the best one is kept and recorded as the new idea. This process is written as $T_s : \Omega^4 \to \Omega$. The probability distribution can be written as

$$P(T_s(x_i, y_i, x_i^{'}, y_i^{'}) = x_i) = \begin{cases} 1 & f(x_i) = \min(f(x_i), f(y_i), f(x_i^{'}), f(y_i^{'})) \\ 0 & f(x_i) \neq \min(f(x_i), f(y_i), f(x_i^{'}), f(y_i^{'})) \end{cases} \quad (9)$$

Therefore, BSO can be formulated with the following equation

$$\{x_i(n+1) = T_r \circ T_m \circ T_c \circ T_s(X(n)), i = 1, 2, ..., Nc\} \quad (10)$$

From the previous description, the next generation of BSO only depends on the ideas just generated, and independent of any other ideas. In the other word, the distribution of $X(n+1)$ is determined if and only if $X(n)$ is identified. The previous description shows that the sequence of BSO $\{X(n); n \in N^+\}$ is a Markov chain. In this article, the N^+ represents the positive set and the indices $n \in N^+$ is interpreted as points of time. Further, T_r, T_m, T_c, T_s is independence of *n*. The sequence of BSO $\{X(n); n \in N^+\}$ is a homogeneous Markov chain.

IV. Convergence Analysis of BSO

BSO algorithm belongs to the category of random algorithm, which means the convergence of BSO can be proved with the random algorithm convergence criteria [30-32].

A. Convergence Criterion

For an optimization problem $<\Omega, f>$, there is a stochastic optimization algorithm denoted as O, and the *k*th iteration solution x_k, thus the next iteration solution is $x_{k+1} = O(x_k, \varsigma)$, where Ω is feasible solution space, f is the fitness function, and ς denoted as $\varsigma = \{x_1, ..., x_{k-1}\}$ is the set of solutions that have been searched in the previous iterations.

The searching infimum is defined in Lebesgue measure space as:

$$\psi = \inf(t : v[x \in \Omega \mid f(x) < t] > 0),$$

where $v[X]$ is the Lebesgue measure in set X. Then the optima area can be defined as

$$R_{\varepsilon,L} = \begin{cases} \{x \in \Omega \mid f(x) < \psi + \varepsilon\}, \psi & \textit{is finitude} \\ \{x \in \Omega \mid f(x) < -L\}, & \psi = -\infty \end{cases},$$

where $\varepsilon > 0$, L is a sufficiently large positive number. If the algorithm seeks out a point in $R_{\varepsilon,L}$, the algorithm is thought to find out the global optimal or the approximate global optimal point.

Assumption 1[30-32]**:** $f(O(x_k, \varsigma)) \le f(x)$ and if $\varsigma \in A$, there is $f(O(x_k, \varsigma)) \le f(\varsigma)$.

Assumption2 [30-32]**:** For any Borel subset B of Ω, s.t. $v[B] > 0$, there is $\prod_{k=0}^{\infty}(1 - u_k[B]) = 0$, where $u_k[B]$ is the probability measure of B at the *k*th iteration searching solution of the algorithm O.

Theorem 1 (The necessary and sufficient conditions of global convergence) [30-32]**:** Let f be measurable, and measurable space A be the measurable subset of R^n. If the algorithm O satisfies the assumption 1 and 2, and $\{x_k\}_{k=0}^{\infty}$ is the solution sequence generated by O, then we have

$$\lim_{k \to \infty} P(x_k \in R_{\varepsilon,L}) = 1,$$

where $P(x_k \in R_{\varepsilon,L})$ is the probability measure at the *k*th iteration solution x_k in $R_{\varepsilon,L}$.

Definition 1: (the BSO population state and the BSO population state space) The BSO population state is all the states of the ideas in BSO population, whilst the BSO population state space is the set of all the BSO population states. From the previous description, we have denoted a BSO population as $X = \{x_i = [x_{i1}, x_{i2}, ..., x_{iD}] \mid x_i \in \Omega, 1 \le i \le Nc\}$, therefore $X = \{x_i \mid x_i \in \Omega, 1 \le i \le Nc\}$ can represent a BSO population state. Then the BSO population state space S can be denoted as $S = \{X = \{x_i \mid x_i \in \Omega, 1 \le i \le Nc\}\}$.

Definition 2: (global optimal state set G) Let the optimal solution of the optimization problem $<\Omega, f>$ be g^*. We denote the global optimal state set G as

$$G = \{X = \{x\} \mid f(x) = f(g^*), X \in S\}$$

It is meaningless to optimize when $G = S$, because in this case, each solution in the feasible solution space is not only the feasible solution but also the optimal solution. Therefore, the following discussion is conducted in the case that $G \subset S$.

B. Convergence Analysis of the Basic BSO

Given by the previous sections, the basic BSO can be expressed as

$$\{x_i(n+1) = T_r \circ T_m \circ T_c \circ T_s(X(n)), i = 1, 2, ..., Nc\}.$$

Lemma1: The basic BSO satisfies the assumption 1. In the other word, in BSO, the evolutionary direction of the population is monotone decreasing, i.e., $f(X(n+1)) \le f(X(n))$.

Proof: From (13), the selecting operator is a greedy selection model, if and only if the fitness of the idea is better than any other ideas, it can be accepted. So the optimal fitness value is monotone decreasing, which means it satisfies assumption 1

Theorem 2: The sequence of the population $\{X(n); n \in N^+\}$ is a finite homogeneous irreducible aperiodic

Markov chain.

Proof: From B Section of III, we can ensure the BSO sequence is finite, i.e. the sequence of the population $\{X(n); n\in N^+\}$ is a finite Markov chain.

Under the definition of the state space *S*, we get

$$\forall X(n)\in S,\quad \sum_{X(n+1)\in S} P\{T_r\circ T_P\circ T_c\circ T_s(X(n))=X(n+1)\}=1$$, i.e., the sequence of the population $\{X(n); n\in N^+\}$ is an irreducible Markov chain.

The transition probability of the BSO population sequence can be written as follow

$$\begin{aligned}&P(T_r\circ T_P\circ T_c\circ T_s(X(n))_i = x_i(n+1))\\&=\sum_S P(T_r(c_k)=c_k^{'})P(T_P(X)=y_i)P(T_c(x_i,y_i)=(x_i^{'},y_i^{'}))\\&\quad P(T_s(x_i,y_i,x_i^{'},y_i^{'})=x_i)=x_i)\end{aligned}$$

where $\sum_S = \sum_{(x_i,y_i,x_i^{'},y_i^{'})\in S^4}\sum_{(x_i,y_i)\in S^2}\sum_{X\in S^{Nc}}\sum_{c_k\in S^M}$

While $\forall X\in S^{Nc},\exists\{x_i,y_i,x_i^{'},y_i^{'}\}\in S^{Nc},\{x_i,y_i\}\in S^{Nc},c_k\in S^{Nc}$, there is

$$P(T_r(c_k)=c_k^{'})>0$$

$$P(T_r(c_k)=c_k^{'})>0$$

$$P(T_P(X)=y_i)>0$$

$$P(T_c(x_i,y_i)=(x_i^{'},y_i^{'}))>0$$

$$P(T_s(x_i,y_i,x_i^{'},y_i^{'})=x_i)>0$$

then $$P(T(X(n))_i = x_i(n+1))>0 \tag{11}$$

which means

$$\begin{aligned}&P(T(X(n))\\&=X(n+1))\\&=\prod_{i=1}^{Nc}P(T(X(n))_i=x_i(n+1))>0\end{aligned},$$

where $T=T_r\circ T_P\circ T_c\circ T_s$ therefore the sequence of the population $\{X(n); n\in N^+\}$ is an aperiodic Markov chain.

In conclusion, the sequence of the population $\{X(n); n\in N^+\}$ is a finite homogeneous irreducible aperiodic Markov chain. Then the transition probability of the idea population can be written as

$$\begin{aligned}&P(X,Y)=P(X(n+1)=Y\mid X(n)=X)=P(T(X)=Y)\\&=\begin{cases}\prod_{i=1}^{Nc}P(T(X(n))_i=x_i(n+1)), & \exists i_0\in[1,Nc],\ s.t. f(Y_{i_0})=\min F(X)\\ 0 & others\end{cases}\end{aligned} \tag{12}$$

Theorem 3: In the basic BSO, for the population state sequence $\{X(n); n\in N^+\}$, the global optimal state set G is a closed set on the state space S.

Proof: Suppose $\forall X_i\in G,\forall X_j\notin G$, for any transfer step $l(l\ge 1)$, by Chapman-Kolmogorov equation [23],

$$\begin{aligned}P_{X_i,X_j}^l &= \sum_{x_{r_1}\in A}\cdots\sum_{x_{r_{l-1}}\in A}P(T(X_i)=X_{r_1})\times\\&\quad P(T(X_{r_1})=X_{r_2})\cdots P(T(X_{r_{l-1}})=X_j)\end{aligned} \tag{13}$$

where P_{X_i,X_j}^l is the probability that the state X_i transfers to the state X_j after l steps. In each product expression of the expansion of (13), there is an item $P(T(X_{c-1})=X_c), 1\le c\le l$ satisfying $X_{c-1}\in G, X_c\notin G$.

By (12), the transition probability of the idea population can be expressed as

$$P(T(X_{r_{c-1}})=X_{r_c})=\prod_{k=1}^{Nc}P(T(X_{r_{c-1}})_k = x_{r_c k}).$$

With $X_{c-1}\in G, X_c\notin G$, there is

$$f(x_c)>f(x_{c-1})=f(g^*)=\inf(f(a)), a\in A,$$

thus at least one item $P(T(X_{c-1})=X_c)=0$, in this case $P_{X_i,X_j}^l=0$. Therefore, the global optimal state set G is a closed set on the state space S.

Theorem 4: There is no non-empty closed set $G_{Ba}^{'}$ in the BSO population state space S which satisfies $G^{'}\cap G=\phi$.

Proof: Proof by contradiction. Assume that there is a non-empty closed set $G^{'}$ in S which satisfies $G^{'}\cap G=\phi$. Let $X_i=(g^*,g^*,\cdots,g^*)\in G,\forall X_j(x_{j1},x_{j2},\cdots,x_{jd})\in G^{'}$, then $f(x_{jc})>f(g^*)$. Thus after infinite iterations, it can satisfy (12) $P(T(x_i)=x_j)>0$. Therefore, if the step l is large enough, there must be one product expression of the expansion of P_{X_i,X_j}^l that satisfies (11), i.e. $P(T(X_{r_{c+i}})=X_{r_{c+i+1}})>0$. Under (13), $P_{X_i,X_j}^l>0$ that means $G^{'}$ is not non-empty closed set, a contradiction. Therefore, there is one and only the closed set G in the state space S.

Theorem 5 [23]: Given a non-empty closed set E, if there is non-empty closed set V satisfies $E\cap V=\phi$, when $j\in E$, $\lim_{n\to\infty}P(X_n=j)=\pi_j$, and $j\notin E$, $\lim_{n\to\infty}P(X_n=j)=0$.

Theorem 6: When the internal iteration of BSO tends to infinity, the state sequence of BSO must get into the global optimal state set G.

Proof: Theorem 6 can be easily proved out under Theorems 5-7.

Lemma 2: The basic BSO satisfies the assumption 2.

Proof: Under the Theorem 8, the probability that the algorithm can't search the global optimal solution of infinite times in a row is 0. Then we get $0 < u_k[B] < 1$, i.e. $\prod_{k=0}^{\infty}(1-u_k[B]) = 0$. The basic BSO satisfies the assumption 2.

Theorem 7: The basic BSO converges to the global optimal.

Proof: Under Lemmas 1, 2 and Theorem 1, the BSO is a global convergent algorithm.

V. Conclusion

In this paper, we developed a theoretical framework based on Markov chains for the brain storm optimization algorithm (operators of grouping, replacing, creating, crossing and selecting). Using the theory of finite element models of continuous optimizers, we have modeled the brain storm optimization. This allows the creation of discrete Markov chain models which approximate the behavior of a BSO exploring a continuous space. The models give the theoretical probability of the occurrence of each possible population as the number of generation count goes to infinity. The convergence of the BSO was proved by Markov models.

References

[1] J. Kennedy, R.C. Eberhart, and Y.H. Shi. Swarm Intelligence*,* San Mateo, CA: Morgan Kaufmann, 2001.

[2] J. Zhang, *et al.* "Evolutionary computation meets machine learning: A Survey," IEEE Comput. Intell. Mag., 2011, vol. 6, pp. 68-75.

[3] H.B. Duan, Q.N. Luo, Y.H. Shi, and G.J. Ma. "Hybrid particle swarm optimization and genetic algorithm for multi-UAV formation reconfiguration," IEEE Comput. Intell. Mag., 2013, vol. 8, pp. 16–27.

[4] H.B Duan, P. Li, and Y.X. Yu. "A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic gametheory," IEEE/CAA Journal of Automatica Sinica, 2015, vol. 2, pp. 11-18.

[5] H.B. Duan, and S.T. Li, "Artificial bee colony based direct collocation for reentry trajectory optimization of hypersonic vehicle," IEEE Trans. on Aero. and Elec. Sys., 2015, vol. 51, pp. 615-626.

[6] B. Zhang, H.B. Duan. "Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment," IEEE/ACM Trans. on Comput. Bi., 2015.

[7] H.B. Duan, H.X. Qiu, and Y.M. Fan. "Unmanned aerial vehicle close formation cooperative control based on predatory escaping pigeon-inspired optimization," Scientia Sinica Technologica, 2015, vol. 45, pp. 559–572.

[8] Y.H Shi. "Brain storm optimization algorithm," in Proceeding of the 2nd International Conference on Swarm Intelligence Chongqing, China, 2011, pp. 303-309.

[9] Y.H. Shi. "An optimization algorithm based on brainstorming process," Int. J of Swarm Intell. Research, 2011, vol. 2, pp. 35-62.

[10] H.B. Duan, and C. Li. "Quantum-behaved brain storm optimization approch to solving loney's solenoid problem," IEEE Trans. on Mag., 2015, vol. 51, Part:2: 7000307.

[11] H.X. Qiu, and H.B. Duan. "Receding horizon control for UAVs formation flight based on modified brain storm optimization," Nonlinear Dynamics, 2014, vol. 78, pp. 1973-1988.

[12] J.Q. Xue, Y.L. Wu, Y.H. Shi, and S. Cheng. "Brain storm optimization algorithm for multi-objective optimization problems," Advances in Swarm Intelligence, Springer Berlin Heidelberg, 2012, pp. 513-519.

[13] D. Zhou, Y. Shi, and S. Cheng. "Brain storm optimization algorithm with modified step-size and individual generation," Advances in Swarm Intelligence, Springer Berlin Heidelberg, 2012, pp. 243-252.

[14] Z.H. Zhan, et al. "Parameter investigation in brain storm optimization," 2013 IEEE Symposium on Swarm Intelligence, IEEE, 2013, pp. 103-110.

[15] Y.H. Shi. "Brain storm optimization algorithm in objective space," In Proceedings of the 2015 IEEE Congress on Evolutionary Computation, 2015, pp. 1227-1234.

[16] Z. Yang, and Y.H. Shi. "Brain Storm Optimization with Chaotic Operation," in Proceedings of the 2015 Seventh International Conference on Advanced Computational Intelligence, 2015, pp. 111-115.

[17] S. Cheng, et al. "Maintaining population diversity in brain storm optimization algorithm," in Proceedings of the 2014 IEEE Congress on Evolutionary Computation, 2014, pp. 3230-3237.

[18] H.B. Duan, D.B. Wang, and X.F. Yu. "Markov chains and martingale theory based convergence proof of ant colony algorithm and its simulation platform," The 6th World Congress on Intelligent Control and Automation, 2006, vol. 1, pp. 3057-3061.

[19] J. Suzuki A. "Markov chain analysis on simple genetic algorithms," IEEE Trans. on Sys., Man, and Cyber., 1995, vol. 25, pp. 655-659.

[20] J. Suzuki. "A further result on the markov chain model of genetic algorithms and its application to a simulated annealing-like strategy," IEEE Trans. on Sys., Man, and Cyber., 1998, vol. 28, pp. 95-102.

[21] J.H. Zhong, M.L. Shen, J. Zhang, H. Chung, Y.H. Shi, and Y. Li. "A differential evolution algorithm with dual populations for solving periodic railway timetable scheduling problem," IEEE Trans. on Evolut. Comput.. 2013, vol. 17, pp. 512-527.

[22] M. Clerc, and J. Kennedy. "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," IEEE Trans. on Evolut. Comput., 2002, vol. 6, pp. 58-73.

[23] W.X. Zhang, and Y. Leung. Mathematical foundation of genetic algorithms. Xi'an Jiaotong University Press, 2000.

[24] D. Simon, M. Ergezer, D.W. Du, and R. Rarick. "Markov models for biogeography-based optimization," IEEE Trans. on Sys., Man, and Cyber., 2011, vol. 41, pp. 299-306.

[25] H.B. Duan, S.T. Li, and Y.H. Shi. "Predator–prey brain storm optimization for DC brushless motor," IEEE Trans. on Mag., 2013, vol. 49, pp. 5336–5340.

[26] Z.H. Zhan, J. Zhang, Y.H. Shi, and H.L. Liu. "A modified brain storm optimization," in Proceeding of 2012 IEEE World Congress on Computational Intelligence, Brisbane, Australia: IEEE, 2012, pp. 1-8.

[27] M. Higashitani, A. Ishigame, and K. Yasuda. "Particle swarm optimization considering the concept of predator-prey behavior," in Proceeding of IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada: IEEE, 2006. pp. 434–437.

[28] R. Poli, and W.B. Langdon. "Markov chain models of bare-bones particle swarm optimizers," in Proceeding of Genetic and Evolutionary Computation Conference, London, England: IEEE, 2007. pp. 142-149.

[29] R. Poli, W.B. Langdon, M. Clerk, and C.R. Stephens. "Continuous optimisation theory made easy? Finite-element models of evolutionary strategies, genetic algorithms and particle swarm optimizers," in Proceedings of the Foundations of Genetic Algorithms workshop, Lecture notes in computer science, Berlin: Springer, 2007, vol. 4436, pp. 165–193.

[30] J.C. Zeng, J. Jie, and Z.H. Cui. Particle swarm algorithm. Beijing: Science Press, 2004.

[31] J.P. Luo, X. Li, and R.M. Chen. "The markov model of shuffled frog leaping algorithm and its convergence analysis," Acta Electronica Sinica, 2010, vol. 38, pp. 2875-2880.

[32] P.A. Ning, and X.Y. Zhang. "Convergence analysis of artificial bee colony algorithm," Control and Decision, 2013, vol. 28, pp.1554-1558.