

## A New Updated Strategy Shuffled Frog Leaping Algorithm based on Gravitation Search Algorithm

YuHong Sun<sup>1, a\*</sup>, Wei Liu<sup>1, b</sup>, YueShan Xie<sup>1, c</sup>, WuJi He<sup>1, d</sup>, Hao Chen<sup>1, e</sup>

<sup>1</sup> School of Applied Mathematics, Guangdong University Of Technology, Guangzhou 510006, China

<sup>a</sup> AbnerSunyh@gmail.com, <sup>b</sup>liuweilw@gdut.edu.cn, <sup>c</sup>406874523@qq.com, <sup>d</sup>360147271@qq.com, <sup>e</sup>howes\_gooogle@gmail.com

**Keywords:** Shuffled Frog Leaping Algorithm, Gravitational search algorithm, Optimizing Performance

**Abstract.** In dealing with the problem that basic SFLA neglect the information exchange between individuals and sub-population in searching optimal solution, this paper proposes a new improved SFLA based on update rule of the Gravitation Search Algorithm. Based on the ideal from gravitation algorithm, the new SFLA regards the frog as a material with real quality and there exists mutual effect between any two frogs, a new update rule is proposed in this paper, which is able to drive the whole population to the optimal solution by prompting the movement of the worst frog under the effect of the gravitation. The simulation results show that the improved SFLA takes full advantage of the information between the sub-population to improve solution accuracy and convergence rate, which has a better practical performance and optimized performance.

### Introduction

Shuffled frog leaping algorithm (SFLA), a new swarm intelligence optimization algorithm, was proposed by Eusuf and Lansey who simulated the foraging behavior of frogs to solve the optimization problem in 2000. In recent years, the algorithm is improved and promoted by some scholars[1]. By Juan Lin and others, reverse operation was added to the process of population initialization and evolve respectively, producing the high quality species to improve optimization ability of the algorithm[1]. By Yongqiang Dai, the frog individual was set with memory function so that the frog is able to remember its last updated step length and the individual neighborhood history optimal value, and their adaptive learning ability improve the optimization performance of the algorithm.

However, the improved SFLA does not use the sharing information between subgroup frog, and ignores the influence from the state of the frogs in the subgroup to the around worst frog's behavior. According to the gravitation algorithm, we take the frog as a material with quality. Attraction between each frog manifest as competition and cooperation, promote the worst frog in the optimization direction under the action of gravity, realize population evolution. Therefore, in this paper, using the frog between gravity information, put forward a new update strategy, and further improve SFLA (noted as GSFLA). Simulation test results show that the improved GSFLA with basic SFLA and improved algorithm based on average for SFLA (noted as SFLA-AV), a new algorithm to speed up the convergence speed, and improves the precision, illustrates the feasibility and effectiveness of the algorithm.

### Basic SFLA

SFLA comes from within the defined space first randomly N initial solution (called the frog), The frog position  $i$  on behalf of the function in the feasible region of a solution  $X_i = (x_i^1, x_i^2, \dots, x_i^r)$ , and  $i = 1, 2, \dots, N$ .  $r$  is the dimensions of the solution space. Counting the frog applicable value (objective function value), then according to the applicable value regarding the frog subgroup, then the worst frog individuals within the groups according to the updated strategy optimization, so as to make the population evolution. The detailed steps of the algorithm are introduced below.

First, we should initialize the population (randomly generated  $N$   $r$ -dimension solutions) and calculate each objective function value of the frog. Then we sort every frog descending order according to the objective function value, write down the best frog for objective function values in a population  $X^g = (x_1^g, x_2^g, \mathbf{L}, x_r^g)$  and equally divide  $N$  frogs into  $s$  groups. Each group contains  $p$  frogs for  $N=p*s$ . In the group, the  $i$ -th frog is assigned to group and position  $[i/s]+1$ . Record best and worst in which individuals  $X^b = (x_1^b, x_2^b, \mathbf{L}, x_r^b)$  and  $X^w = (x_1^w, x_2^w, \mathbf{L}, x_r^w)$ . Then we will make a local search, updates the worst individual  $X^w = (x_1^w, x_2^w, \mathbf{L}, x_r^w)$  in each frog group according to the following policy in each search iterate:

The frog step calculation formula:

$$D^j = rand(1) * (X^b - X^w) \quad (1)$$

Updated frog position calculation formula:

$$X^{w'} = X^w + D^j \quad X^{w'} = X^w + D^j \quad (2)$$

$D_{max}$  is the maximum frog step length. If the worst frog according to the formula (2) can move to a better position, replace the worst individual position of the original subgroup with the position after the movement.; If not, replace the subgroups best frog position in the formula (1) with optimal frog position groups .and repeat calculation (1) (2); Otherwise, the random generation of a new position will replace the worst frog position. Continue to implement partial update operations within subgroup specified number until local search for a stipulated numbers, each subgroup of mixed updated original population, in descending order according to the objective function value, to delimit molecular group, this completes the SFLA one iteration.

From the basic steps of the algorithm we can know that SFLA not only combines the advantages of global optimization with local accurate searching but also can improve the operation efficiency greatly, jump out of local optimal and finds the optimal solution, is suitable for the optimization of continuous and discrete problem.

## The SFLA based on gravity regeneration strategy

### Gravitation

The interaction between two substances makes material move towards the more massive one. Here, the position of the material is the key because good location means greater mass. What' s more, the location of the substance will be continuously updated until it satisfies the algorithm' s the ending condition. Supposing  $fit_i(t)$  represents the  $t$ -th generation' s fitness value of the  $i$ -th frog,  $best(t)$  and  $worst(t)$  represents the best and worst fitness value for the subgroup.  $P$  represents the number of frog subgroup for the current algebra,  $t$  represents current generation . $T$  represents the maximum number of iterations for the population, and  $d$  represents the spatial dimension in the  $d$ -th direction for parameters.  $a$ ,  $G_0$ ,  $e$  are parameters for algorithm.

$$G(t) = G_0 \times e^{-\frac{a}{T}} \quad (3)$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (4)$$

$M_i(t)$ , the quality of the  $i$ -th frog in the  $t$ -th generation, is calculated as:

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^p m_j(t)} \quad (5)$$

$R_{ij}(t)$ , the distance between the  $i$ -th frog and  $j$ -th frog, is calculated as:

$$R_{ij}(t) = \sqrt{(x_i^1 - x_j^1)^2 + (x_i^2 - x_j^2)^2 + \dots + (x_i^d - x_j^d)^2} \quad (6)$$

$F_{ij}^d(t)$ , the attraction between the  $i$ -th frog and  $j$ -th frog in the  $d$ -th dimension direction, is calculated as:

$$F_{ij}^d(t) = G(t) \times \frac{M_i(t) \times M_j(t)}{R_{ij}(t)} \times (X_j^d(t) - X_i^d(t)) \quad (7)$$

$TF_i^d(t)$ , the resultant get by the  $i$ -th frog of the  $t$ -th generation, coming from  $P$  frogs in subgroup in the direction of the  $d$ -th dimension, is calculated as:

$$TF_i^d(t) = \sum_{j=1}^p F_{ij}^d(t) \quad (8)$$

Considering the mass of the worst frog of the subgroup, calculated by the equation (5),  $M_w$  is 0, so this article particularly restricts: the worst( $t$ ) as the worst adaptation value in subgroup frog, and makes  $worst(t) = worst(t) - (best(t) - worst(t))/p$ . And from the equation (1), we can know that the moving step of the frog containing the best position information of subgroup. Supposing the best frog in subgroups is the  $k$ -th frog, then when we calculate the resultant get by the  $i$ -th frog of the  $t$ -th generation, coming from  $P$  frogs in subgroup in the direction of the  $d$ -th dimension according to the formula (8), we should satisfied the condition of  $1 \leq j \leq p$ , and  $j \neq i, j \neq k$ , at the same time we should let  $R_{ij}(t) = 0.001$  for the calculation.

### The update strategy of GSFLA

SFLA update strategy only uses the worst local optimum and global optimum individual information only and ignores the exchange of information among subgroups of individuals and be among groups, which makes it miss a lot of useful information; The frog is seen as a material owns the quality, so according to the principle of gravity, there is a correlation attracted between frogs, and the worst frog is attracted by resultant from the other frog within the subgroup and its changing location has a great impact on it. It is this attractive control that guide the worst frog individuals to improve constantly. In this way, it optimizes the entire frog populations. Based on this, in order to take full advantage of sharing information in a group, we put forward some new updating policies.

From (9) we can calculate the resultant  $TF_w$  of the worst frog  $X^w$ :

$$TF_w(t) = \begin{pmatrix} TF_w^1(t) \\ TF_w^2(t) \\ \mathbf{M} \\ TF_w^r(t) \end{pmatrix} \quad (9)$$

So the new updating policies is:

$$X^{w'} = X^w + D^j + rand(1) * TF_w \quad (10)$$

The new updating policies increase the diversity of the population. The worst frog forced by the resultant of other frogs which changes with the number of iterations. In early iterations, the convergence accelerates and continuously expands the search coverage, In late iterations, the careful search will be happened in the excellent neighborhood of the solution domain, and it improves the convergence precision by combining the expanding and exploring ways.

**GSFLA algorithm.**

The specific algorithm steps of GSFLA are as follows:

1) Group initialization and parameter settings: population size Popsizе=N, dimension VectorSize=r, number of subgroups TeamSize=s, subgroups of individuals p, the upper Lb and lower bounds Ub, subgroup update times TeamIter, the maximum number of iterations MaxIter, the current generation t.

2) Calculate the applicable values of each frog (objective function value), mark as  $fit_i(t)$ .

3) Sort these N frog according to objective function value in descending order, and then line the sorted frog by  $j = 1, i$  ordered from 1 to s; then the  $j = 2, i$  ordered from 1 to s; and so on until it's averaged divided into s subgroups. Write down the overall best individual  $X^g(t)$  and the best individual  $X^b(t)$  and worst  $X^w(t)$  of each subgroup.

4) Calculate the resultant  $TF^w$  forcing the worst frog  $X^w$  by the formula (9).

5) Updating the worst frog according to equation (10) partial within the specified number in Subgroups.

6) Mix subgroups mixed into a new group, and update the best individual.

7) Algorithm termination condition judgment: if it does not satisfy  $t = t + 1$ , jump to 3); otherwise, the algorithm ends.

**Numerical experiment results and analysis**

**Test function and conditions**

We selected six standard test functions, with the basic SFLA (SFLA) and improved algorithm based on average for SFLA (noted as SFLA-AV) for comparison, to verify the validity of GSFLA.

Tab.1 Test function

Function	Expression	Theoretical optimal value	Interval
f1	$f(x) = \sum_{j=1}^5 x_j^2$	0	[-5,5]
F2	$f(x) = \sum_{i=1}^5 [x_i^2 - 10 \cos(2\pi x_i) + 10]$	0	[-10,10]
F3	$f(x) = \sum_{i=1}^4 (x_i^2 + x_{i+1}^2)^{0.25} [\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1]$	0	[-4,4]
F4	$f(x) = \frac{1}{4000} \sum_{i=1}^5 x_i^2 - \prod_{i=1}^5 \cos(\frac{x_i}{\sqrt{i}}) + 1$	0	[-30,30]

In the Tab.1, the f1 is the Sphere Model, f2 is Rastrigin Function, f3 is Schaffer1 Function, and f4 is Griewank Function.

Parameters settings: N = 200, dimension VectorSize=5, S = 20, TeamIter=10, MaxIter=100. Each function run 30 times for test independently, and write down the experiment results.

Tab.2 Computational results

Function	Algorithm	Parameters value	The average optimal value	Best value	Target Precision	Success rate
f1	SFLA	$G_0 = 3$	3.41024E-05	2.64E-07	$1 \times 10^{-37}$	0
	SFLA-AV	$a = 4$	3.60068E-07	3.44E-21		0
	GSFLA	$e = 0.001$	9.36162E-37	2.37E-42		100%
F2	SFLA	$G_0 = 0.5$	0.004052394	1.07E-05	$1 \times 10^{-36}$	0
	SFLA-AV	$a = 4$	2.61192E-05	0		3%
	GSFLA	$e = 0.001$	0	0		100%
F3	SFLA	$G_0 = 0.7$	0.306463333	0.1569	$1 \times 10^{-6}$	0
	SFLA-AV	$a = 3.5$	0.128026667	0.0107		0
	GSFLA	$e = 0.001$	0.000292777	4.62E-08		70%
F4	SFLA	$G_0 = 2$	3.19102E-07	2.04E-07	$1 \times 10^{-12}$	0
	SFLA-AV	$a = 4$	3.23236E-11	3.23E-11		0
	GSFLA	$e = 0.001$	0	0		100%

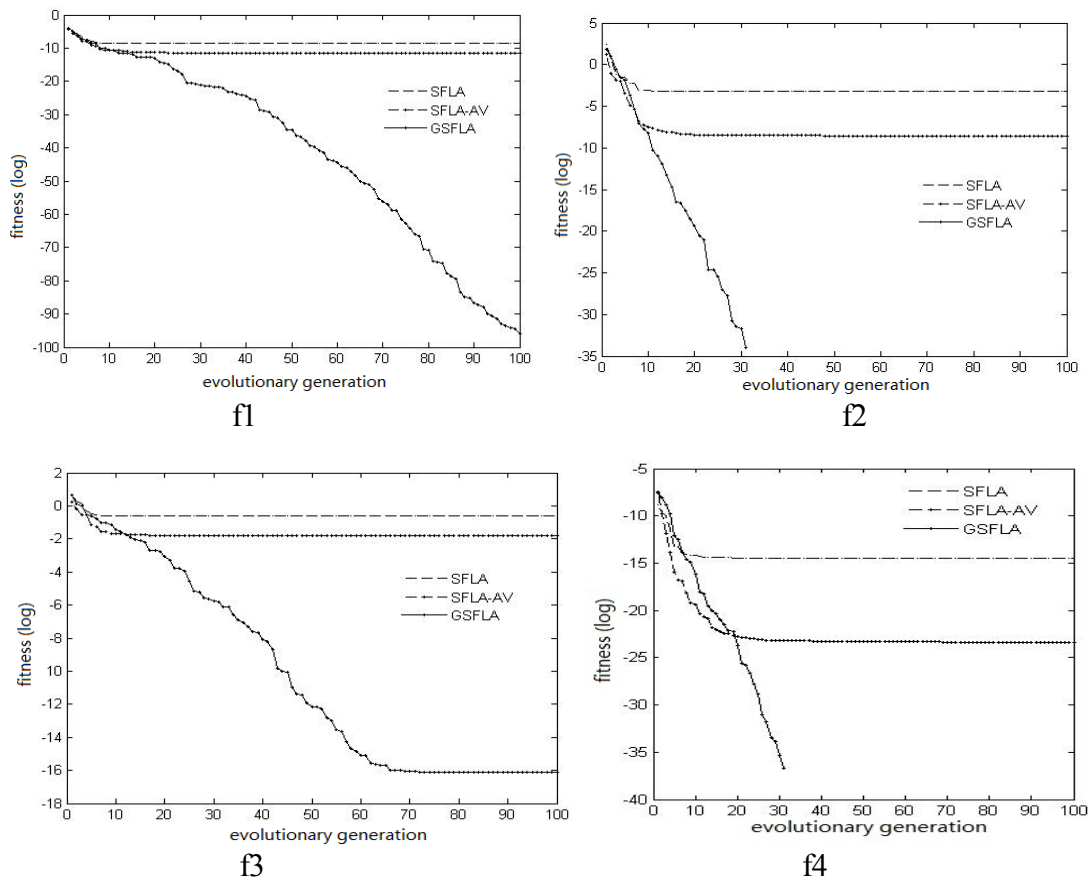


Fig.1 convergence curve of f1-f4

### **Analysis of experiment results**

From Table 2, we can see that in the f1, f2, f3, f4 functions, compared them SFLA. We can know that the solving accuracy of GSFL have increased by  $1 \times 10^{35}$  times,  $1 \times 10^{\infty}$  times,  $1 \times 10^4$  times, and  $1 \times 10^{\infty}$  times each. In addition, from Table 2 we can see that in the function f2, f4, GSFLA can always find the theoretical optimal solution. And it finally shows that the GSFLA have better optimization ability than SFLA and SFLA-AV significantly.

From the evolution of the curve f1-f4, GSFLA shows a better ability of finding optimal solution than SFLA SFLA-AV in the early time. And it can fully use effective information for communication and co-evolution and it will be optimized constantly as the number of iterations increases. While in the early time SFLA and SFLA-AV has fall into the trap of local optimum. Finally it shows that GSFLA has a better effect and obviously has a more local search ability and local optimum out of local. And it has the advantage to constantly find the larger range of solutions and search finely in the local site.

### **Summary**

SFLA is a new intelligent bionic optimization algorithm. In this paper, by using the idea of gravity algorithm, making full use of effective information about the interaction and mutual influence among frog populations and then improve the updating policy. The new algorithm combines the advantages of opening up and fine search in each iteration, so that the new algorithm has the better performance. And the simulation results can show the effectiveness and stability of GSFLA.

### **Acknowledgement**

This work was financially supported by the National Natural Science Foundation (61202269), College students' Innovative Training Program of Guangdong (1184513259) and College students' Entrepreneurship Training Program (201411845033).

### **References**

- [1] Juan Lin, YiWen Zhong, ShenLin Ma. Improved opposition-based shuffled frog leaping algorithm for function optimization problems[J].Application Research of Computers, 2013, 30(3):760-763.
- [2] YongQiang Dai, LianGuo Wang. A Shuffled Frog Leaping Algorithm With Memory Function[J].Computer Engineering and Design, 2011, 37(9):3170-3175:
- [3] Lei Song, LiangGuo Wang, YouHua Zhang. Novel shuffled frog leaping algorithm based on average value[J].JOURNAL OF GANSU AGRICULTURAL UNIVERSITY, 2014, 49(3):176-180.
- [4] XueHui Luo, Ye Yang, Xia Li. Modified shuffled frog-leaping algorithm to solve traveling salesman problem[J].Journal on Communications, 2009, 30(7):130-135
- [5] JianPing Luo, Xia Li. For TSP improved by shuffled frog-leaping algorithm[J].JOURNAL OF SHENZHEN UNIERSITY SCIENCE AND ENGIEERIGVNN, 2010, 27(2):173-179
- [6] JinRong Huang, Yun Li, Haitao Liu. A Hybr id genetic an t colony algorithm for travel in GSA lesman problem[J].Computer Applications, 2008, 28(8):2084-2087.
- [7] Zhen Zeng, Lin Lu, Yuanqing Wang. Enhancement of low-light-level image based on FSFLA algorithm[J].Infrared and Laser Engineering, 2014, 43(7):2318-2323:
- [8] GuoYan Wang, XueSong Chu, MingWu Dou. Frog Leaping Algorithm for the Critical Slip Surfaces of Slopes[J].YELLOW RIVER, 2012, 34(1):128-130