



AN OPERANT CONDITIONING APPROACH FOR LARGE SCALE SOCIAL OPTIMIZATION ALGORITHMS

¹Seyit Alperen CELTEK , ²Akif DURDU 

¹Karamanoğlu Mehmetbey University, Engineering Faculty, Energy Systems Engineering, Karaman, TURKEY

²Konya Technical University, Engineering Faculty, Electric and Electronic Engineering, Konya, TURKEY

¹salperenceltek@kmu.edu.tr, ²adurdu@ktun.edu.tr

(Geliş/Received: 05.11.2020; Kabul/Accepted in Revised Form: 15.12.2020)

ABSTRACT: The changes that positive or negative results cause in an individual's behavior are called Operant Conditioning. This paper introduces an operant conditioning approach (OCA) for large scale swarm optimization models. The proposed approach has been applied to social learning particle swarm optimization (SL-PSO), a variant of the PSO algorithm. In SL-PSO, the swarm particles are sorted according to the objective function and all particles are updated with learning from the others. In this study, each particle's learning rate is determined by the mathematical functions that are inspired by the operant conditioning. The proposed approach adjusts the learning rate for each particle. By using the learning rate, a particle close to the optimum solution is aimed to learn less. Thanks to the learning rate, a particle is prevented from being affected by particles close to the optimum point and particles far from the optimum point at the same rate. The proposed OCA-SL-PSO is compared with SL- PSO and pure PSO on CEC 13 functions. Also, the proposed OCA-SL-PSO is tested for large-scale optimization (100-D, 500-D, and 1000-D) benchmark functions. This paper has a novel contribution which is the usage of OCA on Social Optimization Algorithms. The results clearly indicate that the OCA is increasing the results of large-scale SL-PSO.

Key Words: Operant Conditioning, Large Scale Optimization, Operant Conditioning Approach, Swarm Optimization Algorithm.

Büyük Ölçekli Sosyal Optimizasyon Algoritmaları İçin Edimsel Koşullandırma Yaklaşımı

ÖZ: Olumlu veya olumsuz sonuçların bir bireyin davranışında neden olduğu değişikliklere Edimsel Koşullandırma denir. Bu makale, büyük ölçekli sürü optimizasyon modelleri için bir edimsel koşullandırma yaklaşımı (OCA) sunar. Önerilen yaklaşım, PSO algoritmasının bir varyantı olan sosyal öğrenme parçacık sürüsü optimizasyonuna (SL-PSO) uygulanmıştır. SL-PSO'da sürü parçacıkları amaç işlevine göre sıralanır ve tüm parçacıklar diğerlerinden öğrenilerek güncellenir. Bu çalışmada, her parçacığın öğrenme hızı, edimsel koşullandırmadan esinlenen matematiksel fonksiyonlar tarafından belirlenir. Önerilen yaklaşım, her parçacık için öğrenme oranını ayarlar. Öğrenme oranını kullanarak, optimum çözüme yakın bir parçacığın daha az öğrenmesi amaçlanmaktadır. Öğrenme oranı sayesinde bir parçacığın çözüme yakın partikül ile çözüme uzak partiküllerden aynı oranda etkilenmesinin önüne geçilmektedir. Önerilen OCA-SL-PSO, CEC 13 işlevlerinde SL-PSO ve saf PSO ile karşılaştırılır. Ayrıca, önerilen OCA-SL-PSO, büyük ölçekli optimizasyon (100-D, 500-D ve 1000-D) karşılaştırma işlevleri için test edilmiştir. Bu yazının, Sosyal Optimizasyon Algoritmalarında OCA'nın kullanımı olan yeni bir katkısı vardır. Sonuçlar açıkça OCA'nın büyük ölçekli SL-PSO sonuçlarını artırdığını göstermektedir.

Anahtar Kelimeler: Edimsel koşullandırma, Büyük Ölçekli Optimizasyon, Edimsel Koşullandırma Yaklaşımı, Sürü Optimizasyon Algoritması.

1. INTRODUCTION

Optimization can be called the process of finding the best solution to a problem or system (Celtek *et al.*, 2020) (Pham and Karaboga, 2012). The optimization's main purpose is to find the parameters that achieve the minimum cost and the shortest time (Karaboğa, 2014). The particle swarm optimization (PSO), which is based on a simple mechanism that mimics societal animals' swarming behaviors, is one of the most used to optimize engineering problems. PSO has given successful results in many engineering problems (Cui and Lee, 2013; Yalcin *et al.*, 2015; ASLAN *et al.*, 2018; Eldem and Ülker, 2020; Song *et al.*, 2020). The PSO consists of particles with a position and velocity and represents a candidate solution for the optimization problem (Eberhart and Kennedy, 1995).

While classical PSO can perform well in low dimensional optimization problems, it is weak in large-scale optimization problems and cannot get expected results (Wang *et al.*, 2013). For this reason, many PSO variations have been proposed to improve the PSO's search performance (Clerc and Kennedy, 2002; Premalatha and Natarajan, 2009; Wang *et al.*, 2020).

The social learning particle swarm optimization (SL-PSO) is one variant of the PSO algorithm (Cheng and Jin, 2015). Social learning plays an important role in behavior learning in the swarm. It provides particles to learn behaviors from other's experiences without incurring the costs of particle trials-and-errors. Unlike the studies in the literature recommended improving the PSO's search performance, no global and local best values are kept in SL-PSO. Instead, the swarm particles are sorted according to the objective function, and all particles except the best are updated with learning from other better particle. Unlike the classic PSO, the update process in SL-PSO is achieved by learning all the particles from the others.

In this study, each particle's learning rate is determined by the mathematical functions that are inspired by the operant conditioning. The changes that pleasant and unpleasant consequences cause in individuals' behavior are called Operant Conditioning (Holland and Skinner, 1961). If an individual's behavior results in something pleasant, the individual tends to do it repeatedly.

It is suggested an operant conditioning approach (OCA) to SL-PSO. The main goal of the proposed method is to arrange the learning rate. This method provides the necessary learning for every particle. For instance, one of the particles with good results is intended to learn less and not be affected by the worst particles. If not, in the bad particle case, the learning rate will be high and the particle will be trained more.

This study consists of four sections. In Section 2, the mathematical background of particle update in PSO and SL-PSO methods is given. Then, Section 3 gives information about operant conditioning -the inspiration source for this work- and explains operant conditioning's mathematical modeling. Section 4 gives results in different dimensions of CEC 13 functions. Of proposed approach. In Section 5, the study is terminated by comparing the proposed approach with pure PSO and SL-PSO.

2. PSO and SL-PSO

The PSO, developed by Eberhart and Kennedy in 1995, is a population-based metaheuristic optimization technique (Eberhart and Kennedy, 1995). PSO, which is inspired by birds, is assumed that each particle can update its position and velocity by using global and local best values. Detailed coverage of PSO can be found in (Eberhart *et al.*, 2001).

The position and velocity equations (1-2) for a conventional PSO are as follows;

$$V_i(t+1) = \omega V_i(t) + c_1 R_1(t)(pbest_i(t) - X_i(t)) + c_2 R_2(t)(gbest(t) - X_i(t)) \quad (1)$$

$$X_{i,j}(t+1) = X_{i,j}(t) + V_i(t+1) \quad (2)$$

In these equations, t , $V_i(t)$ and $X_i(t)$ indicate respectively the number of iterations, the speed of the i th particle and the position of the i th particle. Also, ω is the inertia weight, c_1 and c_2 are acceleration vectors, and R_1 and R_2 are random vectors generated at 0-1 interval.

As mentioned above, classical PSO can perform well in low dimensional optimization problems. Unfortunately, as the dimension of the problem increases, the PSO success rate falls. The social learning particle swarm optimization (SL-PSO), which is one variant of the PSO algorithm, has proposed improving the search performance (Cheng and Jin, 2015).

Like the classical PSO, in the SL-PSO approach, a randomly initialized initial vector (X_{ij}) is created, which first constitutes a candidate solution. Then, the swarm particles are sorted according to the ascending order of the fitness function results. As a result, every particle learns from other particles (as it is in the same social learning mechanism) and corrects its behavior. The social learning between the particles takes place as follows;

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}(t) + \Delta X_{i,j}(t+1), & \text{if } p_i(t) \leq P_i^L \\ X_{i,j}(t), & \text{otherwise} \end{cases} \quad (3)$$

where $X_{(i,j)}(t)$ is the particle that learns, X_{ij} is the particle that teaches, P_i is the probability of learning. In detail, $\Delta X_{i,j}(t+1)$ is constructed as follows;

$$\Delta X_{i,j}(t+1) = r_1(t)\Delta X_{i,j}(t) + r_2(t)I_{i,j}(t) + r_3(t)C_{i,j}(t) \quad (4)$$

$$I_{i,j}(t) = X_{k,j}(t) - X_{i,j}(t) \quad (5)$$

$$C_{i,j}(t) = \bar{X}_j(t) - X_{i,j}(t) \quad (6)$$

As can be seen in Equation 4, there are three components for updating to position in SL-PSO. First one is $X_{i,j}(t)$ the same as the inertia component in the canonical PSO. Second component is the called as demonstrators $I_{i,j}(t)$ (Equation 5). As mentioned above, SL-PSO uses demonstrators for position correction (Equation 6) instead of gbest or pbest in PSO. The third one is called social influence factor. There are three random coefficients r_1 , r_2 and r_3 , which will be randomly generated within [0; 1].

Specifically, the j -element in the behavior vector of the particle i ; $X_{ij}(t)$ simulates $X_{kj}(t)$, the j -th element in the behavior vector of the particle k . In SL-PSO, the particle learns from different indicators in the current swarm, and this behavior is controlled randomly.

As the higher the search size, the harder it is to solve the problem, and the lower the likelihood of a particle being willing to learn from others. In SL-PSO, the relationship between learning possibility and problem dimensionality is inversely related to the following relation;

$$P_i^L = \left(1 - \frac{i-1}{m}\right)^{\alpha \log\left(\frac{n}{M}\right)} \quad (7)$$

Where P_i^L is the learning probability for each particle i , M is the dimension of problem, m is the swarm size, n is the number of the dimensionality and α is the smooth coefficient ($\alpha = 0.5$).

3. OPERANT CONDITIONING and OCA-SL-PSO

The main challenge of social swarm methods is learning from just better particles can cause the results to stick to the local optimum points. This study aims at a social learning network that a particle learns from other particles with the learning rate. Thus, the learning rate controls each particle's learning from the other particles.

It is inspired by the actual condition used in social life for the determination of learning rates. In operant conditioning, for every situation that an individual has experienced before, positive or negative consequences arise. These positive or negative consequences arise from past experiences and are largely influential in their future behavior. The changes that positive or negative consequences lead to an individual's behavior are called "Operant Conditioning." If the individual has already had an attitude and has a positive outcome due to this behavior, he or she is directed to the same behavior again. Similarly, if

the individual has had a previous behavior and has received a negative reaction due to this behavior, he/she is striving for similar behavior.

In this study, the operant conditioning is applied to the swarm, not to the individual. In this way, it is possible to benefit from the experiences of all the individuals. The pseudo code of the OCA-SL-PSO is given in Algorithm 1.

Algorithm 1. The pseudo code of the OCA-SL-PSO

- 1 Initialization
 - 2 $t = 0$ (generation index), $n =$ dimension.
 - 3 $M = 100$ (swarm size).
 - 4 Divide into groups like in SL-PSO [11]
 - 5 Set the social impact factor like in SL-PSO [11]
 - 6 Create a solution vector for each group.
 - 7 Sort solutions from large to small.
 - 8 Set the best solutions as a pointer.
 - 9 Find the best, worst and average value of each particle.
 - 10 Calculate the learning rate of each particle.
 - 11 Train the particles.
 - Calculate P_{best} .
 - Calculate the random coefficients.
 - Train the particles with new components.
 - 12 Run the objective function for the solution and create the solution vector.
 - 13 If the FEs number is not completed, go back to Step 7.
 - 14 End.
-

The learning rates in this study are determined by a function that depends on the best distance and the particle's worst distance. If the particle yields a good result, there are two situations; in the first case, the particle should achieve better by imitating the better particles. Another case, the result of the particle is that it is local rather than global, which is overcome by imitating worst particles. The learning rate are defined as P_{best} ;

$$P_{best} = \frac{|X_i - X_{best}|}{|X_{best} - X_{worst}|} \quad (8)$$

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}(t) + \Delta X_{i,j}(t+1), & \text{if } p_i(t) \leq P_{best} \\ X_{i,j}(t), & \text{otherwise} \end{cases} \quad (9)$$

As mentioned above, Equation 9 is used for finding $\Delta X_{i,j}(t+1)$. In our scenario, the coefficients will be randomly generated within [0,1] under one condition. The condition is;

$$r_2(t) > r_3(t) > r_1(t) \quad (10)$$

4. RESULTS

The proposed approach in this study was tested with CEC 13 functions. The CEC 13 benchmark functions consist of 28 benchmark functions. 13 of 28 benchmark functions are using in this paper. The used functions are given in Table 1. The proposed approach first tries with $n = 100$ dimensions and FEs = 3E6 and the results are given in Table 2. The number of fitness evaluations (FEs) is the number of times you evaluate a solution in the meta-heuristic algorithm. The proposed approach has tried with $n = 500$ dimensions and FEs = 3E6 and the results are given in Table 3. The program has been run 10 times in order to get all results on the tables.

Table 1. CEC 13 Test Functions

Number	Function Name
f1	Sphere Function
f2	Rotated High Conditioned Elliptic Function
f3	Rotated Bent Cigar Function
f4	Rotated Discus Function
f5	Different Powers Function
f6	Rotated Rosenbrock's Function
f7	Rotated Schaffers F7 Function
f8	Rotated Ackley's Function
f9	Rotated Weierstrass Function
f10	Rotated Griewank's Function
f11	Rastrign's Functions
f12	Rotated Rastrign's Function
f13	Non-Contious Rotated Rastrign Function

Table 2. OCA-SL-PSO 100 D Results, FEs=6E5

	Best	Worst	Mean	STD
f1	0,00E+00	0,00E+00	0,00E+00	0,00E+00
f2	6,27E-229	4,07E-221	4,07E-222	0,00E+00
f3	9,18E+00	2,53E+04	1,36E+04	9,26E+03
f4	3,21E-36	9,47E-34	1,68E-34	2,96E-34
f5	9,57E+01	3,06E+02	1,38E+02	6,44E+01
f6	0,00E+00	2,00E+00	7,00E-01	8,23E-01
f7	5,72E-02	8,28E-02	6,97E-02	8,30E-03
f8	5,30E+03	8,58E+03	6,80E+03	1,01E+03
f9	7,96E+01	2,02E+02	1,12E+02	3,59E+01
f10	1,33E-14	2,04E-14	1,62E-14	2,80E-15
f11	0,00E+00	1,48E-02	2,22E-03	4,99E-03
f12	4,71E-33	4,71E-33	4,71E-33	0,00E+00
f13	1,35E-32	1,47E-32	1,39E-32	5,95E-34

Table 3. OCA-SL-PSO 500 D Results, FEs=6E5

	Best	Worst	Mean	STD
f1	5,88E-27	7,32E-11	7,32E-12	2,31E-11
f2	9,59E-01	3,39E+01	2,64E+01	1,26E+01
f3	4,46E+05	9,57E+05	6,58E+05	1,78E+05
f4	5,96E+01	7,82E+01	7,00E+01	5,18E+00
f5	1,07E+03	1,77E+03	1,36E+03	2,38E+02
f6	1,00E+01	7,50E+02	2,66E+02	2,87E+02
f7	8,06E-01	9,67E-01	8,80E-01	5,57E-02
f8	3,84E+04	4,41E+04	4,07E+04	2,11E+03
f9	1,46E+03	2,22E+03	1,94E+03	2,46E+02
f10	9,83E-01	2,30E+00	1,67E+00	4,56E-01
f11	6,66E-16	1,01E-02	2,56E-03	4,17E-03
f12	2,89E+00	6,29E+00	4,58E+00	1,08E+00
f13	7,79E-23	6,78E-02	9,07E-03	2,12E-02

The proposed approach model is tested for $n = 1000$ dimensions and FEs=1,2E5, FEs=6E5, FEs=3E6 and the results are given in Table 4, Table 5 and Table 6, respectively.

Table 4. OCA-SL-PSO 1000 D Results, FEs=1,2E5

	Best	Worst	Mean	STD
f1	2,09E+05	3,16E+05	2,81E+05	4,27E+04
f2	1,00E+20	1,00E+20	1,00E+20	
f3	9,48E+06	1,28E+07	1,09E+07	1,61E+06
f4	8,26E+01	8,71E+01	8,53E+01	1,77E+00
f5	1,39E+07	3,61E+07	1,94E+07	9,35E+06
f6	2,03E+05	2,85E+05	2,43E+05	3,81E+04
f7	1,80E+02	6,64E+02	4,61E+02	2,31E+02
f8	3,93E+05	4,02E+05	3,99E+05	3,48E+03
f9	1,21E+04	1,24E+04	1,23E+04	1,36E+02
f10	1,91E+01	1,92E+01	1,92E+01	6,51E-02
f11	2,20E+03	3,41E+03	2,63E+03	4,73E+02
f12	6,90E+06	1,72E+07	1,12E+07	4,07E+06
f13	2,35E+07	5,77E+07	3,99E+07	1,49E+07

Table 5. OCA-SL-PSO 1000 D Results, FEs=6E5

	Best	Worst	Mean	STD
f1	1,15E+04	2,43E+04	1,84E+04	4,65E+03
f2	1,00E+20	1,00E+20	1,00E+20	
f3	4,85E+06	9,65E+06	6,97E+06	2,20E+06
f4	8,05E+01	8,54E+01	8,28E+01	1,87E+00
f5	1,30E+04	4,37E+05	9,95E+04	1,89E+05
f6	8,60E+03	2,95E+04	1,67E+04	8,69E+03
f7	1,68E+00	1,85E+00	1,78E+00	6,68E-02
f8	2,69E+05	2,94E+05	2,80E+05	1,08E+04
f9	8,28E+03	1,20E+04	1,05E+04	1,90E+03
f10	1,56E+01	1,72E+01	1,64E+01	7,30E-01
f11	1,11E+02	1,95E+02	1,53E+02	3,27E+01
f12	1,19E+02	1,29E+04	3,24E+03	5,42E+03
f13	1,42E+03	2,53E+03	1,80E+03	4,28E+02

Table 6. OCA-SL-PSO 1000 D Results, FEs=3E6

	Best	Worst	Mean	STD
f1	4,33E+00	5,17E+02	1,80E+02	1,84E+02
f2	1,00E+20	1,00E+20	1,00E+20	
f3	2,12E+06	3,80E+06	2,68E+06	5,99E+05
f4	7,69E+01	8,21E+01	8,01E+01	1,59E+00
f5	4,35E+03	6,41E+03	5,00E+03	5,65E+02
f6	7,91E+02	6,77E+03	2,80E+03	1,93E+03
f7	1,29E+00	1,52E+00	1,42E+00	9,11E-02
f8	1,08E+05	1,21E+05	1,13E+05	4,71E+03
f9	3,25E+03	6,38E+03	4,78E+03	1,12E+03
f10	5,19E+00	6,37E+00	5,66E+00	4,47E-01
f11	1,33E+00	1,53E+01	4,71E+00	5,98E+00
f12	5,90E+00	7,48E+00	6,73E+00	6,25E-01
f13	2,26E+00	4,42E+00	3,63E+00	1,19E+00

The comparison between the proposed approach in Table 7 and SL-PSO (Cheng and Jin, 2015) and classical PSO (Zambrano-Bigiarini *et al.*, 2013) is given. It is clear from Table 7 that the proposed approach

achieves a more efficient result than the standard PSO or SL-PSO. Moreover, the SL-PSO and PSO data in the table are the results of 50-dimensional studies. The success of the proposed method comes to the forefront if SL-PSO and pure PSO are driven from the idea that the results will get worse when operated in more sizes. From the 13 test functions, OCA-SL-PSO produces the most successful result in 9 functions. The proposed 500-dimensional approach works better than SL-PSO and PSO at 8 out of 13 test functions. The proposed 1000-dimensional approach is more successful than SL-PSO and PSO in five of the 13 test functions.

Table 7. The Comparison Results

	Proposed (n=50)	Proposed (n=100)	Proposed (n=500)	SL-PSO (n=50)	PSO (n=50)
f1	0,00E+00	0,00E+00	5,88E-27	7,19E-13	-1,400E+3
f2	0,00E+00	6,27E-229	9,59E-01	5,90E+5	3,776E+05
f3	4,12E+00	9,18E+00	4,46E+05	2,13E+07	1,99E+07
f4	2,11E-117	3,21E-36	5,96E+01	3,11E+6	3,113E+04
f5	8,61E+00	9,57E+01	1,07E+03	6,23E-11	-1,000E+03
f6	0,00E+00	0,00E+00	1,00E+01	2,33E0	-8,816E+02
f7	0,00E+00	5,72E-02	8,06E-01	1,51E0	-7,439E+02
f8	1,65E+04	5,30E+03	3,84E+04	3,44E-02	-6,790E+02
f9	3,54E+00	7,96E+01	1,46E+03	1,75E0	-5,548E+02
f10	0,00E+00	1,33E-14	9,83E-01	1,06E-1	-4,999E+02
f11	0,00E+00	0,00E+00	6,66E-16	5,87E0	-2,498E+02
f12	0,00E+00	4,71E-33	2,89E+00	2,11E+1	-1,378E+02
f13	1,25E-227	1,35E-32	7,79E-23	2,45E+1	1,196E+02

5. CONCLUSION AND FUTURE WORKS

This study proposes a new approach to swarm-based optimization methods inspired by operant conditioning. The focus point is creating better learning for each particle. Thus, the proposed approach adjusts the learning rate for each particle. The learning rate encourages a particle close to the optimum solution to learn from better particles and learn less from the worse particle. Thanks to the learning rate, a particle is prevented from being affected by particles close to the optimum point and particles far from the optimum point at the same rate. The experiment results on CEC 13 functions clearly prove that our strategy is successful for large scale problem.

Furthermore, we are now investigating our approach on different swarm-based methods like as applied to SL-PSO. We will also apply the proposed method to real engineering problems. In our future work, we will consider other colony-based optimization techniques

6. ACKNOWLEDGEMENT

Authors are thankful to RAC-LAB (www.rac-lab.com) for providing the trial version of their commercial software for this study.

REFERENCES

- Aslan, S., Aksoy, A. and Gunay, M., 2018, Performance of parallel artificial bee colony algorithm on solving probabilistic sensor deployment problem, *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 1-5.
- Celtek, S. A., Durdu, A. and Alı, M. E. M., 2020, Real-time Traffic Signal Control with Swarm Optimization Methods, *Measurement*.

- Cheng, R. and Jin, Y. J. I. S., 2015, A social learning particle swarm optimization algorithm for scalable optimization, 291, 43-60.
- Clerc, M. and Kennedy, J., 2002, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, 6 (1), 58-73.
- Cui, C.-Y. and Lee, H.-H., 2013, Distributed traffic signal control using PSO based on probability model for traffic jam, In: *Intelligent Autonomous Systems 12*, Eds: Springer, p. 629-639.
- Eberhart, R. and Kennedy, J., 1995, A new optimizer using particle swarm theory, *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43.
- Eberhart, R. C., Shi, Y. and Kennedy, J., 2001, *Swarm intelligence*, Elsevier, p.
- Eldem, H. and Ülker, E., 2020, A Hierarchical Approach Based on ACO and PSO by Neighborhood Operators for TSPs Solution, *International Journal of Pattern Recognition and Artificial Intelligence*, 2059039.
- Holland, J. G. and Skinner, B. F., 1961, *The analysis of behavior: A program for self-instruction*.
- Karaboğa, D., 2014, *Yapay Zeka Optimizasyon Algoritmaları*, Nobel Akademik Yayıncılık, p.
- Pham, D. and Karaboga, D., 2012, *Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks*, Springer Science & Business Media, p.
- Premalatha, K. and Natarajan, A., 2009, Hybrid PSO and GA for global maximization, *Int. J. Open Problems Compt. Math*, 2 (4), 597-608.
- Song, B., Wang, Z. and Zou, L., 2020, An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve, *Applied Soft Computing*, 106960.
- Wang, H., Sun, H., Li, C., Rahnamayan, S. and Pan, J.-S., 2013, Diversity enhanced particle swarm optimization with neighborhood search, *Information Sciences*, 223, 119-135.
- Wang, Z.-J., Zhan, Z.-H., Kwong, S., Jin, H. and Zhang, J., 2020, Adaptive Granularity Learning Distributed Particle Swarm Optimization for Large-Scale Optimization, *IEEE transactions on cybernetics*.
- Yalcin, N., Tezel, G. and Karakuzu, C., 2015, Epilepsy diagnosis using artificial neural network learned by PSO, *Turkish Journal of Electrical Engineering & Computer Sciences*, 23 (2), 421-432.
- Zambrano-Bigiarini, M., Clerc, M. and Rojas, R., 2013, Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements, *2013 IEEE Congress on Evolutionary Computation*, 2337-2344.