International Journal of
Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: http://dx.doi.org/10.7324/IJASRE.2017.32518

Vol.3 (9) Oct - 2017

Improved Shuffled Frog Leaping Algorithm

Kanika Sharma, R S Sharma

Rajasthan Technical University

Kota, India

ABSTRACT

Shuffled frog leaping algorithm is a memetic metaheuristic and population based intelligent inquiry metaphor impacted by normal memetics. Predominantly SFLA has been utilized for arrangement of combinative streamlining issues. In SFL algorithm there are two basic things one is population which is divided into several memeplexes and another is information between these memeplexes has been exchanged. In Shuffled frog leaping algorithm one issue is available which is slow convergence. To resolve this problem Improved shuffled frog leaping algorithm is proposed. In both phase local best and global best phase (1- it/Maxit) term is multiplied to get better solution. The proposed strategy is tested more than 12 benchmark functions and com- pared with different algorithms like basic shuffled frog leaping algorithm (SFLA), gravitational search algorithm (GSA), spider monkey optimization (SMO) and differential evolution (DE).

Keywords: Shuffled Frog Leaping Algorithm, Memetic, Memeplex.

I. INTRODUCTION

Nature-inspired algorithms (NIAs) are algorithms, which takes acutation from nature. These type of algorithms are used to resolve divergent complex real world problems, whose definite clarification doesn't exist [1]. Swarm intelligence based algorithms are basically based on swarms e.g. particle swarm optimization (PSO), artificial bee colony algorithm (ABC), shuffled frog leaping algorithm (SFLA) and bacterial foraging algorithm (BFO) etc [2]. Shuffled Frog Leaping Algorithm is a swarm intelligence based algorithm. In SFLA frogs are divided into meme that inspired from the foraging behaviour of frogs [3]. In SFLA, population (frogs) is divided into several memeplexes. In SFLA frogs exchange their memes with other frogs by using memetic evolution procedure. That memetic evolution procedure helps to improve the performance of individual frog towards its global optimum solution. Basic SFLA coincide slowly at the last stage and easily falls into local minima [4]. So for improvement in basic SFLA, Improved shuffled frog leaping algorithm is originated [5]. In Improved algorithm worst solution is updated through local best solution, global best solution or random initialization in search space [6]. The remaining paper is organized as shown: In section II, a brief overview of basic SFLA is mentioned. In section III Improved shuffled frog leaping algorithm is revealed [7]. In section IV, performance of ISFLA is tested with several benchmark functions. In section V, conclude the whole work.

II.BASIC SHUFFLED FROG LEAPING ALGORITHM

Shuffled Frog Leaping Algorithm is right off the bat created by Eusuff and Lansey in 2003. Shuffled Frog Leaping Algorithm is mainly used for solving combinatorial optimization problems [8]. SFLA is a population based cooperative search metaphor which is inspired as a result of foraging behaviour of frogs [9]. In SFLA worst solution is updated by local best solution, global best solution or frogs are randomly initialize in search space [10]. A shuffling is a good approach which is used for exchanging thoughts among local searchers with the purpose of leads them toward a global optimum [11]. A pseudo code of shuffled Frog Leaping Algorithm is described in algorithm is as follows :

1: Firstly, we set initial values such as size of the population (frogs) N, the number of memeplexes M, the number frogs in each memeplexes F and maximum number of iterations;
2: For each individual frog, calculate objective value and sort the population N in the descending order of their objective value;
3: After this, we divide N population into M memeplexes;
4: for each memeplex: do
5: Calculate triangular probability distribution using Eq.(1);

$$\text{Prob}(j)=2(n+1-j)/(n(n+1)) \qquad (1)$$

International Journal of
Advances in Scientific Research and Engineering (ijasre)
E-ISSN : 2454-8006
DOI: http://dx.doi.org/10.7324/IJASRE.2017. Vol.3 (9) Oct - 2017

where j = 1, ..., n, represents rank of frogs within the memeplex, n is the total population of the swarm.
6: Sort the frogs in the descending order of their probability and elect the best and worst frogs;
7: After this, improve the worst frog position using Eq.(2) with respect local best frog;

$$U_{new} = P_W + R(0,1) \times (P_{LB} - P_W) \qquad (2)$$

8: if $U_{new} < P_w$ then;
9: $P_{new} = U_{new}$
10: else
11: If worst frog position is not improved by local best frog then we update worst frog position by using Eq.(3)with respect global best frog;

$$U_{new} = P_W + R(0,1) \times (P_{GB} - P_W) \qquad (3)$$

12: if $U_{new} < P_w$ then
13: $P_w = U_{new}$
14: else
15: censorship = true
16: end if
17: end if
18: Repeat for a specific number of iteration;
19: end for
20: Combine the evolved memeplexes and sort the population N according to their objective value
21: Check if termination condition is true then stop, otherwise partition the frogs into the memeplexes;

III.IMPROVED SHUFFLED FROG LEAPING ALGORITHM(ISFLA)

In Shuffled Frog Leaping Algoritm (SFLA), there is problem of abundant probability present for the solutions to get stuck and slow convergence of population. In Improved Shuffled Frog Leaping Algorihm (ISFLA), for updation of the worst solution we use three ways:
(1) Learning through the local best solution
(2) Learning through the global best solution and
(3) Randomly initialization of solution in the search space.
In each phase of position update process the step size is calculated with respect to local best solution or global best solution or it is randomly initialize in the search space. But one problem is present with three strategies, when we use these three strategies may generated the high step size, this high step size leads to the situation of skipping true optima.This type of search phenomenon enhance the exploration capability of the algorithm but reduce the exploitation. One condition is that for an efficient solution strategy exploration and exploitation should be balanced. Therefore to improve the convergence and exploitation capability of SFLA, following modifications are proposed.
(1). Learning through Local Best Solution-
In this phase the position of worst solution is updated. In the process of local search, the worst solution is get updated either by taking motivation from local best solution or retain the previous knowledge of the solution. The updated step size and position update equations is defined as equations (4 and 5).

$$Step = (R(0,1) * (P_{LB} - P_W)) \times (1 - it/Maxit) \qquad (4)$$

$$U_{new} = P_W + Step \qquad (5)$$

here, Unew is the updated position of worst solution and step shows the step size. (PLB) represents the local best solution. R(0,1)is a uniformly distributed random number in the range between [0,1], it(iteration) shows current iteration and Maxit shows maximum number of iteration. If Unew lies in the feasible space, compute the new objective value. Greedy selection strategy is applied for improving the position of worst solution in the search space. If the position of worst solution gets better than the previous position then the position of worst solution is updated otherwise it goes to next phase. Algorithm 1 shows position update procedure in local best learning phase.

International Journal of
Advances in Scientific Research and Engineering (ijasre)
E-ISSN : 2454-8006
DOI: http://dx.doi.org/10.7324/IJASRE.2017. Vol.3 (9) Oct - 2017

(2). Learning through Global Best Solution-

In this phase chance the position of worst frog is updated by taking motivation from global best solution or retain the previous knowledge. The position update process of worst solution is defined as equations (6 and 7).

$$Step = (R(0,1) * (P_{GB} - P_W)) \times (1 - it/Maxit) \quad (6)$$

$$U_{new} = P_W + Step \quad (7)$$

here Unew is the updated position of worst solution and step shows the step size. (PGB) represents the global best solution.

R(0,1)is a uniformly distributed random number in the range between [0,1], it (iteration) shows current iteration and Maxit shows maximum number of iteration.

(3).Randomly initialization of solution in the search space (Censorship)-

If new position of worst solution is infeasible means worst solution exist outside the range of search space and old position, which is calculated by global best solution is not better. Meme of this frog not spread no longer it means that worst frog does not have good meme so, generate a new solution randomly within the range of feasible search space to replace the frog whose new position was not so good to evolution.

Like SFLA, the ISFLA algorithm is also divided into two phases, namely global exploration phase and local exploration phase. ISFLA algorithm is described below:

The flowchart of ISFLA is described in Fig. 1.

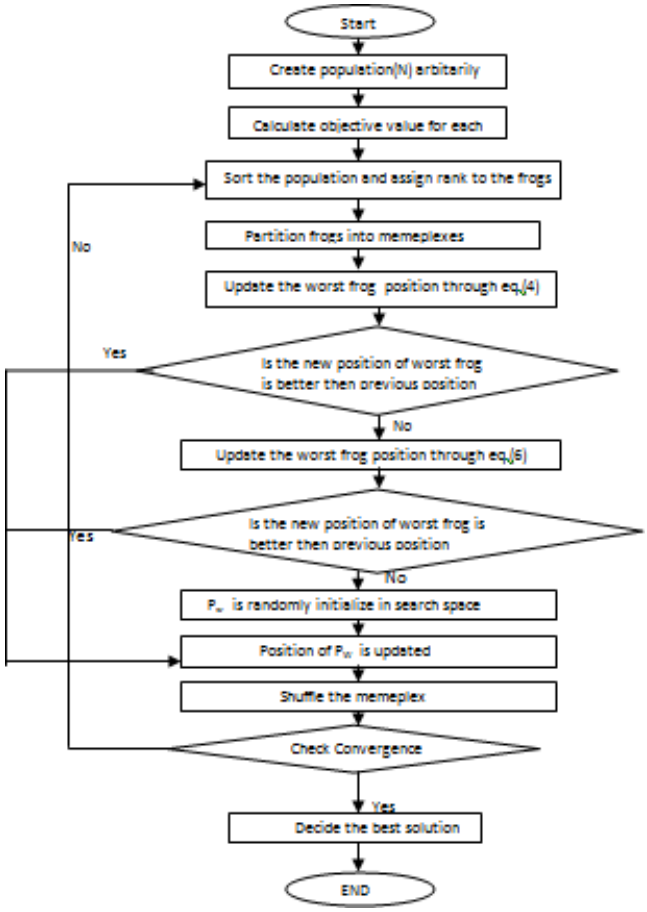

Fig. 1: ISFLA Flowchart

A.*Experimental setting*

To verify the improvement of the proposed algorithm ISFLA, a relative estimation is done among ISFLA, GSA, SMO and DE.The following experimental setting is used:

International Journal of
Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: http://dx.doi.org/10.7324/IJASRE.2017.

Vol.3 (9) Oct - 2017

TABLE I: Test problems,S:Dimension,AE:Acceptable Error

Test Problem	Objective function	Search Range	Optimum Value	S	AE
Sphere	$f_1(d) = \sum_{i=1}^{S} d_i^2$	[-5.12, 5.12]	$f(0) = 0$	30	$1.0E-05$
De Jong f4	$f_2(d) = \sum_{i=1}^{D} i.(d_i)^4$	[-5.12, 5.12]	$f(\vec{0}) = 0$	30	$1.0E-05$
Griewank	$f_3(d) = 1 + \frac{1}{4000}\sum_{i=1}^{D} d_i^2 - \prod_{i=1}^{D}\cos(\frac{d_i}{\sqrt{i}})$	[-600, 600]	$f(\vec{0}) = 0$	30	$1.0E-05$
Ackley	$f_4(d) = -20 + e + exp(-\frac{0.2}{S}\sqrt{\sum_{i=1}^{S} {d_i}^3})$	[-30, 30]	$f(0) = 0$	30	$1.0E-05$
Exponential	$f_5(d) = -(exp(-0.5\sum_{i=1}^{S} {d_i}^2)) + 1$	[-1, 1]	$f(0) = -1$	30	$1.0E-05$
brown3	$f_6(d) = \sum_{i=1}^{S-1}({d_i}^{2({d_{i+1}}+1)^2+1} + {d_{i+1}}^{2{d_i}^2+1})$	[-1, 4]	$f(0) = 0$	30	$1.0E-05$
Axis parallel hyper-ellipsoid	$f_7(d) = \sum_{i=1}^{S} i.d_i^2$	[-5.12, 5.12]	$f(0) = 0$	30	$1.0E-05$
Sum of different powers	$f_8(d) = \sum_{i=1}^{D} \lvert d_i\rvert^{i+1}$	[-1 1]	$f(\vec{0}) = 0$	30	$1.0E-05$
Rotated hyper-ellipsoid	$f_9(d) = \sum_{i=1}^{S}\sum_{j=1}^{i} d_j^2$	[-65.536, 65.536]	$f(0) = 0$	30	$1.0E-05$
Levy montalvo 2	$f_{10}(d) = 0.1(\sin^2(3\Pi d_1) + \sum_{i=1}^{D-1}(d_i-1)^2 \times (1+\sin^2(3\Pi d_{i+1})) + (d_D-1)^2(1+\sin^2(2\Pi d_D))$	[-5, 5]	$f(1) = 0$	30	$1.0E-05$
Ellipsoidal	$f_{11}(d) = \sum_{i=1}^{D}(d_i - i)^2$	[-D, D]	$f(1, 2, 3, ..., D) = 0$	30	$1.0E-05$
Moved axis parallel hyper-ellipsoid	$f_{12}(d) = \sum_{i=1}^{D} 5i \times d_i^2$	[-5.12, 5.12]	$f(d) = 0; d(i) = 5 \times i, i = 1 : D$	30	$1.0E-12$

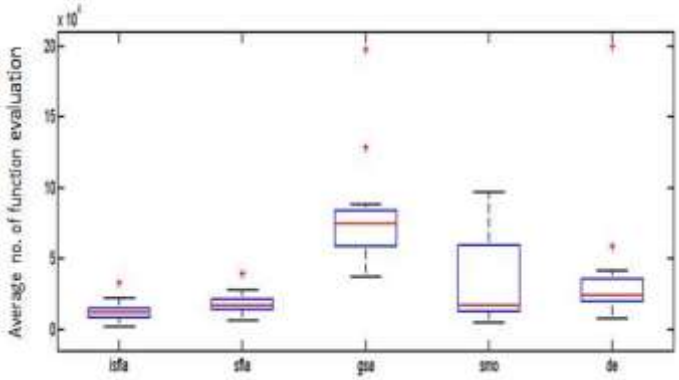

Fig. 2: Boxplots graphs (Average number of function evaluation)

TABLE II: Comparison results of test Problem, TP:

TP	Algorithm	SD	ME	AFE	SR
	ISFLA	9.832E-07	8.971E-06	9922.633	30
f_1	GSA	8.016E-07	8.996E-06	76470.000	30
	SMO	9.200E-07	8.760E-06	12563.100	30
	DE	7.410E-07	9.091E-06	22601.666	30
	ISFLA	1.493E-06	8.099E-06	7470.566	30
f_2	GSA	1.321E-06	8.400E-06	50305.000	30
	SMO	1.090E-06	8.560E-06	10672.200	30
	DE	1.157E-06	8.582E-06	18770.000	30
	ISFLA	7.302E-07	9.081E-06	11672.900	30
f_3	GSA	7.376E-07	8.929E-06	37148.333	30
	SMO	1.840E-03	5.200E-04	85981.600	27
	DE	1.033E-16	0.758	200000.000	0
	ISFLA	5.521E-07	9.271E-06	22345.333	30
f_4	GSA	4.350E-07	9.446E-06	128835.000	30

International Journal of
Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: http://dx.doi.org/10.7324/IJASRE.2017.

Vol.3 (9) Oct - 2017

	SMO	1.440E-06	9.010E-06	97122.233	30
	DE	4.282E-07	9.493E-06	42043.333	30
	ISFLA	6.825E-07	9.132E-06	6870.166	30
f_5	GSA	8.889E-07	8.974E-06	73110.000	30
	SMO	9.660E-07	8.950E-06	86265.233	30
	DE	6.914E-07	9.024E-06	17165.000	30
	ISFLA	8.428E-07	8.948E-06	12505.233	30
f_6	GSA	1.019E-06	8.736E-06	79390.000	30
	SMO	1.070E-06	8.800E-06	12635.700	30
	DE	6.871E-07	9.208E-06	22221.666	30
	ISFLA	5.265E-07	9.190E-06	12347.800	30
f_7	GSA	7.297E-07	9.159E-06	88296.666	30
	SMO	6.140E-07	8.930E-06	14770.800	30
	DE	7.810E-07	9.117E-06	25906.666	30
	ISFLA	1.990E-06	7.371E-06	2145.733	30
f_8	GSA	2.518E-06	6.965E-06	37470.000	30
	SMO	1.730E-06	7.980E-06	5181.000	30
	DE	2.278E-06	7.258E-06	7665.000	30
	ISFLA	0.179	9.069E-06	14260.133	30
f_9	GSA	9.581E-07	8.470E-06	76086.666	30
	SMO	8.300E-07	8.940E-06	18819.900	30
	DE	9.633E-07	8.954E-06	29205.000	30
	ISFLA	7.186E-07	9.182E-06	10780.400	30
f_{10}	GSA	1.053E-06	8.635E-06	70403.333	30
	SMO	1.800E-06	8.840E-06	24788.033	30
	DE	1.015E-06	8.877E-06	23088.333	30
	ISFLA	8.089E-07	9.131E-06	15740.966	30
f_{11}	GSA	4.792E-05	0.000	67561.666	30
	SMO	7.290E-07	8.990E-06	15160.200	30
	DE	5.872E-07	9.079E-06	26216.666	30
	ISFLA	9.446E-17	8.936E-16	33325.366	30
f_{12}	GSA	6.237E-17	9.201E-16	197323.333	30
	SMO	8.240E-17	8.970E-16	34019.700	30
	DE	6.213E-17	9.081E-16	58980.000	30

- The number of runs = 30,
- Total number of iterations = 4000,
- Size of population = 50,
- Parameter perception for the algorithms GSA, SMO and DE are taken from the essential papers,

Benchmark functions f_1 to f_{12} are taken to inspect the execution of ISFLA as appeared in table I.

B. Analysis of Results

Experimental outcomes are appeared in table II for the considered algorithms. Table II shows the analysis of average number of function evaluations (AFE), mean error (ME), standard deviation (SD)and success rate (SR). Here results depict that ISFLA is better at accuracy and efficiency level in comparison of GSA, SMO and DE.

Boxplot analysis of average number of function evaluations (AFE) can also be designed for the comparison of considered algorithms ISFLA, GSA, SMO and DE [12]. The boxplot is a graphical representation of data, in which rectangle is drawn to represent the interquartile with a vertical line indicating the median value. The Fig. 2 depicted the boxplots for ISFLA, GSA [13], SMO [14] and DE [15]. The outcomes demonstrate that interquartile range and medians of ISFLA are low in the correlation of GSA, SMO and DE.

IV.CONCLUSION

In this paper, a new position update strategy for worst solution is mentioned, that is Improved shuffled frog leaping algorithm. By this new algorithm slow convergence and exploitation capability are improved. This variant algorithm is successfully tested over 12 benchmark functions with various statistical measurements. The proposed algorithm is compared to GSA, SMO and DE.

REFERENCES

[1] Parul Agarwal and Shikha Mehta. Nature-inspired al- gorithms: state-of-art, problems and prospects. *Nature*, 100(14), 2014.
[2] Jagdish Chand Bansal, Harish Sharma, Shimpi Singh Jadon, and Maurice Clerc. Spider monkey optimization algorithm

for numerical optimization. *Memetic comput- ing*, 6(1):31–47, 2014.
[3] Jagdish Chand Bansal, PK Singh, Mukesh Saraswat, Ab- hishek Verma, Shimpi Singh Jadon, and Ajith Abraham. Inertia weight strategies in particle swarm optimization. In *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, pages 633–640. IEEE, 2011.
[4] Iztok Fister Jr, Xin-She Yang, Iztok Fister, Janez Brest, and Dus̆an Fister. A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*, 2013.
[5] Chen Liu, Xinquan Lai, Jianguo Jiang, Longjie Zhon and Longbin Wang. An adaptive shuffled frog leaping algorithm JOURNAL OF INFORMATION &COMPU- TATIONAL SCIENCE, 12(17):6621–6628.
[6] Mohammad Rasoul Narimani. A new modified shuffle frog leaping algorithm for non-smooth economic dis- patch. World Applied Sciences Journal, 12(6):803–814, 2011.
[7] Mohammad Pourmahmood, Mohammd Esmaeel Akbari, and Amin Mohammadpour. An efficient modified shuffled frog leaping optimization algorithm. Int. J. Comput. Appl, 32(1):0975–8887, 2011.
[8] A Kai Qin, Vicky Ling Huang, and Ponnuthurai N Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE transactions on Evolutionary Computation, 13(2):398– 417, 2009.
[9] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. Information sciences,179(13):2232–2248, 2009.
[10] Arezoo Sarkheyli, Azlan Mohd Zain, and Safian Sharif. The role of basic, modified and hybrid shuffled frog leaping algorithm on optimization problems: a review. Soft Computing, 19(7):2011–2038, 2015.
[11] Pragya Sharma, Nirmala Sharma, and Harish Sharma. Binomial crossover embedded shuffled frog leaping algorithm. In Computing, Communication and Automation (ICCCA), 2016 International Conference on, pages 321–326. IEEE, 2016.
[12] Pragya Sharma, Nirmala Sharma, and Harish Sharma. Elitism based shuffled frog leaping algorithm. In Ad- vances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on, pages 788– 794. IEEE, 2016.
[13] Shweta Sharma, Tarun K Sharma, Millie Pant, Jitendra Rajpurohit, and Bhagyashri Naruka. Centroid mutation embedded shuffled frog-leaping algorithm. Procedia Computer Science, 46:127–134, 2015.
[14] Anurag Tripathi, Tarun K Sharma, and Vipul Singh. Be- spoke shuffled frog leaping algorithm and its engineering applications.International Journal of Intelligent Systems & Applications, 7(4), 2015.
[15] Jia Zhao and Li Lv. Two-phases learning shuffled frog leaping algorithm. International Journal of Hybrid Information Technology, 8(5):195–206, 2015.