Locally Informed Shuffled Frog Leaping Algorithm

Pragya Sharma(✉), Nirmala Sharma, and Harish Sharma

Rajasthan Technical University, Kota, India
pragya.gecj@gmail.com

Abstract. Shuffled Frog-Leaping Algorithm (SFLA) is a memetic meta-heuristic approach for solving complex optimization problems. Like other evolutionary algorithms, it may also suffer from the problem of slow convergence. To elevate the convergence property of the algorithm, locally informed search strategy is incorporated with SFLA. To improve the intensification and diversification capabilities of SFLA, locally informed search strategy is embedded by calculating the mean of local best and one randomly selected neighbour solution of memeplex while updating the position of worst solution in local best updating phase. Similarly, mean of global best and a randomly selected neighbour solution is used to improve the position of worst solution while updating the position of worst solution in global best updating phase. The proposed algorithm is named as Locally Informed Shuffled Frog-Leaping Algorithm (LISFLA). The modified algorithm LISFLA is analysed over 15 distinct benchmark test problems and compared with conventional SFLA, its recent variant, namely Binomial Crossover Embedded Shuffled Frog-Leaping Algorithm (BC-SFLA) and three other nature inspired algorithms, namely Gravitational Search Algorithm (GSA), Differential Evolution (DE) and Biogeography-Based Optimization Algorithm (BBO). The results manifest that LISFLA is an antagonist variant of SFLA.

Keywords: Meta-heuristic optimization techniques · Swarm intelligence · Shuffled frog leaping algorithm · Locally informed

1 Introduction

Nature-inspired algorithms (NIAs), that take inspiration from nature and its foundation is biological components of nature i.e. human and nature. The main objective of developing such algorithm is to solve distinct complex real world problems whose absolute solution doesn't exist and is to optimize engineering problems [1]. Swarm intelligence based algorithms [5] are based on mimicking collective behavior of natural swarm's e.g. particle swarm optimization (PSO) [2], artificial bee colony algorithm (ABC) [8], shuffled frog-leaping algorithm (SFLA) [7] and bacterial foraging algorithm (BFO) [9] etc.

K. Deep et al. (eds.), *Proceedings of Sixth International Conference on Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 546, DOI 10.1007/978-981-10-3322-3_13

SFLA takes inspiration from the grubbing behavior of frogs that replicate contagious information pattern with the natural and social behavior of species. In SFLA, population (frogs) is partitioned into several memeplexes. Frogs exchange their memes with other frogs using memetic evolution procedure which helps to improve the performance of individual frog towards its global optimum solution. There is always a presence of odds with all the evens, basic SFLA converges slowly at the last stage and easily falls into local minima. To elevate the performance of the conventional SFLA algorithm researchers are continuously working on this algorithm [6,10,15].

To improve the convergence, intensification and diversification proficiency of basic SFLA, locally informed search strategy is embedded in the local exploration phase of the conventional SFLA to ameliorate the position of the worst solution. In LISFLA, worst solution is take good memes either from local best and one local random solution of the memeplex or global best and a randomly chosen neighbour solution of the entire feasible search space. In this process, worst solution is locally informed through the global best or local best with one randomly selected neighbour to ameliorate the knowledge of worst solution of memeplex. The contemplated algorithm is titled as Locally Informed Shuffled Frog Leaping Algorithm (LISFLA).

The remaining paper is organized as shown: In Sect. 2, a brief overview of standard SFLA is described. Locally Informed Shuffled Frog-Leaping Algorithm (LISFLA) is proposed in Sect. 3. In Sect. 4, performance of LISFLA is tested with several numerical benchmark functions. Finally, Sect. 5 conclude the work.

2 Overview of Shuffled Frog-Leaping Algorithm

Eusuff invented Shuffled Frog Leaping Algorithm (SFLA) in 2003 [7] for solving distinct complex optimization problems. SFLA is a population-based cooperative search metaphor inspired by foraging behaviour of frogs [13]. Memetic evolution is used in SFLA for the purpose of spreading ideas among the solutions in a local exploration which is same as PSO [2]. A shuffling approach helps for exchanging ideas among local searchers that lead them toward a global optimum. SFLA contains elements of global exploration, local exploration and shuffling procedures. In general, a SFLA works as follows. Firstly, the parameters for the SFLA are total number of memeplex (Mmpx), the number of frogs in each memeplex (Fm) and the range of feasible search space are initialized. Therefore, the total population size (N) of swarm is denoted as N = Mmpx * Fm. Afterwards, objective value of each frog is calculated. Rank is assigned accordingly to their objective value and sort them in the descending order of their objective values. Then, N frogs are partitioned into memeplexes (M), each containing frogs (F), like that first rank frog goes to first memepelex, second rank frog goes to second memeplex and third rank frog goes to third memeplex and so on. To construct submemeplex, memplexes are divided into the submemeplex with having the goal is that true solution to move towards its optimum solution by elevating their ideas. Submemeplex selection process assigns weights to frogs. The weights are assigned with a triangular probability distribution ($prob_k$) using Eq. 1.

$$prob_k = \frac{2(q+1-k)}{q(q+1)} \tag{1}$$

Here, q is total number of population and $k = 1, 2, 3, ..., q$, represents rank of frogs within the memeplex. The frog with the higher objective value has the higher probability of being selected for the submemeplex. The frogs with the lower objective value has lower probability. The position of best frogs and worst frogs is recorded.

The worst solution is updated their position by using three phases: (1) Local best updating phase (2) Global best updating phase and (3) Randomly initialization of solution in the search space (Censorship).

1. **Local best updating phase**: To improve the position of worst frog. The position update equation for worst solution is defined in Eq. 2.

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

   Here, U_{new} is the new position of worst frog, P_{LB} and P_W are position of local best frog and worst frog respectively. $R(0,1)$ is a random number in the range [0, 1]. If U_{new} lies in the feasible space, compute the new objective value. Greedy selection strategy is applied for improving the position of worst solution. If the position of worst solution gets better than the previous position then position is updated otherwise it goes in next phase i.e. global best updating phase.
2. **Global best updating phase**: In this phase, the worst frog get chance to update its position with the help of global best frog as defined in Eq. 3.

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

   Here, P_{GB} represents the global best frog found so far. Again greedy selection strategy is applied for improving the position of worst solution. If worst solution does not update its position then it is randomly initialized in the feasible search space. After this phase memeplexes are updated with the new position of worst frog solution.
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, randomly generate a new frog within the range of feasible search space to replace the frog whose new position was not so good to progress.

After the memetic evolutionary steps within each memeplexes are to be shuffled and the population is to be sorted in decreasing order of their objective value. Position of the best frog P_{GB} is get updated. To check convergence, repeat the above procedure until the stopping criteria is met.

3 Locally Informed Shuffled Frog Leaping Algorithm (LISFLA)

In the working of SFLA algorithm, there are enough possibilities for the solutions to get stuck in local optima. It also suffers from the problem of slow convergence. To reduce such problems, locally informed search process is incorporated by taking mean of both local best and one randomly selected local neighbour of memeplex and global best and one random neighbour for improving the position of the worst solution in local best updating phase and global best updating phase of basic SFLA respectively.

As it is clear from the solution's search process of conventional SFLA that the Eqs. (2, 3), the worst solution is updated during each iteration by using three phases: (1) Local best updating phase (2) Global best updating phase and (3) Randomly initialization of solution in the search space. Further, it is to be noted that worst solution is simply influenced by the local best or global best solution, which may lead to trap in local optima and leads to loss of intensification and diversification capability.

To avoid such possibilities (stagnation or converging in local optima), in the proposed strategy, the step size is calculated by taking the mean of both local best or global best solution and one randomly selected neighbour solution of memeplex otherwise it is randomly initialized in the search space. This type of search phenomenon elevates the intensification and diversification proficiency of the algorithm that are chief characteristics of the population-based optimization algorithms. Therefore, to improve the convergence and to maintain the intensification and diversification capability of SFLA, following modifications are proposed.

3.1 Local Best Learning Phase with Random Neighbour

The position of worst solution is updated using the locally informed search process. In the process of local search, the worst solution is get updated (informed) by taking mean of both local best solution and one randomly selected local neighbour solution of particular memeplex . The updated step size and position update equations is defined as Eqs. (4 and 5).

$$Step = R(0,1) * (\frac{P_{LB} + P_{KL}}{2} - P_W) \tag{4}$$

$$U_{new} = P_W + Step \tag{5}$$

Here, U_{new} is the updated position of the worst solution and $Step$ shows the step size. (P_{LB}) and (P_{KL}) represents the local best solution and a local neighbour solution of P_W respectively. $R(0,1)$ is a uniformly distributed random number in the range between [0, 1]. If U_{new} 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 the worst solution is updated otherwise it goes to next phase. In the modified equation a new term is added that is containing the information received from the local best solution and a random neighbour solution of memeplex. In Eqs. (4, 5), worst solution P_W is informed through the both both P_{LB} and P_{KL}. In basic SFLA Eq. (2), sometimes the worst solution (P_W) are not updated through the local best solution that leads to loss of intensification and diversification proficiency of the search space. Therefore, in Eqs. (4, 5) to improve the position of worst solution mean of both P_{LB} and P_{KL} are taken that leads to move P_W toward its optimum solution.

3.2 Global Best Learning Phase with Random Neighbour

In this phase the worst solution is get update its position by taking mean of both global best and a random neighbour solutions of the feasible search space. The position update process of worst solution is defined as Eqs. (6 and 7).

$$Step = R(0,1) * (\frac{P_{GB} + P_{KG}}{2} - P_W) \tag{6}$$

$$U_{new} = P_W + Step \tag{7}$$

Here, U_{new} is the updated position of worst solution and $Step$ shows the step size. P_{GB} and P_{KG} represents the global best solution and neighbour solution of P_W respectively. If U_{new} is exist in the feasible space, compute its objective value and apply greedy selection strategy between the new and worst solution. If worst solution does not update its position then it is randomly initialized in the feasible search space. In Eqs. (6, 7) both P_{GB} and P_{KG} has the ability to attract the worst solution P_W. In the modified equation a new term is added that is containing the mean of the information received from the global best solution and a randomly selected neighbour solution of the search space. According to Eqs. (6, 7), in place of only global best solution, mean of both randomly chosen neighbour solution and global best solution is used that commute the worst solution in the direction of the global best solution with neighbour solution that enhance the intensification and diversification potential of SFLA algorithm.

3.3 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 inplace the frog whose new position was not so good to evolution.

After the memetic evolutionary steps within each memeplex, the memeplexes are to be shuffled and the population is to be sorted in decreasing order of their objective value. Position of best frog P_{GB} is get updated. Then we check the stopping criteria of algorithm if it is satisfied then stop the process. Otherwise, again partition the frogs into memeplexes.

Like SFLA, the LISFLA algorithm is also divided into two phases, namely global exploration phase and local exploration phase. The locally informed search strategy is embedded in the local exploration phase of the algorithm, while global exploration phases are kept same as in the basic SFLA.

4 Results and Discussions

To analyze the validity of LISFLA algorithm, 15 distinct global optimization functions (f_1 to f_{15}) are used here, demonstrated in Table 1.

To accredit the pursuance of the proposed algorithm LISFLA, a comparative experiment is carried out among LISFLA, SFLA [7], BC-SFLA, GSA [11], DE and BBO [14]. LISFLA is tested with the basic SFLA, BC-SFLA, BBO, DE and GSA over considered optimization test functions. Following experimental parameters are adopted:

- The number of simulations/run = 30,
- Total number of memeplexes ($Mmpx$) = 5
- Number of frogs in each memeplex (Fm) = 10 and total population Size (Mmpx * Fm) (N) = 50
- Parameter settings for the algorithms basic SFLA, BC-SFLA, GSA, DE and BBO are imitated from their elementary research papers. [7,11,14]

Table 2 shows the experimental results of the SFLA, BC-SFLA, GSA, DE and BBO algorithms and also furnishes about the standard deviation (SD), average number of function evaluations (AFE), mean error (ME), and success rate (SR). Results in Table 2 reflects that most of the time LISFLA outrun in terms of reliability, robustness, efficiency as well as accuracy in comparison to the SFLA, BC-SFLA, GSA, DE and BBO.

Besides, boxplots [3,12] analysis is carried out for comparing the examined algorithms in the form of combined performance though it can efficiently depict

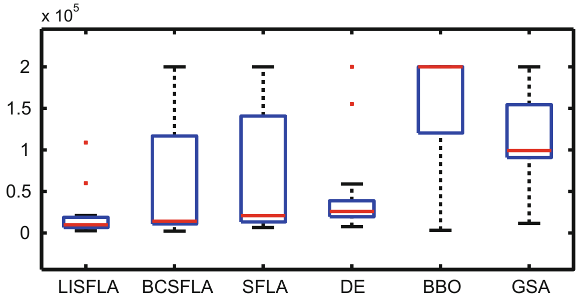

Fig. 1. Boxplots graphs (Average Function Evaluation)

Table 1. TP: Test Problems, OF: Objective Function, SR: Search Range, OV: Optimum Value, D: Dimension, AE: Acceptable Error for LISFLA

Test problem	Objective function	Search range	Optimum value	D	AE
De Jong f4	$f_1(x) = \sum_{i=1}^{n} i.(x_i)^4$	$[-5.12\ ,\ 5.12]$	$f(0) = 0$	30	$1.00E-05$
Griewank	$f_2(x) = 1 + \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})$	$[-600\ ,\ 600]$	$f(0) = 0$	30	$1.0E-05$
Ackley	$f_3(x) = -20 + e + exp(-\frac{0.2}{D}\sqrt{\sum_{i=1}^{D} {x_i}^3}) - exp(\frac{1}{D}\sum_{i=1}^{D}\cos\ (2\pi.x_i)x_i)$	$[-1\ ,\ 1]$	$f(0) = 0$	30	$1.0E-05$
Cosine mixture	$f_4(x) = \sum_{i=1}^{D} {x_i}^2 - 0.1(\sum_{i=1}^{D}\cos\ 5\pi x_i) + 0.1D$	$[-1\ ,\ 1]$	$f(0) = -1 \times 0.1$	30	$1.0E-05$
Exponential	$f_5(x) = -(exp(-0.5\sum_{i=1}^{D} {x_i}^2)) + 1$	$[-1\ ,\ 1]$	$f(0) = -1$	30	$1.0E-05$
brown3	$f_6(x) = \sum_{i=1}^{D-1}({x_i}^{2(x_{i+1})^2+1} + {x_{i+1}}^{2{x_i}^2+1})$	$[-1\ ,\ 4]$	$f(0) = 0$	30	$1.0E-05$
Salomon problem	$f_7(x) = 1 - cos(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}) + 0.1(\sqrt{\sum_{i=1}^{D} x_i^2})$	$[-100\ ,\ 100]$	$f(0) = 0$	30	$1.0E-01$
Axis parallel hyper-ellipsoid	$f_8(x) = \sum_{i=1}^{D} i.x_i^2$	$[-5.12\ ,\ 5.12]$	$f(0) = 0$	30	$1.0E-05$
Sum of different powers	$f_9(x) = \sum_{i=1}^{D} \lvert x_i\rvert^{i+1}$	$[-1\ ,\ 1]$	$f(0) = 0$	30	$1.0E-05$
Step function	$f_{10}(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5\rfloor)^2$	$[-100\ ,\ 100]$	$f(-0.5 \le x \le 0.5) = 0$	30	$1.0E-05$
Rotated hyper-ellipsoid	$f_{11}(x) = \sum_{i=1}^{D}\sum_{j=1}^{i} x_j^2$	$[-65.536\ ,\ 65.536]$	$f(0) = 0$	30	$1.0E-05$
Levy montalvo 1	$f_{12}(x) = \frac{\Pi}{D}(10\sin^2(\Pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2 \times (1 + 10\sin^2(\Pi y_{i+1})) + (y_D - 1)^2)$, where $y_i = 1 + \frac{1}{4}(x_i + 1)$	$[-10\ ,\ 10]$	$f(-1) = 0$	30	$1.0E-05$
Levy montalvo 2	$f_{13}(x) = 0.1(\sin^2(3\Pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2 \times (1 + \sin^2(3\Pi x_{i+1})) + (x_D - 1)^2(1 + \sin^2(2\Pi x_D))$	$[-5\ ,\ 5]$	$f(1) = 0$	30	$1.0E-05$
Ellipsoidal	$f_{14}(x) = \sum_{i=1}^{D} (x_i - i)^2$	$[-30\ ,\ 30]$	$f(1, 2, 3, ..., D) = 0$	30	$1.0E-05$
Moved axis parallel hyper-ellipsoid	$f_{15}(x) = \sum_{i=1}^{D} 5i \times x_i^2$	$[-5.12\ ,\ 5.12]$	$f(x) = 0; x(i) = 5 \times i, i = 1 : D$	30	$1.0E-15$

Table 2. Comparative result of TP, TP: Test Problem for LISFLA

TP	Algorithm	SD	ME	AFE	SR
f_1	LISFLA	9.80E−07	8.32E−06	5999.47	30
	BCSFLA	9.67E−07	8.24E−06	8546.30	30
	SFLA	1.12E−06	8.73E−06	12333.10	30
	DE	1.16E−06	8.58E−06	18770.00	30
	BBO	8.01E−07	8.98E−06	46193.33	30
	GSA	1.34E−06	8.03E−06	63223.33	30
f_2	LISFLA	1.03E−06	8.85E−06	8502.20	30
	BCSFLA	1.77E−02	8.05E−01	200000.00	0
	SFLA	2.19E−02	8.25E−01	200000.00	0
	DE	1.03E−16	7.59E−01	200000.00	0
	BBO	1.13E−03	7.63E−01	200000.00	0
	GSA	3.21E−03	7.60E−01	200000.00	0
f_3	LISFLA	5.23E−07	9.38E−06	14840.93	30
	BCSFLA	5.95E−01	2.77E−01	56603.13	24
	SFLA	5.62E−01	2.69E−01	61437.80	24
	DE	4.28E−07	9.49E−06	42043.33	30
	BBO	1.00E−02	4.53E−02	200000.00	0
	GSA	5.84E−07	9.37E−06	161030.00	30
f_4	LISFLA	3.69E−02	9.86E−03	20234.33	28
	BCSFLA	1.61E−01	1.67E−01	136836.07	10
	SFLA	2.47E−01	5.52E−01	193757.90	1
	DE	7.96E−07	9.02E−06	21763.33	30
	BBO	1.87E−01	1.58E−01	200000.00	0
	GSA	8.13E−07	8.66E−06	111176.67	30
f_5	LISFLA	8.41E−07	8.90E−06	5554.03	30
	BCSFLA	8.74E−07	8.96E−06	8074.27	30
	SFLA	7.73E−07	8.92E−06	10541.43	30
	DE	6.91E−07	9.02E−06	17165.00	30
	BBO	3.62E−07	9.54E−06	93826.67	30
	GSA	7.56E−07	8.86E−06	91298.33	30
f_6	LISFLA	9.45E−07	8.87E−06	9168.40	30
	BCSFLA	6.82E−07	9.02E−06	10797.23	30
	SFLA	7.06E−07	8.94E−06	18280.53	30
	DE	6.87E−07	9.21E−06	22221.67	30
	BBO	1.65E−05	4.26E−05	200000.00	0
	GSA	1.07E−06	8.84E−06	99265.00	30

(*continued*)

Table 2. (*continued*)

f_7	LISFLA	6.19E−02	2.50E−01	108844.53	17
	BCSFLA	1.29E−01	5.17E−01	200000.00	0
	SFLA	1.47E−01	3.60E−01	167215.30	7
	DE	3.00E−02	2.10E−01	155318.33	26
	BBO	5.37E−02	4.67E−01	200000.00	0
	GSA	5.82E−02	8.00E−01	200000.00	0
f_8	LISFLA	8.56E−07	8.66E−06	8698.07	30
	BCSFLA	6.53E−07	9.11E−06	12437.80	30
	SFLA	9.55E−07	8.84E−06	16133.77	30
	DE	7.81E−07	9.12E−06	25906.67	30
	BBO	4.33E−04	1.28E−03	200000.00	0
	GSA	1.02E−06	8.91E−06	109881.67	30
f_9	LISFLA	2.73E−06	6.81E−06	2738.93	30
	BCSFLA	1.93E−06	7.11E−06	2266.53	30
	SFLA	1.36E−06	8.09E−06	6626.67	30
	DE	2.28E−06	7.26E−06	7665.00	30
	BBO	2.20E−06	7.48E−06	3241.67	30
	GSA	1.81E−06	6.49E−06	46398.33	30
f_{10}	LISFLA	0.00E+00	0.00E+00	3844.73	30
	BCSFLA	4.76E−01	2.00E−01	38038.30	25
	SFLA	0.00E+00	0.00E+00	9004.93	30
	DE	0.00E+00	0.00E+00	10886.67	30
	BBO	0.00E+00	0.00E+00	5351.67	30
	GSA	0.00E+00	0.00E+00	11583.33	30
f_{11}	LISFLA	1.07E−06	8.58E−06	9897.87	30
	BCSFLA	8.49E−07	8.66E−06	14172.93	30
	SFLA	8.00E−07	8.95E−06	18385.47	30
	DE	9.63E−07	8.95E−06	29205.00	30
	BBO	5.39E−03	1.44E−02	200000.00	0
	GSA	9.47E−07	8.75E−06	95278.33	30
f_{12}	LISFLA	3.96E−02	2.39E−02	59982.17	22
	BCSFLA	6.44E−01	5.53E−01	149901.03	8
	SFLA	3.15E+00	4.85E+00	194013.27	1
	DE	7.17E−07	8.90E−06	26943.33	30
	BBO	7.07E−01	8.54E−01	200000.00	0
	GSA	8.82E−07	8.81E−06	90630.00	30

(*continued*)

Table 2. (*continued*)

f_{13}	LISFLA	1.79E−02	3.32E−03	14040.03	29
	BCSFLA	8.89E−07	8.78E−06	11064.47	30
	SFLA	1.79E−02	3.32E−03	20893.87	29
	DE	1.02E−06	8.88E−06	23088.33	30
	BBO	3.09E−05	1.04E−04	200000.00	0
	GSA	6.13E−07	9.00E−06	95498.33	30
f_{14}	LISFLA	5.87E−07	9.07E−06	12680.57	30
	BCSFLA	1.03E−06	8.59E−06	13154.93	30
	SFLA	6.76E−07	9.16E−06	25357.17	30
	DE	5.87E−07	9.08E−06	26216.67	30
	BBO	6.60E−04	2.33E−03	200000.00	0
	GSA	2.33E−05	7.78E−04	134665.00	28
f_{15}	LISFLA	9.31E−17	8.72E−16	21009.73	30
	BCSFLA	9.95E−17	8.45E−16	28845.97	30
	SFLA	4.86E−17	9.30E−16	37419.80	30
	DE	6.21E−17	9.08E−16	58980.00	30
	BBO	2.02E−03	5.94E−03	200000.00	0
	GSA	1.34E−12	1.34E−11	200000.00	0

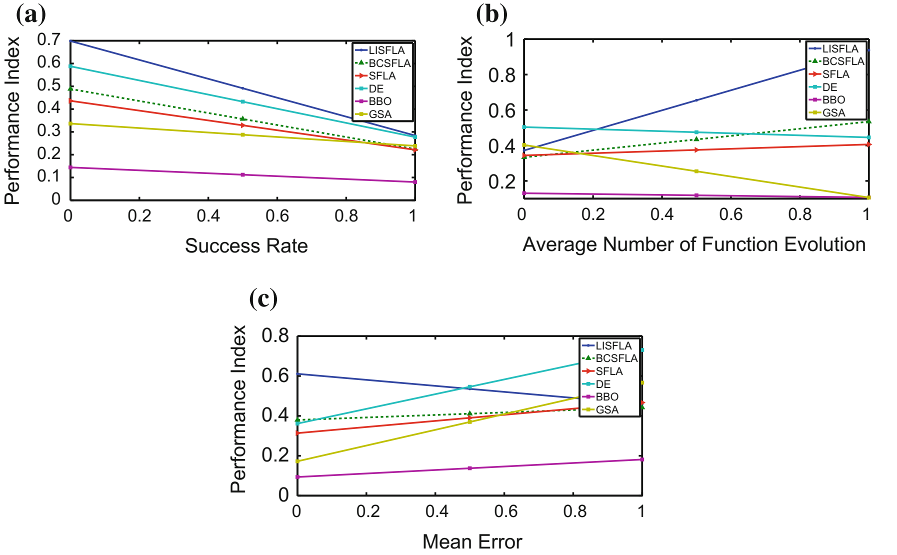

Fig. 2. Performance index for test problems; (a) for weighted importance to SR, (b) for weighted importance to AFE and (c) for weighted importance to ME.

the empirical dispersion of data graphically. The boxplots for LISFLA, SFLA, BC-SFLA, GSA, DE and BBO are displayed in Fig. 1. The results manifest that interquartile range and medians of LISFLA are comparatively low.

Nextly, all considered algorithms are also compared by giving weighted importance to the ME, SR, and AFE. This comparison is measured using the performance indices which is described in [3,4]. The resultant values of PI for the LISFLA, BC-SFLA, SFLA, GSA, DE and BBO are computed and corresponding PIs graphs are demonstrated in Fig. 2.

The graphs belonging to each of the cases i.e. giving weighted importance to AFE, SR and ME (as explained in [3,4]) are depicted in Figs. 2(a), (b), and (c) respectively. In these figures, horizontal axis represents the weights while vertical axis expresses the PI.

It is clear from Fig. 2 that PI of LISFLA are superior than the other considered algorithms in each case. i.e. LISFLA performs better on the considered test problems as compare to the BCSFLA, SFLA, GSA, DE and BBO.

5 Conclusion

In this paper, a new variant of SFLA algorithm is presented, namely Locally Informed Shuffled Frog Leaping Algorithm (LISFLA). In the proposed LISFLA, a new position update strategy for the worst solution is proposed and that is embedded in local exploration phase of primary SFLA. In the proposed locally informed update process, the step size of the worst solution is decided on the basis local best or global best and a local randomly selected neighbour solution of memeplex. In this proposed LISFLA, local best and global best solutions are intensified the search space while, randomly selected neighbour solution is diversified the search area. Further, the proposed algorithm is compared with basic SFLA, its recent variant, namely Binomial Crossover Embedded Shuffled-Frog Leaping Algorithm (BC-SFLA) and three other nature inspired algorithms, namely Gravitational Search Algorithm (GSA), Differential Evolution (DE) and Biogeography-Based Optimization Algorithm (BBO). Experiments over the test functions, depicts that the LISFLA outplays to the considered algorithms.

References

1. Agarwal, P., Mehta, S.: Nature-inspired algorithms: state-of-art, problems and prospects. Int. J. Comput. Appl. **100**(14), 14–21 (2014)
2. Banks, A., Vincent, J., Anyakoha, C.: A review of particle swarm optimization. part ii: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. Nat. Comput. **7**(1), 109–124 (2008)
3. Bansal, J.C., Sharma, H.: Cognitive learning in differential evolution and its application to model order reduction problem for single-input single-output systems. Memetic Comput. **4**(3), 209–229 (2012)
4. Bansal, J.C., Sharma, H., Arya, K.V., Nagar, A.: Memetic search in artificial bee colony algorithm. Soft Comput. **17**(10), 1911–1928 (2013)

5. Blum, C., Li, X.: Swarm intelligence in optimization. In: Blum, C., Merkle, D. (eds.) Swarm Intelligence. Natural Computing Series, pp. 43–85. Springer, Heidelberg (2008)
6. Elbeltagi, E., Hegazy, T., Grierson, D.: A modified shuffled frog-leaping optimization algorithm: applications to project management. Struct. Infrastruct. Eng. **3**(1), 53–60 (2007)
7. Eusuff, M., Lansey, K., Pasha, F.: Shuffled frog-leaping algorithm: a memetic metaheuristic for discrete optimization. Eng. Optim. **38**(2), 129–154 (2006)
8. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department (2005)
9. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. **22**(3), 52–67 (2002)
10. Pourmahmood, M., Akbari, M.E., Mohammadpour, A.: An efficient modified shuffled frog leaping optimization algorithm. Int. J. Comput. Appl. **32**(1), 0975–8887 (2011)
11. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. Inf. Sci. **179**(13), 2232–2248 (2009)
12. Sharma, H., Bansal, J.C., Arya, K.V.: Self balanced differential evolution. J. Comput. Sci. **5**(2), 312–323 (2014)
13. Sharma, S., Sharma, T.K., Pant, M., Rajpurohit, J., Naruka, B.: Centroid mutation embedded shuffled frog-leaping algorithm. Procedia Comput. Sci. **46**, 127–134 (2015)
14. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**(6), 702–713 (2008)
15. Zhao, J., Lv, L.: Two-phases learning shuffled frog leaping algorithm. Int. J. Hybrid Inf. Technol. **8**(5), 195–206 (2015)