



Trends in
**Applied Sciences
Research**

ISSN 1819-3579



Academic
Journals Inc.

www.academicjournals.com

Dynamic Parameters Optimization for Enhancing Performance and Stability of PSO

¹Amir Khosravani-Rad, ²Ramin Ayanzadeh and ³Elaheh Raisi

¹Department of Applied Mathematics, Parand Branch, Islamic Azad University, Parand, Iran

²Department of Computer Science, Bojnord Branch, Islamic Azad University, Bojnord, Iran

³Department of Computer, Science and Research Branch, Islamic Azad University, Tehran, Iran

Corresponding Author: Ramin Ayanzadeh, Department of Computer Science, University of Maryland, Baltimore County, United State of America

ABSTRACT

In this study, a new method for optimal control of parameters in particle swarm optimization based on fuzzy rules, is presented. In proposed method, to prevent premature convergence, social and personal learning coefficients are updated according to the convergence rate of the algorithm. In other words, fuzzy linguistic variables and membership functions are employed to conduct the swarm toward global optimum point. Several computational simulations are carried out to demonstrate high performance and stability of this method. Simulation results reveal superior optimality and stability and lower computational cost of the new algorithm compared to the traditional metaheuristics such as standard particle swarm optimization, genetic algorithms and standard particle swarm optimization which justifies its advantages for particle swarm optimization algorithms.

Key words: Adaptive parameter adjustment, fuzzy sets, metaheuristics, numerical, optimization, particle swarm optimization, swarm intelligence

INTRODUCTION

Owing to the significant importance of optimization in real world applications, it is widely used in many research areas such as engineering, industry, biology, medicine and agriculture (Setayeshi and Fadaei, 2011; Jang *et al.*, 2005). Optimization problems are defined when words like best, worst, cheapest, lowest, etc are involved and their purpose is to find out the optimal or semi-optimal solution with respect to some goals among all feasible and potential solutions (Shahamatnia *et al.*, 2011). Optimization problems have fundamental importance in mathematics and computer science areas, including multi processor scheduling systems, job-shop scheduling, network and telecommunication routing, large scale integrated circuit design, time-cost trade-off, training neural networks and time tabling (Jang *et al.*, 2005; Russel and Norving, 2006). To solve an optimization problem following steps must be taken.

Problem formulation: In this stage, state space (search space) which involves all states of problem variables or in a looser language, the space of all feasible solutions is created. Each point in the search space represents one potential solution. In training artificial neural network, for instance, system variables are assumed as weights of neurons (Ayanzadeh *et al.*, 2011a;

Jang *et al.*, 2005). In scheduling problems, orders of tasks are system variables and set of all possible orders are task graph. In fact, search space includes all possible and some times impossible permutations (Russel and Norving, 2006).

Defining an objective function: Objective function or cost function receives a point in state space as input and returns a real number as output and then different solutions are compared regarding these outputs. In other words, objective function estimates a quantitative value to measure qualitative features in optimization problems. Purpose of optimization is to find a solution that minimize/maximize objective function. In multi-objective optimization problems, several features must be maximized or minimized to find a solution. These features are mostly incompatible which lead to many difficulties (Ayanzadeh *et al.*, 2009b, 2011b; Marzban *et al.*, 2014; Russel and Norving, 2006).

Finding optimum solution: Optimization is finding the best solution among all possible (and impossible) answers. If the state space size is finite, searching among all potential solutions to find the best one will be performed in acceptable time; however, for large scale problems with many solutions (state space is infinite or continues). Thus, finding optimum solution will be somehow impossible (Haupt and Haupt, 2004; Russel and Norving, 2006). For example, for search space size equal to n , finding optimum solution in real world problems would take several years even with high performance computational machines. In computer science realm, these problems are known as NP-complete (Russel and Norving, 2006). Moreover, optimization methods may have to satisfy some constraints in which parts of state space are infeasible. These constraints are divided into hard and soft constraints. Hard constraints “must” be satisfied; however, soft constraints are “advised” to be satisfied (Ayanzadeh *et al.*, 2009a; Jaberi *et al.*, 2012; Russel and Norving, 2006).

A lot of methods have been introduced to solve optimization problems (Eiben and Smith, 2003; Engelbrecht, 2007; Mitchell, 2002). Most of mathematical methods in applied mathematics compute the 1st and 2nd derivations of objective function (Jang *et al.*, 2005). Obviously, there may be no derivable objective function in real world application. Thus, mathematical approaches would encounter some limitations. To overcome these restrictions, numerical approaches have been proposed including but not limited to Heuristics such as hill climbing and simulated annealing (Cano *et al.*, 2007; Hernandez *et al.*, 2008; Jacobson *et al.*, 2006; Johnson and Jacobson, 2002), metaheuristic methods like evolutionary algorithms (Eiben and Smith, 2003; Haupt and Haupt, 2004; Mitchell, 2002), swarm intelligence techniques (Engelbrecht, 2007), cellular and learning automata (Setayeshi and Fadaei, 2011), artificial immune systems (Engelbrecht, 2007), memetic algorithms (Shahamatnia *et al.*, 2011) are some other examples for numerical optimization algorithms.

Particle Swarm Optimization (PSO) is a popular swarm intelligence method (like honey bees optimization and ant colony optimization) that applies both personal and global/social knowledge to optimize a cost function (Engelbrecht, 2007). However, particle swarm optimization finds more optimum solutions in continues state spaces compared to traditional metaheuristics such as genetic algorithms (Jaberi *et al.*, 2013; Kennedy and Eberhart, 1995).

Beside all advantages that PSO brings, it suffers from some drawbacks such as premature convergence. In other words, despite the fact that PSO serves desirable functionality in continuous search spaces, convergence speed is quiet high so the algorithm is more likely to be trapped in local optimums (Ayanzadeh *et al.*, 2010).

In this sense, several modifications have been performed. Settles and Soule (2005) injected genetic operators such as crossover and mutation into the structure of standard PSO to introduce breeding PSO technique. In the same way, Ratnaweera *et al.* (2004) proposed hierarchical PSO (HPSO) and mutation PSO (MPSO) to prevent premature convergence. Employing a memory buffer to make a delay on impacting global knowledge on particles' position vector was another study that provided desirable enhancement on standard PSO (Xiang *et al.*, 2007). On this basis, in this study, a novel fuzzy based approach is introduced to avoid premature convergence in particle swarm optimization. This approach enjoys some fuzzy sets for optimum and dynamic parameter adjustment. In effect, fuzzy sets are exploited to dynamic control of personal and social learning coefficients.

Particle swarm optimization: Particle swarm optimization has been developed by Kennedy and Eberhart (1995) for problem solving and optimization applications. Particle swarm optimization is a population based technique which has been inspired from social behavior and living style of animals. In some sense, this algorithm is the most reputable swarm intelligent approach for optimization purposes more precisely in continuous state spaces (Engelbrecht, 2007).

Similar to all other metaheuristics and swarm intelligence approaches, PSO method optimizes the objective function in order to find the optimal solution. Swarm (population) is made up of potentially possible solutions, called particles. PSO is initialized with random population as such other numerical optimization techniques. Particles seek state space, during this movement, each particle updates their own velocity and position vectors based on the best experience of their own and the whole population (Engelbrecht, 2007). Updating particles to accelerate the convergence rate is performed similar to arithmetic crossover operator in evolutionary algorithms. Despite PSO is a general metaheuristics for optimization applications, it is mainly applicable for continuous search spaces. In other words, to apply this algorithm in discrete state spaces, some modifications on standard formulation are required (Engelbrecht, 2007).

Each particle in this method consists of position (x), velocity (v) and personal memory vectors. Personal memory keeps track of the best position of each particle since initial iteration. Velocity and position vectors are updated as follow:

$$v_j(t+1) = v_j(t) + r_1 C_1 (x_j^{pBest}(t) - x_j(t)) + r_2 C_2 (x_j^{gBest}(t) - x_j(t)) \quad (1)$$

$$X_j(t+1) = x_j(t) + v_j(t+1) \quad (2)$$

where, $v_j(t)$ represents the velocity of particle j in time t . In the same way, $x_j(t)$ denotes the position of particle j in time t , $x_j^{gBest}(t)$ is the best global position of particle j until time t and $x_j^{pBest}(t)$ is the best personal position of particle j until time t based on its memory, r_1 and r_2 are uniform random numbers, C_1 and C_2 are personal and social learning coefficients which are positive and constant. Equation 1 consists of 3 major parts. The 2nd and the 3rd one specify personal and social learning. Therefore, C_1 and C_2 parameters define personal and social impact of knowledge in optimizing process (Engelbrecht, 2007).

FUZZY BASED ADAPTIVE PARTICLE SWARM OPTIMIZATION

Previous experiments show that when complexity of problem (state space size) increases, efficiency of particle swarm optimization reduces noticeably. In fact, PSO's performance has exponential relationship with complexity of search space. Furthermore, it has been proved that static coefficients (offline parameter adjustment) could not prevent premature convergence (Ratnaweera *et al.*, 2004; Settles and Soule, 2005; Xiang *et al.*, 2007). To deal with this drawback, a novel approach is proposed to enhance the performance and stability of particle swarm optimization by dynamic and online adjusting the personal and social learning coefficients during optimization process.

In this method, rate of convergence in each iteration is compared to the rate of convergence in previous iteration. That is, decrement/increment rate of objective function for global best particle in current iteration comparing to the previous iteration will be calculated. In order to evaluate coefficients, 2 fuzzy sets named "low convergence" and "high convergence" are defined. Figure 1 illustrates membership functions of these two fuzzy sets (Ayanzadeh *et al.*, 2012; Dahmani and Benyettou, 2004; Mamlook, 2006; Gholami *et al.*, 2014; Jaberi *et al.*, 2011).

In the case of low convergence rate, social and personal learning coefficients are increased and decreased, respectively. On the contrary, when convergence rate is high, personal and social learning coefficients are increase and decrease consequently. These modifications are performed on the basis of low and high membership functions.

Clearly, personal and social learning parameters must be initialized in advance, so several scenarios are introduced. For example, learning coefficients can be initialized with constant or random values (usually with uniform distribution). However, according to the online coefficient adjustment, initial values seem to have no remarkable impact on the performance and stability of the algorithm.

EXPERIMENT RESULTS

Simulated results are discussed to evaluate performance of the proposed method. For this purpose, 2 types of experiments have been performed; visual and statistical evaluations.

Visual evaluations: In this section optimization progress for minimizing Ackley function in terms of different variables is shown. Figure 2-5 depict progress of optimizing Ackley function by

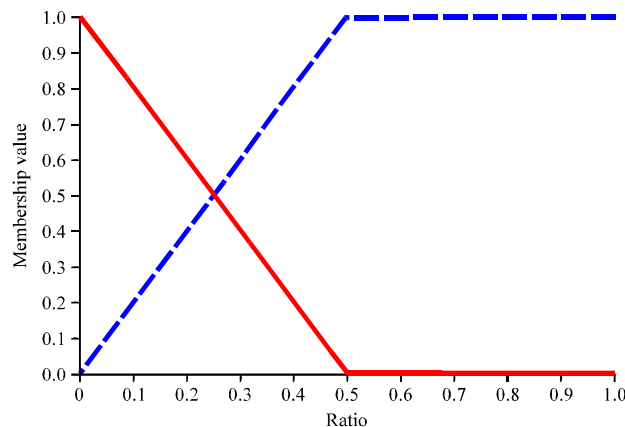


Fig. 1: Membership functions of low and high convergence

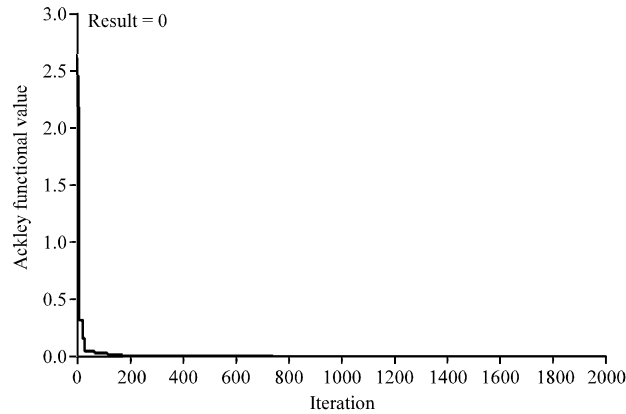


Fig. 2: Minimization process for Ackley function with $n = 2$

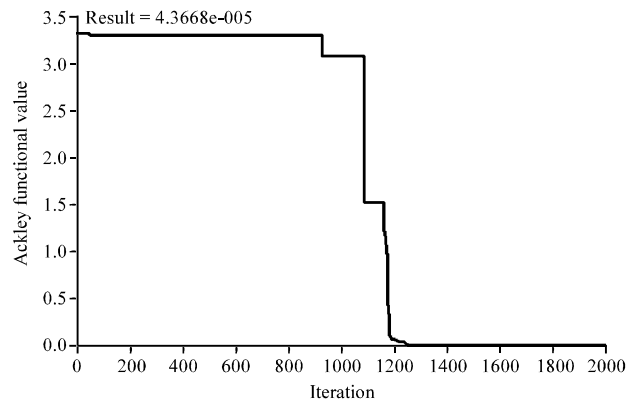


Fig. 3: Minimization process for Ackley function with $n = 5$

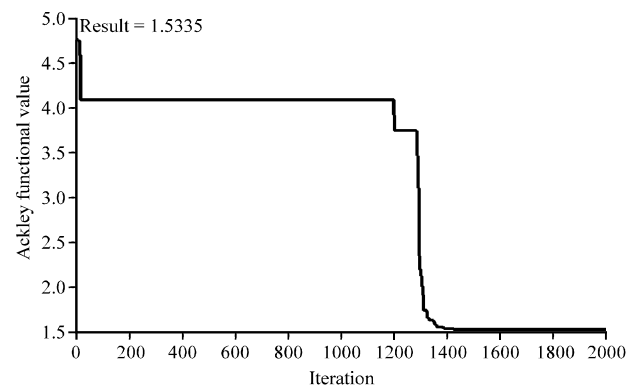


Fig. 4: Minimization process for Ackley function with $n = 10$

employing proposed method for $n = 2, 5, 10$ and 20 , respectively. It can be seen that utilizing fuzzy sets for online and dynamic parameter adjustment in particle swarm optimization ebbs the probability of premature convergence. Strictly speaking, the proposed method subsides velocity of convergence when complexity is rised.

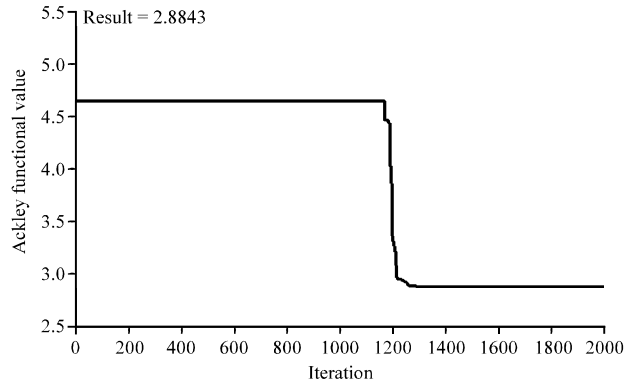


Fig. 5: Minimization process for Ackley function with $n = 20$

Table 1: Statistical indicators for Ackley function optimization with $n = 2$

Parameters	Best	Average	Variance
Genetic algorithms	6.71096e-008	6.7160e-003	2.174508e-004
Particle swarm optimization	0	0	0
Proposed method	0	0	0

Table 2: Statistical indicators for Ackley function optimization with $n = 5$

Parameters	Best	Average	Variance
Genetic algorithms	1.9923e-000	3.8824e-000	4.7600e-000
Particle swarm optimization	2.1463e-002	1.7353e-000	6.2367e-002
Proposed method	7.68315e-006	1.3391e-005	2.0210e-008

Table 3: Statistical indicators for Ackley function optimization with $n = 10$

Parameters	Best	Average	Variance
Genetic algorithms	2.5346e-000	8.3463e-000	5.0087e-000
Particle swarm optimization	1.2276e-000	3.7239e-000	1.1134e-000
Proposed method	1.8459e-002	2.1094e+000	5.2087e-003

Table 4: Statistical indicators for Ackley function optimization with $n = 20$

Parameters	Best	Average	Variance
Genetic algorithms	5.9453e-000	11.3357e-000	7.1146e-000
Particle swarm optimization	4.1324e-002	8.7543e-000	3.1132e-000
Proposed method	2.2938e+000	2.8690e+000	1.5427e-001

Statistical evaluations: Due to the stochastic behaviour of particle swarm optimization, different results will be obtained in every running time. In statistical experiments, performance and stability of proposed approach is compared with other traditional techniques. In this experiment, each simulation is rerun 30 times to extract statistical indicators. Specifically, best (to evaluate accuracy and efficiency), average (to evaluate quality) and variance (to evaluate stability) indicators are estimated.

Table 1-4 explain the evaluated statistical indicators in minimizing multi-variable Ackley function for $n = 2, 5, 10$ and 20 , respectively. As it is shown, new proposed method generates more desirable and stable results comparing to genetic algorithms and standard particle swarm optimization.

CONCLUSION

Particle swarm optimization is a reputable solution for optimization problems which attracted lots of attentions in recent decades thanks to its special properties. PSO with continuous range is more effective, easy to implement. There are a few parameters to be adjusted and no overlapping and mutation calculation required compared to traditional metaheuristics such as genetic algorithm. One major flaw of this method is premature convergence. Expressly, the algorithm estimates local and global optimums at initial iterations, but converges to one of these initial optimums.

In this study, a novel method is presented to dynamically adjust social and personal learning coefficients. In the proposed method, personal learning coefficients are grown when algorithm trapped to a local optimum. When convergence of algorithm is high, social learning coefficients are decreased. In this sense, several computational simulations were performed to verify and validate performance and stability of the proposed method. Simulation results show that proposed new method is more stable and effective in comparison with standard particle swarm optimization and genetic algorithms.

REFERENCES

- Ayanzadeh, R., K. Hassani, Y. Moghaddas, H. Gheiby and S. Setayeshi, 2009a. Innovative approach to generate uniform random numbers based on a novel cellular automata. *J. Applied Sci.*, 9: 4071-4075.
- Ayanzadeh, R., E. Shahamatnia and S. Setayeshi, 2009b. Determining optimum queue length in computer networks by using memetic algorithms. *J. Applied Sci.*, 9: 2847-2851.
- Ayanzadeh, R., K. Hassani, Y. Moghaddas, H. Gheiby and S. Setayeshi, 2010. Multi-layer CA for normal random number generation. *Proceedings of the 18th Iranian Conference on Electrical Engineering*, May 11-13, 2010, Isfahan, Iran.
- Ayanzadeh, R., A.S.Z. Mousavi and S. Setayeshi, 2011a. Fossil fuel consumption prediction using emotional learning in Amygdala. *Proceedings of the 19th Iranian Conference on Electrical Engineering*, May 17-19, 2011, Tehran, Iran, pp: 1-6.
- Ayanzadeh, R., A.S.Z. Mousavi and H. Navidi, 2011b. Honey bees foraging optimization for mixed nash equilibrium estimation. *Trends Applied Sci. Res.*, 6: 1352-1359.
- Ayanzadeh, R., A.S.Z. Mousavi and E. Shahamatnia, 2012. Fuzzy cellular automata based random numbers generation. *Trends Applied Sci. Res.*, 7: 96-102.
- Cano, A., M. Gómez, S. Moral and J. Abellán, 2007. Hill-climbing and branch-and-bound algorithms for exact and approximate inference in credal networks. *Int. J. Approximate Reasoning*, 44: 261-280.
- Dahmani, Y. and A. Benyettou, 2004. Parameter tuning of fuzzy subsets inertia influence in navigation case. *J. Applied Sci.*, 4: 384-387.
- Eiben, A.E. and J.E. Smith, 2003. *Introduction to Evolutionary Computing*. Springer, New York, USA., ISBN-13: 9783540401841, Pages: 199.
- Engelbrecht, A.P., 2007. *Computational Intelligence: An Introduction*. 2nd Edn., John Wiley and Sons Ltd., New York, USA., ISBN-13: 9780470035610, Pages: 597.
- Gholami, A.A., R. Ayanzadeh and E. Raisi, 2014. Fuzzy honey bees foraging optimization: swarm intelligence approach for clustering. *J. Artif. Intell.*, (In Press).
- Haupt, R.L. and S.E. Haupt, 2004. *Practical Genetic Algorithms*. 2nd Edn., John Wiley and Sons, New York, ISBN-13: 9780471455653, Pages: 272.

- Hernandez, D., R. Gras and R. Appel, 2008. Neighborhood functions and hill-climbing strategies dedicated to the generalized ungapped local multiple alignment. *Eur. J. Operat. Res.*, 185: 1276-1284.
- Jaberi, A., R. Ayanzadeh and E. Shahamatnia, 2011. A novel fuzzy cellular automata for uniform random number generation. *Proceedings of the 2nd International Conference on Contemporary Issues in Computer and Information Sciences*, May 31-June 2, 2011, Zanjan, Iran.
- Jaberi, A., R. Ayanzadeh and A.S.Z. Mousavi, 2012. Two-layer cellular automata based cryptography. *Trends Applied Sci. Res.*, 7: 68-77.
- Jaberi, A., R. Ayanzadeh, E. Raisi and A. Sadighi, 2013. Central limit theorem based cellular automata for generating normal random numbers. *Inform. Technol. J.*, 12: 2440-2446.
- Jacobson, S.H., L.A. McLay, S.N. Hall, D. Henderson and D.E. Vaughan, 2006. Optimal search strategies using simultaneous generalized hill climbing algorithms. *Math. Comput. Modell.*, 43: 1061-1073.
- Jang, J.S.R., C.T. Sun and E. Mizutani, 2005. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall of India, New Dehli, India.
- Johnson, A.W. and S.H. Jacobson, 2002. A class of convergent generalized hill climbing algorithms. *Applied Math. Comput.*, 125: 359-373.
- Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. *Proceedings of the International Conference on Neural Networks*, Volume 4, November 27-December 1, 1995, Perth, WA., USA., pp: 1942-1948.
- Mamlook, R., 2006. Fuzzy set methodology for evaluating alternatives to compare between different power production systems. *J. Applied Sci.*, 6: 2117-2125.
- Marzban, F., R. Ayanzadeh and P. Marzban, 2014. Discrete time dynamic neural networks for pre-dicting chaotic time series. *J. Artif. Intell.*, (In Press).
- Mitchell, M., 2002. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA., USA.
- Ratnaweera, A., S.K. Halgamuge and H.C. Watson, 2004. Self-organizing hierarchical particle swarm optimizer with time varying acceleration coefficients. *IEEE Trans. Evol. Comput.*, 8: 240-255.
- Russel, S. and P. Norving, 2006. *Artificial Intelligence: A Modern Approach*. 2nd Edn., Prentice Hall of India Pvt. Ltd., New Dehli, India.
- Setayeshi, S. and A.H. Fadaei, 2011. *CA for Optimization*. Nova Science Publishers Inc., New York, USA., ISBN-13: 9781611227437.
- Settles, M. and T. Soule, 2005. Breeding swarms: A GA/PSO hybrid. *Proceeding of the Conference on Genetic and Evolutionary Computation*, June 25-29, 2005, Washington, DC., USA., pp: 161-168.
- Shahamatnia, E., R. Ayanzadeh, R.A. Ribeiro and S. Setayeshi, 2011. Adaptive imitation scheme for memetic algorithms. *Adv. Inform. Commun. Technol.*, 349: 109-116.
- Xiang, T., K.W. Wong and X. Liao, 2007. A novel particle swarm optimizer with time-delay. *Applied Math. Comput.*, 186: 789-793.