Improved Frog Leaping Algorithm Using Cellular Learning Automata

Somayeh Ranjkesh
Islamic azad university, Qazvin Branch, Iran
s.ranjkesh@yahoo.com

Mohammad Reza Meybodi
Department of Computer Engineering and Information Technology
Amirkabir University of Technology
Tehran, Iran
mmeybodi@aut.ac.ir

***Abstract*—In this paper, a new algorithm which is the result of the combination of cellular learning automata and frog leap algorithm (SFLA) is proposed for optimization in continuous, static environments.At the proposed algorithm, each memeplex of frogs is placed in a cell of cellular learning automata. Learning automata in each cell acts as the brain of memeplex, and will determine the strategy of motion and search.The proposed algorithm along with the standard SFLA and two global and local versions of particle swarm optimization algorithm have been tested in 30-dimensional space on five standard merit functions. Experimental results show that the proposed algorithm has a very good performance.**

Keywords-Frog Leaping Algorithm;Optimization; cellular learning automata

I. INTRODUCTION

Optimization is considered one of the MOST important problems in engineering and mathematics. Great importance and application of optimization have caused many scientists to engage in research in its various areas.Optimization is a problem that all people deal with in their daily lives. When the parameters and constraints of optimization problems are small in number, they can be easily solved; however if the constraints and parameters increase, these problems will be very difficult to solve in the way that they become NP-hard problems.Collective intelligence algorithms are one of the most popular methods to solve these problems .All these algorithms are meta-heuristic methods, the most famous of which are: particle swarm optimization algorithm [1], bee colony optimization algorithm [2], frog leaping algorithm [3], ant colony optimization algorithm [4], and artificial fish batch algorithm [5].Frog leap algorithm (SFLA) was presented by Eusuff and Lansey in 2003 [6]. This algorithm, which is derived from the particle swarm optimization algorithm and memetic algorithms, can simulate the behavior of frogs in the wetlands to find food.In this algorithm, the advantages of memetic algorithms, which are based on genetics and social behaviors of particles in PSO, is used. A cellular learning automata (CLA) [7] is a model designed for systems that are composed of simple components that can show complex behaviors through interaction with each other.The cellular learning automata is composed of a cellular automata in which each cell is equipped with one or more learning automata that specify the status of this cell. A local rule governs the environment that determines the reward or penalty of the action chosen by the learning automata.In this paper, a new algorithm called SFLA-CLA is proposed, in which the performance of SFLA is substantially improved with the help of the CLA.In this algorithm, each memeplex of frogs is placed in a cell of the CLA. LA in each cell is responsible for controlling the memeplex in the cell, and acts as its brain.The proposed algorithm along with the standard SFLA and two global and local versions of particle swarm optimization algorithm is used to optimize six famous functions which are used to measure the performance of optimization algorithms in constant, static environments.The experimental results show that the proposed algorithm has very high performance and ability.This paper is organized as follows,The second section presents the standard SFLA algorithm; in the third section, the proposed SFLA-CLA algorithm will be described; the fourth section provides the experimental results, and the final section deals with conclusions.

II. SHUFFLED FROG LEAPING ALGORITHM (SFLA)

Frog leap algorithm was presented by Eusuff and Lansey in 2003 [7]. This algorithm works on the basis of the probability rules, random search and population. The process of running this algorithm is derived from the behavior of frogs in wetlands to find food.In optimization problems, the situations with more merits have more food, and aim to find more food. Thus, frogs try to get more food with the implementation of the SFLA.At the beginning of the implementation of the SFLA, there is a population of frogs that are randomly distributed within the problem space. In the SFLA, every frog represents a feasible solution within the problem space. After positioning the frogs, the value of their merit is measured.Position of *ith* frog in the D-dimensional space is equal to xi= (*xi ,1, xi, 2, …, xi, D*); and the amount of food available in the position of the frog, which is value of its merit, is equal to f (xi).Like other collective intelligence algorithms, the SFLA also works in an iterative manner, which the following steps are performed at any iteration,1. Members of the population are first sorted according to their merit value (the best frog is placed in index “1”).2. Frogs are divided into *m* groups, which are called memeplexes.Each memeplex includes *n* frogs. The population is divided as follows,First frog will be given to the first memeplex, second frog to second memeplex, and *mth* frog to *mth* memeplex. Then, *(m+1)th* frog is given to 1th memeplex; and similarly the classification process continues until nth frog is placed in each memeplex. 3. After frogs are classified in *m* memeplexes, each memeplex will repeat the following process *itr* times. In each memeplex, position of the best and the worst frogs are found in terms of merit levels, which are equal to *Xb* and *Xw*, respectively. Then a candidate position is obtained by using the following equations,

$$D = Rand.\left(X_b - X_W\right) \qquad (1)$$

$$X'_W = X_W + D\,, \qquad D_{\min} \le D \le D_{\max} \qquad (2)$$

where in "(1)" and "(2)", rand is a function generating random number (with uniform distribution over the interval [0,1]), and is equal to the length of displacement step: its value in each dimension of the problem should be in the range of [*Dmin*, *Dmax*].After the position is obtained, its value of merit will be measured. If the value of merit is better than the previous position, *Xw* will move to its new position, and otherwise "(1)" and "(2)" are performed again, except that the position of the best frog among all memeplexes (i.e. *Xg*) is used instead of the position of *Xb*.After this step, if the merit value of the position obtained by from "(2)" is not better than the previous position of Xw, then this frog will be initialized; i.e. its new position within the problem space is determined randomly. After step 3 is performed by all memeplexes for *itr* times, the termination condition of the algorithm is measured.If this condition is not met, all the frogs will be placed again in a group (shuffling) regardless of the memeplex to which it belongs; and they perform steps 1 to 3.

III. The Proposed Algorithm

In this section, a new combined method is proposed based on the frog leap algorithm and cellular learning automata in which each of the existing memeplexes in the SFLA algorithm is placed in a cell of cellular learning automata. Each cell is equipped with a learning automaton which becomes the brain of the memeplex in the cell.In the SFLA algorithm, each frog acts within the problem space for each displacement as follows: It first tries to go to the best frog of its memeplex; and if failed, it moves to find the best frog in the whole group. If failed in doing this, it moves to a random point within the problem space.In the SFLA algorithm, when the worst frog of each memeplex moves to look for the best frog of its memeplex, it can cause a search to find a better position by memeplex Y and improve the ability of global search to find the best peak.However, when the frogs move in search of the best frog of group, the group will be gathered around the best position found by the algorithm, and the local search is increased around it. On the other hand, the random motion of failed frogs within the problem space causes the escape from local optimum and thus increases in the ability of global search.With three actions listed, the LAFSA algorithm tries to do global search and local search at acceptable level. But it does not strengthen or weaken one or both of the abilities depending on the conditions within the problem space. For example, the conditions of the problem may be in a such way that a strong local search is needed to achieve better results, while some frogs look for the best frog of their memeplex and typically conduct a global search.In fact, a part of the potential of the algorithm is lost in such a case. To fix this problem and improve the results of the algorithm, LA in each cell acts as a decision maker for the memeplex in the same cell, and will determine the types of motion of frogs in it according to the conditions of memeplex and its type of group within the problem space.The performance trend of LA in each cell is as follows: in the proposed algorithm, there is an LA in each cell which contains two actions.The first action is to follow the best frog of each memeplex by the worst frog of the memeplex, and if failed, to perform a random motion within the problem space. The second action is to follow the best frog of each memeplex by the worst frog, and if failed, to perform a random motion within the problem space.In fact, with the first action, it is tried to do better the ability of global search for algorithm; and with the second behavior of the algorithm, it tries to improve its ability of local search.In any iteration of the algorithm, this automaton will determine how frogs may act. For each frog that intends to move within the problem space (i.e. the worst frog of the memeplex), the automata chooses an action based on its probability vector, and then, does this action.LA in each cell performs the action of learning and decision making under the local rule governing CLA.In the proposed algorithm, the local rule governing CLA determines whether the action selected by LA in each cell should be rewarded or fined.In the proposed algorithm, there are *m* cells (equal to the number of memeplexes); and the neighborhood in CLA is in line with periodic boundary so that successive memeplexes can be located in the neighboring cells.There is the same

LA in all cells (i.e. they have the same actions and parameters, and all are updated together; CLA is of homogeneous type.At any iteration of the proposed algorithm, each LA acts independently of LA in other cells and selects an action based on its probability vector, and then implements it.After the action selected by LA in all cells was implemented, the LA of each cell can evaluate the action done according to local rule, and encourages or punishes its action based on the result of evaluation.The local rule governing CLA is as follows:1. If a memeplex manage to find a better position in an iteration of its implementation (i.e. position of the best frog of the improved memeplex) and the merit of its best frog is higher in amount than the merit of the best frog in its two neighboring memeplexes, then the selected action will be rewarded.2. If a memeplex have had a progress that is at least higher in amount than the progress of one of its neighboring memeplexes, then the selected action will be encouraged (the amount of progress of a memeplex, i.e. the difference in the merit value of the frog before and after implementation of the action chosen by the automata, which should be a negative number in minimization problems).3.

If a memeplex has conditions other than the two above conditions (1 and 2), then the selected action will be punished (i.e. it has no progress or a progress less than that of its neighboring memeplexes)

IV. Results Of Experiments

Experiments were performed on five standard functions that are generally used as a measure of the optimization algorithms in continuous, static spaces.Experiments were performed in 30-dimensional space, and the results of the proposed algorithm are compared with the standard SLFA, global version of the particle swarm optimization [8] and the local version of the algorithm [9].Parameters of the two PSO algorithms are determined based on their references. In the SFLA algorithm and the proposed algorithm, the number of memeplexes is considered equal to 10, and frog population size is considered equal to 30.Length of D can be at most equal to 0.1 times the length of the problem space in each dimension. In addition, the value of *itr* parameter is considered equal to 10.The values listed for the parameters of the two algorithms (SFLA and the proposed algorithm) were determined on the basis of several experiments conducted in different areas.Structure of the learning automata used in the proposed algorithm varies; environment has been considered to be of P type; learning algorithm is used; values "a" and "b" are equal; and algorithm is of LARP type.In the proposed algorithm, the coefficients of rewards and penalties ("a" and "b" for learning automata) are considered equal to 0.01.Experiments are repeated 50 times, and 200,000 evaluations have continued each time. The average results, the best results and their standard deviations are given in Table 1 in 30-dimensional space.

TABLE I.

Function	**Criteria**	**GPSO**	**LPSO**	**SFLA**	**SFLA-CLA**
Sphere	***Best***	4.56e-03	1.31e-01	9.42e-02	1.46e-03
	Mean	7.72e-03	3.72e-00	4.44e-01	7.06e-03
	Std-Dev	2.22e-02	3.59e-00	1.72e-01	1.26e-03
Schwefel P2.22	***Best***	20.00	0.0001	1.94e-014	1.55e-02
	Mean	256.21	82.97	9.59e-01	7.82e-01
	Std-Dev	512.63	96.43	3.00e-01	1.96e-01
Generalized Penalized	***Best***	1.11e-03	1.64e-00	3.70e-02	1.57e-03
	Mean	0.04	5.39e-00	8.37e-01	2.00e-02
	Std-Dev	0.11	0.02	1.94e-01	8.95e-02
Griewank	***Best***	2.22e-01	2.00e-00	3.34e-01	2.11e-01
	Mean	0.14	7.43e-00	0.21	0.05
	Std-Dev	0.14	9.02e-00	0.26	0.11
Ackly	***Best***	6.22e-01	8.39e-00	1.00e-01	2.04e-01
	Mean	1.20e-01	4.18e-00	1.14e-00	3.35e-01
	Std-Dev	3.00e-01	2.09e-00	4.47e-00	7.29e-01

Comparison of the best and average results of running the algorithms for 50 times on the test functions in 30-dimensional space.

As is shown in Table 1, the results of the proposed algorithm are better than the SFLA standard algorithm in all cases. The improved performance of the proposed algorithm is due to the control of search behavior of frogs during the optimization process.Indeed, with the help of LAs in CLA cells, each memeplex can do a local search or global search according to the position of frogs within the problem space and the progress of the optimization process.The main task of the CLA, in fact, is to create a balance between capabilities of local and global search and use them in time. According to the results of algorithms, the proposed algorithm did not get the best results only in two Griewangk and Ackley functions.The result obtained in the Ackley function is almost equal to the GPSO algorithm; and the difference in results is minimal. However, the results obtained in Griewangk function have considerable difference with the results obtained from the LPSO algorithm.Performance degradation of the proposed algorithm in this function is due to the specific form of space of problem in this function, in which it is very hard to escape the local optimum.Since very strong global search is carried out due to the type of the loop neighborhood of the particles, the LPSO algorithm can be removed from the local optimum of this function. In "Fig. 1", the average figure of merit function in algorithm run 50 times on the test functions are shown.

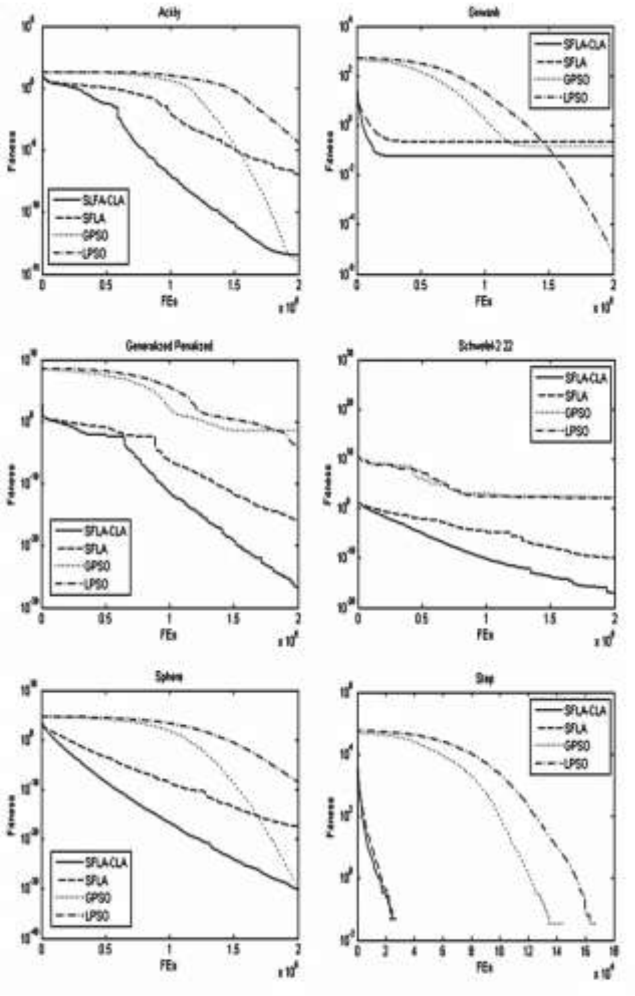

Figure1. Convergence behavior of algorithms on the functions in Table1

V. Conclusion

In this paper, a new combined algorithm was proposed based on the frog leap algorithm and cellular learning automata. In the proposed algorithm, learning automata in each cell acts as the brain of memeplexes in the cells.The main task of the cellular learning automata used in the proposed algorithm is to create a balance between global and local search capabilities and control them.Performance of the proposed algorithm was compared to the shuffled frog leap algorithms and two global and local versions of particle swarm optimization algorithm on five known benchmark functions.In all, results showed that the performance of the proposed algorithm is better than the other algorithms tested (especially standard SFLA).

References

[1] J. kennedy and R. Eberhart, “Particle Swarm Optimization”, in IEEE International Conference on Neural Networks, Vol. 4, pp. 1942-1948, Perth, November 1995.

[2] D. Karaboga and B. Akay, “A Comparative Study of Artificial Bee Colony Algorithm”, in Elsevier Journal on Applied Mathematics and Computation, 2009.

[3] M. Eusuff and K. Lansey, “Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm”, in Journal of Water Resource Plan and Management, Vol. 3, pp. 10-25, 2003.

[4] M. Dorigo, M. Birattari, T.Stutzle, “Ant Colony Optimization”, in IEEE Computational Intelligent Magazine, Vol. 1, pp. 28-39, 2006.

[5] L. X. Li, Z. J. Shao and J. X. Qian, “An Optimizing Method Based on Autonomous Animate: Fish Swarm Algorithm”, In Proceeding of System Engineering Theory and Practice, Vol. 11, pp. 32-38 , 2002.

[6] E. Elbeltagi, T. Hegezy and D. Grierson, “A Modified Shuffled Frog-Leaping Optimization Algorithm: Applications to Project Management”, in Journal of Taylor & Francis on Structure and Infrastructure Engineering, Vol. 3, pp. 53-60, 2007.

[7] M. R. Meybodi, H. Beygi and M. Taherkhani, “Cellular Learning Automata”, In Proc. of CSICC01, 2001.

[8] J. von Neumann, “Theory of Self-Reproducing Automata”, University of Illinois Press, 1966.

[9] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” in IEEE Congress on Evolutionary Computation, Honolulu, HI, pp. 1671–1676, 2002.