

# Shuffled frog leaping algorithm based on quantum rotation angle

*Bibo Hu\**

Guangzhou College of Technology and Business, Guangzhou, China

**Keywords:** Shuffled frog leaping algorithm, Quantum rotation angle, Artificial bee colony algorithm.

**Abstract.** In this paper, through the analysis of the artificial intelligence algorithm, shuffled frog leaping algorithm is effectively improved, and the position of the frog is determined by the quantum rotation angle, so as to improve the performance of the algorithm. Compared with the artificial bee colony algorithm and the shuffled frog leaping algorithm, the improved algorithm has a significant improvement in the convergence speed of the algorithm and the ability to jump out of the local area.

## 1 Introduction

With the continuous progress of society, more and more nonlinear extremum problems need to be solved urgently. The traditional mathematical extremum methods which must be continuously derivable are greatly restricted, so many optimization algorithms emerge as the times require[1]. Some bionic cluster intelligent algorithms are widely used, such as genetic algorithm, particle swarm optimization algorithm(pso), shuffled frog leaping algorithm and so on. Among them, the shuffled frog leaping algorithm proposed by American scholars Eusuff and Lansey in 2003 has attracted extensive attention in recent years because of its simple algorithm, few parameters and easy to understand. Compared with genetic algorithm and particle swarm optimization algorithm, under the continuous optimization model, the convergence speed and jumping out of the local optimal solution , the shuffled frog leaping algorithm are better than the genetic algorithm, close to the particle swarm optimization algorithm[2].

However, shuffled frog leaping algorithm also has the common problems of the bionic intelligent algorithm, which is mainly manifested in that the local search ability weakens gradually with the progress of the algorithm, the population diversity decreases rapidly, and the effect is not obvious in solving the high-dimensional continuous optimization problem. Many scholars have improved Shuffled Frog Leaping Algorithm in order to improve the performance of the algorithm, mainly in the aspects of parameter adjustment, population update mode and the combination of intelligent algorithms. This study is mainly about shuffled frog leaping algorithm based on quantum Rotation Angle.

---

\* Corresponding author: [amone222@163.com](mailto:amone222@163.com)

## 2 The principle of frog jumping algorithm

Consists of a frog population, with a total of  $p$  individuals, each individual represents a solution vector of the solution space. Each frog is sorted according to its fitness value and arranged in descending order. Then, according to the need, the frogs are divided into  $m$  subgroups, and there are  $n$  frogs in each subgroup. The first to  $m$  frogs were placed in your  $1$  to  $m$  groups, the  $m + 1$  frogs were re-placed in the first group, and so on. In each iteration of the algorithm, the frogs with the highest and lowest fitness in each group of frogs are recorded as  $x_b$  and  $x_w$  respectively, the frog mobile location for  $S_i = rand \times (x_b - x_w)$ , and every frog's new position is: the  $x_w = x_w + S_i$ ;  $S_{max} \geq S_i \geq -S_{max}$ , generating more optimal solution, if the iteration to replace the optimal solution is  $x_w$ , otherwise the entire population of optimal solutions to replace  $x_w$ , and the global optimal solution will be found gradually after many iterations. The flow chart of the algorithm is shown in the following figure:

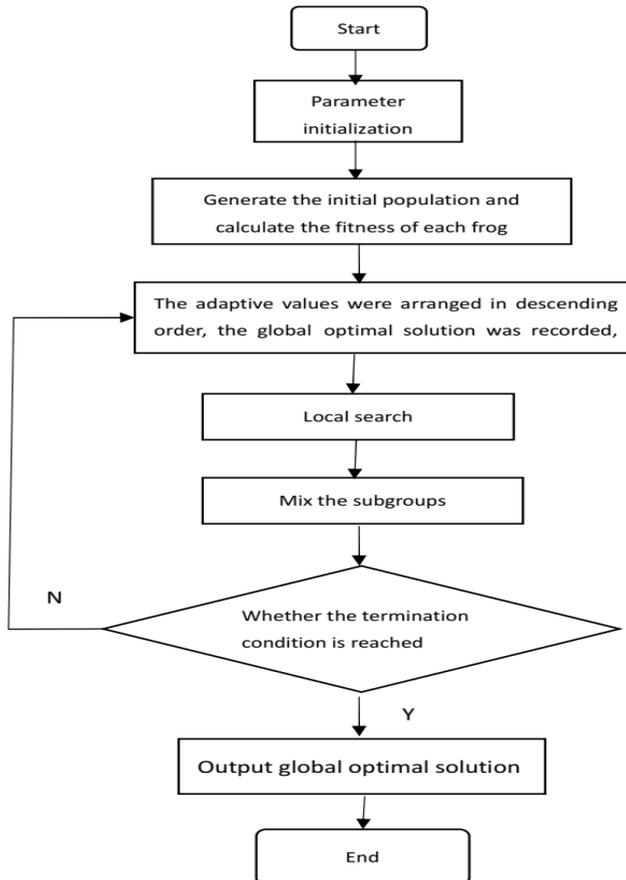


Fig. 1. Flow chart.

## 3 Shuffled frog leaping algorithm based on quantum rotation Angle

As an important branch of quantum theory, shuffled frog leaping algorithm based on quantum rotation angle has attracted the attention of many scholars. Compared with traditional computation, quantum computing has many advantages. At present, the research on quantum computing is mainly divided into two aspects: one is to design a new quantum algorithm, and

the other is to study the combination of quantum computing and traditional computing methods. The idea of quantum computing is introduced into the traditional computing to improve the performance of the traditional computing algorithm. In recent years, there have been a large number of quantum swarm intelligence optimization algorithms, such as quantum particle swarm optimization (QPSO) algorithm[3], quantum ant colony algorithm (QACA) [4], quantum fish swarm algorithm (QAFSA) [5] and so on.

Quantum computation is a new type of calculation based on the theory of quantum mechanics. On traditional computers, the basic unit of information is bit, Each bit is either "1" or "0". The smallest information unit stored on a quantum computer is called qubit. The state of a qubit is a vector of a two-dimensional complex space, which can be in the "1" state (denoted as  $|1\rangle$ ), "0" state denoted as  $|0\rangle$ ).

In addition, it can also be in a state that is neither  $|0\rangle$  nor  $|1\rangle$ , but above the so-called intermediate state of a linear combination of state  $|0\rangle$  and  $|1\rangle$ , that is, the linear superposition state of state  $|0\rangle$  and  $|1\rangle$ .  $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$  Oh, here.  $\alpha$  and  $\beta$  is arbitrary complex and satisfies the requirement of normalization  $|\alpha|^2 + |\beta|^2 = 1$ , where  $|\alpha|^2$  and  $|\beta|^2$  gives the probability that the qubit in the superposition state is "0" and "1".

Because of the superposition property of data and the unitary transformation nature of operation in quantum computing, the evolutionary operation of quantum system is actually the linear combination of its basic state  $|0\rangle$  state and  $|1\rangle$  state evolutionary operation. This property of quantum computing is called quantum parallelism. Quantum computing can greatly improve the operation speed because of quantum parallelism. In quantum computing, quantum states are used as the basic information unit, and the complex optimization problems can be effectively solved by quantum parallel computation by making use of the superposition, entanglement and interference of quantum states.

Shuffled frog leaping algorithm based on quantum Rotation Angle uses qubits to encode the current position of the frog, uses the quantum rotating gate to search the optimal position of the frog, and uses the quantum non-gate to realize the mutation of the frog position to avoid premature convergence<sup>[6]</sup>.

### 3.1 Frog position coding

Because of  $|\alpha|^2 + |\beta|^2 = 1$ , it is assumed that  $\alpha = \cos(\theta)$ ,  $\beta = \sin(\theta)$  qubits can be expressed in the following form:  $|\varphi\rangle = \cos(\theta)|0\rangle + \sin(\theta)|1\rangle = [\cos(\theta), \sin(\theta)]^T$ . The probability amplitude of the qubits in the above formula is used to represent the position of the frog. The position of the frog in quantum space can be expressed as the following form

$$B\_posit_i = \begin{bmatrix} \cos(\theta_{i1}) & \cos(\theta_{i2}) & \dots & \cos(\theta_{in}) \\ \sin(\theta_{i1}) & \sin(\theta_{i2}) & \dots & \sin(\theta_{in}) \end{bmatrix}, \text{ here } \theta_{ij} = 2\pi \times rand(), i=1,2,\dots,m; j=1,2,\dots,n; m$$

is the population size, n is the dimension of the solution space, and the frog position B\_posit represents 2n solutions in the solution space. For simplicity, only two solutions are taken, which correspond to the probability amplitudes of quantum states  $|0\rangle$  and  $|1\rangle$  respectively.

$$B\_posit_1 = (\cos(\theta_{i1}), \cos(\theta_{i2}), \dots, \cos(\theta_{in}))$$

$$B\_posit_2 = (\sin(\theta_{i1}), \sin(\theta_{i2}), \dots, \sin(\theta_{in}))$$

, For ease of expression, B\_posit1 is called cosine

position and B\_posit2 is called sine position. The frog position coding mentioned above can be further expressed in the form of polar coordinates in 2D Hilbert space, so that the updating and variation of frog position are carried out in 2D Hilbert space.  $B\_posit = (\theta_{i1}, \theta_{i2}, \dots, \theta_{in})$ .

### 3.2 Solution space transformation

The search space of each dimension is  $[-1,1]$  for the shuffled frog leaping algorithm Based on quantum rotation angle. In order to calculate the fitness value of the frog, it is necessary to transform the solution space. Suppose the  $j$  dimension qubit of the  $i$  frog is  $[\cos(\theta_{ij}), \sin(\theta_{ij})]^T$ , then convert it to the corresponding variable in the solution space is:

$$X_1^i = \frac{1}{2} [b_j(1 + \cos(\theta_{ij})) + a_j(1 - \cos(\theta_{ij}))]$$

$$X_2^i = \frac{1}{2} [b_j(1 + \sin(\theta_{ij})) + a_j(1 - \sin(\theta_{ij}))]$$

Here,  $a_j$  and  $b_j$  are the maximum and minimum values of the  $j$  dimension variable respectively.

### 3.3 Frog position update

In the shuffled frog leaping algorithm Based on quantum rotation angle, the quantum revolving gate is used to update the frog's position. Suppose that the  $j$  dimension qubit of frog  $i$  in the  $t$  generation is  $[\cos(\theta_{ij}(t)), \sin(\theta_{ij}(t))]^T$ , the  $j$  dimension qubit of frog  $i$  in the  $t+1$  generation is  $[\cos(\theta_{ij}(t+1)), \sin(\theta_{ij}(t+1))]^T$ , and the rotation Angle is  $\Delta\theta_{ij}(t+1)$ . The determination of rotation Angle is a key issue in the position update of frog group, and rotation Angle includes size and direction. The update of qubit amplitude and Angle increment is determined by frog swarm algorithm as  $\Delta\theta_{ij}(t+1) = \Delta\theta_{ij}(t) + \Delta\varphi_{ij}(\Delta\theta_{ij})$

$$\Delta\theta_{ij} = \begin{cases} 2\pi + \theta_{kj} - \theta_{ij}, & \theta_{kj} - \theta_{ij} < -\pi \\ \theta_{kj} - \theta_{ij}, & -\pi \leq \theta_{kj} - \theta_{ij} \leq \pi \\ \theta_{kj} - \theta_{ij} - 2\pi, & \theta_{kj} - \theta_{ij} > \pi \end{cases}$$

After the update, the two new positions are:

$$B\_posit_1 = (\cos(\theta_{i1}(t) + \Delta\theta_{i1}(t+1)), \dots, \cos(\theta_{in}(t) + \Delta\theta_{in}(t+1)))$$

$$B\_posit_2 = (\sin(\theta_{i1}(t) + \Delta\theta_{i1}(t+1)), \dots, \sin(\theta_{in}(t) + \Delta\theta_{in}(t+1)))$$

It can be seen that the quantum revolving gate realizes the simultaneous movement of two positions by changing the phase of describing qubits. Therefore, under the condition that the population size remains unchanged, the adoption of qubit encoding can expand the ergodicity of search and improve the optimization efficiency of the algorithm.

### 3.4 The steps of the shuffled frog leaping algorithm based on quantum rotation angle

The specific implementation steps of the shuffled frog leaping algorithm Based on quantum rotation angle are shown below. The main results are as follows:

(1) The frog group is initialized to form the initial population composed of the position of the frog group.

(2) The solution space is transformed to calculate the fitness of each frog. If the current fitness of the frog is better than that of the original frog, the current frog is used to replace the original frog position. If the current global optimal value is better than the current searched optimal position, the current global optimal value is also used to replace the current global optimal position.

(3) To update all frog positions

(4) To return to step (2) until the convergence condition is reached or the maximum number of iterations is reached. To get the final results.

### 4 Experimental results and analysis

To test the proposed algorithm is applied in the strives for the optimal value performance, choose the three commonly used functions as a test function, including f1 is unimodal functions, use it mainly to test the algorithm convergence speed and precision, optimization f2 and f3 is multimodal function, mainly used to test the algorithm's ability to jump out of local optimal and the ability to avoid premature.

**Table 1.** Three test functions.

Function	Name	Search Space	MinValue
f1	Rosenbrock	[-30,30]	0
f2	Rastrigin	[-5.12,5.12]	0
f3	Ackley	[-32,32]	0

$$f_1(x) = \sum_{i=1}^n [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$$

$$f_2(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$$

$$f_3(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{j=1}^n x_j^2}) - \exp(\frac{1}{n}\sum_{j=1}^m \cos(2\pi x_j)) + 20 + \exp(1)$$

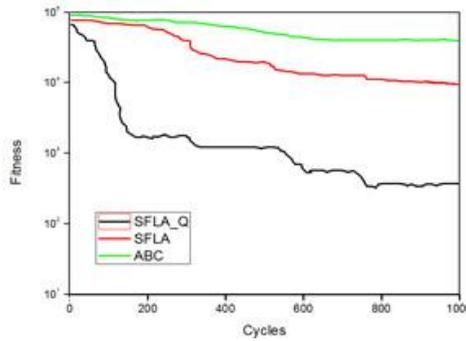
With SFLA algorithm, SFLA\_Q algorithm and ABC algorithm three algorithms for each test function independently 50 times experiment, get the corresponding optimal value, the worst value, mean value and variance, the test results are shown in Table 2:The test results. Among them, the best values and the worst values reflect the quality of the algorithm, the mean reflect the algorithm convergence speed and precision, the variance reflects stability and data dispersion degree of the algorithm.

As can be seen from table 2, the SFLA\_Q algorithm proposed in this paper has high optimization accuracy in both single-peak function and multi-peak function. Compared with ABC algorithm and SFLA algorithm, the best value and the worst value of the algorithm are obviously improved, indicating that SFLA\_Q algorithm is more stable. Under the same number of iterations, the improved algorithm obtains the minimum mean and minimum variance on each test function, which indicates that the convergence speed is faster, the optimization precision is higher and the performance is better.

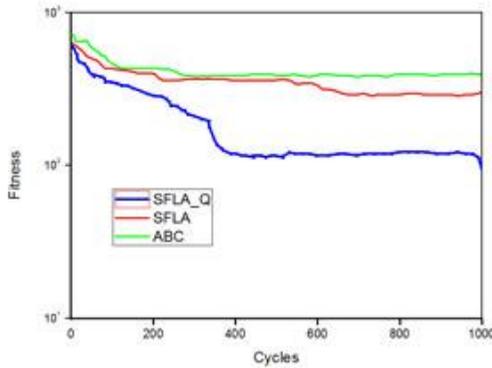
**Table 2.** The test results.

Function	algorithm	best value	worst value	mean	variance
f1 Rosenbrock	ABC	7.1023e+04	2.1396e+05	1.2142e+04	4.7065e+02
	SFLA	4.0297e+03	5.6843e+05	6.6954e+04	1.4642e+02
	SFLA_Q	1.7003	3.6876e+02	7.5970	1.1503e -01
f2 Rastrigin	ABC	1.0648e+03	9.4287e+06	4.7296e+05	8.1976e+06
	SFLA	7.4105e+02	3.3297e+05	1.1765e+04	1.3967e+04
	SFLA_Q	1.2297	2.3692e+02	8.2003	7.2364e+01
f3 Ackley	ABC	9.2285e+01	1.1496	1.1768e+01	2.9695
	SFLA	7.6894e-01	7.7805e+01	3.1326	1.5062
	SFLA_Q	1.9732e	4.8658e-02	3.7468e-04	3.7426e-03

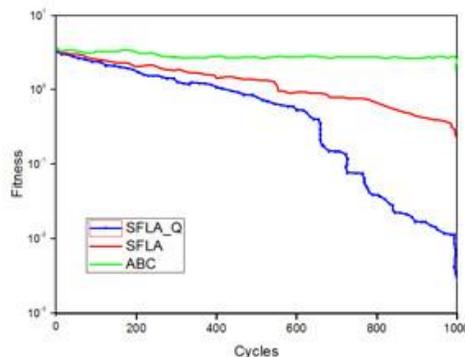
The following group of graphs shows the evolution curve of average fitness value of SFLA\_Q algorithm, ABC algorithm and SFLA algorithm that have run independently for 50 times. As can be seen from figure 2, the number of SFLA\_Q algorithm reaching stable convergence is far less than that of ABC algorithm and SFLA algorithm. As can be seen from figure 3, ABC algorithm and SFLA algorithm converge faster, have fewer iterations, but stop prematurely, and the optimization precision is not as good as SFLA\_Q algorithm. As can be seen from figure 4, SFLA\_Q algorithm shows strong advantages in the later stage of search, while ABC algorithm and SFLA algorithm are easy to fall into local optimization. From the above analysis, it can be seen that the optimization accuracy of SFLA\_Q algorithm is much higher than ABC algorithm and SFLA algorithm when it reaches stable convergence, especially the convergence speed of single-peak function is more obvious.



**Fig. 2.** F1 Rosenbrock.



**Fig. 3.** F2 Rastrigin.



**Fig. 4.** F3(Ackley).

## Reference

1. Liu, Chao ; Niu, Peifeng ; Li, Guoqiang ; Ma, Yunpeng ; Zhang, Weiping ; Chen, Ke, Enhanced shuffled frog-leaping algorithm for solving numerical function optimization problems, *Journal of Intelligent Manufacturing*, 2018, Vol.29(5), pp.1133-1153
2. Li, Rongbo ; Jiang, Zhiqiang; Li, Anqiang ; Yu, Shan; Ji, Changming, An improved shuffled frog leaping algorithm and its application in the optimization of cascade reservoir operation *Hydrological Sciences Journal*, 10 December 2018, Vol.63(15-16), pp.2020-2034
3. Sun J, Xu W B,Feng B. A global search strategy of quantum-behaved particle swarm optimization // *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems*. 2004,111-116.
4. Li Panchi, Li Shiyong, Quantum Ant Colony algorithm for solving continuous Space Optimization problems. *Control theory and application*. 2008,25 (2); 237-241
5. Jiang Mingyan, Yuan Dongfeng. *Artificial fish swarm algorithm and its application*. Beijing: science Press, 2012
6. Liu, Kui; Cai, Chunxiao; Li, Juan ; Ma, Long ; Sun, Hengxin ; Gao, Jiangrui, Squeezing-enhanced rotating-angle measurement beyond the quantum limit, *Applied Physics Letters*, 24 December 2018, Vol.113(26)