

# Convergence and Parameters Analysis of Shuffled Frog Leaping Algorithm

Lianguo Wang

College of Information Science & Technology  
Gansu Agricultural University  
Lanzhou, China  
e-mail: wanglg@gsau.edu.cn

Yaxing Gong

College of Engineering  
Gansu Agricultural University  
Lanzhou, China  
e-mail: yaxing918@126.com

**Abstract**—Markov chain is an effective tool for convergence analysis of intelligence optimization algorithms. This paper briefly studies the state space of the basic Shuffled Frog Leaping Algorithm (SFLA) and theoretically analyzes the convergence behavior of SFLA by using Markov chain. It is proved that the SFLA has global convergence. Besides, the impact of key parameters on algorithm performance is discussed through simulation experiments. It provides the theoretical foundation and basis for using the algorithm to solve practical optimization problem.

**Keywords**—swarm intelligence; shuffled frog leaping algorithm; Markov chain; convergence; parameters analysis

## I. INTRODUCTION

Shuffled Frog Leaping Algorithm (SFLA) is one of biological evolutionary algorithms based on swarm intelligence which was proposed by Eusuff and Lansey in 2003 [1]. Because of its characteristics such as simple concept, lesser parameters, powerful optimal performance, fast calculation speed and easy realization, it has been primarily applied to many fields. The theoretical analyses about SFLA, however, were not very comprehensive. Some scholars have studied its convergence and the effects of algorithm parameters on its performance. Combined with global convergence criterion of stochastic optimization algorithm, reference [2] analyzed the convergence performance of SFLA. Reference [3] examined the trajectory of SFLA and studied its global convergence by adopting convergence criterion of stochastic algorithm. An improved SFLA was presented by learning from the evolutionary operator of particle swarm optimization and deviation evolution algorithm in [4]. Besides, the convergence of this novel algorithm had been proved by Markov chain theory. Reference [5] considered how to set the population size of meme triangular probability distribution shuffled frog leaping algorithm. The effects of primary parameters on shuffled frog leaping algorithm based on local orthogonal crossover operator had been analyzed in [6]. Based on previous researches made by [7~9], this paper analyzed the convergence performance of SFLA with the same method which was used for genetic algorithm in [10]. Besides, we discussed the impacts of algorithm parameters settings on SFLA by simulation experiments.

## II. FUNDAMENTAL PRINCIPLES AND MATHEMATICAL MODEL

In the process of simulating frog swarm to find food, SFLA spreads the ideas by grouping the population. It combines the communication of global information and deep search of local area to realize evolution. Local search is helpful to transmit information among individuals. Hybrid strategy can realize the exchange of local ideas and the sharing of global information. By this balanced strategy between global communication and local exploitation, SFLA could jump out the local extremum and finally find the global optimum.

Here we set the  $i$ 'th frog to  $X_i=(x_{i1},x_{i2},\dots,x_{in})$ ; the dimensions of variable to  $n$ ; the fitness of current frog's location to  $Y=f(X)$ . Frogs in the swarm are sorted in descending order by fitness values. Then we divide these frogs into  $m$  subgroups and each group contains  $s$  frogs, which meets the formula:  $F=m \times s$ . In this process, the first frog is assigned to the first subgroup; the second frog to the second subgroup; the  $m$ 'th frog to the  $m$ 'th subgroup; the  $(m+1)$ 'th frog to the first subgroup; the  $(m+2)$ 'th frog to the second subgroup. And so on, until all frogs are assigned to subgroups. In each subgroup, the best individual (having best fitness in each subgroup) is set to  $X_b=(x_{b1},x_{b2},\dots,x_{bn})$ ; the worst is  $X_w=(x_{w1},x_{w2},\dots,x_{wn})$ . In global view, the best individual (having best fitness in whole population) is set to  $X_g=(x_{g1},x_{g2},\dots,x_{gn})$ . Then local searches will be performed in each subgroup. In this stage, the  $X_w$  in each subgroup is circularly updated by (1,2):

$$d_j = \text{rand}() \times (x_{bj} - x_{wj}), \quad (1)$$

$$\begin{aligned} \text{new}x_{wj} &= x_{wj} + d_j, & (2) \\ D_{\max} &\geq d_j \geq -D_{\max}, j=1,2,\dots,n, \end{aligned}$$

where  $\text{new}X_w=(\text{new}x_{w1},\text{new}x_{w2},\dots,\text{new}x_{wn})$  is a new individual generated by update strategy;  $d_j$  is each moving step length of the  $j$ 'th dimensional component;  $\text{rand}()$  is a random number between 0 and 1;  $D_{\max}$  is the maximum allowable step. After updating, if the newly generated  $\text{new}X_w$  is better than the old  $X_w$ ,  $X_w$  will be substituted by  $\text{new}X_w$ ; otherwise,  $X_b$  will be substituted by  $X_g$  and strategy (1,2) will be repeatedly executed. If  $X_w$  still have no improvement, it will be replaced by a randomly generated individual. This update process will repeat until reaching the pre-set iterations. When local deep searches of all subgroups have finished, these subgroups will be mixed, sorted, and divided into components again. Then

repeat local searches until the stopping condition is reached. The detailed algorithm process of SFLA is represented as follow.

Step1: Randomly initialize  $F$  frogs as algorithm population and set internal iteration number and hybrid iteration number.

Step2: Calculate the individual fitness and find out the global best individual  $X_g$ .

Step3: Sort  $F$  frogs by their fitness values in descending order and divide them into  $m$  subgroups and each group contains  $s$  frog.

Step4: Determine the best individual  $X_b$  and the worst  $X_w$ . Update  $X_w$  by (1,2) in allowable iterations.

Step5: Calculate the individual fitness and find out the global best individual  $X_g$ .

Step6: Judge whether the stopping criterion is met and if so, the program will be terminated and the optimum  $X_g$  is outputted; otherwise the execution process will jump to step3.

### III. BASIC KNOWLEDGE

**Definition 1**  $P$  is a transition probability matrix of size  $k \times k$ ,  $\{p_{ij}; i, j=1, 2, \dots, k\}$ . The stochastic process  $(X_0, X_1, \dots)$  of a finite state space  $S = \{s_1, s_2, \dots, s_k\}$  is called a time-homogeneous Markov chain of  $P$  if for any given  $n$ ,  $i, j \in \{1, 2, \dots, k\}$ ,  $i_0, i_1, \dots, i_{n-1} \in \{1, 2, \dots, k\}$ , eq.(3) is true.

$$P(X_{n+1}=s_j | X_0=s_{i_0}, X_1=s_{i_1}, \dots, X_{n-1}=s_{i_{n-1}}, X_n=s_{i_n}) = P(X_{n+1}=s_j | X_n=s_i) = p_{ij} \quad (3)$$

**Definition 2** The time-homogeneous Markov chain  $(X_0, X_1, \dots)$  of transition probability matrix  $P$  is called an irreducible one, if for any given  $s_i, s_j \in S$ , eq.(4) is true.

$$P(X_{m+n}=s_j | X_m=s_i) > 0 \quad (4)$$

If the Markov chain  $(X_0, X_1, \dots)$  is time-homogeneous and irreducible, then

$$P(X_{m+n}=s_j | X_m=s_i) = (P^n)_{ij} > 0 \quad (5)$$

**Definition 3** The Markov chain  $(X_0, X_1, \dots)$  is said to be nonperiodic, if for any given  $s_i \in S$ , eq.(6) is true.

$$d(s_i) = \gcd\{n \geq 1: (P^n)_{ii} > 0\} = 1 \quad (6)$$

where  $\gcd\{a_1, a_2, \dots\}$  is the greatest common divisor of  $a_1, a_2, \dots$ . A state  $s_i \in S$  has period  $d(s_i)$  if any return (existing a positive probability) to state  $s_i$  must occur in multiples of  $d(s_i)$  time steps. Besides, the corresponding transition probability matrix  $P$  is primitive.

**Lemma 1** Strictly positive Markov chains are irreducible and aperiodic.

**Lemma 2** If  $P$  is a  $n$  order, reducible and stochastic matrix which can be switched to

$$P' = \begin{pmatrix} C & 0 \\ R & T \end{pmatrix}$$

by same row and column transformation, where  $C$  is a  $m$  order primitive matrix, and  $R, T \neq 0$ , then

$$P^{\infty} = \lim_{k \rightarrow \infty} P^{nk} = \lim_{k \rightarrow \infty} \begin{pmatrix} C^k & 0 \\ \sum_{i=0}^{k-1} T^i R C^{k-i} & T^k \end{pmatrix} = \begin{pmatrix} C^{\infty} & 0 \\ R^{\infty} & 0 \end{pmatrix} \quad (7)$$

is a stable stochastic matrix and  $P^{\infty} = 1' p^{\infty}$ . Here,  $p^{\infty} = p^0$ ;  $P^{\infty}$  is determined uniquely and independent of initial distribution;  $p_i^{\infty} > 0, 1 \leq i \leq m; p_i^{\infty} = 0, m < i \leq n$  [11].

### IV. CONVERGENCE ANALYSIS OF SFLA

#### A. State space analysis

If we express component  $x_i$  of the frog  $X = (x_1, x_2, \dots, x_n)$  as  $M$ -bit binary series—quantize  $[\underline{x}_i, \bar{x}_i]$  to  $2^M$  discrete values, the precision can be expressed as  $\varepsilon = (\bar{x}_i - \underline{x}_i) / 2^M$ . Therefore, the convergence of SFLA could be analyzed directly by real-number encoding. If the needed precision is  $\varepsilon$ , the searching space  $S$  could be considered as a discrete space. Its size is

$$|S| = \prod_{i=1}^n \lceil (\bar{x}_i - \underline{x}_i) / \varepsilon \rceil.$$

The fitness of each element  $X \in S$  is  $Fitness(X)$ . If  $F = \{Fitness(X) | X \in S\}$ , obviously  $|F| \leq |S|$ , then  $F$  can be expressed as  $F = \{F_1, F_2, \dots, F_{|F|}\}$ , where  $F_1 > F_2 > \dots > F_{|F|}$ .  $S$  can be divided into several nonempty subsets  $\{S_i | i=1, 2, \dots, |F|\}$  by fitness values, and where  $S_i$  has the form

$$S_i = \{X | X \in S \text{ and } Fitness(X) = F_i\}; \quad (8)$$

$$\sum_{i=1}^{|F|} |S_i| = |S|; S_i \neq \emptyset, \forall i \in \{1, 2, \dots, |F|\}; S_i \cap S_j = \emptyset, \forall i \neq j; \bigcup_{i=1}^{|F|} S_i = S. \quad (9)$$

For any given two elements  $X_i \in S_i$  and  $X_j \in S_j$ , eq.(10) must be true.

$$\begin{aligned} Fitness(X_i) &> Fitness(X_j), \text{ if } i < j \\ Fitness(X_i) &= Fitness(X_j), \text{ if } i = j \\ Fitness(X_i) &< Fitness(X_j), \text{ if } i > j \end{aligned} \quad (10)$$

Obviously,  $F_1$  is the global optimum  $F^*$ , and set  $S_1$  contains all individuals whose fitness is equal to  $F^*$ .

In the whole evolutionary process of SFLA, the individual number  $N$  of population  $p = \{X_1, X_2, \dots, X_N\}$  is remain unchanged.

Let  $P$  be the whole swarm sets. Since the individuals in a swarm could be same, the possible population number is:

$$|P| = \binom{|S| + N - 1}{N} \quad (11)$$

In order to measure the population quality, we define the fitness of swarm  $P$  as:

$$Fitness(P) = \max \{f(X_i) | i=1, 2, \dots, N\} \quad (12)$$

Then,  $F_{|F|} \leq Fitness(p) \leq F_1, \forall p \in P$ . Therefore, we can divide  $P$  into nonempty subsets  $\{P_i\}$ :

$$P_i = \{p | p \in P \text{ and } Fitness(p) = F_i\}, i=1, 2, \dots, |F|; \quad (13)$$

$$\sum_{i=1}^{|F|} |P_i| = |P|; P_i \neq \emptyset, \forall i \in \{1, 2, \dots, |F|\}; P_i \cap P_j = \emptyset, \forall i \neq j; \bigcup_{i=1}^{|F|} P_i = P. \quad (14)$$

The set  $P_1$  contains all individuals whose fitness is  $F^*$ .

Let  $P_{ij}$  be the  $j$ 'th swarm of  $P_i$ ,  $i=1,2,\dots,|F|$ ,  $j=1,2,\dots,|P_i|$ . The state transitions from  $P_{ij}$  to  $P_{kl}$  can be expressed as  $P_{ij} \rightarrow P_{kl}$  under the influence of the evolutionary operators of SFLA. Let  $p_{ij,kl}$  be the transition probability from  $P_{ij}$  to  $P_{kl}$ ;  $p_{ij,k}$  be the transition probability from  $P_{ij}$  to any swarm of  $P_k$ ;  $p_{i,k}$  be the transition probability from any swarm of  $P_i$  to any swarm of  $P_k$ . Obviously, eq.(15) is true.

$$P_{ij,k} = \sum_{l=1}^{|P_k|} p_{ij,kl}, \sum_{k=1}^{|F|} P_{ij,k} = 1, p_{i,k} \geq p_{ij,k} \quad (15)$$

### B. Convergence analysis of SFLA

**Definition 4** An evolutionary algorithm converges to the global optimization solution if and only if:

$$\lim_{t \rightarrow \infty} Pr\{Fitness(P^t) = F^*\} = 1. \quad (16)$$

Here,  $Pr$  is probability;  $P^t$  is the  $t$ 'th generation population.

**Theorem 1** The searching process of SFLA is a time-homogeneous Markov chain.

*Proof.*

According to the operating principle of SFLA, this algorithm conduct searches in a finite dimensional population state  $S$ . And in the searching process, the next generation swarm  $P^{t+1}$  is dependent only on the current population and independent of previous generations ( $P^{t-1}, P^{t-2}, \dots, P^1, P^0$ ). Therefore, the conditional probability from a given swarm  $P^t$  to a specific one  $P^{t+1}$  is not affected by the previous change at any time. Learning from the definition of Markov chain, the searching process of algorithm have Markov property—stability ineffectiveness which meet Markov criterion. Thus, the searching process of SFLA can be described by time-homogeneous Markov chains with finite state space.

**Theorem 2** In SFLA,  $\forall i, k \{1, 2, \dots, |F|\}$ , eq.(17) is true.

$$\begin{cases} p_{i,k} > 0, k \leq i \\ p_{i,k} = 0, k > i \end{cases} \quad (17)$$

*Proof.*

$\forall P_{ij} \in P_i$ ,  $i=1,2,\dots,|F|$ ,  $j=1,2,\dots,|P_i|$ ,  $\exists X^*=(x_1^*, x_2^*, \dots, x_n^*) \in P_{ij}$ ,  $Fitness(X^*)=F_i$ ;

$\forall P_{kl} \in P_k$ ,  $k=1,2,\dots,|F|$ ,  $l=1,2,\dots,|P_k|$ ,  $\exists X'=(x_1', x_2', \dots, x_n') \in P_{kl}$ ,  $Fitness(X')=F_k$ .

Let  $P_{ij}$  switch to  $P_{kl}$  under the updating behavior of SFLA. If  $P_{ij}$  is the  $t$ 'th generation population and denoted as  $P^t$ , then  $P_{kl}$  is the  $(t+1)$ 'th one and denoted as  $P^{t+1}$ .

At first, we use the elitist reserve strategy to make  $X_g$  to save the previous best states of population in SFLA. According to the above definitions and assumptions, the best individual of  $P^t$  is  $X^*$  and it is saved in  $X_g$ :  $X_g=X^*$ . At the same time, the composition of swarm  $P^t$  is not changed.  $P^{t+1}$  is generated by  $P^t$  through the updating behavior of SFLA. We make comparisons between the best individual  $X'$  of  $P^{t+1}$  and  $X_g$ . If  $X'$  is better than  $X_g$ , then  $X_g=X'$ ; otherwise  $X_g$  remain unchanged. Put  $X_g$  on the far left side of the population and it doesn't participate in the evolutionary process. Thus, we could deduce the following process.

$$Fitness(X_g) \geq Fitness(X')$$

$$\Rightarrow Fitness(P^{t+1}) \geq Fitness(P^t)$$

$$\Rightarrow k \leq i$$

$$\Rightarrow \forall k > i, p_{ij,kl} = 0$$

$$\Rightarrow \forall k > i, p_{ij,k} = \sum_{l=1}^{|P_k|} p_{ij,kl} = 0$$

$$\Rightarrow \forall k > i, p_{i,k} = 0$$

Next, each individual of subgroups could update its position with the iterations in SFLA. The algorithms have three update strategies for searching optimum in each iteration. The detailed analysis is showed as follow:

(1) Individual  $X'$  learns from the best individual  $X_b$  in a subgroup and then  $X'$  is generated. If  $Fitness(X') \geq Fitness(X^*)$ —the probability  $p_b$  of this case is bigger than 0, then  $\forall k \leq i$ ,  $p_{i,k} > 0$  and the proposition is verified; otherwise  $X^*$  will be updated by learning from the global best individual  $X_g$ .

(2) Individual  $X'$  learns from the global best individual  $X_g$  in the whole population and then  $X'$  is generated. If  $Fitness(X') \geq Fitness(X^*)$ —the probability  $p_g$  of this case is bigger than 0, then  $\forall k \leq i$ ,  $p_{i,k} > 0$  and the proposition is verified; otherwise  $X^*$  will be replaced by a randomly generated individual.

(3) The probability  $p_r$  of  $X^*$  is substituted by randomly generated individual  $X'$  in the search range is:

$$p_r = \prod_{i=1}^n \frac{\varepsilon}{\bar{x}_i - \underline{x}_i} > 0,$$

then the proposition is verified.

Combining foregoing analysis, we know that  $p_{i,k} > p_{ij,k} > 0$ ,  $\forall k \leq i$ . Thus, the proof is finished.

This theorem shows that the swarm which have lower fitness could transit to the one which have same or higher fitness, but not from higher one to lower one. Therefore, the algorithm cannot get out once it enters set  $P_1$ .

**Theorem 3** SFLA has global convergence.

*Proof.*

According to the result of theorem 1, each  $P_i$ ,  $i=1,2, \dots, |F|$ , could be regarded as a state of the finite Markov chain. We can get the transition matrix of this Markov chain as (18) from the conclusion of theorem 2.

$$P' = \begin{pmatrix} P_{1,1} & P_{1,2} & \dots & 0 \\ P_{2,1} & P_{2,2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ P_{|F|,1} & P_{|F|,2} & \dots & P_{|F|,|F|} \end{pmatrix} = \begin{pmatrix} C & 0 \\ R & T \end{pmatrix} \quad (18)$$

Here,  $R=(p_{2,1} \ p_{3,1} \ \dots \ p_{|F|,1})^T > 0, T \neq 0, C=(p_{1,1})=(1) \neq 0$ .

We can get (19) from lemma 2.

$$P'^{\infty} = \lim_{k \rightarrow \infty} P'^{k} = \lim_{k \rightarrow \infty} \left( \sum_{i=0}^{k-1} T^i R C^{k-i} \quad T^k \right) = \begin{pmatrix} C^{\infty} & 0 \\ R^{\infty} & 0 \end{pmatrix} \quad (19)$$

Here,  $C^{\infty}=1, R^{\infty}=(1 \ 1 \ \dots \ 1)^T$ . So  $P'^{\infty}$  is a stable stochastic matrix, and

$$P^{\infty} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 1 & 0 & \dots & 0 \end{pmatrix}.$$

This illustrates that the probability of individual staying at non-optimal solution approaches zero, that is to say, as the time tends to infinity, SFLA will converge to optimum with probability 1.

Therefore, we can get the following formula from definition 4:

$$\lim_{t \rightarrow \infty} \Pr\{\text{Fitness}(P^t) = F^*\} = 1,$$

which completes the proof.

## V. PARAMETERS ANALYSIS OF SFLA

In this paper, we took five standard test functions of optimization problems to discuss the effects of step, subgroup number, and population size on SFLA. The parameters settings of these functions were shown in Table 1.

$$f_1(x) = \sum_{i=1}^n x_i^2$$

$$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1}^2 - x_i)^2 + (1 - x_i)^2)$$

$$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$f_5(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$$

TABLE I. THE PARAMETERS OF BENCHMARK TEST FUNCTIONS

Function	Dimension	Search range	Theoretical optimum	Objective precision
$f_1$	30	[-5.12,5.12]	0	$10^{-5}$
$f_2$	30	[-30,30]	0	$10^2$
$f_3$	30	[-5.12,5.12]	0	$10^1$
$f_4$	30	[-600,600]	0	$10^{-2}$
$f_5$	30	[-32,32]	0	$10^{-2}$

### A. The effects of step on algorithm performance

In these experiments, swarm size was set to 200 frogs. The swarm was divided into 20 subgroups and each group contained 10 frogs. The maximum number of iterations was 500 and the number of internal iterations in the subgroup was 10. Use different step to experiment and each test is repeated thirty times. The average results of all experiments were listed in Table 2~6, where  $D_{max}$  is the maximum search range.

TABLE II. EXPERIMENTAL RESULTS ABOUT STEP OF  $F_1$

Step	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
$D_{max}$	0.014481	0.001971	0.038850	0.008619	0.4430
$D_{max}/2$	0.013348	0.004070	0.039540	0.009156	0.4349

Step	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
$D_{max}/5$	0.006478	0.001864	0.014920	0.003581	0.4303
$D_{max}/10$	0.009530	0.001603	0.023551	0.005511	0.4305
$D_{max}/15$	0.011772	0.001989	0.034278	0.007797	0.4318
$D_{max}/20$	0.015247	0.001644	0.045563	0.010697	0.4312

TABLE III. EXPERIMENTAL RESULTS ABOUT STEP OF  $F_2$

Step	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
$D_{max}$	234.3173	118.3145	455.9867	84.1175	0.4664
$D_{max}/2$	203.8843	49.3228	492.7374	93.0078	0.4421
$D_{max}/5$	138.6388	52.6125	285.4543	56.6302	0.4423
$D_{max}/10$	190.3842	61.5119	368.3378	70.4728	0.4378
$D_{max}/15$	160.7564	92.9695	328.9869	56.0892	0.4421
$D_{max}/20$	158.5924	75.4645	278.5388	54.9013	0.4407

TABLE IV. EXPERIMENTAL RESULTS ABOUT STEP OF  $F_3$

Step	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
$D_{max}$	14.4916	6.6531	24.2175	4.8170	0.5881
$D_{max}/2$	14.3147	3.7151	21.4093	4.8058	0.5699
$D_{max}/5$	11.3535	2.1147	24.9701	4.9772	0.5694
$D_{max}/10$	17.1678	3.8959	31.7710	6.3704	0.6282
$D_{max}/15$	143.7956	112.6888	176.2406	18.6704	0.7663
$D_{max}/20$	190.8729	172.4977	213.0382	12.0297	0.7874

TABLE V. EXPERIMENTAL RESULTS ABOUT STEP OF  $F_4$

Step	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
$D_{max}$	1.0390	0.7594	1.2250	0.0849	0.6281
$D_{max}/2$	0.9043	0.3005	1.0847	0.2135	0.6001
$D_{max}/5$	0.9004	0.5698	1.0951	0.1600	0.6100
$D_{max}/10$	0.9098	0.3665	1.1569	0.2296	0.6073
$D_{max}/15$	0.9845	0.4471	1.0812	0.1417	0.6068
$D_{max}/20$	0.9797	0.4854	1.4347	0.1782	0.6054

TABLE VI. EXPERIMENTAL RESULTS ABOUT STEP OF  $F_5$

Step	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
$D_{max}$	1.3180	0.2038	2.5891	0.5841	0.6418
$D_{max}/2$	1.3145	0.2003	2.1441	0.5721	0.6274
$D_{max}/5$	1.0359	0.1194	2.3274	0.6252	0.6311
$D_{max}/10$	0.6320	0.1331	1.7174	0.3983	0.6232
$D_{max}/15$	0.5409	0.1096	1.5711	0.4076	0.6244
$D_{max}/20$	0.4663	0.1349	1.3999	0.2916	0.6228
$D_{max}/25$	0.5540	0.2189	1.2206	0.2832	0.6233
$D_{max}/30$	0.4897	0.0972	1.0293	0.2590	0.6234

From Table 2~6, we can see that the results for function  $f_1$ ,  $f_2$ , and  $f_3$  is, as the step reduces, the optimization precision rise gradually at first and then start to decrease when step meets certain range (the range refers to step is  $D_{max}/5$  in our experiment). For function  $f_4$ , the impact of step change on precision is small. For function  $f_5$ , as the step reduces, the

optimization precision rise gradually at first and then start to keep stable when the step meets certain range (the range refers to step is  $D_{max}/10$  in our experiment). Besides, for every function, the effects of step change on operating time are less obvious. Generally speaking, for  $f_1 \sim f_4$ ,  $step = D_{max}/5$  is recommended; for  $f_5$ ,  $step = D_{max}/20$  is suggested.

### B. The effects of subgroup number on algorithm performance

According to the previous experimental results, for  $f_1 \sim f_4$ , let step be  $D_{max}/5$ ; for  $f_5$ , let step be  $D_{max}/20$ . And let swarm size be 200 and maximum number of iterations was 500. Use different subgroup numbers  $m$  which respectively are 2, 4, 5, 10, 20, 40, 50, and 100, correspondingly, the individual numbers  $s$  of each subgroup are 100, 50, 40, 20, 10, 5, 4, and 2 in order. Let the number of internal iterations be 10. Every algorithm is also repeated 30 times. The relevant results were shown in Table 7~11.

TABLE VII. EXPERIMENTAL RESULTS ABOUT SUBGROUP OF  $F_1$

$m \times s$	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
2×100	2.022E-01	4.948E-02	4.644E-01	8.762E-02	0.3377
4×50	7.978E-02	2.750E-02	2.234E-01	4.332E-02	0.4103
5×40	9.632E-02	2.253E-02	2.420E-01	5.553E-02	0.4363
10×20	4.560E-02	9.343E-03	9.159E-02	2.091E-02	0.4763
20×10	8.776E-03	2.074E-03	3.891E-02	7.533E-03	0.4383
40×5	2.372E-04	2.282E-05	9.347E-04	2.532E-04	0.4742
50×4	1.651E-05	2.244E-06	5.821E-05	1.433E-05	0.5085
100×2	1.082E-08	2.301E-09	2.802E-08	7.837E-09	0.7332

TABLE VIII. EXPERIMENTAL RESULTS ABOUT SUBGROUP OF  $F_2$

$m \times s$	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
2×100	1550.268	439.9518	8720.922	1465.545	0.3157
4×50	745.6922	213.8186	3429.119	632.4862	0.4191
5×40	525.4500	203.7292	1650.482	314.5442	0.4363
10×20	262.8235	93.2583	662.2425	138.0759	0.4586
20×10	174.6654	46.9399	468.1013	87.6370	0.4456
40×5	77.8993	28.0122	185.1117	36.8798	0.4914
50×4	67.9580	23.6875	139.2855	32.9119	0.5486
100×2	62.8352	22.7532	133.9588	40.1691	0.7852

TABLE IX. EXPERIMENTAL RESULTS ABOUT SUBGROUP OF  $F_3$

$m \times s$	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
2×100	44.2180	27.6512	64.0645	10.2445	0.3182
4×50	33.8031	9.7029	49.3681	9.7167	0.4352
5×40	33.2514	16.1965	62.6480	10.5324	0.4680
10×20	19.0451	2.7875	35.8327	7.9599	0.5340
20×10	10.2092	1.6119	23.7418	5.1527	0.5725
40×5	6.9082	2.0639	14.8921	3.4088	0.7420
50×4	8.9895	1.9934	22.2454	4.0299	0.8439
100×2	12.3753	5.9699	18.9047	3.1348	1.3816

TABLE X. EXPERIMENTAL RESULTS ABOUT SUBGROUP OF  $F_4$

$m \times s$	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
2×100	1.6009	1.1526	2.3011	0.2671	0.3338
4×50	1.3476	1.1428	2.0174	0.1683	0.4508
5×40	1.2222	1.0654	1.5539	0.1007	0.4820
10×20	1.1190	1.0571	1.2678	0.0560	0.5434
20×10	0.8600	0.3833	1.1248	0.2083	0.6084
40×5	0.0947	0.0163	0.2933	0.0680	0.8190
50×4	0.0351	0.0007	0.1559	0.0351	0.9589
100×2	0.0170	0.0001	0.0788	0.0197	1.6421

TABLE XI. EXPERIMENTAL RESULTS ABOUT SUBGROUP OF  $F_5$

$m \times s$	Average optimum	Minimum	Maximum	Standard deviation	Operating time(s)
2×100	3.3155	2.6673	4.0267	0.4402	0.3255
4×50	2.3595	1.6525	3.2281	0.4394	0.4441
5×40	2.1207	1.2811	2.9736	0.4369	0.4773
10×20	1.4405	0.5096	2.2784	0.4634	0.5340
20×10	0.5333	0.1172	1.3875	0.3095	0.6058
40×5	0.0468	0.0111	0.1483	0.0344	0.7997
50×4	0.0154	0.0049	0.0393	0.0087	1.0286
100×2	0.0004	0.0002	0.0014	0.0002	1.5444

From Table 7~11, we get that for all functions, basically, the optimization precision increases with the subgroup number decreases. Therefore, considering the time cost, for  $f_1$ ,  $m$  could be 20, 40, and 50; for  $f_3$ ,  $m=20$  is recommended; for  $f_2$ ,  $f_4$ ,  $f_5$ ,  $m$  could be 20,40.

### C. The effects of population size on algorithm performance

In our experiments, population size was set to different values; the detailed settings were shown in Table 12.

To clearly represent the impact of population size on algorithm performance, here, the maximum number of iterations was set to 2000. We used the common log of the average extremum and the average time reached the given convergence precision (Table 1) in 30 trial times as evaluation criteria. The final results were shown in Fig.1~2.

TABLE XII. SUBGROUP NUMBER SETTINGS ABOUT TESTING POPULATION SIZE

Population size	Subgroup number	Individual number of each subgroup
10	5	2
50	10	5
100	10	10
150	15	10
200	20	10
250	25	10
300	20	15
350	25	14
400	25	16
450	25	18
500	25	20

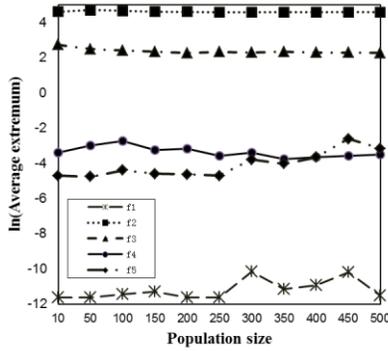


Figure 1. The relationship of population size and function value.

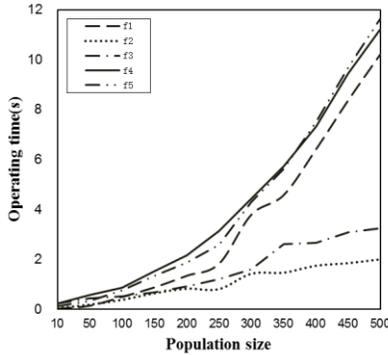


Figure 2. The relationship of population size and operating time.

Fig.1 shows that for  $f_1$ , the best optimization is gained at the population size is 200~250; for  $f_2, f_3$ , the algorithm performance is not obviously change with the swarm size increases; for  $f_4$ , the precision maintains at a better level after the size reaches 250; for  $f_5$ , the accuracy maintains at a better level before the size is 250 and start to decrease after 250.

Fig.2 illustrates that for  $f_1 \sim f_5$ , the operating time increases with the swarm size increases. Thus, the population size of SFLA should be 200~250.

## VI. CONCLUSIONS

Markov chain is an effective tool for convergence analysis of intelligence optimization algorithms. In this paper, we described the fundamental definitions and theorems and briefly analyzed the state space of SFLA. The global convergence of SFLA based on elitist reserve strategy has been proved by theoretically analyzing the convergence behavior of SFLA through Markov chain. Those provide theoretical foundation and basis for solving practical problems better. Besides, in order to help the algorithm to realize effectively, we used benchmark test functions to discuss the impacts of algorithm parameters settings on SFLA and give the recommended parameters values.

## ACKNOWLEDGMENT

Thanks for the support from the Natural Science Foundation of China (61063028) and the Education Informatization Development Strategy Research Project in Gansu (2011-2).

## REFERENCES

- [1] M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Water Resources Planning and Management*, vol.129, no.3, pp.210-225, 2003.
- [2] J. P. Luo, X. Li, and M. R. Chen, "The markov model of shuffled frog leaping algorithm and its convergence analysis," *Acta Electronica Sinica*, vol.38, no.12, pp.2875-2880, 2010. (in Chinese)
- [3] J. P. Luo and M. R. Chen, "Study on trajectory and convergence analysis of shuffled frog leaping algorithm and its improved algorithm," *Signal Processing*, vol.26, no.9, pp.1428-1433, 2010. (in Chinese).
- [4] Y. C. He, W. L. Qu, and J. W. Xu, "Improved shuffled frog leaping algorithm and its convergent analysis," *Computer Engineering and Applications*, vol.47, no.22, pp.37-40, 2011. (in Chinese)
- [5] G. Y. Zhu, "Meme triangular probability distribution shuffled frog leaping algorithm," *Computer Integrated Manufacturing Systems*, vol.15, no.10, pp. 1979-1985, 2009. (in Chinese)
- [6] Meng Q Y, "The research of shuffled frog leaping algorithm based on local orthogonal crossover operator and its application," Lanzhou: Gansu Agricultural University, 2011. (in Chinese)
- [7] W. C. Zhong, M. Z. Xue, J. Liu and L. C. Jiao, "Multiple intelligent agents genetic algorithm used for optimizing hyper-high dimensional functions," *Progress in Natural Science*, vol.13, no.10, pp.1078-1083, 2003. (in Chinese)
- [8] J. Liu, W. C. Zhong, F. Liu, and L. C. Jiao, "An organizational evolutionary algorithm for constrained and unconstrained optimization problems," *Chinese Journal of Computers*, vol.27, no.2, pp.157-167, 2004. (in Chinese)
- [9] R. Jiang, Y. P. Luo, D. C. Hu, and G. Y. Situ, "A genetic algorithm by coordinating exploration and exploitation-convergence properties and performance analyses," *Chinese Journal of Computers*, vol.24, no.12, pp.1233-1241, 2001. (in Chinese)
- [10] B. Dinabandhu, C. A. Murthy, and K. P. Sankar, "Genetic algorithm with elitist model and its convergence," *International Journal of Pattern Recognition and Artificial Intelligence*, vol.10, no.6, pp.731-747, 1996.
- [11] M. Iosifescu, "Finite Markov Processes and Their Application," Wiley: Chichester, 1980.