An Improved Shuffled Frog Leaping Algorithm with Comprehensive Learning for Continuous Optimization

Liping Xue, Yinglong Yao, Hong Zhou, Zhiqiang Wang
College of Computer Science and Software Engineering, Shenzhen University
Shenzhen, 518060, China
E-mail: xuelp@szu.edu.cn

***Abstract*—This paper presents a shuffled frog leaping algorithm (SFLA) with comprehensive learning strategy (SFLA-CL) for global optimization. This algorithm uses a novel learning strategy whereby all other frogs' information of the memplex is used to update the worst frog's position. The strategy enables the diversity of the memplex to be preserved to discourage premature convergence. SFLA-CL also introduces a new search learning coefficient into the formulation of the original SFLA to enhance the convergence performance of SFLA. SFLA-CL has been evaluated, in comparison with existing evolutionary algorithm, such as SFLA, particle swarm optimization (PSO) and fast evolutionary programming (FEP), on five mathematical benchmark functions. Experimental results demonstrate that the SFLA-CL performs much better than SFLA, PSO, and FEP in optimizing these benchmark functions, particularly, in terms of its convergence rates and robustness.**

Keywords-Evolutionary compution; Shuffled forg leaping algorithm; Comprehensive learning strategy; Particle swarm optimization; Continuous Optimization

I. INTRODUCTION

Shuffled frog-leaping algorithm (SFLA) is originally developed by M. Eusuff and K. Lansey in 2003[1]. It combines the advantages of the genetic-based memetic algorithm (MA) and the social behavior-based particle swarm optimization (PSO) [2]. Recently, there have been a few papers that have reported results of the application of SFLA to various problems [3-7], ranging from the classical combinatorial optimization problem, such as the Travelling Salesman Problem (TSP), to speaker recognition, assembly line sequencing, water distribution network design and reactive power dispatch. The SFLA, in its original form, is easy to implement and has been empirically shown to perform well on unconstrained problems, but detailed analysis and suitable revisions are still needed to further explore its potential.

In this paper, we propose a shuffled frog leaping Algorithm with comprehensive learning strategy (SFLA-CL). In order to improve SFLA performance on complex continuous optimization problems, we apply a new learning strategy which is involved comprehensive learning PSO [8]. Instead of using the frog with the best fitness as the exemplars, all frogs' information of the memplex can potentially be used as the exemplars to guide the frog with the worst fitness leaping direction. Instead of learning from the same exemplar frog for all dimensions, each dimension of the worst frog in general can learn from different frog for different dimensions. In other words, each dimension of the worst frog may learn from the corresponding dimension of different frog. To further improve the search ability of SFLA, the frog's leaping step size is adjusted by adding a search learning coefficient that pull the worst frog toward to speed up convergence. To demonstrate the merits of SFLA-CL, we have evaluated it on five mathematical benchmark functions which cover a range of optimization problems from uni-modal and multi-modal to high dimensions. The algorithm evaluation has been undertaken in comparison with SFLA, PSO and fast evolutionary programming (FEP)[9]. The proposed algorithm is shown to be more superior in performance. Its strong global exploration ability makes its convergence speed very fast, and at the same time it is able to escape from local optima to obtain the global optimum.

II. SHUFFLED FROG LEAPING ALGORITHM

In SFLA, there is a population of possible solutions defined by a set of frogs (solutions) that is divided into different subgroups called memeplexes, each performing a local search. Within each memeplex, the individual frogs hold ideas that can be affected by the ideas of other frogs. After a defined number of memetic evolution steps, ideas are passed among memeplexes in a shuffling process. The local search and the shuffling process continue until defined convergence criteria are satisfied.

In a D-dimension target searching space, generate randomly P frogs (solution) to compose initial population. The ith frog represents the solution of the problem $\boldsymbol{X}_i=(X_{i1}, X_{i2}, \dots, X_{iD})$. Frogs are sorted in a descending order based on their fitness. Afterwards, the frogs are separated into m memeplexes, each containing n frogs (i.e. $P=m\times n$). In this procedure, the first frog is distributed to the first memeplex, the second frog to the second memeplex, the mth frog to the mth memeplex, and the $(m+1)$th frog to the first memeplex and so on.

In each memeplex, the frogs with the best and worst fitness are determined as $\boldsymbol{X}_b$ and $\boldsymbol{X}_w$, respectively. Also, the frog with the global best fitness among the memeplexes is determined as $\boldsymbol{X}_g$. Then, a process is applied to improve only the frog with the worst fitness $\boldsymbol{X}_w$ (not all frogs) in each cycle. Accordingly, each frog updates its position to catch up with the best frog as follows:

Frog leaping step update:

$$\boldsymbol{D}_i = rand(\,)\bullet(\boldsymbol{X}_b-\boldsymbol{X}_w) \quad (-D_{\max}\le \boldsymbol{D}_i\le D_{\max})$$

© the authors

Position update:

$$\text{new } \boldsymbol{X}_w = \boldsymbol{X}_w + \boldsymbol{D}_i \quad .$$

Where *rand*( ) is a random number between 0 and 1; D_{max} represents the maximum of update step allowed. If this process produces a better solution, it replaces $\boldsymbol{X}_w$. Otherwise, $\boldsymbol{X}_b$ of (1) is changed to $\boldsymbol{X}_g$ and adapted to (1).

$$\boldsymbol{D}_i = rand(\,)\bullet(\boldsymbol{X}_g - \boldsymbol{X}_w) \quad (-D_{\max} \le \boldsymbol{D}_i \le D_{\max}) \quad .$$

If the fitness of new $\boldsymbol{X}_w$ still hasn't been improved, a new $\boldsymbol{X}_w$ will be generated randomly.

$$\text{new}\boldsymbol{X}_w = rand(\,)\bullet(\boldsymbol{X}_{\max} - \boldsymbol{X}_{\min}) + \boldsymbol{X}_{\min} \quad .$$

Repeat this update operation until satisfying the update number.

After the local area deep-searching of all memeplex have been finished, to ensure global exploration, the whole memeplexes are mixed in the shuffling process. The local search and the shuffling continue until convergence criteria are satisfied.

III. Shuffled Frog Leaping Algorithm with Comprehensive Learning

SFLA-CL uses a novel learning strategy whereby all other frogs' information of the memplex is used to update the worst forg's position. This strategy enables the diversity of the memplex to be preserved to discourage premature convergence. SFLA-CL also introduces a new search learning coefficient into the formulation of the original SFLA to pull the worst frog toward to speed up convergence.

A. Comprehensive Learning Strategy

In the original SFLA, the worst frog $\boldsymbol{X}_w$ learns from the best frog $\boldsymbol{X}_b$ in each memeplex or the global best frog $\boldsymbol{X}_g$. However, because the worst frog $\boldsymbol{X}_w$ learns from the best frog $\boldsymbol{X}_b$ even if the current is far from the global optimum, frogs may easily be attracted to the region and get trapped in a local optimum if the search environment is complex with numerous local solutions. Liang proposed comprehensive learning strategy to improve the original PSO [8]. In [8], all particles' are used to update the velocity of any one particle. This novel strategy ensures that the diversity of the swarm is preserved to discourage premature convergence. In order to make better use of the beneficial information, we propose a new learning strategy to improve the original SFLA based on [8].

In this new learning strategy, equation (2) can be modified as (5).

$$D_{ij} = rand(\,)\bullet\left[X_{s(j)} - X_{wj}\right] \quad (-D_{\max j} \le D_{ij} \le D_{\max j}) \quad (5)$$

Where $s = [s(1), s(2), \cdots s(D)]$ defines which the worst frog $\boldsymbol{X}_w$ should follow. $X_{s(j)}$ can be the corresponding dimension of any frog's position including $\boldsymbol{X}_b$. We employ the tournament selection procedure when the frog's dimension $\boldsymbol{X}_w$ learns from another frog's as follows.

1) We first randomly choose two frogs out of the memplex.

2) We compare the fitness of these two frogs and select the better one.

3) We use the winner as the exemplar to learn from for that dimension. The details of choosing are given in Fig. 1.

All these $X_{s(j)}$ can generate new positions in the search space using the information derived from different frogs' positions. To ensure that the worst frog learns from good exemplars and to minimize the time wasted on poor directions, we allow the worst frog to learn from the exemplars. We observe two main differences between the SFLA-CL and the original SFLA.

1) Instead of using memplex' $\boldsymbol{X}_b$ and as the exemplars, all frogs' information can potentially be used as the exemplars to guide the worst frog's leaping direction.

2) Instead of learning from the same exemplar frog for all dimensions, each dimension of the worst frog in general can learn from different frog for different dimensions. In other words, each dimension of the worst frog may learn from the corresponding dimension of different frog.

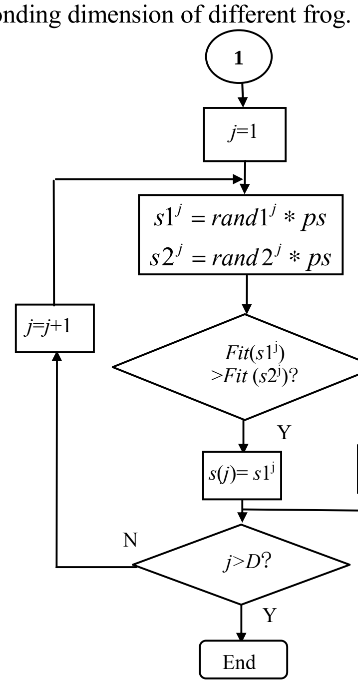

Figure 1. Selection of exemplar dimensions for the frog with worst fitness.

B. Search learning coefficient

In the original SFLA, restricting the frog with worst fitness to jump toward $\boldsymbol{X}_b$ or $\boldsymbol{X}_g$ with a random number between 0 and 1 makes SFLA converge slow. In order to make SFLA converge faster, equation (5) can be modified as (6), and equation (3) can be modified as (7).

$$D_{ij} = C\bullet rand(\,)\bullet\left[X_{s(j)} - X_{wj}\right] \quad (-D_{\max j} \le D_{ij} \le D_{\max j}) \quad (6)$$

© the authors

$$\boldsymbol{D}_i = C \cdot rand(\,) \cdot (\boldsymbol{X}_g - \boldsymbol{X}_w) \quad (-D_{\max} \le \boldsymbol{D}_i \le D_{\max}) \quad (7)$$

where C is a search learning coefficient. It is constant greater than 1 that represents the searching scale for frogs' leaping step size. It is obvious that C cannot be set too large; otherwise, the local search tends to be lost in the random search with little improvement or even cause premature convergence.

IV. EXPREMENTS AND RSULTS

A. *The benchmark functions*

In order to evaluate the performance of SFLA-CL, five widely used benchmark continuous functions selected from [9], are listed below.

Sphere's Function:

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{30} x_i^2$$

where $x \in [-100, 100]^{30}$

Generalized Rosenbrock's Function:

$$f_2(\boldsymbol{x}) = \sum_{i=1}^{29} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$$

where $x \in [-2.048, 2.048]^{30}$

Generalized Rastrigin's Function:

$$f_3(\boldsymbol{x}) = \sum_{i=1}^{30} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$$

where $x \in [-5.12, 5.12]^{30}$

Ackley's Function:

$$f_4(\boldsymbol{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} x_i^2} \right) - \exp\left(\frac{1}{30}\sum_{i=1}^{30} \cos 2\pi x_i \right) + 20 + e$$

where $x \in [-32, 32]^{30}$

Griewank's Function:

$$f_5(\boldsymbol{x}) = \frac{1}{4000}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$$

where $x \in [-600, 600]^{30}$

In the experiment studies, SFLA-CL is evaluated on the benchmark functions in comparison with SFLA, PSO with constriction factors (PSO-cf)[10], and FEP. Here, the evaluation number of an objective function in each algorithm is adopted for comparison purpose. The total evaluation number for each algorithm, taken in a complete optimization process, is 200,000.

B. *Initialisation parameters setting*

For the shuffled frog-leaping algorithm, we follow the parameter settings in [11]. There are 20 memeplexes, each containing 10 frogs. The local exploration in each memeplex is executed for 10 iterations. The parameters settings for SFLA-CL are the same as those of SFLA, with Search learning coefficient C equal to1.5, which is was obtained through trial. For the PSO-cf, the cognitive and social scaling parameters, i.e., $c1$ and $c2$, are both equal to 1.4962; the inertia weight w is 0.7298; and the population size is set to 40. For the FEP, we follow the parameter settings in [9].

C. *Experimental results*

SFLA-CL, SFLA, FEP and PSO-cf are used to optimize the benchmark functions respectively. Each algorithm ran 50 times to give a mean value of the best solutions and a standard deviation obtained from the 50 runs. Table I demonstrates the results obtained by four algorithms applied on the five benchmark functions respectively. From Table I, it can be seen clearly that SFLA-CL can provide a better optimization solution with a much smaller deviation for three out of five benchmark functions, encompassing both uni-modal and multi-modal problems with high dimensions. The merits and characteristics of SFLA-CL are discussed in comparison with SFLA, FEP and PSO-cf as follows.

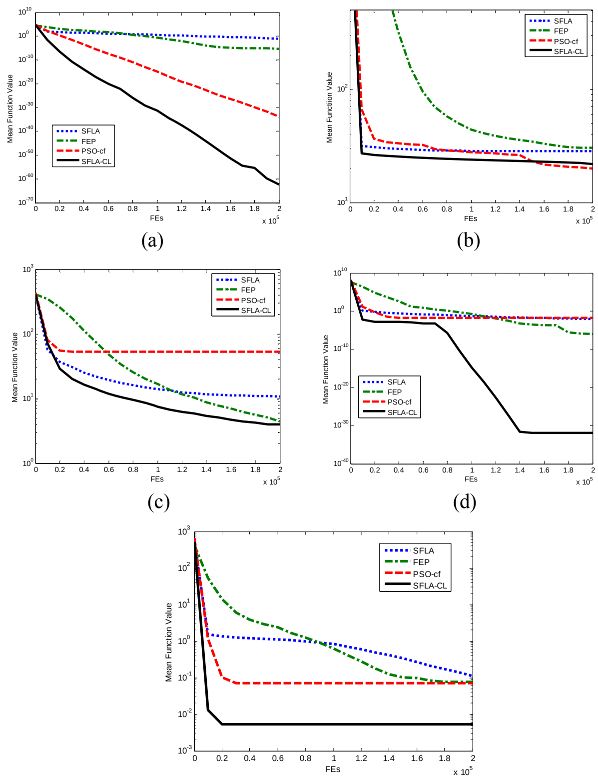

Figure 2. The median convergence charateristics of test functions. (a) Sphere function. (b) Rosenbrock's function. (c) Rastrigin's function. (d) Ackley's function. (e) Griewank's function.

1) *Convergence*

Figures 2 show the convergence process of SFLA-CL, SFLA, FEP and PSO-cf respectively, conducted on the five benchmark functions. For a comparison purpose, we use the number of evaluations to plot the convergence performance of the four algorithms. All the figures illustrate the average best fitness in a population obtained from 50 runs of the program, which is plotted using a logarithmic scale in order to reduce the biggest and smallest values in the whole optimization process.

SFLA-CL has also demonstrated a better ability of global searching for functions Sphere, Rastrigin, Ackley and Griewank. SFLA-CL performs almost as well as PSO-cf in

© the authors

Rosenbrock. From the results presented in Table I, it can be seen that SFLA-CL performs the best in average for most of the benchmark functions. This is because that, the comprehensive learning and search learning coefficient of SFLA-CL enables its capability of searching global optimum in multi-modal continuous functions.

2) Robustness

In most of evolutionary algorithms (EAs), the algorithm robustness is a crucial issue, as EAs are based on stochastic research and random selections. Although the sensitivity to the initial positions and the instability in different functions have been noted during the course of the experiment studies, the experimental results show that the standard deviations obtained, for most of benchmark functions, by SFLA-CL are smaller than those obtained by SFLA, FEP and PSO-cf.

V. CONCLUSIONS

The novel SFLA-CL algorithm has been developed to improve the stability and global search ability for high-dimensional continuous function optimization. In this algorithm, a novel learning strategy whereby all other frogs' information of the memplex is used to update the worst forg's position. This strategy enables the diversity of the memplex to be preserved to discourage premature convergence. And the frog's leaping step size is adjusted by adding a search learning coefficient that pulls the worst frog toward to speed up convergence. SFLA-CL has been evaluated on five benchmark problems, which include uni-modal and multi-modal functions in high dimension domains. The convergence rates and robustness of SFLA-CL have been well discussed in this paper. The experimental results have shown that SFLA-CL has superior performance in comparison with SFLA, FEP and PSO. Through the SFLA-CL experiment studies, it has been seen that SFLA-CL possesses a great potential for global optimization of complex problems.

ACKNOWLEDGMENT

This work is supported by Foundation for Distinguished Young Talents in Higher Education of Guangdong, China (No. LYM1113) and National Natural Science Foundation of China Project (No. 61103055).

REFERENCES

[1] M. Eusuffm, K. E. Lansey, "Optimization of Water Distribution Network Design Using Shuffled Frog Leaping Algorithm," Journal of Water Resources Planning and Management, Vol. 129, Mar. 2003, pp. 210-225.

[2] J. Kennedy, R. Eberhart, "Particle swarm optimization," Proc. IEEE Int Conf on Neural Networks 1995, IEEE Press, 1995, pp.1942-1948.

[3] X. H. Luo, Y. Yang, and X. Li, "Modified shuffled frog-leaping algorithm to solve traveling salesman problem," Journal of Communications, Vol. 30 , Jul. 2009, pp. 130-135.

[4] L. P. Xue, J. X. Yin, J. R. Zhou, and Z. Ji, "A novel shuffled particle-pair optimizer for speaker recognition," Journal of Electronics & Information Technology, Vol. 6, Jun. 2009, pp. 1359-1362.

[5] M. Eusuffm, K. E. Lansey, and F. Pasha, "Shuffled frog leaping algorithm: a memetic meta-heuristic for discrete optimization," Engineering and Technology, Mathematics and Optimization, Vol. 38, Feb. 2006, pp. 129-154.

[6] A. Rahimi-Vahed, M. Dangchi, and H. Rafiei, "A novel hybrid multi-objective shuffled frog leaping algorithm for a bi-criteria permutation flow shop scheduling problem," International Journal of Advanced Manufacturing Technology, Vol. 41, Dec. 2009, pp. 1227-1239.

[7] A. Khorsandi, A. Alimardani, B. Vahidi, and S.H. Hosseinian, "Hybrid shuffled frog leaping algorithm and Nelder–Mead simplex search for optimal reactive power dispatch," IET Genetation Transmission & Distribution, Vol. 5, Iss.2, 2011, pp. 249-256.

[8] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baker, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," IEEE Transactions on Evolutionary Computation, Vol. 10, Jun. 2006, pp. 281-295.

[9] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster," IEEE Transactions on Evolutionary Computation, Vol. 3, Jul. 1999, pp. 82-102.

[10] R. C. Eberhart, Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," Proceedings of the IEEE International Conference on Evolutionary Computation, IEEE Press, 2000, pp. 84-88.

[11] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary based optimization algorithms," Advanced Engineering Informatics, Vol. 19, Jan. 2005, pp. 43- 53.

TABLE I. COMPARISON RESULTS FOR FEP, PSO-CF, SFLA, AND SFLA-CL

Function	FEP Mean Best ± Std Dev	PSO-cf Mean Best ± Std Dev	SFLA Mean Best ± Std Dev	SFLA-CL Mean Best ± Std Dev
Sphere	5.84e-06±2.13e-06	1.73e-34±9.56e-34	8.79e-02±1.04e-01	4.20e-63±2.94e-62
Rosenbrock	3.03e+01±1.77e+01	1.99e+01±1.36e+01	2.84e+01±2.37e-01	2.20e+01±1.64e-01
Rastrigin	4.44e+00±2.78e+00	5.25e+01±1.72e+01	1.07e+01±3.98e+00	7.87e+00±3.67e+00
Ackley	9.87e-07±5.71e-07	1.49e-02±3.74e-02	8.85e-03±9.51e-03	1.35e-32±2.74e-48
Griewank	7.79e-02±1.19e-01	1.49e-02±3.74e-02	1.09e-01±8.19e-02	5.53e-03±7.06-03

© the authors