Iranian Journal of Fuzzy Systems
Volume 17, Number 1, (2020), pp. 65-75

Adaptive particularly tunable fuzzy particle swarm optimization algorithm

N. Bakhshinezhad[1], S. A. Mir Mohammad Sadeghi[2], A. R. Fathi[3] and H. R. Mohammadi Daniali[4]

[1,2,3,4] *Department of Mechanical Engineering, Babol Noshirvani University of Technology, Mazandaran, Iran, P.O. Box: 484.*
[3] *Department of Mechanical Engineering, Babol Noshirvani University of Technology, Shariati Ave., Babol, Mazandaran, Iran.*

n.bakhshinezhad@yahoo.com, salimmsadeghi@yahoo.com, fathi@nit.ac.ir, mohammadi@nit.ac.ir

Abstract

Particle Swarm Optimization (PSO) is a metaheuristic optimization algorithm that owes much of its allure to its simplicity and its high effectiveness in solving sophisticated optimization problems. However, since the performance of the standard PSO is prone to being trapped in local extrema, abundant variants of PSO have been proposed by far. For instance, Fuzzy Adaptive PSO (FAPSO) algorithms have been being studied extensively in recent years. In this study, a modified version of PSO algorithms is presented and is named as Adaptive Particularly Tunable Fuzzy Particle Swarm Optimization (APT-FPSO). In it, the global and personal learning coefficients of every single particle are tuned adaptively and particularly, at an individual extent, within each iteration with the aid of fuzzy logic concepts. Ample statistical evidence is provided indicating that the proposed algorithm further improves the potentialities and capabilities of the standard PSO.

Keywords: Particle Swarm Optimization (PSO), fuzzy logic, meta-heuristics.

1 Introduction

Proposed by Kennedy and Eberhard in 1995, Particle Swarm Optimization (PSO) algorithm is among swarm-based metaheuristic optimization algorithms which mimic the social behavior of insects, birds, fishes, etc. in a stochastic, yet intelligent, manner [9]. What has made PSO a successful algorithm and appealing to scientists of the relevant fields, is the particles' being able to communicate and learn simultaneously, yet maintaining a very simple mechanism while doing so. However, this algorithm is still prone to being trapped in local extrema. With the aim of avoiding premature convergence, a growing number of variants of the standard PSO have been being proposed so far. Among them are those that dynamically adapt the global and personal learning coefficients to achieve a fair, well-balanced paradigm toward conducting exploration and exploitation. Clerc and Kennedy proposed a generalized model of PSO algorithm containing a set of coefficients that control the convergence tendencies of the algorithm[8]. Yue et al. represented the swarm model by iterated functions systems and derived the dynamic trajectory of the swarm [22]. Liu et al. investigated the chaotic dynamic characteristics of the swarm model, and they calculated Lyapunov exponent and correlation dimension of the dynamic system [12].

Studied extensively in the recent years, fuzzy set strategies have been increasingly being used for dynamic adaptation of parameters. Ameli et al. have proposed a Fuzzy Discrete Harmony Search (FDHS) algorithm to adjust harmony memory considering rate and pitch adjusting rate [3]. Amador-Angulo and Castillo presented a Fuzzy Bee Colony Optimization (FBCO) algorithm with dynamic adaptation ability in order to control the trajectory stability of a unicycle mobile robot [1]. By having a fuzzy system change dynamically the setting parameters of Bee Colony Optimization (BCO) algorithm, Caraveo et al. presented a Fuzzy BCO algorithm [5]. They concluded that FBCO performed better in comparison with traditional BCO. Amador-Angulo and Castillo have compared statistically Type-1 and interval Type-2 Fuzzy Logics in adapting the setting parameters of BCO in order to tune membership functions of a fuzzy logic

Corresponding Author: A. R. Fathi
Received: July 2018; Revised: January 2019; Accepted: April 2019.

controller [2]. They found that, compared to classical BCO, the both Fuzzy BCO algorithms performed better, with a higher diversity of solutions, in solving the test problem. Olivas et al. proposed an Ant Colony Optimization (ACO) algorithm with a fuzzy system for its parameter adaptation. In doing so, they tried to take control over balancing between exploration and exploitation [16]. Pérez et al. introduced a new approach for dynamic adaptation of a Bat Algorithm's parameters using a fuzzy system [17]. Castillo et al. introduced a Fuzzy Differential Evolution (FDE) algorithm with the goal that a fuzzy system dynamically adjust the crossover and mutation parameters of a Differential Evolution (DE) algorithm, [6]. Cheng and Prayogo presented a Fuzzy Adaptive Teaching-learning-based Optimization (FATLBO) algorithm by performing three modifications on the standard TLBO [7].

In addition to the works reviewed so far, a large number of similar studies have been conducted to improve the performance of the standard PSO algorithm. Valdez et al. have written a survey article on coupling fuzzy logic with bio-inspired optimization algorithms regarding dynamic parameter adaptation of them [19]. Niknam has applied a combination of Fuzzy Adaptive Particle Swarm Optimization (FAPSO) algorithm and Nelder-Mead simplex algorithm [14]. Juang et al. have introduced an Adaptive Fuzzy PSO algorithm that was further enhanced by being been incorporated with Quadratic Interpolation, called as AFPSO-QI [11]. The results showed that AFPSO-QI algorithm offered better solutions with higher accuracy and efficiency compared to six other powerful variants of PSO. Yang and Liao have proposed a new Mutation Fuzzy Adaptive PSO (MF-APSO) algorithm [21]. It was shown that the proposed method could mitigate the overvoltage matter more effectively and highly reduced the total line loss. Tang et al. surmounted the problem of optimal operation and schedule model of active distribution networks (ADNs) using a novel algorithm called Kriging Model Assisted Modified Fuzzy Adaptive PSO (KMA-MFAPSO) [18]. Comparison of the results obtained with those obtained by the others, confirmed the great robustness and fast convergence rate of KMA-MFAPSO. Olivas et al. have analyzed the effects of different membership functions types- i.e. triangular; Gaussian; trapezoidal; and generalized bell- in an interval type-2 fuzzy system in tuning the acceleration parameters of PSO algorithm [15]. They found that using different types of membership functions would result in solutions with almost the same accuracy. Melin et al. have improved the standard PSO through fuzzy logic with the aim of designing optimal fuzzy classifiers [13]. The proposed algorithm was, then, evidenced to be of greater potential applicability compared to the standard PSO. Aminian and Teshnelab have introduced a Fuzzy PSO algorithm in which the inertia weight, global, and personal learning coefficients were adjusted individually, called Fuzzy Logic Controller On Particles Level (FLCOPL) [4, 23].

In this work, a modified Fuzzy PSO algorithm is presented in order to further improve upon the performance of the standard PSO. The major contribution of this work is to develop a strategy in which the learning coefficients of the standard PSO are tuned particularly; that is, within each iteration, the fuzzy operator assigns learning coefficients to "each" of the individual particles with regard to a) their corresponding fitness values and b) the number of iteration. This resulted into an algorithm with more enhanced exploitation capability compared to the standard PSO. Seven benchmark test functions were chosen to carry out in-depth statistical analysis. Meanwhile, each of the test functions was run in four dimensions, any one of which was run for 1000 times, i.e. an overall of 28000 ($7 * 4 * 1000$) simulations for the rival algorithms apiece. Thereafter, maximum, minimum, mean, and standard deviation of the statistical results of the APT-FPSO algorithm were compared to those of the standard PSO.

The organization of the rest of the article is as follows:
In the next section, a brief review of the standard PSO algorithm is presented. Section 3 elaborates on the execution mechanism of APT-FPSO. Next, in Section 4, ample statistical analysis is provided to indicate that the developed APT-FPSO algorithm outperforms the standard PSO. Section 5 concludes the article.

2 Overview of PSO algorithm

Since it was first introduced in 1995, PSO algorithm has enjoyed an incredible reputation among scientists from a broad realm of study. This algorithm performs in such a way that, at first, a number of random positions, i.e. candidate solutions, are assigned to each of the particles. Next, these positions are evaluated and the initial values for personal and global fittest are selected. Thereafter, the algorithm main loop begins in which positions and velocities of the particles are updated, and the new personal and global fittest are stored. The previously mentioned loop is reiterated until desired termination criteria are met, and the latest global fittest is selected as the final answer to the problem.

In this algorithm, Eqs. 1 and 2 represent, respectively, the position and velocity updating mechanisms:

$$x^i_{(t+1)} = x^i_t + v^i_{(t+1)} \tag{1}$$

$$v^i_{(t+1)} = w_(i) \times v^i_t + c_1 \times \text{rand1} \times \left(p^i_t - x^i_t\right) + c_2 \times \text{rand2} \times \left(p^g_t - x^i_t\right) \tag{2}$$

Where denotes the personal best record of the i^{th} particle in the t^{th} iteration, and p_t^g indicates the global best in the t^{th} iteration; rand1 and rand2 are two random numbers within the range $[0, 1]$; w is the inertia weight; c_1 and c_2 and are two constants called personal and global learning coefficients, respectively. It is worth noting that and , respectively, account for controlling the bias toward exploitation and exploration during execution of the algorithm.

3 The developed adaptive particularly tunable fuzzy PSO algorithm

As mentioned before, a great deal of endeavor has been being made in order to extricate the standard PSO algorithm from premature convergence phenomenon, resulting in plenteous modified variants of this algorithm. What all these algorithms pursue is achieving an adaptive, dynamic, well-balanced strategy for tuning PSO's setting parameters while performing exploration and exploitation. In this regard, Fuzzy Adaptive PSO algorithms have been evidenced to be of an excessive effectiveness compared to other variants of PSO. This is chiefly due to their ability which enables them to adaptively and dynamically tune the setting parameters of PSO algorithm using fuzzy logic concepts. However, the presented algorithm in this study possesses a unique trait that virtually all the other modified FAPSO algorithms are deprived of. This trait enables the algorithm to adaptively tune the learning coefficients of the algorithm, i.e. and , particularly for the particles apiece within every iteration, and, by doing so, it achieves the best strategy possible in balancing between exploration and exploitation. To this end, in the current study, the Mamdani Fuzzy Inference System (FIS) of interest has two inputs and two outputs.

The first input is dedicated to normalized iteration (NIt) from the beginning (NIt ≈ 0) to the end of the algorithm (NIt $= 1$). Eq. 3 describes the normalized iteration

$$\text{NIt} = \frac{\text{Current iteration}}{\text{Maximum number of iterations}} \tag{3}$$

By using three linguistic variables of "Low", "Medium", and "High", and assigning three corresponding Gaussian membership functions (MFs), the authors have fuzzified the normalized iteration.

The second input of the developed FIS has to do with the fitness value of each particle within each iteration. Here, an important point, which needs to be paid careful attention, reveals. At the first glance, the paradigm followed by the current study may seem extremely similar to that followed by [4], yet once one pays careful attention, he/she will find the difference between the two far from absolute. The major difference, however, is in fitness normalization strategy followed by the two works. In [4], the fitness values of every particle are normalized with respect to the lowest and highest fitness values experienced by the particle from the very first iteration. In other words, suppose a particle experiences a huge fitness value of the order 10^4 at the very first iterations of the algorithm. The work presented in [4] unforgivingly normalizes the subsequent fitness values of particles with respect to that huge value; thus, all fitness values of order $10^1, 10^0, 10^{-1}$ etc., will be, linguistically, referred to as "low", regardless of the substantial intermediate errors. Obviously, the mentioned flaw makes the algorithm performance extremely dependent on the objective function contour, and destructively dominates the rule-base inference system, leading to huge malfunctioning of the algorithm. On the contrary, forgetful of the great fitness values that once a particle has experienced in prior iterations, the proposed APT-FPSO algorithm normalizes the particles' fitness with respect to the best and worst fitness values of the swarm within each iteration. Eq. 4 formulates the normalized fitness index (NFI):

$$\text{NFI} = \frac{\text{fitness}_{t,i} - \min(\text{fitness}_t)}{\max(\text{fitness}_t) - \min(\text{fitness}_t)} \tag{4}$$

where $\text{fitness}_{t,i}$ indicates the fitness value of the i^{th} particle in the t^{th} iteration; besides, $\max(\text{fitness}_t)$ and $\min(\text{fitness}_t)$ denote, respectively, maximum and minimum fitness values of the swarm in the t^{th} iteration. Similarly, three Gaussian MFs are used to define the second input NIt linguistically named as "low", "medium", and "high", respectively. Figure 1 (a) and (b) indicates the inputs of the developed FIS.

The outputs of the designed FIS, however, are personal and global learning coefficients, i.e. $c_{1,t}^i$ and $c_{2,t}^i$, whose recommended values have been reportedly introduced in the literature to be within the range $[0.5, 2.5]$ [13, 10]. It is to be noted that $c_{1,t}^i$ and $c_{2,t}^i$, respectively, denote the learning coefficients of c_1 and c_2 for the i^{th} particle in the t^{th} iteration. Meanwhile, each of the outputs $c_{1,t}^i$ and $c_{2,t}^i$ is described via five triangular MFs, linguistically named as "Very low", "Low", "Medium", "High", and "Very high". In short, the developed FIS has two inputs, i.e. normalized iteration and normalized fitness index, and two outputs, i.e. c_1 and c_2. The outputs of the developed FIS are shown in Figure 1 (c) and (d).

There may be other distributions of MFs that would result in better solutions; however, this is not within the scope of the current study and is left as a future work.

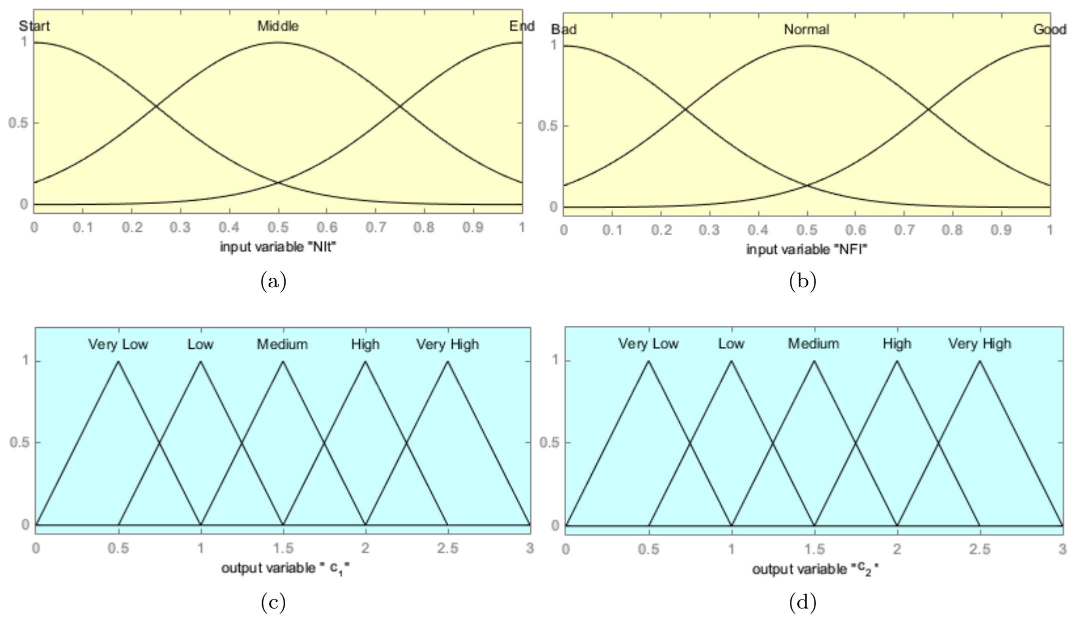


Figure 1: *Membership functions of inputs (a, b) and outputs (c, d) of the developed FIS. (a) Number of iterations; (b) normalized fitness index; (c) personal learning coefficient; (d) global learning coefficient.*

Given the mentioned inputs and outputs, one can form many rule bases for the developed FIS, any one of which provides a unique performance. However, the rule-base surfaces of one of these four are shown in Figure 2.

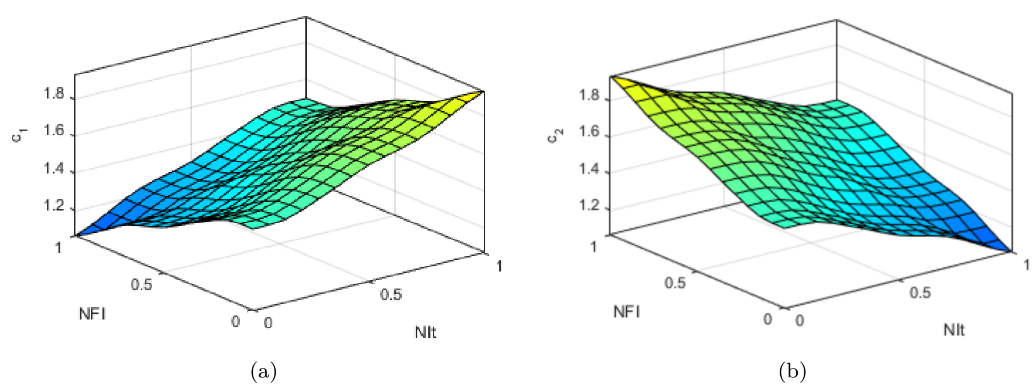


Figure 2: *Surfaces of the inputs and the outputs for the rule-base structure No.1.*

Defined inputs and outputs shown in Figure 1 are related to each other via rule-base structures. These four rule-base structures are tabulated in Table 1. The rule-base surfaces shown in Figure 2 are related to the rule-base structure No. 1 in the Table 1.

4 Results and discussion

As mentioned previously, the principal contribution of the current study is to further enhance the potentiality of the standard PSO algorithm. In order to find out how superior the proposed APT-FPSOs are, a thorough statistical analysis is conducted over seven benchmark functions. The benchmark functions, over which the statistical analysis is performed, are introduced in Table 2.

Additionally, using the statistical analysis, one can judge more fairly about the intermediate superiority of the developed APT-FPSO algorithms having four different rule-base formations. In this paper, for each of the five competitor algorithms, there are seven benchmark functions in four dimensions- i.e. $N = 2, 4, 8,$ and 20- any one of which was

Table 1: The four rule-base structure for the outputs c_1 and c_2

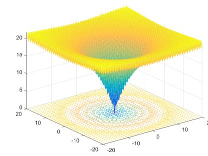
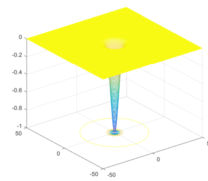
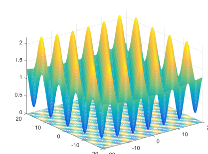
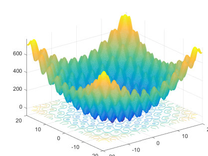
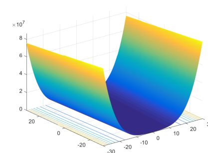
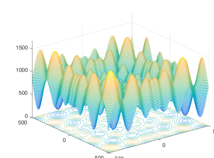
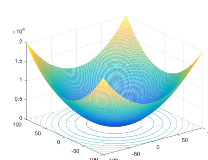
	c_1				c_2			
	NIt \ NFI	Bad	Normal	Good	NIt \ NFI	Bad	Normal	Good
Rule-base structure No. 1	Start	Medium	Low	Very Low	Start	Medium	High	Very High
	Middle	High	Medium	Low	Middle	Low	Medium	High
	End	Very High	High	Medium	End	Very Low	Low	Medium
Rule-base structure N0. 2	Start	Medium	High	Very High	Start	Medium	Low	Very Low
	Middle	Low	Medium	High	Middle	High	Medium	Low
	End	Very Low	Low	Medium	End	Very High	High	Medium
Rule-base structure N0. 3	Start	Very Low	Low	Medium	Start	Very High	High	Medium
	Middle	Low	Medium	High	Middle	High	Medium	Low
	End	Medium	High	Very High	End	Medium	Low	Very Low
Rule-base structure N0. 4	Start	Very High	High	Medium	Start	Very Low	Low	Medium
	Middle	High	Medium	Low	Middle	Low	Medium	High
	End	Medium	Low	Very Low	End	Medium	High	Very High

run for 1000 times. Thus, an overall of 140000 ($5 * 7 * 4 * 1000$) simulations are conducted, i.e. 28000 (140000/5) simulations for the five algorithms apiece. It should be noted that all the simulations are done with 50 number of particles, and the learning coefficients of the standard PSO are equal to $c_1 = c_2 = 2$ and the inertia weight (w) is set equal to 0.72 for all the five algorithms. The statistical results for the corresponding 140 ($5*7*4$) cases are reported in Table 3 to Table 6, including the maximum, minimum, mean, and standard deviation of the final global best cost values after 1000 times of simulations. Besides, a flowchart is provided in Figure 3 that demonstrates, diagrammatically, how APT-FPSO algorithm minimizes the benchmark functions; indeed, an algorithm of such flowchart was run, for each of the benchmark functions in each of the mentioned dimensions, for 1000 times. Accordingly, the algorithm procedure can be described as follows. Firstly, APT-FPSO setting parameters are defined; these parameters include the maximum number of iterations (MaxIt), number of particles (NumPart), inertia weight, details associated with the FIS, etc. Thereafter, the swarm is initialized, and random positions are assigned to particles apiece. Afterward that each one of the positions (candidate solutions) is evaluated, the global best of the initialized population is selected. After initialization, the algorithm main loop begins. In it, global and personal records are checked and selected; in addition, with the aid of fitness vector of the particles, the normalized fitness index (NFI) is calculated considering the minimum and maximum fitness values of the swarm within each iteration. Next, the normalized iteration (NIt) in conjunction with the NFI for each of the particles are fed into the FIS in order to obtain the corresponding outputs of $c_{1,t}^i$ and $c_{2,t}^i$ corresponding to each of the particles. These two learning factors along with personal and global best records are used to update particles' positions and velocities. The previously described main loop is reiterated until desired termination criteria be met.

Variations of c_1 and c_2 with respect to the number of iterations are plotted in Figure 4 for the Sphere benchmark, in the 2-dimension case for the rule-base structure No. 1, whose if-then rules are given in Table 1 and the rule-base surfaces are plotted in Figure 2. It is to be noted that, without loss of generality, for a better (more clear) illustration of the variations' trend, the number of particles was reduced to 10. Obviously, for the other benchmarks in the other dimensions, c_1 and c_2 vary with a similar trend.

Concerning Table 3 to Table 6, a low value for standard deviation is not something necessarily appropriate. That is to say, if the mean value happens to be greater (worse) than what one expects, then low values of standard deviation offer the results not much better than the insufficient mean value. Figure 5 depicts and compares the error values distributions of the most successful APT-FPSO, i.e. the bold values in Table 3 to Table 6, versus the standard PSO for all the seven benchmark functions and all the four dimensions. In Figure 5, the X axis of all the 28 diagrams is the error value and the Y axis is the error frequency. Virtually, what reported in Table 3 to Table 6 can be seen from Figure 5; indeed, Figure 5 is a visual representation of Table 3 to Table 6. However, it is to be noted that, in some of the cases, there may exist slight discrepancies between Min/Max of the data reported in Table 3 to Table 6, and that

Table 2: Benchmark test functions

Name	Function	Domain	Minimum	
Ackley	$f(x) = 20 - 20\text{exp}\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)$	$x \in [-30, 30]^n$	0	
Exponential	$f(x) = -\exp\left(-0.5\sum_{i=1}^{n} x_i^2\right)$	$x \in [-100, 100]^n$	-1	
Griewangk	$f(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n}\cos\left(x_i/\sqrt{i}\right)$	$x \in [-600, 600]^n$	0	
Rastrigin	$f(x) = \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	$x \in [-5.12, 5.12]^n$	0	
Rosenbrock	$f(x) = \sum_{i=1}^{n-1}\left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$	$x \in [-30, 30]^n$	0	
Schwefel	$f(x) = n * 418.98291 - \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$	$x \in [-500, 500]^n$	0	
Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	$x \in [-100, 100]^n$	0	

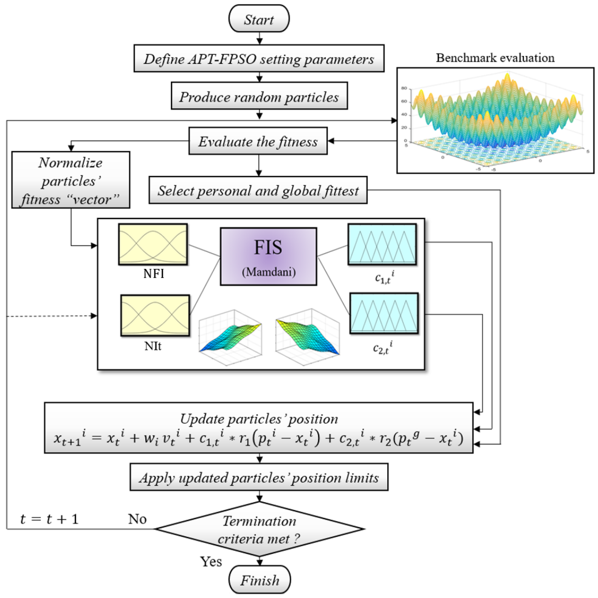
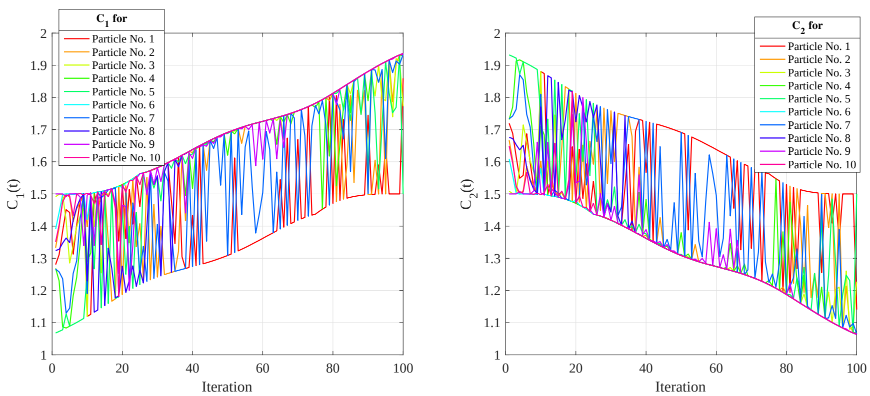


Figure 3: *Flowchart of the APT-FPSO algorithm while solving benchmark functions.*



(a) (b)

Figure 4: *The variation of the learning coefficients with respect to the number of iterations; for the Sphere benchmark, with 10 particles, in the 2D case.*

given in Figure 5. That is due to the fact that in those cases, some of the missing data may be ignored for the sake of clarity and for better representation of the "pattern" of the error distribution. In other words, in Figure 5, the focus is put on the illustration of the errors' distribution patterns rather than errors' quantitative values.

Concerning Table 3 to Table 6, the underlined mean values are merely related to APT-FPSOs and indicate the superiority of the corresponding APT-FPSO compared to the standard PSO. In other words, if the mean value of an APT-FPSO is lower (better) than that of the standard PSO, then it will be underlined. Accordingly, it can be noticed that APT-FPSO1, APT-FPSO2, APT-FPSO3, and APT-FPSO4 are superior to the standard PSO in 89.28% $(25/28 * 100)$, 82.14% $(23/28 * 100)$, 75% $(21/28 * 100)$, and 85.71% $(24/28 * 100)$ of the cases, respectively. It is to be noted that the fraction denominator (28) corresponds to the seven benchmarks all in four dimensions $(7 * 4)$.

Moreover, it might be interesting to compare the performance of APT-FPSOs, with different rule-base structures, with respect to each other. In this regard, the bold numbers indicate the intermediate superiority of APT-FPSO algorithms. For all APT-FPSOs in each of the dimensions, the lowest (the best) benchmarks' mean value is bolded. Accordingly, it can be observed that APT-FPSO1, APT-FPSO2, APT-FPSO3, and APT-FPSO4 are superior, with respect to each other, in 42.85% $(12/28 * 100)$, 17.85% $(5/28 * 100)$, 14.28% $(4/28 * 100)$, and 25% $(7/28 * 100)$ of the cases, respectively.

Generally, it can be concluded that APT-FPSO outperformed the standard PSO in an overall of 83.03% $(93/112*100)$ of the cases. It is worth pointing out that the fraction denominator (112) corresponds to the four APT-FPSO algorithms

Table 3: Statistical results for the 2-dimensional case.

$N = 2$		Bench mark fit function						
Algorithm	Statistical parameter	Ackley	exponential	Griewangk	Rastrigin	Rosenbrock	Schwefel	sphere
PSO	Mean	1.533e-04	-4.440e-01	4.660e-03	5.349e-03	8.980e-03	6.443e+01	5.588e-08
	St.D	1.503e-04	2.888e-01	4.016e-03	6.766e-02	1.055e-01	6.945e+01	1.230e-07
	Min	1.453e-06	-9.989e-01	2.702e-09	5.998e-11	2.697e-08	2.545e-05	2.383e-12
	Max	1.303e-03	-2.528e-05	2.727e-02	9.952e-01	2.429e00	2.368e+02	1.785e-06
APT-FPSO1	Mean	**4.082e-06**	-4.480e-01	**3.124e-03**	5.799e-03	8.934e-03	**6.249e+01**	4.386e-11
	St.D	3.680e-06	2.874e-01	4.139e-03	7.068e-02	6.876e-02	7.158e+01	8.587e-11
	Min	6.346e-08	-9.987e-01	4.096e-14	1.278e-13	1.298e-09	2.545e-05	9.995e-16
	Max	2.399e-05	-1.661e-04	3.947e-02	9.988e-01	1.116e00	3.553e+02	1.022e-09
APT-FPSO2	Mean	4.862e-06	-4.436e-01	3.874e-03	**1.152e-03**	4.211e-03	6.362e+01	5.172e-11
	St.D	4.341e-06	2.929e-01	4.172e-03	3.154e-02	5.619e-02	7.242e+01	9.145e-11
	Min	5.847e-08	-9.994e-01	1.315e-13	1.492e-13	5.625e-12	2.545e-05	5.798e-16
	Max	2.986e-05	-1.278e-05	1.977e-02	9.949e-01	1.182e00	3.355e+02	9.960e-10
APT-FPSO3	Mean	4.464e-06	**-4.498e-01**	3.311e-03	5.220e-03	1.263e-02	6.405e+01	5.224e-11
	St.D	4.206e-06	2.947e-01	4.064e-03	6.714e-02	1.077e-01	6.800e+01	1.321e-10
	Min	3.092e-08	-9.988e-01	4.195e-13	7.105e-15	4.908e-10	2.545e-05	1.964e-14
	Max	4.977e-05	-1.636e-04	2.958e-02	9.949e-01	1.781e00	2.368e+02	2.851e-09
APT-FPSO4	Mean	4.095e-06	-4.321e-01	4.070e-03	2.414e-03	**2.071e-03**	6.458e+01	**4.071e-11**
	St.D	3.678e-06	2.971e-01	4.494e-03	4.517e-02	2.969e-02	7.025e+01	8.648e-11
	Min	1.481e-08	-9.999e-01	9.725e-14	1.243e-13	3.437e-10	2.545e-05	4.331e-15
	Max	2.685e-05	-4.960e-04	2.958e-02	9.949e-01	6.743e-01	2.368e+02	1.100e-09

Table 4: Statistical results for the 4-dimensional case.

$N = 4$		Bench mark fit function						
Algorithm	Statistical parameter	Ackley	exponential	Griewangk	Rastrigin	Rosenbrock	Schwefel	sphere
PSO	Mean	6.378e-03	-1.272e-02	9.301e-02	1.304e00	2.367e00	3.765e+02	1.407e-04
	St.D	3.852e-03	5.600e-02	4.182e-02	9.887e-01	3.623e00	1.481e+02	1.667e-04
	Min	2.158e-04	-8.335e-01	6.246e-03	1.284e-04	0.003520	8.423e-04	6.364e-08
	Max	3.479e-02	-4.830e-15	2.328e-01	6.014e00	2.710e+01	7.896e+02	1.597e-03
APT-FPSO1	Mean	1.031e-04	**-1.556e-02**	5.360e-02	1.252e00	**2.066e00**	**3.745e+02**	3.542e-08
	St.D	5.660e-05	6.643e-02	3.416e-02	9.800 e-01	3.468e00	1.489e+02	4.530e-08
	Min	1.232e-05	-7.876e-01	3.665e-06	4.429e-06	8.669e-05	5.153e-05	3.650e-10
	Max	3.625e-04	-4.811e-17	1.936e-01	5.973e00	2.130e+01	7.716e+02	4.955e-07
APT-FPSO2	Mean	1.474e-04	-1.265e-02	5.400e-02	**1.236e00**	2.124e00	3.871e+02	8.002e-08
	St.D	8.771e-05	5.391e-02	3.357e-02	9.653e-01	3.488e00	1.521e+02	1.271e-07
	Min	1.481e-05	-6.794e-01	4.368e-05	3.471e-06	1.329e-05	5.481e-05	4.954e-10
	Max	6.117e-04	-4.206e-15	2.275e-01	5.976e00	2.180e+01	8.290e+02	2.403e-06
APT-FPSO3	Mean	1.243e-04	-1.407e-02	**5.351e-02**	1.317e00	2.391e00	3.785e+02	5.913e-08
	St.D	6.689e-05	6.033e-02	3.455e-02	9.973e-01	3.786e00	1.529e+02	7.337e-08
	Min	8.858e-06	-7.480e-01	1.381e-04	4.879e-07	1.270e-06	5.193e-05	7.262e-10
	Max	4.234e-04	-4.692e-15	2.572e-01	5.970e00	2.183e+01	8.093e+02	7.198e-07
APT-FPSO4	Mean	**1.021e-04**	-1.413e-02	5.419e-02	1.307e00	2.079e00	3.763e+02	**3.409e-08**
	St.D	6.114e-05	6.256e-02	3.434e-02	1.009e00	3.406e00	1.499e+02	4.381e-08
	Min	8.805e-06	-8.331e-01	4.969e-06	2.632e-05	6.996e-05	5.384e-05	3.499e-10
	Max	4.651e-04	-1.519e-14	2.258e-01	5.985e00	2.013e+01	7.698e+02	3.671e-07

applied to seven benchmarks, all in four dimensions ($4 * 7 * 4$). In sum, based on the statistical results revealed, the proposed APT-FPSO method significantly and statistically outperformed the standard PSO.

According to Table 2 to Table 6, it can be observed that APT-FPSO algorithms excessively outperform the standard PSO specifically while minimizing the benchmark test functions with more concavity, e.g. Ackley, Sphere, and so on. For the other test functions of a different type, which require being more explored rather than be exploited, APT-FPSO algorithms do not share the great effectiveness as they do for functions with more concavity. Nevertheless, they still excel the standard PSO in the other test functions as well. That is to say, without jeopardizing the exploration ability, owing to the individually and particularly tunable mechanism of APT-FPSO, this algorithm further enhances its exploitation ability. In doing so, as most of the engineering applications and problems are of such type, which requires being further exploited, APT-FPSO algorithm can be taken advantage to surmount a huge number of sophisticated engineering optimization problems.

It should be mentioned that compared to the standard PSO, APT-FPSO requires more of CPU time; however, this fact is not far from expectations since no free lunch theorem applies in optimization algorithms [20].

Table 5: Statistical results for the 8-dimensional case.

$N = 8$		Bench mark fit function						
Algorithm	Statistical parameter	Ackley	exponential	Griewangk	Rastrigin	Rosenbrock	Schwefel	sphere
PSO	Mean	1.275e-01	-1.540e-05	4.809e-01	7.589e00	1.586e+01	1.211e+03	5.847e-02
	St.D	6.127e-02	4.812e-04	0.1224e-01	3.649e00	3.970e+01	2.725e+02	4.415e-02
	Min	2.305e-03	-1.521e-02	1.031e-01	1.167e-01	7.375e-01	3.358e+02	4.842e-03
	Max	6.407e-01	-3.814e-40	7.924e-01	3.047e+01	3.811e+02	2.034e+03	3.099e-01
APT-FPSO1	Mean	**2.274e-03**	**-5.730e-05**	1.756e-01	7.718e00	1.071e+01	1.209e+03	**2.664e-05**
	St.D	1.013e-03	01.267e-03	9.990e-02	3.940e00	2.880e+01	2.652e+02	2.282e-05
	Min	5.502e-04	-3.758e-02	2.479e-02	1.037e00	1.791e-03	2.171e+02	1.352e-06
	Max	9.547e-03	-3.973e-37	7.007e-01	2.721e+01	4.060e+02	1.974e+03	1.849e-04
APT-FPSO2	Mean	4.280e-03	-2.301e-06	1.805e-01	**6.995e00**	**8.228e+00**	1.213e+03	1.167e-04
	St.D	2.102e-03	5.327e-05	9.810e-02	3.301e00	1.920e+01	2.613e+02	1.191e-04
	Min	4.137e-04	-1.609e-03	1.058e-02	1.028e00	1.737e-02	3.553e+02	4.834e-06
	Max	1.635e-02	-9.416e-39	6.077e-01	2.312e+01	1.821e+02	1.875e+03	1.395e-03
APT-FPSO3	Mean	3.403e-03	-7.614e-06	1.838e-01	7.320e00	1.003e+01	1.216e+03	7.069e-05
	St.D	1.600e-03	2.098e-04	9.956e-02	3.682e00	2.411e+01	2.541e+02	6.922e-05
	Min	8.662e-04	-6.581e-03	1.691e-02	9.988e-01	1.834e-03	4.540e+02	1.583e-06
	Max	1.262e-02	-1.732e-40	5.964e-01	2.499e+01	1.985e+02	1.993e+03	6.408e-04
APT-FPSO4	Mean	2.400e-03	-8.097e-07	**1.655e-01**	7.318e00	8.894e+00	**1.197e+03**	3.040e-05
	St.D	1.099e-03	1.980e-05	9.586e-02	3.634e00	2.177e+01	2.785e+02	2.930e-05
	Min	5.259e-04	-5.937e-04	1.321e-02	9.118e-02	8.536e-03	2.369e+02	8.143e-07
	Max	7.858e-03	-5.841e-39	5.812e-01	2.592e+01	1.998e+02	1.877e+03	3.525e-04

Table 6: Statistical results for the 20-dimensional case.

$N = 20$		Bench mark fit function						
Algorithm	Statistical parameter	Ackley	exponential	Griewangk	Rastrigin	Rosenbrock	Schwefel	sphere
PSO	Mean	1.532e00	-5.013e -47	1.056e00	2.875e+01	1.613e+02	3.864e+03	6.420e00
	St.D	3.098e-01	1.538e-45	2.146e-02	1.102e+01	1.415e+02	5.572e+02	2.090e00
	Min	6.798e-01	-4.863e-44	7.6710e-01	6.958e00	4.638e+01	1.954e+03	1.758e00
	Max	2.479e00	-4.649e-110	1.124e00	7.917e+01	1.436e+03	5.574e+03	1.477e+01
APT-FPSO1	Mean	4.621e-01	-1.013e-44	**2.967e-01**	3.241e+01	**5.363e+01**	3.861e+03	**7.465e-02**
	St.D	5.512e-01	2.374e-43	1.678e-01	1.568e+01	5.552e+01	5.610e+02	5.023e-02
	Min	2.931e-02	-6.659e-42	4.152e-02	8.038e00	7.058e00	2.330e+03	9.959e-03
	Max	3.250e00	-3.212e-115	9.763e-01	1.101e+02	3.781e+02	5.615e+03	5.750e-01
APT-FPSO2	Mean	5.334e-01	-1.814e-46	5.494e-01	**2.693e+01**	5.917e+01	3.851e+03	3.109e-01
	St.D	5.189e-01	5.737e-45	1.987e-01	1.052e+01	5.613e+01	5.671e+02	1.828e-01
	Min	7.073e-02	-1.814e-43	1.174e-01	7.737e00	1.226e+01	1.678e+03	3.138e-02
	Max	3.034e00	-2.468e-113	9.920e-01	1.028e+02	4.554e+02	5.519e+03	1.402e00
APT-FPSO3	Mean	**4.388e-01**	**-1.873e-40**	4.796e-01	2.973e+01	5.782e+01	3.830e+03	1.948e-01
	St.D	4.891e-01	5.533e-39	1.988e-01	1.362e+01	5.877e+01	5.622e+02	1.180e-01
	Min	5.337e-02	-1.745e-37	6.825e-02	6.569e00	1.085e+01	1.802e+03	3.352e-02
	Max	2.528e00	-3.951e-111	9.568e-01	1.118e+02	4.732e+02	5.417e+03	1.206e00
APT-FPSO4	Mean	5.153e-01	-7.896e-41	3.506e-01	2.866e+01	5.523e+01	**3.805e+03**	9.824e-02
	St.D	5.790e-01	2.497e-39	1.820e-01	1.185e+01	6.042e+01	5.408e+02	6.709e-02
	Min	4.194e-02	-7.896e-38	4.221e-02	6.873e00	5.818e00	1.846e+03	1.100e-02
	Max	3.891e00	-3.961e-119	9.260e-01	8.560e+01	4.985e+02	5.278e+03	6.092e-01

There exist a large number of works for further expansion of this study in future. For instance, in order to further improve the performance of the optimization algorithm, one can consider other indices to the input, e.g. diversity of the swarm, or to the output, e.g. inertia weight (w), of the corresponding FIS. Not only that, but also one may take advantage of an Adaptive Neuro-fuzzy Inference System (ANFIS) to tune the MFs associated with the developed FIS. In addition, the distribution of the MFs can be designed using optimization techniques; this will most likely lead to better performance of the algorithm.

5 Conclusion

In this study, in order to improve upon the performance of PSO algorithm and for the premature convergence phenomenon to be avoided, the authors presented a modified version of fuzzy PSO algorithms, namely Adaptive Particularly Tunable Fuzzy Particle Swarm Optimization (APT-FPSO) algorithm. This algorithm follows such a paradigm in which

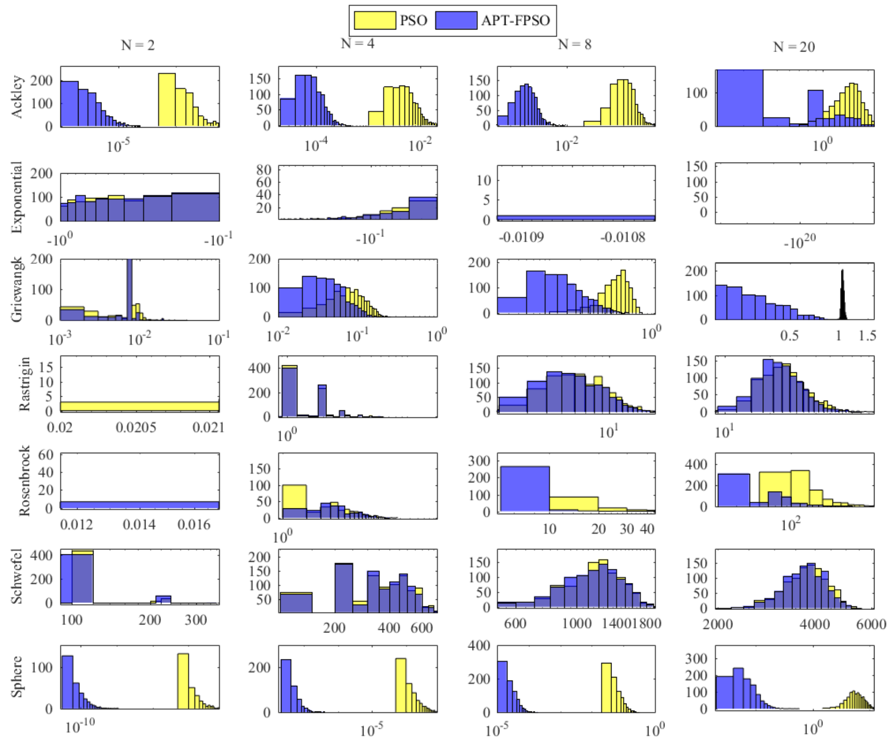


Figure 5: *Error bar charts of APT-FPSO vs. PSO, resulted from the statistical analysis.*

the adaptive setting parameters of FPSO algorithm are tuned particularly, at an individual extent, for particles apiece within each iteration. Ample statistical evidence was provided to confirm the superior capability of the presented algorithm compared to the standard PSO. Having been run each of the seven benchmark functions all in four dimensions for 1000 times, the authors observed that, in the best case scenario, APT-FPSO outperformed the standard PSO in 89.28% of the cases. More specifically, it was observed that two of the APT-FPSOs, with specific rule-base structures, promisingly exceled the standard PSO, and the two other performed almost as the same level as did the standard PSO. The excessive effectiveness of APT-FPSO was reconfirmed, guaranteeing its qualified candidacy in surmounting sophisticated optimization problems. By scrutinizing the statistical results, one can conclude that APT-FPSO shares greater effectiveness in solving problems required to be more exploited rather than be explored. That is, because of being been tuned particularly, APT-FPSO further enhances its exploitation ability, without jeopardizing exploration, compared to the standard PSO. As a result, APT-FPSO can be classified as a powerful meta-heuristic optimization algorithm and an excellent candidate for solving sophisticated optimization problems from an eclectic realm.

References

[1] L. Amador-Angulo, O. Castillo, *A fuzzy bee colony optimization algorithm using an interval type-2 fuzzy logic system for trajectory control of a mobile robot,* Mexican International Conference on Artificial Intelligence, Springer, (2015), 460–471.

[2] L. Amador-Angulo, O. Castillo, *Statistical analysis of Type-1 andinterval Type-2 fuzzy logic in dynamic parameter adaptation of the BCO,* Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15), Atlantis Press, 2015.

[3] K. Ameli, A. Alfi, M. Aghaebrahimi, *A fuzzy discrete harmony search algorithm applied to annual cost reduction in radial distribution systems,* Engineering Optimization, **48**(9) (2016), 1529–1549.

[4] E. Aminian, M. Teshnehlab, *A novel fuzzy particle swarm optimization,* 2013 13th Iranian Conference on Fuzzy Systems (IFSC), IEEE, (2013), 1–6.

[5] C. Caraveo, F. Valdez, O. Castillo, *Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation,* Applied Soft Computing, **43** (2016), 131–142.

[6] O. Castillo, P. Ochoa, J. Soria, *Differential evolution with fuzzy logic for dynamic adaptation of parameters in mathematical function optimization,* Imprecision and Uncertainty in Information Representation and Processing, Springer, Cham, (2016), 361–374.

[7] M. Y. Cheng, D. Prayogo, *Fuzzy adaptive teaching–learning-based optimization for global numerical optimization,* Neural Computing and Applications, **29**(2) (2018), 309–327.

[8] M. Clerc, J. Kennedy, *The particle swarm-explosion, stability, and convergence in a multidimensional complex space,* IEEE Transactions on Evolutionary Computation, **6**(1) (2002), 58–73.

[9] R. Eberhart, J. Kennedy, *A new optimizer using particle swarm theory,* IEEE MHS'95, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, (1995), 39–43.

[10] A. P. Engelbrecht, *Fundamentals of computational swarm intelligence,* John Wiley & Sons, 2005.

[11] Y. T. Juang, S. L. Tung, H. C. Chiu, *Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions,* Information Sciences, **181**(20) (2011), 4539–4549.

[12] H. Liu, A. Abraham, M. Clerc, *Chaotic dynamic characteristics in swarm intelligence,* Applied Soft Computing, **7**(3) (2007), 1019–1026.

[13] P. Melin, *Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic,* Expert Systems with Applications, **40**(8) (2013), 3196–3206.

[14] T. Niknam, *A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem,* Applied Energy, **87**(1) (2010), 327–339.

[15] F. Olivas, F. Valdez, O. Castillo, *A comparative study of membership functions for an interval Type-2 fuzzy system used for dynamic parameter adaptation in particle swarm optimization,* In Recent Developments and New Direction in Soft-Computing Foundations and Applications, Springer, (2016), 373–385.

[16] F. Olivas, F. Valdez, O. Castillo, *Ant colony optimization with parameter adaptation using fuzzy logic for TSP problems, in design of intelligent systems based on fuzzy logic,* Neural Networks and Nature-Inspired Optimization, Springer, (2015), 593–603.

[17] J. Peŕez, F. Valdez, O. Castillo, *A new bat algorithm with fuzzy logic for dynamical parameter adaptation and its applicability to fuzzy control design,* In Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics, Springer, (2015), 65–79.

[18] J. Tang, et al., *Study on day-ahead optimal economic operation of active distribution networks based on Kriging model assisted particle swarm optimization with constraint handling techniques,* Applied Energy, **204** (2017), 143–162.

[19] F. Valdez, P. Melin, O. Castillo, *A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation,* Expert Systems With Applications, **41**(14) (2014), 6459–6466.

[20] D. H. Wolpert, W. G. Macready, *No free lunch theorems for optimization,* IEEE Transactions on Evolutionary Computation, **1**(1) (1997), 67–82.

[21] H. T. Yang, J. T. Liao, *MF-APSO-based multiobjective optimization for PV system reactive power regulation,* IEEE Transactions on Sustainable Energy, **6**(4) (2015), 1346-1355.

[22] B. Yue, H. Liu, A. Abraham, *Dynamic trajectory and convergence analysis of swarm algorithm,* Computing and Informatics, **31**(2) (2012), 371–392.

[23] Y. Zhang, S. Wang, G. Ji, *A comprehensive survey on particle swarm optimization algorithm and its applications,* Mathematical Problems in Engineering, 2015.