A New Hybrid Algorithm for Optimization Based on Artificial Fish Swarm Algorithm and Cellular Learning Automata

Danial Yazdani
Islamic azad university, Shirvan Branch, Iran
d.yazdani@iau-shirvan.ac.ir

Sara Golyari
Islamic azad university, Shirvan Branch, Iran
S.golyari@iau-shirvan.ac.ir

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

Abstract— **In this article, a new algorithm which is obtained by hybridizing cellular learning automata and artificial fish swarm algorithm (AFSA) is proposed for optimization in continuous and static environments. In the proposed algorithm, each dimension of search space is assigned to one cell of cellular learning automata and in each cell a swarm of artificial fishes are located which have the optimization duty of that specific dimension. In fact, in the proposed algorithm for optimizing D-dimensional space, there are D one-dimensional swarms of artificial fishes that each swarm is located in one cell and they contribute with each other to optimize the D-dimensional search space. The learning automata in each cell is responsible for making diversity in artificial fishes swarm of that dimension and equivalence between global search and local search processes. The proposed algorithm with standard AFSA, Cooperative Particle swarm optimization (PSO) and global version of PSO in 10 and 30-dimensional spaces are practiced on six standard fitness functions. Experimental results show that presented method has an acceptable performance.**

Keywords; Artificial fish swarm algorithm, cellular leaning automata, particle swarm optimization, cooperative approach.

I. INTRODUCTION

Optimization concept is that among parameters of a function we seek for values which minimize or maximize the function. All proper values for this are called feasible solutions and the best value of these values is called optimal solution. Optimization algorithms cover both maximization and minimization problems. Optimization has various applications such as resource allocation, scheduling, decision making, etc and there are different methods for optimization [1][2][3]. Finding solution for NP problems is too hard. Algorithms such as swarm intelligence algorithms have solved this problem to some extent. Some of characteristics of these algorithms are scalability, fault tolerance, compatibility, speed, internal independence, parallelism and so on. One of the most famous and applicable swarm intelligence algorithms is Particle Swarm Optimization (PSO) algorithm which was presented by Kennedy and Eberhart in 1995 [4]. This algorithm is an efficient technique for optimization problems that works based on probability laws and population. One of the other swarm intelligence algorithms which were presented so far is Artificial Fish Swarm Algorithm (AFSA). AFSA is an algorithm that is inspired from nature which was presented by Dr. Li Xiao Li [5] in 2002. This algorithm is a new technique based on swarm behaviors that are inspired from social behaviors of fishes swarm in the nature. It has high convergence rate, insensitive to initial values, flexibility and high fault tolerance. This algorithm has been used in applications of optimization such as data clustering [6][7], PID control [8], data mining [9], DNA sequence encoding [10] and so on. AFSA has major differences to PSO algorithm, in fact the structure and performance of AFSA is completely different from PSO's. The biggest difference between these two algorithms is that particles in PSO are dependent to their past for their next movement, indeed for next movement, every particle uses the best personal experience, best global experience and previous velocity but in AFSA, artificial fishes for next movement act completely independent from past and next movement is just related to current position of artificial fish and its other companions. Since AFSA had much higher complexity to its similar algorithms such as PSO and however its efficiency was not better than them, was not much considered.

A learning automata [11], is a machine which can perform finite number of actions. Each selected action is evaluated by a probability environment and evaluated result in the form of a positive or negative signal is given to learning automata and learning automata uses this response to select next action and thus approaches to selection of an action which gets the most reward from environment. In other words, automata learn the action which receives the most reward from the environment. Learning automata has been used for improvement of many algorithms that of them, it could refer to artificial fish swarm algorithm [12], genetic algorithms [13] and particle swarm optimization [14]. Cellular leaning automata (CLA) [15], is a model for systems which are constituted of simple elements that by interaction with each other can show complex behavior of themselves. Each cellular learning automata consists of a single cellular automata which each of its cells is equipped with one or more learning automata that determine cell condition. A local rule dominates on environment which is determining reward or penalty of selected action by learning automata.

In this article, a new algorithm called AFSA-CLA is proposed in that with aid of CLA the efficiency of AFSA was considerably improved. In the proposed algorithm, a cell of CLA is assigned to each dimension of problem space that has a population of artificial fishes. Each population has the responsibility of optimizing one dimension. Available learning automata in each cell is controlling the parameters of artificial fishes of that cell and avoiding of preceding convergence. Each learning automata by using the experiences of neighbor cells determines its form of cell's artificial fish's movement.

The rest of this article is organized as follows: second section introduces standard AFSA algorithm, section three briefly introduces learning automata, cellular automata and cellular learning automata, in section four describes proposed algorithm of AFSA-CLA, section five represents experimental results and final section is concluding.

II. Artificial Fish Swarm Algorithm

In water world, fishes can find areas that have more foods, which it is done with individual or swarm search by fishes. According to this characteristic, artificial fish model is represented by prey, free move, swarm movement and follow behaviors which the problem space is searched by them. The environment, which artificial fish lives in, substantially is solution space and other artificial fishes domain. Food consistence degree in water area is AFSA objective function. Finally, artificial fishes reach to a point which its food consistence degree is maximum (global optimum).

As in figure 1 is observed, artificial fish perceive external concepts with sense of sight. Current situation of artificial fish is shown by vector $X=(x_1, x_2, \ldots, x_n)$. The visual is equal to sight field of artificial fish and X_v is a position in *visual* where the artificial fish wants to go there. Then if X_v has better food consistence than current situation, we go one step toward it which causes change in artificial fish situation from X to X_{next}, but if the current position is better than X_v, we continue browsing in visual area. The step is equal to maximum length of the movement, the distance between two artificial fishes which are in X_i and X_j positions is shown by $d_{ij}=\| X_i-X_j\|$ (Euclidean distance).

Artificial fish model consists of two parts of variables and functions that variables include X (current artificial fish situation), step (maximum length step), visual (sight field), try-number (maximum test interactions and tried) and crowd factor δ ($0<\delta<1$). Also functions consist of prey behavior, free move behavior, swarm behavior and follow behavior.

In each step of optimization process, artificial fish look for locations with better fitness values in problem search space by performing these four behaviors based on algorithm procedure[6][9][10][12].

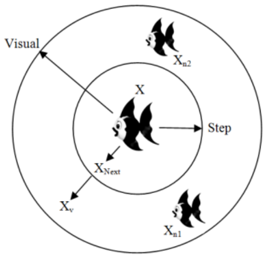

Figure 1. Artificial Fish and the environment around it.

III. Cellular Automata, Learning Automata and Cellular Learning Automata

Cellular automata (CA) are mathematical models for systems consisting of large numbers of simple identical components with local interactions. CA is non-linear dynamical systems in which space and time are discrete. It is called *cellular* because it is made up cells like points in a lattice or like squares of checker boards, and it is called *automata* because it follows a simple rule. The simple components act together to produce complicated patterns of behavior. Cellular automata perform complex computations with a high degree of efficiency and robustness. They are especially suitable for modeling natural systems that can be described as massive collections of simple objects interacting locally with each other. Informally, a d-dimensional CA consists of an infinite d-dimensional lattice of identical cells [15].

In other hand, learning automata (LA) are simple agents for doing simple things. The learning automata have finite set of actions and at each stage choose one of them. The choice of an action depends on the state of automaton which is usually represented by an action probability vector. For each action chosen by the automaton, the environment gives a reinforcement signal with fixed unknown probability distribution, which specified the goodness of the applied action. Then upon receiving the reinforcement signal, the learning automaton updates its action probability vector by employing a learning algorithm [15].

CLA is a mathematical model for dynamical complex systems that consists of large number of simple components. The simple components, which have learning capability, act together to produce complicated behavioral patterns. A CLA is a CA in which a LA will be assigned to its every cell. The learning automaton residing in each cell determines the state of the cell on the basis of its action probability vector. Like CA, there is a rule that CLA operates under it. The rule of CLA and the actions selected by the neighboring LAs of any cell determine the reinforcement signal to the LA residing in that cell [15].

IV. Proposed Algorithm

In this section an artificial fish swarm algorithm based on cellular learning automata called AFSA-CLA is proposed which in that both ability of local search and ability of global search to standard AFSA has been increased and improved remarkably. In the proposed algorithm, there are swarms of one dimensional artificial fishes equal to the number of search space dimensions that each swarm is placed in a cell of cellular learning automata. Each cell is equipped with a learning automaton that determines and adjusts the corresponding next movement of artificial fishes. Before describing the performance of proposed algorithm, it is essential to impose a change to improve AFSA's performance.

In standard AFSA, the best situation is stored which has been found by swarm so far. Because even artificial fish that has found this situation, in the next iteration by doing a freely movement may lose its current situation and stick in worse situation so it is necessary to store the best result in each iteration which has been found so far. But this is considered a great weakness in AFSA, since the best found swarm's result is not used to develop swarm's member. Before this, in algorithms with similar efficiency such as PSO, for movement and improvement of swarm's members in next iterations the best result is used which the swarm has experienced so far.

In AFSA in order to be able of using the best swarm experience we must make a change in the structure of AFSA. In AFSA when an artificial fish could not find better situation in food search behavior, with doing a freely movement, moves a random step and loss its previous situation. Doing this movement even could result in worse situation of fish in problem space. But performing this movement on artificial fishes is necessary because causes diversity in swarm so artificial fishes could search other situations in next iterations. Thus it can conclude that omitting this movement is wrong. Therefore we will not perform this movement only for the best artificial fish in the swarm. So the best artificial fish in the swarm, doesn't lose its position even if doesn't find better position. Therefore the best point that has been found by swarm during previous iterations is the position of the best artificial fish. Thus, the best artificial fish of swarm acts like the leader of swarm and in each iteration the other artificial fishes by performing hierarchical follow-behavior, follow that artificial fish toward better positions.

In AFSA, artificial fishes, by performing AFSA behaviors, search problem space in their visual space and in each iteration move toward their goal some random percent of step length. In standard AFSA, determination of initial value of these two parameters has a remarkable effect on the quality of final result. The values of these two parameters remain constant and equal to their initial value up to the end of performing the algorithm. If we consider large initial values for these two parameters artificial fish swarm can move faster toward global optimum, since artificial fishes can explorer larger space around themselves and move with a larger step length in each iteration. In this condition, artificial fish swarm will have more ability to pass local optimums because being large of these two parameters causes the swarm to be developed and the distance between artificial fishes with each other will be more. Thus fishes that get trapped in local optimum, by following other swarm's members that are out of local optimum go out of there.

Also, large value of visual parameter causes search space of artificial fishes to become large and they can explore further distances which this results they could search spaces out of local optimums where they are trapped and get out of there. However, large values of visual parameter and step length parameter have some disadvantages, in this case, the accuracy and stability of algorithm is decreased in local search. In fact, in this case, the algorithm performs well global search but after artificial fish swarm gets near global optimum, algorithm is not capable of performing an acceptable local search since in this case, parameter value of visual is larger than space where it is necessary artificial fish searches, therefore the probability of finding places with better fitness value is decreased. Also because of being large of step length, artificial fishes pass global optimum and even go far it that this causes decrease in accuracy and stability of algorithm near global optimum.

If we consider small values for these two parameters, algorithm can do local search with more stability and accuracy, but in this condition fishes move slower toward goal and their swarm becomes move compact that as a result the capability to pass local optimums is decreased. Thus, it is necessary for improving results, a balance made between global search and local search process so the algorithm could perform both acceptably that this is done in the proposed algorithm.

AFSA-CLA performance is as follow: instead of using of one swarm with n D-dimensional artificial fishes, we use D swarm of n one-dimensional artificial fishes. The objective of each swarm is to optimize a component of optimization vector which corresponds to that swarm. To achieve the desirable result, all these swarms should cooperate with each other. In order to cooperate, the best fish of each swarm would be chosen. The vector that is totally being optimized for all swarms is called "main vector". Each swarm introduces it's the best artificial fish as representative of that dimension. In fact, main vector which is D-dimensional vector consists of the values of the best artificial fish of each of D swarms. In each iteration of algorithm performance, for artificial fishes of swarm *i*, just the element of that *i*-th dimension of main vector is changed. Indeed, for cooperation among swarms, for calculating the fitness of an artificial fish in *j*-th swarm, vector value for that artificial fish is replaced in *j*-th element of main vector to obtain a D-dimensional vector, and then its fitness is calculated in D-dimensional space.

In AFSA-CLA, every cell is assigned to one of components of main vector, in fact, the structure of cellular learning automata is one dimensionally (linear) with Moor neighborhood with periodic boundary (D-th cell is the neighbor of first cell). Therefore *i*-th cell is corresponding to *i*-th component of main vector and includes *i*-th swarm of artificial fishes. Available learning automata is equal in all cells and all are updated together (cellular learning automata is homogenous). Available learning automata in each cell includes two action α_1 and α_2 and has variable structure with linear learning algorithm $p(n+1)=T[\alpha(n),\beta(n),p(n)]$ Actions of learning automata are as follows:

1- Decrease in visual and step parameters with multiplying their values by smaller value than one.

2- Performing Reset action for defined percentage of available artificial fishes in swarm.

First action of learning automata is for balancing the global search ability and local search ability of proposed algorithm. For this purpose, first we consider large parameters value of visual and step so the global search ability of algorithm to be high. Thus, the swarm moves faster toward global optimum and passes local optimums with more ability. With performing first action of learning automata, simultaneously with artificial fishes movement toward global optimum, gradually parameters value of step and visual are decreased so finally the algorithm could perform acceptable local search near global optimum.

In the proposed algorithm, each of the artificial fishes has corresponding visual and step for itself, that is the values of these parameters can be unequal in two different fishes. Second action of learning automata is considered for avoiding preceding convergence in order to increase the capability of escaping from local optimums. For this purpose, by doing this action, a defined percentage of artificial fishes of swarm randomly disperse in problem space and their visual and step parameters values are become equal to their initial values (Reset action). Thus, artificial fishes that have leaved the swarm, return to swarm by doing AFSA swarm behaviors but in return way to swarm, they search spaces of problem which were remote to find better values of fitness that it can cause to discover positions with better fitness.

In the cellular learning automata that used in proposed algorithm, each cell is neighbor with two previous and next cells. Dominated local rule on cellular learning automata determines whether the action learning automata has done should receive reward or penalty. After that learning automata in cells do an action, each cell will evaluate the amount of its neighbor's improvement in previous iteration. If the ratio of at least one of neighbors is higher than defined percentage of swarm improvement corresponding to considered cell, performed action by learning automata will be fined, otherwise it'll be rewarded. In fact, if neighbor have considerable improvement to considered swarm, i.e. in corresponding dimension of this swarm, the algorithm is trapped in local optimum or the kind of selected movement is not suitable. Thus in proposed AFSA-CLA algorithm, there is a balance between global search and local search ability as well as the capability of avoiding preceding convergence is increased. Experimental results are given in next section.

V. Experimental Results

Experiments are done on six standard test functions which are often used as measurement criterion of optimization algorithms in continuous and static spaces. Applied functions together with their search space range and acceptable result are presented in Table 1 [16]. It should be noted that optimal value of all these function equals zero.

Experiments are performed in 10 and 30 dimensional spaces and the result of proposed algorithm of AFSA-CLA, standard AFAS, global version of PSO (GPSO) [17] and cooperative PSO (CPSO) [18] are compared. Initial value of visual parameter and step parameter are considered 50 and 30 percent of the length of fitness function's variables range, respectively (for instance, the length of Akley fitness function's variables range is 64). Maximum number of iterations is 1000 in all experiments. In standard AFSA and proposed algorithm crowd factor is 0.5, and maximum number of attempts (try_number) is 10. In AFSA-CLA, population size in each swarm is 30 and also in standard AFSA the population is 30. Reward and punishment (a and b) coefficients for learning automata in AFSA-CLA assumed 0.01. For first action of learning automat, the value multiplied by visual and step for reducing them is considered 0.9. Performing second action of learning automata resets 20 percent of swarm population. In local rule, if neighbors improve more than 10 times, performed action by automata is punished. For GPSO, c1 and c2 values are 2, inertia weight is linearly from 0.9 to 0.4 is reduced during performance and the populations for 30-dimensional space and 10-dimensional space are considered 150 and 50, respectively. In CPSO, GPSO is applied and in D-dimensional space, D one-dimensional groups cooperate with each other. Inertia weight and other applied parameters adjustments are same as GPSO in CPSO.

Experiments were repeated 100 times and average results, best results and their standard deviation in 10-dimensional space and 30-dimensional space is presented in tables 2 and 3, respectively. In figure 2, average value graph of fitness function in 100 attempts of performing standard AFSA, GPSO, CPSO and proposed algorithm of AFSA-CLA is presented for test functions in 30-dimensional space for 1000 iterations.

As it can be seen in tables 2 and 3, in 10 and 30-dimensional spaces, AFSA-CLA and CPSO results have better efficiency rather than standard AFSA and GPSO, since in both algorithms for cooperative property among various swarms, convergence rate was much higher and their efficiency was not decreased significantly by raising problem space dimensions. Also, standard deviation of AFSA-CLA and CPSO results in various performance was much more acceptable than that of standard AFSA and GPSO, in other words, these two method resistance for cooperative property between different groups caused convergence to global optimum further guaranteed and difference between results of various performances to lesser.

In AFSA-CLA, cellular learning automata acts as the thinking brain of each of swarms and by using of existing collaborative property between groups, it controls behavior of available fishes in each swarm. This causes that global search and local search to be balanced and this algorithm reaches to better results than that of CPSO.

Among tested functions, Griewank function has some properties that algorithms with cooperative properties cannot work on it as the same power in optimizing other functions. This function has much complexity such that algorithms with cooperative property cannot cross local optimums and eventually converge to one of local optimums. So, as it can be seen in tables 2 and 3 and also graph e of figure 2, efficiency of AFSA-CLA and CPSO is less than that of GPSO on Griewank test function, since unlike AFSA-CLA and CPSO that only swarms cooperate each other for optimizing a single vector, in GPSO, there are vectors as the size of group population which is optimized by using GPSO behaviors that increase their ability in crossing local optimums. Among tested functions, step function, is optimized in much less iterations by using CPSO and AFSA-CLA algorithms.

TABLE I. SIX STANDARD TEST FUNCTIONS USED IN THIS PAPER.

Test Function	Search Space	Acceptable Result	Name of Function
$f_1(x)=\sum_{i=1}^{D}x_i^2$	$[-100,100]^D$	0.01	Sphere
$f_2(x)=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	$[-5.12,5.12]^D$	50	Rastrigin
$f_3(x)=\sum_{i=1}^{D}\left(100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right)$	$[-50,50]^D$	100	Rosenbrock
$f_4(x)=\sum_{i=1}^{D}\left(\lfloor x_i+0.5\rfloor\right)^2$	$[-100,100]^D$	0	Step
$f_5(x)=\sum_{i=1}^{n}\left(\frac{x_i^2}{4000}\right)-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600,600]^D$	0.01	Griewank
$f_6(x)=20+e-20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}-e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$	$[-32,32]^D$	0.01	Ackly

TABLE II. COMPARISON OF THE BEST, AVERAGE AND STANDARD DEVIATION OF RESULTS OF 100 PERFORMANCES OF STANDARD AFSA, GPSO, CPSO AND PROPOSED ALGORITHM OF AFSA-CLA ON SIX TEST FUNCTIONS IN 10-DIMENSIONALL SPACE.

Function	Criteria	Std-AFSA	GPSO	CPSO	AFSA-CLA
Sphere	***Best***	0.2354	5.3421e-29	2.4373e-101	1.3675e-105
	Mean	2.0285	4.2920e-25	1.8709e-96	2.6440e-105
	Std-Dev	1.5621	1.7007e-24	4.0624e-96	9.6107e-106
Rastrigin	***Best***	0.0033	1.7860e-09	0	0
	Mean	2.7421	2.6951	0	0
	Std-Dev	2.1856	1.3950	0	0
Rosenbrock	***Best***	9.5455	4.5222e-04	1.1007e-04	1.9865e-04
	Mean	33.0628	1.6954e+04	27.5790	4.3539
	Std-Dev	30.9376	5.7883e+04	13.1006	9.1519
Step	***Best***	0	0	0	0
	Mean	3.7500	0	0	0
	Std-Dev	2.7120	0	0	0
Griewank	***Best***	0.0359	2.5538e-11	4.8849e-15	1.1102e-15
	Mean	0.0953	0.0058	0.0720	0.1031
	Std-Dev	0.0468	0.0235	0.1362	0.1331
Ackly	***Best***	0.9329	2.6645e-15	2.6641e-15	2.6641e-15
	Mean	1.8405	1.2523e-13	5.5067e-15	4.7501e-15
	Std-Dev	0.4888	1.3657e-13	2.7519e-15	2.6473e-15

TABLE III. COMPARISON OF THE BEST, AVERAGE AND STANDARD DEVIATION OF RESULTS OF 100 PERFORMANCES OF STANDARD AFSA, GPSO, CPSO AND PROPOSED ALGORITHM OF AFSA-CLA ON SIX TEST FUNCTIONS IN 30-DIMENSIONALL SPACE.

Function	Criteria	Std-AFSA	GPSO	CPSO	AFSA-CLA
Sphere	***Best***	410.2043	2.9765e-07	4.7452e-98	3.1917e-104
	Mean	728.1900	6.3979e-06	3.9427e-94	5.2775e-104
	Std-Dev	183.2601	1.0412e-05	1.3722e-93	8.5608e-105
Rastrigin	***Best***	77.5419	20.0214	0	0
	Mean	162.0464	38.4314	0	0
	Std-Dev	34.2185	12.2041	0	0
Rosenbrock	***Best***	8.5567e+04	21.9464	0.0016	0.0033
	Mean	1.7816e+05	1.2242e+07	29.2735	9.6354
	Std-Dev	6.6751e+04	3.6004e+07	13.2391	12.7490
Step	***Best***	383	0	0	0
	Mean	667.6500	0	0	0
	Std-Dev	184.6787	0	0	0
Griewank	***Best***	4.0010	1.7090e-05	1.1102e-15	4.4408e-16
	Mean	6.9563	0.0701	0.4278	0.3500
	Std-Dev	1.5048	0.1258	0.3954	0.3737
Ackly	***Best***	5.7284	9.4971e-05	1.3322e-14	1.3322e-14
	Mean	6.8145	7.9296e-04	2.5401e-14	2.3092e-14
	Std-Dev	0.5602	7.0334e-04	6.2695e-15	5.5128e-15

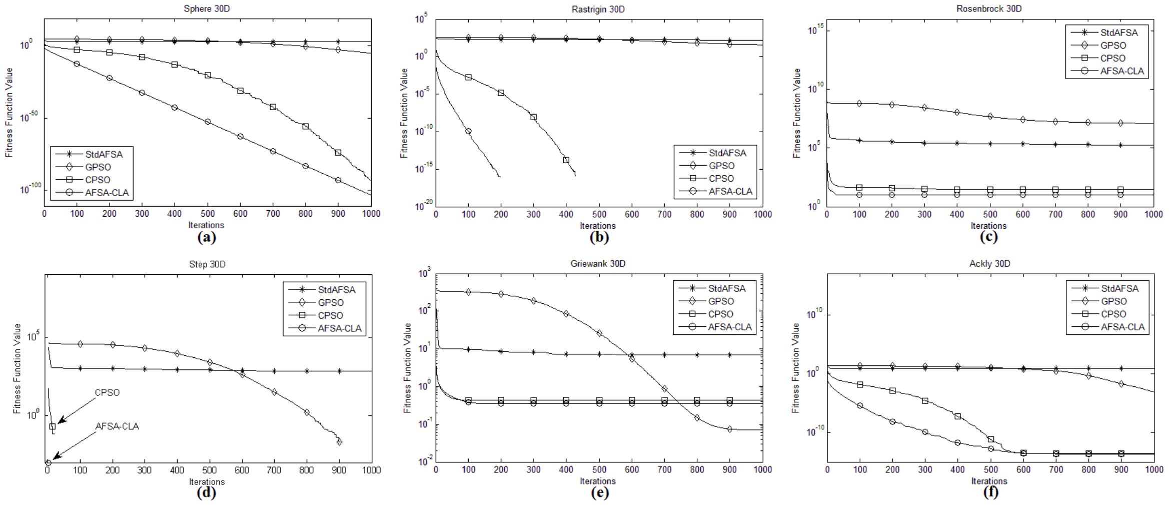

Figure 2. Comparison of results average obtained from 100 performances of standard AFSA, GPSO, CPSO and proposed algorithm of AFSA-CLA on (a) sphere, (b) rastrigin, (c) Rosenbrock, (d) step, (e) griewank and (f) Ackly functions in 30-dimensional space for 1000 iterations.

In general, according to graphs of figure 2, AFSA-CLA has higher convergence rate rather than CPSO based on number of iterations. It should be noted that in AFSA-CLA for existing different swarms and establishing cooperative property between groups to optimize main vector, the size of algorithm computation becomes more but obtained results are much more acceptable and appropriate than that of standard AFSA. In general, experiments show that proposed algorithm of AFSA-CLA has an acceptable efficacy.

VI. Conclusion

In this paper, a new algorithm was proposed for optimization based on two algorithms of artificial fish swarm algorithm (AFSA) and cellular learning automata (CLA). Proposed algorithm allocated one cell of CLA to each dimension of problem space that each of them consisted of an artificial fish swarm. Available learning automata in each cell has the responsibility of making balanced between global search and local search, preventing from preceding convergence and making variety in group. In the proposed algorithm, since every swarm optimizes one dimension and cooperative property exists among swarms, ability and convergence rate of the algorithm is acceptable and a considerable improvement than standard AFSA. Simulating results showed that proposed algorithm is of better efficacy than standard AFSA, GPSO and CPSO.

References

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

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

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

[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] S. Hi, N. Belacel, H. Hamam, and Y. Bouslimani, “Fuzzy Clustering with Improved Artificial Fish Swarm Algorithm,” In International Joint Conference on Computational Sciences and Optimization 09, Vol. 2, pp. 317-321, Hainan, China, 2009.

[7] L. Xiao, “A Clustering Algorithm Based on Artificial Fish School,” in 2nd International Conference on Computer Engineering and Technology, Vol. 7, pp. 766-769, Chengdu, china, aprile 2010.

[8] Y. Luo, J. Zhang, and X. Li, “The Optimization of PID Controller Parameters Based on Artificial Fish Swarm Algorithm,” In IEEE International Conference on Automation and Logistics, pp. 1058-1062, Jinan, china, 2007.

[9] M.Zhang, C.Shao, M.Li, and J.Sun, “Mining Classification Rule with Artificial Fish Swarm,” In 6th World Congress on Intelligent Control and Automation, Vol. 2, pp. 5877-5881, Dalian, China, 2006.

[10] G.Cui, X.Cao, J. Zhou, and Y.Wang, “The Optimization of DNA Encoding Sequences Based on Improved Artificial Fish Swarm Algorithm,” In IEEE International Conference on Automation and Logistics, pp. 1141-1144, Jinan, China, 2007.

[11] K. S. Narendra, and M. A. L. Thathachar, “Learning Automata: An Overview,” Prentice Hall, 1989.

[12] D. Yazdani, and M. R. Meybodi, “AFSA-LA: A New Model for Optimization,” In 15th Annual CSI Computer Conference (CSICC10), Tehran, 2010.

[13] M. N. Howell, T. J. Gordon, and F. V. Brandao, “Genetic Learning Automata for Function Optimization,” in IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, 2002.

[14] M. Sheybani and M. R. Meybodi, “PSO-LA: A New Model for Optimization”, In 11th Annual CSI Computer Conference (CSICC06), 2006.

[15] M. R. Meybodi, H. Beygi, and M. Taherkhani, “Cellular Learning Automata,” In 6th Annual CSI Computer Conference (CSICC01), 2001.

[16] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, “Adaptive Particle Swarm Optimization,” In IEEE Transaction on System, Man and Cybernetics, Part B: Cybernetics, Vol. 39, No. 6, pp. 1362-1381, 2009.

[17] Y.Shi, and R. Eberhart, “A Modified Particle Swarm Optimization,” In IEEE International Conference on Evolutionary Computation Proceedings, pp. 69-73, Anchorage, 1998.

[18] F. van den Bergh, and A. P. Engelbrecht, “A Cooperative Approach to Particle Swarm Optimization,” in IEEE Transaction on Evolutionary Computation, Vol. 8, No. 3, pp. 225-239, June 2004.