International Journal of Engineering & Technology, 7 (3) (2018) 1725-1729

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i3.14742
Research paper

Solving large-scale problems using multi-swarm particle swarm approach

Sinan Q. Salih [1, 2] *, AbdulRahman A. Alsewari [1]

[1] IBM Centre of Excellence, Faculty of Computer Systems & Software Engineering, University Malaysia Pahang, 26300, Gam bang, Pahang, Malaysia
[2] Computer Science Department, College of Computer Science and Information Technology, University of Anbar, Ramadi, Iraq
**Corresponding author E-mail: sinan_salih@computer-college.org*

Abstract

Several metaheuristics have been previously proposed and several improvements have been implemented as well. Most of these methods were either inspired by nature or by the behavior of certain swarms such as birds, ants, bees, or even bats. In the metaheuristics, two key components (exploration and exploitation) are significant and their interaction can significantly affect the efficiency of a metaheuristic. How-ever, there is no rule on how to balance these important components. In this paper, a new balancing mechanism based on multi-swarm approach is proposed for balancing exploration and exploitation in metaheuristics. The new approach is inspired by the concept of a group(s) of people controlled by their leader(s). The leaders of the groups communicate in a meeting room where the overall best leader makes the final decisions. The proposed approach applied on Particle Swarm Optimization (PSO) to balance the exploration and exploitation search called multi-swarm cooperative PSO (MPSO). The proposed approach strived to scale up the application of the (PSO) algorithm towards solving large-scale optimization tasks of up to 1000 real-valued variables. In the simulation part, several benchmark functions were performed with different numbers of dimensions. The proposed algorithm was tested on several test functions, with four different number of dimensions (100, 500, and 1000) it was evaluated in terms of performance efficiency and compared to standard PSO (SPSO), and master-salve PSO algorithm. The results showed that the proposed PSO algorithm outperformed the other algorithms in terms of the optimal solutions and the convergence.

Keywords: *Particle Swarm Optimization; Multi-Swarm Optimization; Meeting Room Approach; Large-Scale.*

1. Introduction

In most real-world optimization tasks, finding optima requires an expensive computation process. Some of the study limitations such as computation resource constraints and project time requirements have necessitated the need to make optimization process less complicated and rapid [1]. Most of the standard optimization frameworks require several function evaluations and usually produce satisfactory results due to their inherent special information transfer mechanism of using several first-choice solutions in a range of fitness evaluations. The need to evaluate each candidate resolution makes these processes to demand much computing resources and execution time. Consequently, efforts have been devoted to the development of efficient optimization algorithms for the evaluation of several functions. Several novel approaches have been proposed in recent times with some satisfactory performances with fewer function evaluations [2-3].

The nature-inspired algorithms are computational frameworks that are inspired by natural occurrences, while swarm intelligence refers to a set of social patterns which are pre-determined by some principles. Most individuals do not behave wisely enough when alone; however, when in a group, they can have a better behavior under swarm cooperation [4-5]. Swarm intelligence (SI) is inspired by the decentralized collective intelligence of self-organized systems.

A swarm refers to a population of individuals that interacts in a solution space to achieve a global objective. The intelligence of the swarm depends on the network of interaction among the participants, as well as their individual interaction with their environment. In the swarm, the individuals generally strive to move toward the center of the population on critical dimensions. This behavior results in the convergence of the individual on an optimum [6].

The particle swarm optimization (PSO) algorithm is one of the researched nature-inspired (NI) swarm intelligence framework [7–9]. The PSO is inspired by hunting or living styles of birds and fish. It has found application in the handling of complex optimization tasks. It was introduced by [8] and since its introduction, it has undergone several modifications which result in several PSO variants which are aimed at finding a better way of handling specific optimization problems. There are four categories of the modifications on the PSO variants: The first category of modification is centered on the parameter settings with emphasis on the inertia weight and acceleration coefficients parameters optimization. In the second variants, the emphasis is on the topology of the neighborhood which expresses the inter-particle connectivity. The third category of modification focus on the learning strategies with an emphasis on teaching and peer learning of the bests and global best positions of the particles. The fourth category of modification is mainly on the hybridization of the PSO variants with other optimization frameworks [10–12].

The metaheuristics in general, and NIs in particular consist of two main components - exploration or diversification and exploitation or intensification. Exploration is the ability of an algorithm to search for new solutions/individuals far from the current best solution in the search space (represents the global search). On the other

hand, exploitation is the ability of an algorithm to search the surrounding area near the current best solution (represents the local search). There is a need to maintain a good balance between these two algorithmic components. This is because, an optimization algorithm with a good exploration capability, but a poor exploitation power can only successfully escape local optimums but cannot converge optimally to the global optimum. Contrarily, metaheuristics with a good exploitation capability but a poor exploration power merely can escape local optimums, hence, has a low chance of finding a global optimum when challenged with a problem with numerous local optimums[6-13].

In this paper, a new multi-swarm cooperative scheme to balance the exploration and exploitation is proposed. The proposed scheme consists of several swarms called 'clans', each clan has its own leader, the leader is the best solution in the clan which represents the local best. All leaders periodically meet, and the best leader among them represents the global best solution, which has the ability to control them and lead them to best positions. The interaction between the best leader – global best – and the normal leaders – local best – influence the balance between the exploration and exploitation and maintain a suitable diversity in the population, even when it is approaching the global solution, thus reducing the risk of converging to local sub-optima. The proposed algorithm is used to solve the large scale problems, which are the optimization problems with large number of variables – or dimensions.

The rest of the paper is organized as follows. Section 2 describes the original PSO (SPSO) along with its variants. Section 3 motivates and describes the proposed approach and gives the pseudocode of the MPSO algorithm. Section 4 defines the benchmark continuous optimization problems used for experimental comparison of the algorithm, and the discussions of the results. The conclusion are given in Section 5.

2. Preliminaries

2.1. Standard particle swarm optimization

The standard PSO is initialized with randomly selected solutions while the algorithm strives to establish an optimum by updating the generations. The potential PSO solutions are called particles; these particles are assumed to move in a D Dimension solution space at a velocity that is adjusted dynamically based on the particles' experience and the experience of its neighbors. In the PSO, the i_{th}particle is expressed as$x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, where $x_{id} \in [LB_d, UB_d]$,$d \in [1, D]$, LB_d, UB_d respectively represent the lower and upper limits of the d_{th}dimension. The particles' velocity i expressed as $v_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD})$, and it is associated with the particles' maximum velocity V_{max} which is predetermined by the user. The particles, in each time step t, are coordinated based on the following relations:

$$v_i(t+1) = v_i(t) + r_1 c_1\left(P_i - x_i(t)\right) + r_2 c_2\left(P_g - x_i(t)\right) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t) \quad (2)$$

Where r_1 and r_2 represents the randomly selected values in the range of 0 and 1. $c_1$and c_2 represents the acceleration constants of the particle. These constants coordinate the movement of the particle per iteration. P_i represents the previous best position of the ith particle. There are 2 versions of the PSO based on the definition of P_g. The global PSO version is obtained if P_g represents the best position (also called gbest) among the swarm, but if P_g is sourced from a smaller number of adjacent particles in the swarm (also called lbest), then, the version is known as a local PSO version. Later, an inertia term w was introduced [14] through the modification of Equation 1:

$$v_i(t+1) = w \times v_i(t) + r_1 c_1\left(P_i - x_i(t)\right) + r_2 c_2\left(P_g - x_i(t)\right) \quad (3)$$

The inertia term was introduced with the assumption that its suitable selection can ensure a trade-off between local and global explorations. Such trade-off will ensure finding a sufficiently optimal solution using fewer iterations. The inertia term w at its introduction was set based on the following relation:

$$w = w_{max} - \frac{w_{max} - w_{min}}{itr_{max}} \times itr \quad (4)$$

where w_{max} and w_{min} respectively represents the initial and final weights, itr_{max} represents the maximum allowable iteration number, and itr refers to the existing iteration number. This PSO version is later referred to as a linearly decrease inertia weight method (LPSO) in this paper.

Besides LPSO, a random inertia weight factor was also introduced for dynamic systems tracking [15]. This factor is set to vary randomly based on the relation:

$$w = 0.5 - \frac{rand()}{2} \quad (5)$$

Where rand() represents a uniformly random distributed number ranging from0 and 1. It is advised that the value of the acceleration coefficients be maintained at 1.494. Later in this report, this method is referred to as random weight method (RPSO). The generalized PSO pseudocode is presented in the figure below.

Algorithm 1: SPSO

```
Input: #Particles, #MaxItr, c1, c2
Initialize all Particles
Calculate inertia weight via eq. 5
Evaluate the fitness for all Particles
While (itr ≤ #MaxItr )
    For each particle p in Particles
        Calculate the velocity for p via eq. 3
        Calculate the new position via eq. 2
        Evaluate the fitness for all Particles
        If p.Fitness is better than pBest.Fitness Then
            pBest = p
        End
    Next p
    gBest = Determine the best in Particles
End While
```

2.2. Challenges of PSO

The PSO is a simple social model which has attracted several research interests since inception in 1995 due to its ease of implementation and simplicity [16]. Although PSO has undergone several developmental modifications, it is still prone to the following problems which demand to be addressed in future studies:

- Premature convergence: The PSO ends up searching early best solutions and this is predominant in multimodal functions.
- Convergence speed: The PSO establishes the best solution in the early search stage but get stagnated in the process of exploiting the global solution.
- Quality of solution. The PSO produces low-quality solutions due to inherent problem complexity, multimodality, and discontinuity. Uncertainty of solutions. The stochastic nature of PSO makes it produce different solutions in different runs.
- Update strategy: The PSO has a simple solution update strategy and hence, cannot achieve better solutions in complex environments.

3. Meeting room approach

The core idea of multi-swarm is the interaction between several groups while searching for a solution. Many multi-swarm schemes have been proposed, each idea inspired by a natural behaviour. In this paper, a new cooperative multi-swarm scheme inspired by the human social behaviour is proposed. Here, the interaction is between groups of people known as 'Clans' and their leaders. The proposed scheme consists of several swarms called 'clans'; each

clan consists of several solutions which represent the group members. The best member is designated as the leader of a clan; the leader controls the members of the clan in terms of the time to move or where to explore. Figure 1 depicts the structure of the individual swarm.

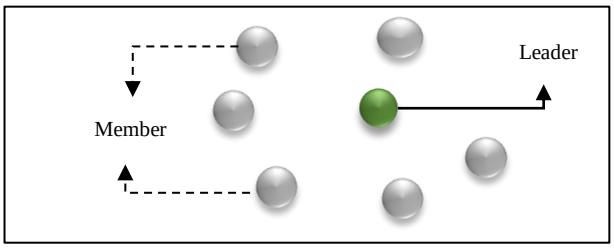

Fig. 1: The Structure of the Individual Swarm.

In each generation, the leaders meet in a room where the overall best leader update the positions of the other normal leaders based on his own positional information. This behaviour of knowledge sharing helps to balance the exploration stage with the exploitation stage of the PSO. The new multi-swarm approach is called 'Meeting Room Approach' (MRA). Figure 2 depicts the MRA model. In this figure, each clan performs a single PSO search, including positional and velocity updating, as well as new local population generation. Having established the new generations for all the clans, each clan sends the leader 'best solution' to the meeting room. The best among all the leaders in the meeting room is selected as the overall best leader. The new overall best leader shares his positional information with the ordinary leaders using the following equations:

$$w^{Ln} = \left(\frac{w^{Lg} - w^{Ln}}{Itr}\right) \times rand() \quad (6)$$

$$v_i{}^{Ln}(t+1) = w^{Ln} \times v_i{}^{Ln}(t) + rc\left(P_g{}^{L} - P_n{}^{L}(t)\right) \quad (7)$$

$$x_i{}^{Ln}(t+1) = x_i{}^{Ln}(t) + v_i{}^{Ln}(t) \quad (8)$$

Where Ln represents the normal leaders, Lg represents the overall best leader, $x_i{}^{L}$ represents the position of the normal leader, $v_i{}^{Ln}$ represents the velocity of the normal leader, w^{Lg} and w^{Ln} represent the inertia weight of the best leader and the normal leader, respectively.

After each generation, a new leader is chosen for each swarm due to the changes in the positions of the members. The new equation of the inertia in the meeting room controls the exploration of the search algorithm. The pseudo-code of the proposed Multi-Swarm Particle Swarm Optimisation (MPSO) algorithm is presented in Figure 3.

Algorithm 2: MPSO

Input: $\#Swarm, \#Particles, \#MaxItr, c_1, c_2, \#Dim$
Initialize all $Leaders$ and their $Swarms$
Calculate inertia weight via $eq.5$
Evaluate the fitness for all $Particles$
While ($itr \leq \#MaxItr$)
  For each c in $Clans$
    For each particle p in $Particles$
      Calculate the velocity for p via $eq.3$
      Calculate the new position via $eq.2$
      Evaluate the fitness for all $Particles$
      If $p.Fitness$ is better than $pBest.Fitness$ Then
        $pBest$ = p
      End
    Next p
  Next c
  Determine the best $Leader$ among all $Leaders$
  Update the inertia weight of the Clan c via $eq.6$
  Update the velocity of the $Leader_c$ via $eq.7$
  Update the position of the $Leader_c$ via $eq.8$
  Determine the $gLeader$ ever as the global best.
End While
Return $Best\ Leader$

4. Results

This results of the benchmarking evaluations commonly used in the evolutionary literature are presented in this section [17]. Each test function varies in terms of modality (unimodal and multimodal) and the number of dimensions (fixed and dynamic). Table 1 shows these test functions with their characteristics.

The performance of the MPSO was evaluated by comparing with that of original PSO (SPSO) [18] and Master-Slave PSO (MCPSO)[19]. The parameters used for SPSO were recommended by [18] with asymmetric initialization method and a linearly decreasing w (changes from 0.9 to 0.4). Several swarms of SPSO were involved in the MPSO and MCPSO (as clans and slaves respectively) during the benchmark function optimization. Both MPSO and MCPSO have the same parameter settings as SPSO1. To investigate the efficiency of the proposed MPSO, different population sizes with different dimensions were used for each function. The maximum iteration number was set to 500, corresponding to the dimensions 100, 500, and 1000. The experiments were conducted for a total of 30 settings. Table 2 presents the parameters setting for all the evaluated algorithms.

Table 1: Benchmark Test Functions

f_n	Function	U_b, L_b	Opt.
f_1	$f_1 = \sum_{i=1}^{D} X^2$	-100, 100	0
f_2	$f_2 = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	-600, 600	0
f_3	$f_3 = \sum_{i=1}^{D} (x_i^2 - 10\ cos(2\pi x_i) + 10\ )$	-5.12, 5.12	0
f_4	$f_4 = -20\,exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2 0}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right) + 20 + e$	-32, 32	0

Table 2: Parameters Settings

Alg.	Parameter	Value
SPSO	W	0.9 – 0.4
	No. of Swarms	1
	c_1,c_2	1.5
	Swarm Size	50
MCPSO	W	0.9-0.6
	No. of Slaves	5
	c_1, c_2, c_3	1.5
	Swarm Size	50
MPSO	w^{Ln}	0.8 – 0.5
	w^{Lg}	0.9 – 0.7
	c_1,c_2	1.5
	No. of Clans	5
	Clan Size	10

The best and mean fitness values of the particles after 30 experimental runs over [4] benchmark functions are presented in Table 3. From the table, MPSO performed better than the benchmarking algorithms in almost all the cases. Generally analysing the table, MPSO has [5] swarms; each swarm consists of 10 particles but only [5] particles are interacting in the meeting room. Hence, it can be said that the MPSO has less computational complexity and a better performance in terms of finding the best solution. Figure 3.a and Figure 3.b illustrate the ability of the MPSO to evolve in situations that the algorithms may have been trapped

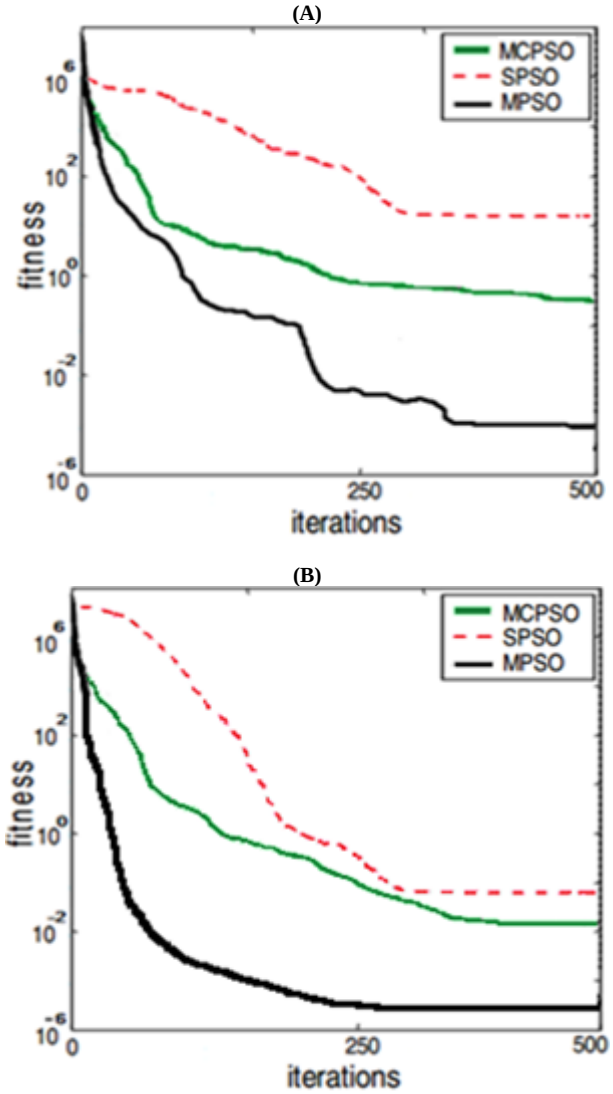

Fig. 3: Convergence Curve: A) Sphere Function B) Griewank Function

Table 3: Results

Dim	f_n	Alg.	Best	Mean	S.D
100	f_1	SPSO	2.5457521	2.7647845	0.0784516
		MCPSO	0.9854126	1.0154784	0.0014784
		MPSO	0.0007845	0.0008748	0.0000184
	f_2	SPSO	0.0884741	0.0964587	0.0078478
		MCPSO	0.0078414	0.0087789	0.0009874
		MPSO	0.0000897	0.0000997	0.0000658
	f_3	SPSO	21.695847	27.947512	0.0847896
		MCPSO	2.0018977	2.6647845	0.0078487
		MPSO	0.0004687	0.0045214	0.0000144
	f_4	SPSO	16.4875218	26.110161	0.0238484
		MCPSO	1.99847	2.5869124	0.0084578
		MPSO	0.0002648	0.0017636	0.0000584
500	f_1	SPSO	7.2456571	2.7647845	0.0784516
		MCPSO	2.4859157	1.0154784	0.0014784
		MPSO	0.0026472	0.0008748	0.0000184
	f_2	SPSO	1.2785781	0.0964587	0.0078478
		MCPSO	0.9045472	0.0087789	0.0009874
		MPSO	0.0041816	0.0000997	0.0000658
	f_3	SPSO	48.995751	27.947512	0.0847896
		MCPSO	7.2214945	2.6647845	0.0078487
		MPSO	0.0784457	0.0045214	0.0000144
	f_4	SPSO	37.125475	26.110161	0.0238484
		MCPSO	3.35847	2.5869124	0.0084578
		MPSO	0.1778499	0.0017636	0.0000584
1000	f_1	SPSO	16.422422	2.7647845	0.0784516
		MCPSO	4.7923729	1.0154784	0.0014784
		MPSO	1.0749752	0.0008748	0.0000184
	f_2	SPSO	3.0899761	0.0964587	0.0078478
		MCPSO	5.2574914	0.0087789	0.0009874
		MPSO	0.9177297	0.0000997	0.0000658
	f_3	SPSO	21.695847	27.947512	0.0847896
		MCPSO	2.0018977	2.6647845	0.0078487
		MPSO	0.9563589	0.0045214	0.0000144
	f_4	SPSO	16.4875218	26.110161	0.0238484
		MCPSO	1.99847854	2.5869124	0.0084578
		MPSO	0.48758311	0.0017636	0.0000584

5. Conclusion

This paper presents a social inspired mechanisms designed to improve the performance of Particle Swarm Optimizer (PSO). The proposed mechanism simulates the behavior of group of people – clans – and the interactions between their leaders. The proposed algorithm MPSO is capable of controlling the balance between exploration and exploitation, and it has been used in order solve large-scale problems with three different sizes of dimensions (100, 500 and 1000). Four benchmark functions were performed in the simulation part using different algorithms. The performance comparisons indicate that MPSO is superior to SPSO in both the high quality of the solution and the robustness of results. In future works, other well-known metaheuristics can be enhanced by using the proposed multi-swarm approach, such as firefly algorithm, grey wolf optimizer, and bat algorithm. The proposed model in this paper may enhance their balancing between the exploration and exploitation. In addition, MRA can be applied as a tuning framework for finding the optimal values of the controlling parameters for the above mentioned metaheuristics.

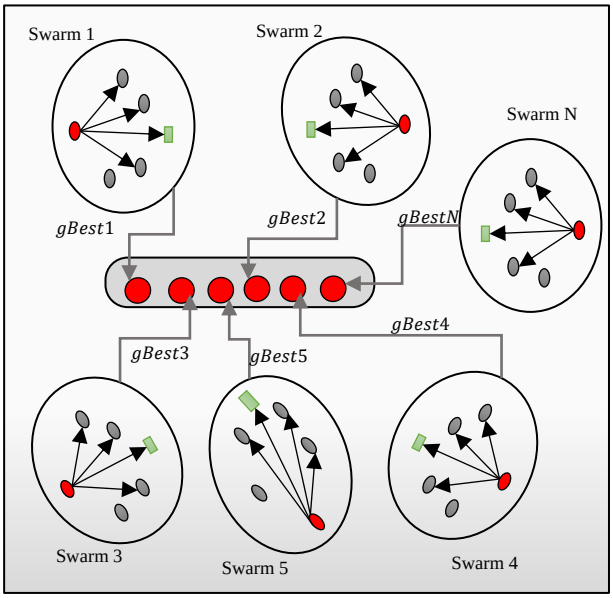

Fig. 2: The Structure of Meeting Room Approach.

Acknowledgement

This research is funded by UMP PGRS170338: Analysis System based on Technological YouTube Channels Reviews, and UMP RDU180367 Grant: Enhance Kidney Algorithm for IOT Combinatorial Testing Problem.

References

[1] I. Boussaïd, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Inf. Sci. (Ny).237 (2013) 82–117. https://doi.org/10.1016/j.ins.2013.02.041.

[2] I. Fister, X.S. Yang, J. Brest, D. Fister, A brief review of nature-inspired algorithms for optimization, Elektroteh. Vestnik/Electrotechnical Rev. 80 (2013) 116–122.

[3] A. Slowik, H. Kwasnicka, Nature Inspired Methods and Their Industry Applications – Swarm Intelligence Algorithms, IEEE Trans. Ind. Informatics.14 (2018) 1004–1015. https://doi.org/10.1109/TII.2017.2786782.

[4] X.S. Yang, Nature-Inspired Optimization Algorithms, 2014. https://doi.org/10.1016/C2013-0-01368-0.
[5] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, 2010. https://doi.org/10.1016/B978-0-12-416743-8.00005-1.
[6] X.S. Yang, S. Deb, Y.-X. Zhao, S. Fong, X. He, Swarm intelligence: past, present and future, Soft Comput. (2017). https://doi.org/10.1007/s00500-017-2810-5.
[7] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms, Appl. Soft Comput. J. 18 (2014) 261–276. https://doi.org/10.1016/j.asoc.2013.09.018.
[8] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, MHS'95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci. (1995) 39–43. https://doi.org/10.1109/MHS.1995.494215.
[9] J. Kennedy, R. Eberhart, Particle swarm optimization, Neural Networks, 1995. Proceedings. IEEE Int. Conf. 4 (1995) 1942–1948 vol.4. https://doi.org/10.1109/ICNN.1995.488968.
[10] İ.B. Aydilek, A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems, Appl. Soft Comput. J. 66 (2018) 232–249. https://doi.org/10.1016/j.asoc.2018.02.025.
[11] K. Premalatha, a M. Natarajan, Hybrid PSO and GA for Global Maximization, Int. J. Open Probl. Compt. Math.2 (2009) 597–608. Doi: 1998-6262.
[12] D. Chen, J. Chen, H. Jiang, F. Zou, T. Liu, An improved PSO algorithm based on particle exploration for function optimization and the modeling of chaotic systems, Soft Comput.19 (2015) 3071–3081. https://doi.org/10.1007/s00500-014-1469-4.
[13] X.S. Yang, Metaheuristic optimization: algorithm analysis and open problems, in: Int. Symp. Exp. Algorithms, 2011: pp. 21–32. https://doi.org/10.1007/978-3-642-20662-7_2.
[14] Y. Shi, R. Eberhart, a Modified Particle Swarm Optimizer, Evol. Comput. Proceedings, 1998. IEEE World Congr. Comput. Intell. 1998 IEEE Int. Conf. (1998) 69–73. https://doi.org/10.1109/ICEC.1998.699146.
[15] Y. Eberhart, R. C. and Shi, Tracking and optimizing dynamic systems with particle swarms, in: Proc. IEEE Congr. Evol. Comput. IEEE, Seoul, Korea, 2001: pp. 94–97.
[16] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Micro Mach. Hum. Sci. 1995. MHS'95. Proc. Sixth Int. Symp., 1995: pp. 39–43. https://doi.org/10.1109/MHS.1995.494215.
[17] M. Jamil, X.S. Yang, H.J.D. Zepernick, Test Functions for Global Optimization: A Comprehensive Survey, in: Swarm Intell. Bio-Inspired Comput. 2013: pp. 193–222. https://doi.org/10.1016/B978-0-12-405163-8.00008-9.
[18] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, Evol. Comput. 1999. CEC 99. Proc. 1999 Congr.3 (1999) 1945–1950 Vol.3. https://doi.org/10.1109/CEC.1999.785511.
[19] B. Niu, Y. Zhu, X. He, H. Wu, MCPSO: A multi-swarm cooperative particle swarm optimizer, Appl. Math. Comput.185 (2007) 1050–1062. https://doi.org/10.1016/j.amc.2006.07.026.