Adaptive Parameter Selection in Comperehensive Learning Particle Swarm Optimizer

Mohammad Hasanzadeh, Mohammad Reza Meybodi and Mohammad Mehdi Ebadzdeh

Computer Engineering and Information Technology Department, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

{mdhassanzd, mmeybodi and ebadzadeh}@aut.ac.ir

Abstract. The widespread usage of optimization heuristics such as Particle Swarm Optimizer (PSO) imposes huge challenges on parameter adaption. One variant of PSO is Comprehensive Learning Particle Swarm Optimizer (CLPSO), which uses all individuals' best information to update their velocity. The novel strategy of CLPSO enables population to read from exemplars for specified generations which is called refreshing gap m. In this paper, we develop two classes of Learning Automata (LA) in order to study the learning ability of automata for CLPSO refreshing gap tuning. In the first class, a learning automaton is assigned to the population and in the second one each particle has its own personal automaton. We also compare the proposed algorithm with CLPSO and CPSO-H algorithms. Simulation results show that our algorithms outperform their counterpart algorithms in term of performance, robustness and convergence speed.

Keywords: Particle Swarm Optimizer (PSO), Comprehensive Learning Particle Swarm Optimizer (CLPSO), Learning Automata (LA), parameter adaption.

1 Introduction

Particle Swarm Optimization (PSO) [1], [2] is an iterative optimization approach that optimizes a problem by producing new feasible solutions in the problem space. In each generation of the PSO a new population produces and evaluates through a fitness function. Since the introduction of PSO, many researchers have developed the original framework of PSO. Having a well-known standard algorithm, Bratton and Kennedy defined a straightforward extension of the PSO algorithm in [3]. Comprehensive Learning Particle Swarm Optimizer (CLPSO) [4] is a PSO variant, which uses a new velocity update strategy based on historical best information of all particles. CLPSO preserves the diversity of the swarm and avoids premature convergence.

Learning Automata (LA) [5] are adaptive decision-making machines operating on unknown environment. Recently, many researchers explore the applications of LA in diverse fields such as: Evolutionary Algorithms (EAs) [6], Grid computing and Cloud computing[7], [8], Wireless Sensor Network (WSN) [9] and image processing [10].

adfa, p. 1, 2013.

LA has been successfully applied to various classes of EAs which consist of: PSO, Genetic Algorithm (GA), Differential Evolution (DE), Ant colony Optimization (ACO), Artificial Bee colony (ABC), Artificial Immune System (AIS), Firefly Algorithm (FA) and Imperialist Competitive Algorithm (ICA).

Standard PSO [3] easily get trapped in local minima when solving complex problems. A lot of research has been done on hybridizing PSO and LA so far. Adjusting the PSO parameters [11], controlling PSO population [12] and incorporating Cooperative PSO (CPSO) [13] mechanism into the standard PSO via the usage of a learning automaton [14] were among them. There is a key parameter in CLPSO [4] named the refreshing gap *m*, which allows a particle to learn from its exemplar for certain number of generations. In this paper, we have proposed learning automata based algorithms for adaptive parameter selection in CLPSO. Two categories of CLPSO parameter adaption are proposed. In the first category the refreshing gap is set for whole population while in the second category, each particle adjust its refreshing gap individually. The new CLPSOs with adaptive refreshing gap are tested on several benchmark functions. The experimental results show that the novel learning strategy has been improved the CLPSO's performance.

The rest of this paper is composed as follow: Section 2 reviews CLPSO and standard PSO. Section 3 gives an introduction to LA and a brief note of its applications. Section 4 describes the proposed algorithms based on LA. Section 5 gives the simulation results and performance analysis. And finally, section 6 concludes the paper.

2 Comprehensive Learning Particle Swarm Optimizer (CLPSO) Versus Standard PSO

Premature convergence on multimodal functions and trapping in local minima are two problems of standard PSO [3]. Comprehensive Learning (CL) strategy of CLPSO [4] is aimed to avoid these phenomena by maintaining the population diversity. In the following we introduce the CLPSO algorithms in term of its differences toward standard PSO: 1) In standard PSO each particle simultaneously learns from its best personal position (*pbest*) and best global position of the swarm (*gbest*). While in CLPSO, an exemplar function is used for each particle of the population in which each dimension of particles learns from its corresponding *pbest* or another particle's *pbest* in that dimension. 2) As $f(x) = f([x^1, x^2, \dots, x^D])$, the fitness value of a particle in standard PSO is determined with D variables. While in CLPSO the $f_i = [f_i(1), f_i(2), \dots, f_i(D)]$ function specifies that each dimension of particle learns from which particle's *pbest*. 3) The velocity update equations of the standard PSO and the new CL strategy of CLPSO are different from each other.

3 Learning Automata (LA) Representation

Reinforcement Learning (RL) is the combination of dynamic programming and supervised learning. Learning automaton [5] is an stochastic optimization machine which is one of the branches of Reinforcement Learning (RL) algorithms. An automaton is composed of an action set and a probability vector. After enough interactions with the unknown environment, the optimal action will emerge which has the highest probability. The detailed formulation of Learning Automaton is reported in [5]. Moreover, Fig. 1 sketches how automaton interacts with its corresponding environment and receives reward and penalty signals.

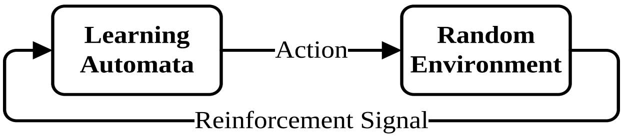

Fig. 1. The interaction between learning automata and environment

Parameter adaption [11] is one of the most sophisticated tasks of EAs. As there are sensitive parameters in PSO, it needs a mechanism to synthesize and analysis them during the evaluation trend. Using LA as the movement engines of the PSO particles is another PSO variant [12]. In this model a learning automaton is designated to each particle and controlled the velocity of the particle. Cooperative PSO [13] is a new version of PSO which assigns an independent swarm to each component of solution vector and optimizes the problem cooperatively. A Hybrid CPSO (CPSO-H) [13] is an algorithm that combines the CPSO algorithm with the PSO [3]. Given that a learning automaton has the ability to learn the structure of benchmark function and that the CPSO-H and PSO algorithms have interleaved execution, it would be ideal to have an adaptive switching mechanism that could exploit both beneficial properties of these two algorithms adaptively. This adaptive approach is called Cooperative PSO – LA (CPSOLA) and is introduced in [14].

4 Proposed Algorithms

4.1 Naïve Tuning of CLPSO's Refreshing Gap

After executing exemplar function of CLPSO algorithm [4], the particles have specific number of generations to learn from their designated exemplar. This counter is refreshing gap *m* which is set to seven for all the benchmark functions. So, this parameter could not scale well with unimodal or multimodal problems. Also, the structure of the objective function may have influence in determining the exact value of this parameter. Fig. 2 shows the refreshing gap changes and its related eligibility. This experiment conducted on Rosenbrock, Quadric, Ackley and Rastrigin benchmark functions with 10 dimensions and 10 particles. All the functions are optimization problems which the lower their fitness values, the better their performance are.

From Fig. 2, we can observe that *m* has a direct impact on problem results. For Rosenbrock, Quadric, Ackley and Rastrigin better results were reached when *m* is around 9, 3, 6 and 2 respectively. But, in CLPSO experiments [4], the refreshing gap *m* is set at seven for all benchmark functions which is a simple choice for this parameter.

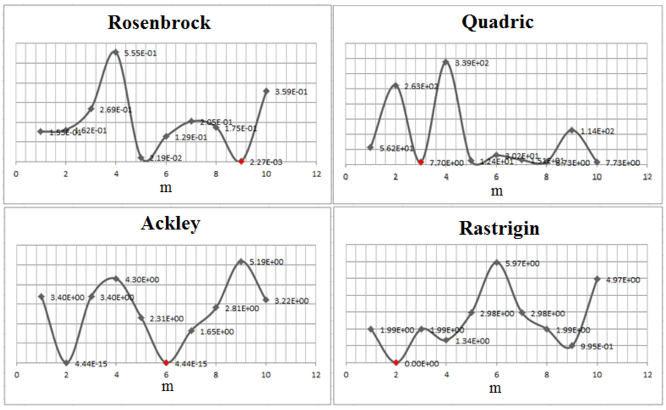

Fig. 2. CLPSO results on four test functions with different refreshing gap

Considering the CLPSO's refreshing gap [4] as a constant in all experiments is a prototype implementation which fades the significance of it. In order to have a reliable vision toward the problem space, we should adaptively adjust this parameter. Fig. 3 depicts that CLPSO's population learn *m* times from exemplar function. In each generation of CLPSO algorithm all particles learn from their exemplar function for seven times. The following scenarios show the low flexibility of CLPSO's refreshing gap.

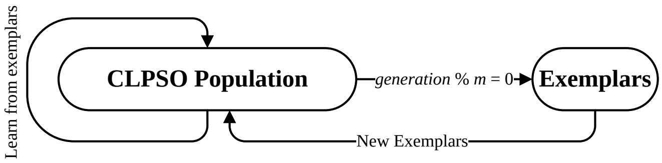

Fig. 3. In every *m* generations new exemplars are reassigned to particles

Scenario 1: In order to perform global search at the early generations of the CLPSO [4], we may consider refreshing gap *m* as a small value. Also while exploiting from local minima, we need larger refreshing gap *m* to perform local search. After exploiting from local minima, a realtime change in diversity is crucial to continue population search path toward global minima (again small *m* value is needed).

Scenario 2: The structure of existing benchmark functions such as unimodal and multimodal problems are coherent to their mathematical equations. For solving these problems, we may need an algorithm which dynamically optimizes the problems. In CLSPO [4] an adaptive mechanism could tune the refreshing gap *m* while solving different benchmark functions instead of considering it as a constant for all problems.

Scenario 3: One could adjust the refreshing gap *m* in the population and particle level. The regulation of this parameter in population level will promise similar explore and exploit abilities for all particles. This uniform refreshing gap *m* solved unimodal problems efficiently, but in order to optimize the multimodal problems more efficiently we will need proper diversity for this parameter. By adjusting this parameter in particles level, we could have proportional refreshing gap *m* for different particles.

4.2 Macroscopic Adaptive CLPSO (MaPSO)

In Macroscopic Adaptive CLPSO (MaPSO) a learning automaton is attached to the whole population. The automaton has the rule of adjusting refreshing gap *m* and has three actions: *increases*, *decreases* and *halts* the *m* value. Each time the automaton chooses an action and set the refreshing gap value, the particles of population will learn from their exemplars for *m* generations. The schematic view of MaPSO is depicted in Fig. 4.

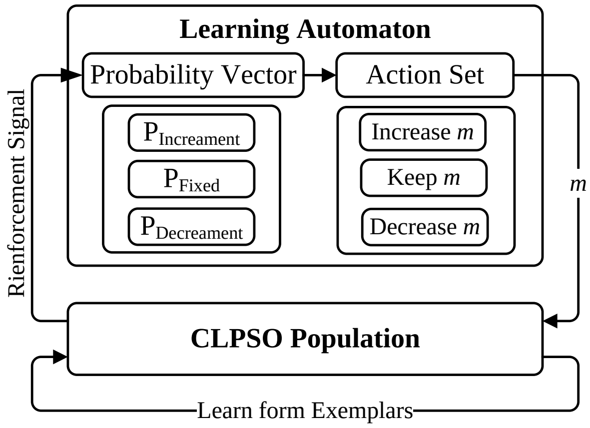

Fig. 4. Structure of the MaPSO

We should define a suitable feedback from CLPSO population for selected action of the automaton. The MaPSO's reinforcement signal is calculated through (1), which *i* is the current generation number. In every *m* generations the *gbest* calculated and compared by previous *gbest* position of population. If the *gbest* fitness improved, then the selected action will get the reward otherwise it will be punished. Adjusting the value of refreshing gap *m*, The MaPSO has tried to balance the global and local search. Also this algorithm could have different search strategies toward solving different objective functions.

$$\beta = \begin{cases} 0 & if \quad fitness\left(gbest_i < gbest_{i-1}\right) \\ 1 & Otherwise \end{cases} \tag{1}$$

4.3 Microscopic Adaptive CLPSO (MiPSO)

The CLPSO individuals exemplars are updated each $m = 7$ iterations, which is not a flexible mechanism for function optimization. The mentioned MaPSO algorithm adjusted the refreshing gap parameter in population level in which all particles read from a common refreshing gap *m*. In multimodal problems where population is scattered on multiple local minima, depending on the position of particles, each particle need a different *m* value to read from its exemplar. Previously in section 4-1, we discussed the necessity of the adaptive refreshing gap *m*.

The Microscopic Adaptive CLPSO (MiPSO) is a variant of CLPSO, which a learning automaton is mounted on each individual. In MiPSO a set of LA is determine the value of refreshing gap for particles. The LA interact to the environment and receive their reinforcement signal from individuals' *pbest*. In multimodal test functions, while the population spread in the problem space, this mechanism will diversify the refreshing gap value. This diversity will be beneficial while trying to optimize complex multimodal problems. The internal structure of MiPSO employing CLPSO is given in Fig. 5.

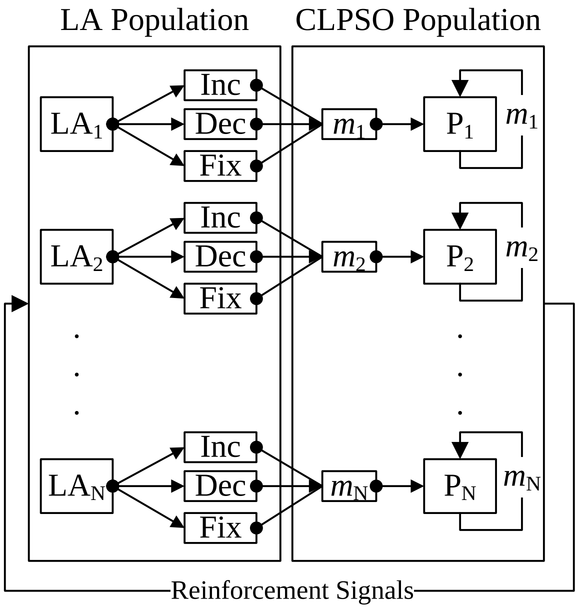

Fig. 5. Structure of the MiPSO where *N* is the population size

In MiPSO each particle learns from its exemplar for *m* generations. If the reminder of division of the current generation number and the particles refreshing gap is equal to zero, then the associating learning automaton will select an action and increase, keep or decrease the value of the refreshing gap *m*. The reward and penalty signals of each automaton are defined based on its associating *pbest*. If the *pbest* of the particle is improved, then the selected action will get the reward otherwise it will be punished. The reinforcement signal is calculating based on (2), which *i* and *j* are representing the current generation and current individual, respectively.

$$\beta = \begin{cases} 0 & if \quad fitness\left(pbest_i^j < pbest_{i-1}^j\right) \\ 1 & Otherwise \end{cases} \tag{2}$$

While giving flexibility to individuals, MiPSO gave different particles flexible exploration and exploitation powers. Population diversity in term of refreshing gap can completely cover the problem space in multimodal problems. In MiPSO each particle could have special exploration and exploitation attributes based on its refreshing gap *m*.

5 Numerical Simulation

The experiments are conducted on TEC 2006 benchmark functions [4] which are proposed to test CLPSO algorithm. Simulations were organized to compare MaPSO and MiPSO with 2 PSO algorithms including CPSO-H [13] and CLPSO [4] on 16 test problems with 10 and 30 dimensions. For 10 dimensional problems, the population size is set at 10 and the maximum number of function evaluations is set at 30000. For the 30 dimensional problems, the population size is set at 40 and the maximum number of function evaluations is set at 200000. For all experiments, 30 independent runs of the MaPSO and MiPSO and the other two algorithms were executed to obtain the average and standard deviation of the results. The results of these simulations are shown in Tables 1-2.

The MaPSO and MiPSO are tested on various kinds of learning algorithms and finally we selected the $L_{R\text{-}P}$ (Linear Reward-Penalty) learning schema with learning parameters *alpha* = *beta* = 0.1. Each automaton corresponding to MaPSO or MiPSO algorithms has an action set which varies the value of refreshing gap. The variation of refreshing gap *m* is in the range of [1,20] and the initial value of refreshing gap is set to 1, for all automata.

Table 1. Fig. **6.** 10-dimensional results after 30000 Function Evaluations

F	**CPSO-H** [13]	**CLPSO** [4]	**MaPSO**	**MiPSO**
f_1	4.98E-45 ± 1.00E-44	5.15E-29 ± 2.16E-28	7.48E-75 ± 3.08E-74	**2.86E-116 ± 1.57E-115**
f_2	**1.53E+00 ± 1.70E+00**	2.46E+00 ± 1.70E+00	3.69E+00 ± 1.56E+00	3.56E+00 ± 2.14E+00
f_3	**1.49E-14 ± 6.97E-15**	4.32E-14 ± 2.55E-14	1.74E+00 ± 1.37E+00	1.64E+00 ± 1.23E+00
f_4	4.06E-02 ± 2.80E-02	**4.53E-03 ± 4.81E+03**	1.72E-02 ± 2.12E-02	3.01E-02 ± 5.03E-02
f_5	1.07E-15 ± 1.67E-15	**0.00E+00 ± 0.00E+00**	6.45E-02 ± 1.74E-01	8.54E-02 ± 2.52E-01
f_6	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	1.03E+00 ± 1.12E+00	1.29E+00 ± 1.61E+00
f_7	2.00E-01 ± 4.10E-01	**0.00E+00 ± 0.00E+00**	1.57E+00 ± 1.19E+00	1.30E+00 ± 1.06E+00
f_8	2.13E+02 ± 1.41E+02	**0.00E+00 ± 0.00E+00**	4.56E+02 ± 2.34E+02	4.88E+02 ± 2.27E+02
f_9	1.36E+00 ± 8.85E-01	**3.56E-05 ± 1.57E-04**	9.78E-01 ± 1.08E+00	8.45E-01 ± 1.14E+00
f_{10}	1.20E-01 ± 8.07E-02	4.50E-02 ± 3.08E-02	**3.23E-02 ± 2.60E-02**	5.01E-02 ± 8.24E-02
f_{11}	4.35E+00 ± 1.35E+00	**3.72E-01 ± 4.40E-01**	8.22E-01 ± 6.44E-01	5.64E-01 ± 5.47E-01
f_{12}	2.67E+01 ± 1.06E+01	**5.97E+00 ± 2.88E+00**	6.43E+00 ± 2.34E+00	6.50E+00 ± 3.04E+00
f_{13}	1.90E+01 ± 9.05E+00	5.44E+00 ± 1.39E+00	**4.93E+00 ± 2.03E+00**	4.51E+00 ± 1.65E+00
f_{14}	9.67E+02 ± 3.67E+02	**1.14E+02 ± 1.28E+02**	9.41E+02 ± 2.48E+02	1.02E+03 ± 3.02E+02

f_{15}	1.65E+02 ± 1.42E+02	**1.64E+01 ± 3.63E+01**	2.42E+01 ± 4.06E+01	3.62E+01 ± 5.52E+01
f_{16}	2.46E+02 ± 2.18E+02	**1.98E+01 ± 2.93E+01**	3.94E+02 ± 7.67E+01	4.11E+02 ± 7.26E+01

Table 2. 30-dimensional results after 200000 Function Evaluations

F	**CPSO-H** [13]	**CLPSO** [4]	**MaPSO**	**MiPSO**
f_1	**1.16E-113 ± 2.92E-113**	4.46E-14 ± 1.73E-14	2.78E-53 ± 1.19E-52	2.30E-64 ± 1.03E-63
f_2	**7.08E+00 ± 8.01E+00**	2.10E+01 ± 2.98E+00	2.31E+01 ± 1.83E+00	2.37E+01 ± 1.17E+00
f_3	4.93E-14 ± 9.17E-14	**0.00E+00 ± 0.00E+00**	1.10E-14 ± 3.00E-15	1.27E-14 ± 2.59E-15
f_4	3.63E-02 ± 3.60E-02	3.14E-10 ± 4.64E-10	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
f_5	7.82E-15 ± 8.50E-15	3.45E-07 ± 1.94E-07	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
f_6	**0.00E+00 ± 0.00E+00**	4.85E-10 ± 3.63E-10	3.32E-02 ± 1.82E-01	**0.00E+00 ± 0.00E+00**
f_7	1.00E-01 ± 3.16E-01	**4.36E-10 ± 2.44E-10**	3.33E-02 ± 1.83E-01	1.00E-01 ± 3.05E-01
f_8	1.83E+03 ± 2.59E+02	**1.27E-12 ± 8.79E-13**	2.40E+02 ± 1.74E+02	1.53E+02 ± 1.43E+02
f_9	2.10E+00 ± 3.84E-01	3.43E-04 ± 1.91E-04	**1.07E-14 ± 3.49E-15**	1.28E-14 ± 2.57E-15
f_{10}	5.54E-02 ± 3.97E-02	**7.04E-10 ± 1.25E-11**	1.08E-08 ± 5.89E-08	1.40E-09 ± 4.78E-09
f_{11}	1.43E+01 ± 3.53E+00	3.07E+00 ± 1.61E+00	9.46E-01 ± 6.52E-01	**7.19E-01 ± 5.42E-01**
f_{12}	1.01E+02 ± 3.53E+00	3.46E+01 ± 1.61E+00	2.97E+01 ± 1.21E+01	**2.87E+01 ± 6.87E+00**
f_{13}	8.80E+01 ± 2.59E+01	3.77E+01 ± 5.56E+00	2.52E+01 ± 6.36E+00	**2.47E+01 ± 8.93E+00**
f_{14}	3.64E+03 ± 7.41E+02	**1.70E+03 ± 1.86E+02**	3.04E+03 ± 4.04E+02	3.16E+03 ± 4.05E+02
f_{15}	1.30E+02 ± 1.64E+02	7.50E-05 ± 1.85E-04	6.50E-04 ± 3.10E-03	**5.01E-07 ± 2.58E-06**
f_{16}	7.83E+01 ± 1.60E+02	**7.86E+00 ± 3.64E+00**	5.21E+02 ± 8.12E+01	5.54E+02 ± 9.90E+01

5.1 10-dimensional Problems

Table 1 shows the averages and standard deviations of the 30 runs of CPSO-H, CLPSO, MaPSO and MiPSO on 16 benchmark functions with 10 variables. The best results are bolded. From the results in Table 1, we can perceive that in Sphere problem (f_1), the MiPSO outperforms the other PSOs. The sphere problem is a simple and convex problem that MiPSO can optimize it faster than other PSOs. In Rosenbrock problem (f_2) the proposed algorithms are got trapped in local minima and the automata couldn't utilize the refreshing gap value properly. Also in unrotated multimodal problems (f_3-f_8), MaPSO and MiPSO couldn't manage the global and local searches simultaneously and the algorithms performance is worse than their rival algorithms. In rotated multimodal problems (f_9-f_{14}), the algorithms that see the problem dimensions independently are failed to optimize these kinds of problems. The MaPSO optimizes 2 out of 6 of rotated problems. Finally in Composition problems CLPSO is the best choice.

5.2 30-dimensional Problems

The same experiments as 10 dimensional problems are performed on 30 dimensional problems and the results showed in Table 2. MaPSO and MiPSO are collectively suppress CPSO-H and CLPSO algorithms on f_4, f_5, f_6, f_9, f_{11}, f_{12}, f_{13}, f_{15} and especially significantly improve the performance on f_4, f_5. The Group A problems are simple unimodal problems which require a simple iterative heuristic to optimize them. Although the number of functions evaluations increased from 30000 to 200000, the proposed algorithms couldn't maintain their performance on simple unimodal problems with respect to 10 dimensional problems. MiPSO algorithm optimizes the f_4,f_5,f_6 better than CLPSO and CPSO-H. Since CPSO-H and CLPSO are considered the variables of problem space independently, in rotated multimodal problems (f_9-f_{14}), because of the fully rotated nature of the problems the performance of CPSO-H and CLPSO drop off and the performance of MiPSO significantly improved. In the two composition problems (f_{15}-f_{16}) with randomly distributed local and global optima, MiPSO significantly improves the results on f_{15}. The set of learning automata associating with refreshing gaps of each particle of MiPSO population could balance the exploration and exploitation attributes of this heuristic. Finally CLPSO performs the best on f_{16}, because of its diverse solutions.

From the results of tables 1 and 2, MaPSO and MiPSO are not the best choices for optimizing unimodal and simple multimodal problems and either they are not solving the 10 dimensional problems efficiently. Since there are many variables which are involved in real world problems and these problems are like multimodal optimization problems with high dimensions, it is reasonable to use an algorithm which performs well on multimodal and high dimensional problems.

6 Conclusion

In this paper we present two variants of Comprehensive Learning PSO (CLPSO) with adaptive learning strategy where learning automata regulate the refreshing gap of CLPSO. Two branches of intelligent algorithms are proposed. In the first category, the same refreshing gap is set for whole swarm while in the other category each particle has its own automaton which adjusts the refreshing gap individually.

According to the results of MaPSO and MiPSO on both 10 dimensional and 30 dimensional problems, we can conclude that the proposed algorithms do not act the best for low dimensional problems. There is a cost for tuning the refreshing gap in order to attain better results in high dimensional problems, and the cost is the slow convergence of the proposed algorithms in 10 dimensional problems. The MaPSO and MiPSO are achieved the best results in three, 10 dimensional problems. However they perform better on rotated multimodal problems where the problem dimensions are correlated together. Also in 30 dimensional problems the proposed algorithms perform the best in 7 out of 16 benchmark functions. From the clear difference of 10 and 30 dimensional problems, we observe that the proposed adaptive strategy enable the MaPSO and MiPSO to make use of the learning ability of the learning automata to optimize the problems more accurately than CPSO-H and CLPSO algorithms.

As long as the surface of 10 dimensional problems is plain, there is no need for extensive search to optimize the problems efficiently. The MaPSO outperforms MiPSO in 10 dimensional problems because of its simple one automaton structure. But in 30 dimensional problems where the problem space is complicated and in some cases the problem space is rotated, the distributed behavior of MiPSO's set of learning automata could appropriately adjust the value of refreshing gap by active feedbacks from the problem's landscape.

References

1. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
2. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: 6th International Symposium on Micro Machine and Human Science, pp. 39-43 (1995)
3. Bratton, D., Kennedy, J.: Defining a Standard for Particle Swarm Optimization. In IEEE Swarm Intelligence Symposium, pp. 120–127 (2007)
4. Liang, J., Qin, A., Suganthan, P. N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transaction on Evolutionary Computing, 10(3), pp. 281–295, (2006)
5. Najim, K., Poznyak, A.S.: Learning automata: theory and applications. Pergamon Press (1994)
6. Hasanzadeh, M., Meybodi, M.R., Ebadzadeh M.M.: Adaptive Cooperative Particle Swarm Optimizer. Applied Intelligence, 39(2), pp. 397-420 (2013)
7. Hasanzadeh, M., Meybodi, M.R.: Grid resource discovery based on distributed learning automata. Computing, pp. 1–14 (2013)
8. Hasanzadeh, M., Meybodi, M.R.: Deployment of gLite middleware: An E-Science grid infrastructure. In 21st IEEE Iranian Conference on Electrical Engineering, pp. 1-6 (2013)
9. Esnaashari, M., Meybodi M.R.: A cellular learning automata-based deployment strategy for mobile wireless sensor networks. Journal of Parallel and Distributed Computing, 71(7), pp. 988-1001 (2011)
10. Abin, A.A., Fotouhi, M., Kasaei, S.: A new dynamic cellular learning automata-based skin detector, Multimedia systems, 15(5), pp. 309-323 (2009)
11. Hashemi, A.B., Meybodi M.R.: A note on the learning automata based algorithms for adaptive parameter selection in PSO. Applied Soft Computing, 11(1), PP. 689-705 (2011)
12. Hasanzadeh, M., Meybodi, M.R., Shiry, S.: Improving Learning Automata based Particle Swarm: An Optimization Algorithm. In 12th IEEE International Symposium on Computational Intelligence and Informatics, pp. 291-296 (2011)
13. Van den Bergh, F., Engelbrecht. A.P.: A Cooperative approach to particle swarm optimization. IEEE Transaction on Evolutionary Computing, 8(3), pp. 225– 239 (2004)
14. Hasanzadeh, M., Meybodi, M.R.: A robust heuristic algorithm for Cooperative Particle Swarm Optimizer: A Learning Automata approach. In 20th IEEE Iranian Conference on Electrical Engineering, pp. 656-661 (2012)