IJE TRANSACTIONS A: Basics Vol. 34, No. 01, (January 2021) 97-109

International Journal of Engineering

Journal Homepage: www.ije.ir

Novel Particle Swarm Optimization Algorithm Based on President Election: Applied to a Renewable Hybrid Power System Controller

M. Yahyazadeh*[a], M. S. Johari[a], S. H. HosseinNia[b]

[a] *Faculty of Engineering, Vali-e-Asr University of Rafsanjan, Iran*
[b] *Faculty 3mE, Departmant Precision and Microsystems Engineering, Mekelweg 2, 2628 CD DELFT, The Netherlands*

PAPER INFO

Paper history:
Received 25 August 2019
Received in revised form 30 August 2020
Accepted 03 September 2020

Keywords:
Hybrid Optimization Algorithm
Chaotic Function
Hybrid Power System
Particle Swarm Optimization Algorithm
President Election Algorithm

ABSTRACT

Particle swarm optimization has been a popular and common met heuristic algorithm from its genesis time. However, some problems such as premature convergence, weak exploration ability and great number of iterations have been accompanied with the nature of this algorithm. Therefore, in this paper we proposed a novel classification for particles to organize them in a different way. This new method which is inspired from president election is called President Election Particle Swarm Optimization (PEPSO). This algorithm is trying to choose useful particles and omit functionless ones at initial steps of algorithm besides considering the effects of all generated particles to get a directed and fast convergence. Some preparations are also done to escape from premature convergence. To validate the applicability of our proposed PEPSO, it is compared with the other met heuristic algorithm including GAPSO, Logistic PSO, Tent PSO, and PSO to estimate the parameters of the controller for a hybrid power system. Results verify that PEPSO has a better reaction in worst conditions in finding parameters of the controller.

***doi**: 10.5829/ije.2021.34.01a.12*

1. INTRODUCTION

Optimization makes an important role in many fields such as social, economic and engineering. It could help us to get more desirable results. These problems include examples such as generating an optimal duty cycle which varies with photovoltaic parameters in order to extract the maximum power, estimating the parameter of a new model of solar cells, returning the system voltages inside the permitted range (for voltage regulation of MV distribution systems), etc. As an inspiring, nature could help us to design the optimization system for complex computational problems [1-4]. Some evolutionary algorithm (which are the most successfully ones and inspired from the nature) are Genetic, Particle Swarm Optimization, Ant Colony Optimization, etc. Among the algorithms, PSO became one of the most popular methods as a solution to solve the optimization problems, due to its efficiency in complex optimization problems in various fields [5]. It can be stated that, the main advantages of the PSO algorithm are: simple concept, easy implementation, relative robustness to control parameters, and computational efficiency [6]. For the first time, PSO algorithm was introduced. In this algorithm, every particle has its own position and velocity. The particles position and velocity are updated according to each particle positions and velocity and the best particles positions and velocity in the group, to find the best solution [5]. However, besides its advantages, the algorithm has problems like getting trapped in local minima or weak convergence rate. Some efforts have been done to overcome these problems includes combining PSO with other algorithm like GA or using modified discrete algorithm of PSO [7-9]. In a study, if no achievement is resulted at the end of a certain number of steps, PSO algorithm is stopped and the final point is considered as the new beginning point. The process is repeated through repositioning until the criterion is

*Corresponding Author Institutional Email: *Yahyazadeh@vru.ac.ir* (M. Yahyazadeh)

Please cite this article as: M. Yahyazadeh, M. S. Johari, S. H. HosseinNia, Novel Particle Swarm Optimization Algorithm Based on President Election: Applied to a Renewable Hybrid Power System Controller, International Journal of Engineering, Transactions A: Basics Vol. 34, No. 01, (2021) 97-109

satisfied to decrease the locally capacity particles [10]. Particle Swarm Optimization – Grey Wolf Optimizer (PSO–GWO) method has been also used to acquire the optimal size of the different system components in order to minimize the total cost of fresh water production [11]. A modified variant, named Repository and Mutation based PSO (RMPSO) is proposed by Jana et al. [12]. In RMPSO variant, two extra repositories have been introduced and maintained. So, it is done for storing personal and global best solutions which has the same fitness values. When the dimension of the problem is scaled up, the performance of the proposed algorithm remains consistent in most of the cases in this method. In the other study, the suitability of the No Speeds and Coefficients Particle Swarm Optimization (NSC-PSO) method is investigated to solve reliability optimization problems [13]. It is done by approaching a set of test problems which comprises two known Redundancy Allocation Problem (RAP) case studies: Fault Tree optimization (FTO) and Event Tree Optimization (ETO). In another study it is suggested that the memory structure of canonical PSO is modified by introducing a multi-leader mechanism to overcome weak exploration ability and premature convergence of PSO [14]. Applying chaotic function besides a Gaussian distribution to give particles more opportunities to jump out of the local optima is also done to overcome these problems. The best advantage of chaotic sequence is their unpredictability, i.e., by their spread-spectrum characteristic, non-periodic, complex temporal behavior, and argotic properties [15]. In fact, incorporating a chaotic map for the random number generation instead of the random number generators (RNG), increases the efficiency of the Basic PSO algorithm besides introducing diversity in the solutions and is used as a compared algorithm in our paper. Therefore, using hybrid algorithm could help to combine the better characteristics of each one to achieve the best approach. For example, GAPSO has been used in constraint optimization problems [8].

According to what has expressed up to now, there are some problems with basic PSO and should be noticed to solved. Therefore, in this paper we propose a new classification for particles called president election to achieve the best solution in less iteration besides escaping from local minima. This work is done by a good and novel filtration on original particles to select the better ones for upper level of the algorithm. This new classification is introduced to give optimized particles to a kind of modified PSO algorithm to accelerate the process of election of proper particles besides avoiding weak convergence rate. In PEPSO, by an original refining, a proposed filtration is done at first. Center of gravity method is used to choose better particles at this step. In addition, by increasing the number of parties (final classifications), which is described in PEPSO algorithm, the chance to get trapped in local minima extremely reduces. So, a fast and directed convergence rate is achieved. On the other hand, because of giving best particles to the modified PSO, a better solution is achieved in less time. A modified PSO is a PSO which has been used particles from president election level and consequently avoiding functionless particles. To have a comprehensive comparison, we used a practical case i.e. hybrid power system with Fractional Order PID (FOPID) controller. The results are also shown better convergence of PEPSO in comparison with the other algorithms including GA-PSO, Logistic-PSO, Tent-PSO, and PSO.

The rest of the article is organized as follows: Section 2 describes the problem by explaining the basic PSO algorithm. The proposed method is also introduced in this section. The optimization description of the algorithm and mathematical analysis explained in section 3. Description of the hybrid power system, FOPID controller, objective function and error signal are represented in section 4. Results demonstrate and validate the effectiveness and robustness of the proposed method represented in Section 5. Sections 6 and 7 are devoted discussion and conclusions, respectively.

2. PROBLEM DESCRIPTION

To overcome problems with inspired algorithms such as Genetic or PSO, some efforts have been done in the literature [8, 11,12]. However, such systems lead to high complexity especially for multi-dimensional systems, which in turn as challenging issue, will demand to propose novel method to modify this methods. In addition, fast convergence and increasing reliability are important issues in improvement of PSO. To this respect, a new classification is proposed in this paper. All of the achievements are based on the application of hybrid power system.

2. 1. Basic PSO Algorithm PSO works with particles contains the solutions of a problem. Each particle has a position and a velocity. The evaluation is achieved by the objective function of the optimization problem. Particle position dimensions are the variables of the optimization problem. Each particle has two criterions. The criterions for updating are called *Pbest* and *Gbest* which are the best position of each particle position itself and the best position among all particles achieved, up to last iteration implemented, respectively. So, basic PSO method is based on moving the particle position to a better position to find the best solution according to the following equation:

$$v_{id}^{j+1} = z_{\max} v_{id}^{j} + b_1 \times Rnd_1 \times (p_{id}^{best,j} - p_{id}^{j}) + b_2 \times Rnd_2 \times (p_{swarm,d}^{best,j} - p_{id}^{j}), \quad (1)$$

$$p_{id}^{j+1} = p_{id}^{j} + v_{id}^{j+1}, \tag{2}$$

where b_1 and b_2 are positive constants and represent the acceleration coefficients, Rnd_1 and Rnd_2 are two random variables within [0, 1]. v_{id} is the velocity of individual i on dimension d. p_{id} is ith current position on dimension d, $p_{id}^{best,j}$ is the location of the best problem solution vector found by i, $p_{swarm,d}^{best,j}$ is the location of the best particle among all the particles in the population on dimension d in iteration j, and $z_{\max}$ is the inertia weight that warrants convergence of the PSO algorithm [1].

A maximum velocity ($v_{\max}$) for each modulus of the velocity vector of the particles is also defined to control excessive roaming of particles outside the user-defined search space. Whenever a v_{id} exceeds the defined limit, its velocity is set to $v_{\max}$.

There are some advantages with PSO algorithm and also there exist some disadvantages to work on such as: getting stuck in local optimum, population variety reduction, increasing its convergence speed problem and so on. Researchers have been trying to improve these problems with basic PSO. As an example, applying chaotic coefficient instead of random numbers is a method to improve the chance of optimal solution selection according to the following equation:

$$v_i^{j+1} = z_{\max} v_i^j + b_1 \times n_1 (p_i^{best,j} - p_i^j) + b_2 n_2 (p_{swarm}^{best,j} - x_i^j), \tag{3}$$

$$p_i^{j+1} = p_i^j + v_i^{j+1}, \tag{4}$$

where the random numbers (b_1 and b_2) is multiplied by chaotic numbers n_1 and n_2. As discussed, various approaches need an improvement. By using the new approach called PEPSO or President Election Particle Swarm Optimization, which is inspired from president election procedure, we are trying to make some improvement in iterations number in convergence to give better solution.

2. 2. President Election Algorithm

Each legal age population has permission to vote their president in every country. The president is a person elected among the population. Every country consists of several provinces or states. In other classification, it consists of several parties (usually 2 parties in many countries). Every party follows its goals in every province or states. According to the fact that each candidate for president election should qualify to introduce for election and each party has at least a candidate for this vital election, an optimization algorithm with a suitable convergence could be introduced by using this model besides a modified PSO algorithm for giving it as a good direction for update. By a new classification inspired from president election, persons who are the agent of population of a country, vote to two or more candidate of parties for election to elect the best particle which denotes the president. Each person means one particle. By this novelty, a good refining is done on all particles. After giving the best particles to a modified PSO, a better solution in less time is achieved.

As shown in Figure 1, population of a country is generated at first. More population means more opportunity. Two left and right parties are marked with gray and red colors. The color of the rectangles shows the distinction between states or provinces. Finally, a person is selected to give a more optimal number. In here, by applying a filter to the whole population, about 50% of it has been removed. In this step, a center of gravity method doses the filtration. Each person´s value which has less than 50% of center of gravity´s value of the population is removed. A random distribution is applied to the remaining population, which we called legal age population in this paper, to distribute and locate at states. The number of states must be more than 2 to have at least two parties for a competition. A similar filtration like step 1 is done again for each state. Persons whose values are in the left part of the center of gravity are called left party and persons on the other side are right parties. The best person of each party in every state is one whose value is nearer to center of gravity. Parties are made by collecting best left and right party member of every state. We have 2 parties here at least. A modified PSO is applied to both parties. A modified PSO is used, because the initial particles of basic PSO is different from what has been researched in this paper. At last after finding the best particles (They are best candidates) of each party, the

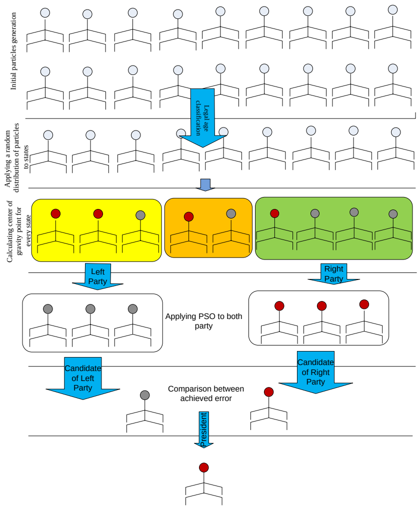

Figure 1. A graphic description of what has been done in PEPSO algorithm. Red persons represent right party members and the other color shows left party members

particle which called president here is elected by a comparison. Algorithm procedure is discussed in detail as following.

Algorithm Procedure:

Step 1. Initial particles generation:
A population is generated similar to PSO algorithm, as a greater number like a country. At this step, variables by its bounds and constants of the problem are determined. This variables and constants could be as follows:
Number of population, positions of particles and best particle, velocity of particles, number of provinces or states, variable number of the problem and its bound, maximum voted mans, inertia coefficient which has been reduced when be running the program, constants b_1 and b_2 which has been considered 1 here.
Step 2. Legal age classification from the whole population by center of gravity method is done in this step. About 50% of persons (particles) are rejected to vote according to their distance from the center of gravity point. Center of gravity point calculations are Equations (5) and (6):

Let us consider particles $p_i;\ i=1,...,n$, each with position m_i (value of fitness function here) that are located in space with coordinates $r_i;\ i=1,...,n$, the coordinate R (the position of the center of gravity mass point here) of the center of mass, which satisfy the following equation.

$$\sum_{i=1}^{n} m_i(r_i - R) = 0 \tag{5}$$

Solving the equation for R yields the following formula:

$$R = \frac{1}{M}\sum_{i=1}^{n} m_i r_i , \tag{6}$$

where M is sum of the masses (fitness function values) of all particles.
Step 3. Applying a random distribution of persons to states:

A number of states are given to program as a constant. More states give more accuracy. At first, 50% persons are distributed to number of states and called maximum voted mans (MVM). Then by a random distribution and a bound which is considered in range of $[1, MVM]$, $E=\{1,2,...,MVM\}$ the number of legal age population in each state is specified according to following equation:

$$N_P = rand_k\ (MVM);\ k=\{1,2,...,h\}, \tag{7}$$

where N_P^k is number of persons that could be vote in one state. The remaining persons are distributed to states in equal ratio as follows:

$$F_N = N_p^k + \frac{N_R}{N_S}, \tag{8}$$

where F_N, N_R, and N_S are final number of persons that could be vote, number of remaining persons and number of states, respectively.

$$F_{N_L} = MVM - \sum(F_{N-1}) \tag{9}$$

where F_{N_L} is final number of persons could vote in last remaining state and F_{N-1} is F_N except last remaining state.
Step 4. Similar to step 2 center of gravity points for every state is calculated.
Step 5. The state population is dividing into two parties. The right party is located in the right position of the center of gravity point and the left party in against position. By this classification in each state, the nearest persons of right and left position to the center of gravity point are selected as the agent of that party at that state according to Equations (10) and (11).

$$P_r < C_r < C_g , \tag{10}$$

where P_r, C_r, and C_g are fitness value of persons in right position of gravity point, fitness value of candidate of right party in each state and fitness value of center of gravity point, respectively.

$$C_g < C_l < P_l , \tag{11}$$

where C_g, C_l, and P_l are fitness value of center of gravity point , fitness value of candidate of left party in each state and fitness value of persons in left position of gravity point, respectively.
Step 6. At this step, in order to have better agents for candidates in each party, we need to modify PSO algorithm and then one is applied to each party agent to find the best candidates. A modified PSO algorithm is defined as follows:

The first and second step of using random number and locating them in the fitness function in basic PSO algorithm is omitted here. It is shown in Figure 2.
Step 7. Each candidate who has the minimum error and proper solution between two parties is elected as the president (best particle).
Step 8. End iteration criterion: If the minimum error is close to zero (or a constant which could be defined in the program) at first iteration the program will end and if not, the next iteration starts again. The end condition for best solution could be end iteration too.

3. OPTIMIZATION DESCRIPTION OF THE ALGORITHM

3. 1. Explanations and Flowcharts of the Algorithm

By using this algorithm, an original refining on the whole population is done by considering

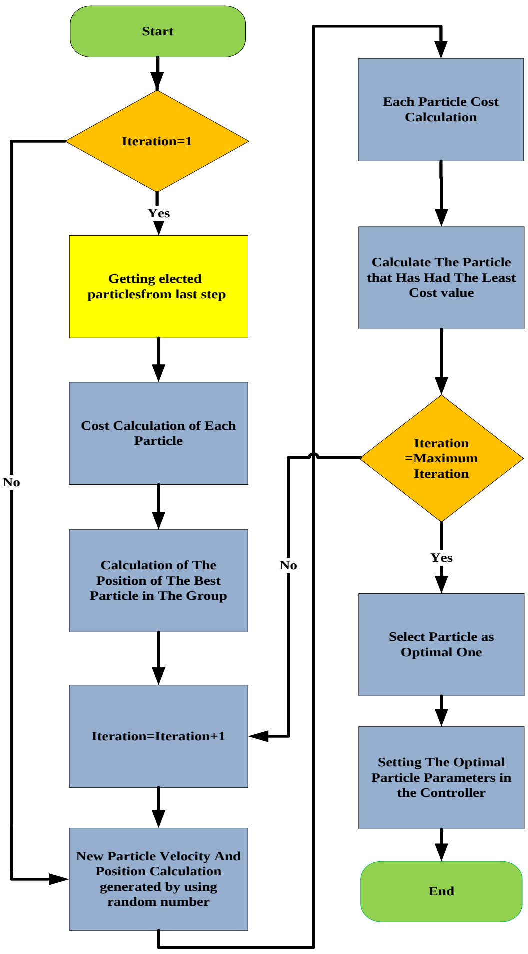

Figure 2. Diagram of modified PSO Algorithm. Yellow color block show the differences between modified and basic PSO algorithm. In PSO algorithm the particles are selected randomly. In here an elected particles are given to the next steps of basic PSO

the whole population experiences to elect the best particles for upper levels. In other word, all particles are acting as a deciding particle and vote the best particles to be candidates. The particles are called intelligent particles. This process which is done by center of gravity method in this new approach, create several search spaces which its number is in accordance with parties. So by locating the best particles to avoid functionless particles and using several search spaces in order to avoiding getting stuck in local minima or an early convergence, a better optimal point and faster convergence is achieved. Flowchart of modified PSO algorithm is shown in Figure 2 and PEPSO algorithm is shown in Figure 3. As illustrated in this figure, a considerable difference between basic PSO and this modified PSO is original refining which is done to achieve the optimal solution. This difference has been shown by yellow color box in this diagram. Since the limitation of calculation time for all particles position in basic PSO algorithm which limits the number of initial particles, the great number of population of PEPSO, with respect to a country population, is the other advantage of the approach that increases the opportunity of an optimal particle selection. In addition by the original refining, few number of best particles elect for applying into modified PSO algorithm. Due to kinds of properties of the new approach, less iteration for every test is needed. Here just to have a better comparison with the other models, the number of iterations is selected 50.

3. 2. Analysis In PSO algorithm, each particle has two criterions for updating called *Pbest* and *Gbest*. *Pbest* is the last best position of each particle position itself and *Gbest* is the best position among all particles achieved, up to last iteration implemented. All particles are updated in each iteration. As clear from its name, PEPSO are trying to choose the competent particle among all population. So by an inspiring from what has been done in president election procedure, it has been trying to find the best and proper solution for the problem (the best person as president for a country). A good

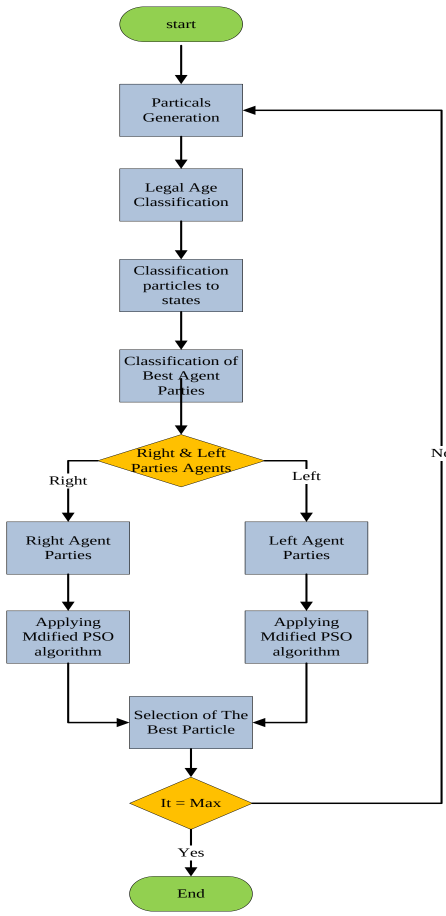

Figure 3. Diagram of President Election Particle Swarm Optimization Algorithm

selection is the final result. In PEPSO, with respect to the original refinery step which is done on primary particles (intelligence particles) the number of functionless particles has been decreasing effectively. Therefore the upper steps in PEPSO algorithm is a kind of competition and cooperation between better particles called candidate here. As shown in Figure 3, in fact PSO is a step of PEPSO algorithm which provides an iteration pass for more proper particles. The number of search space could also increased by an increase in number of parties. A multi-party mechanism (two-party here) enhances diversity of particles' search pattern to escape from local minima and increase weak convergence rate of basic PSO. So a greater chance space is created to find minima. Fast convergence by escaping from local minima is also achieved.

4. DESCRIPTION OF HYBRID POWER SYSTEM

4. 1. Hybrid Power System Due to the increasing use of fossil fuels and amount of required energy, a hybrid power system which benefited renewable energies will become a necessity. In a hybrid power system there are several energy components. Among them wind and solar renewable energy depends on weather conditions and the consuming load may at times exceed the production values. So the use of energy storage devices such as batteries, Ultra capacitors and flywheel along with other equipments at a power grid, seems to be necessary [16]. The dependence of each of the energies on seasonal weather and climate conditions and also using solar and wind energy together which are connected to a power grid, a decline in one generation could be compensated by the other. For these reasons, researchers have been interested in using the hybrid power system.

4. 2. Fractional-order Fuzzy Logic PID Controller Selecting a superior controller for this hybrid power system is also a question. Among the controllers, FOPID (Fractional Order PID) controller is getting more interested between researchers due to the design performance and flexibility of fractional calculus [17]. So according to a research on the controller of hybrid power system, a FOPID controller is selected [16]. The schematic of the hybrid power system using fractional-order fuzzy PID controller is illustrated in Figure 4.

A structure of Fractional-order fuzzy logic PID controller is shown in Figure 5. The controller parameters are $\{K_E, K_D\}$ and $\{K_{PI}, K_{PD}\}$ as an input and output scaling factor, respectively. Parameters $\{\lambda, \mu\}$ determine the fractional order differential-integrals respectively. The heart of system controller is formed of fuzzy membership functions as shown in Figure 6.

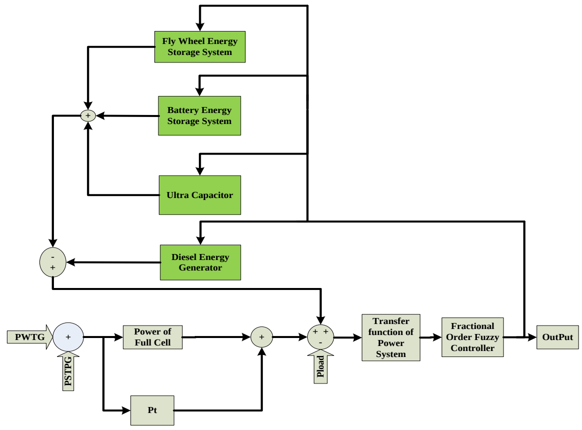

Figure 4. Schematic of hybrid power system with its components. This figure consists of 4 parts: inputs which includes power of the wind, sun and Diesel generator, saving parts which consists of battery, fly wheel and ultra capacitor, control system and output delivered to loads. Parts of input energy are given to fuel cell to run it [16]

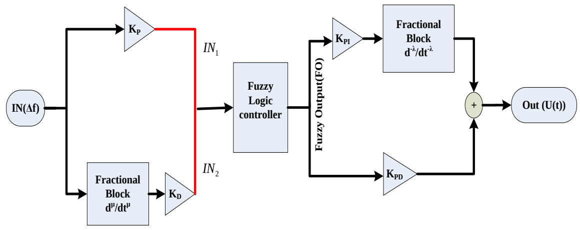

Figure 5. Schematic of the fractional-order fuzzy PID controller (The red line marks the use of a multiplexer which acts as a fuzzy switch to select one of the fuzzy inputs. Δf is input signal)

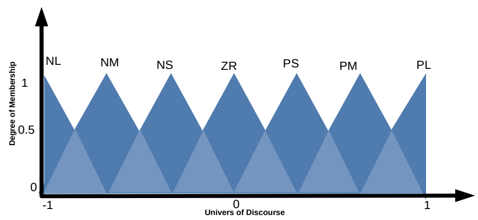

Figure 6. Membership functions of the fuzzy PID controller

Fuzzy linguistic variables NL, NM, NS, ZR, PS, PM and PL represent a negative large, negative medium, negative small, zero, positive small, positive medium and positive large respectively. The rule base considered for the fuzzy controller is depicted in Table 1 and the corresponding membership functions in Figure 6. The method used to calculate the output of the fuzzy controller has been chosen center of gravity defuzzification. Fuzzy system consists of two input variables and one output variable. In order to balance the computational complexity and at the same time having a high degree of certainty, the number of membership functions of fuzzy control was elected seven [18]. For a simple analysis, Triangular membership functions have been used. Figure 7 shows plot surface of fuzzy controller for the control parameters.

4. 3. Objective Function and the Controller Parameters An integral performance index has been considered as the objective function for optimization in Equation (12). The simulation period is also considered $T_{Max} = 120\,s$. At this equation the weighted sum of squared frequency deviation and the deviation of controlled signal v from its expected steady state value v_{ss} are used as follows:

$$J = \int_0^{T_{Max}} \left(q_1 (\Delta f)^2 + q_2 (v - v_{ss})^2 \right) dt. \tag{12}$$

The first term represents the Integral of Squared Error (ISE) of grid frequency deviation and the second one is the Integral of Squared Deviation of Controller Output for the disturbance rejection task of the controller. The positive weight coefficients q_1 and q_2 determines the relative importance of the first and second term and considered $q_1 = q_2 = 1$ here.

4. 4. Error Signal For all the cases (generated and demand powers independent of controller structure and Δf, frequency deviation) there is a sudden jumps and stochastic component superimposed on a base value at arbitrary instants of time to show a sudden large change in the power at different time instants (40s and 80s in this case). So the steady state control signal v_{ss} changes after each switching in the load and generation and shows a proper performance of the control system. The ideal and achieved output control signal has been depicted in Figures 8 and 9, respectively [19]. Figure 8 shows the ideal signal which expected to be achieved by some controlled loads that have been given to the system. This steady state output signal which varies after each switching on generated power and consumption load is based on the following equation:

$$v_{ss}(t) = 0.81G(t) + 0.17G(t-40) + 1.12G(t-80) \tag{13}$$

$G(t)$ is a step function. Figure 9 shows the real signal which has been resulted by the controller. Error has been resulted by a subtraction between these two signals (ideal and achieved output by the controller) according to Equation (14). Figure 10 shows error signal after minimization by the controller.

$$E = I_o - A_o \tag{14}$$

which E, I_o, and A_o are error signal, ideal output of the controller, and achieved output of the controller, respectively.

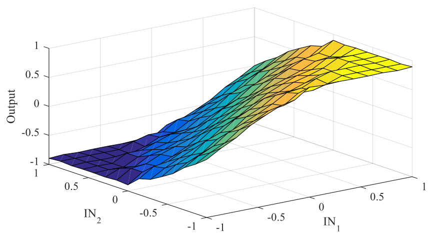

Figure 7. Fuzzy controller surface

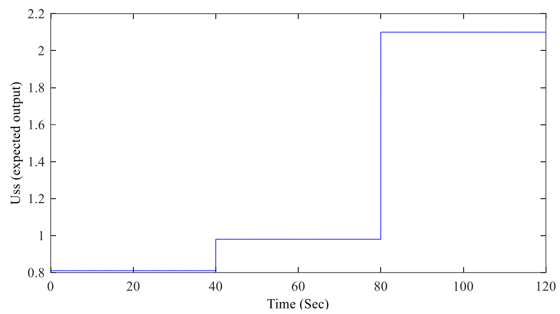

Figure 8. Desired output (reference signal). The ideal output of the controller which should varies by controlled loads that have been given to hybrid system

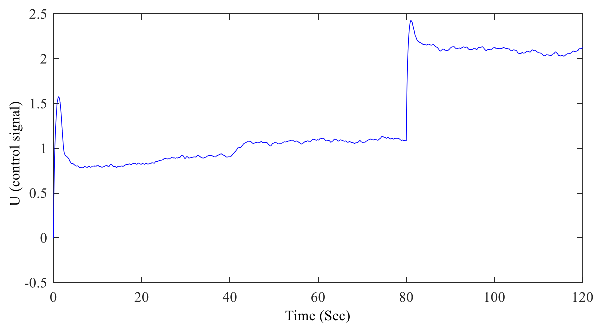

Figure 9. Real output of control signal produced by controller after optimization

TABLE 1. Rule base for error, fractional rate of error and FLC output

e ($\frac{d^u e}{dt^u}$)	NL	NS	NM	ZR	PS	PM	PL
PL	ZR	PS	PM	PL	PL	PL	PL
PM	NS	ZR	PS	PM	PL	PL	PL
PS	NM	NS	ZR	PS	PM	PL	PL
ZR	NL	NM	NS	ZR	PS	PM	PL
NM	NL	NL	NM	NS	ZR	PS	PM
NS	NL	NL	NL	NM	NS	ZR	PS
NL	NL	NL	NL	NL	NM	NS	ZR

5. RESULTS AND ALNALYSIS

In this section the application of PEPSO for parameter estimation of FOPID controller will be evaluated and corresponding results is presented. To validate the superior performance of PEPSO, a comparison is done with a couple of state of art algorithm including GAPSO [19], Logistic [15], Tent [17] and Basic PSO [17].

Parameters estimation and optimization of a FOPID controller of a hybrid power system have been used as the case studies of this research. These parameters estimation is done for all introduced above algorithm. The number of parameters decision (dimension in an optimization problem) is 6. For a fair comparison the number of initial particles is set to 100 for all algorithm all algorithm uses a same cost and fitness function. The compared algorithm has been run for 50 times. For all algorithm $b_1 = b_2 = 1$ and z is linearly decreased from 0.9 to 0.1. Used ranges for FOPID parameter controller of a hybrid power system have also been tabulated in Table 2.

Basic PSO was introduced by its equations. Logistic and Tent PSO are as follows:

Logistic chaotic function: This function is described by Equation (15). α and x_0 are equal to 0.4 and 0.2027, respectively.

$$x_{n+1} = ax_n(1 - x_n), \tag{15}$$

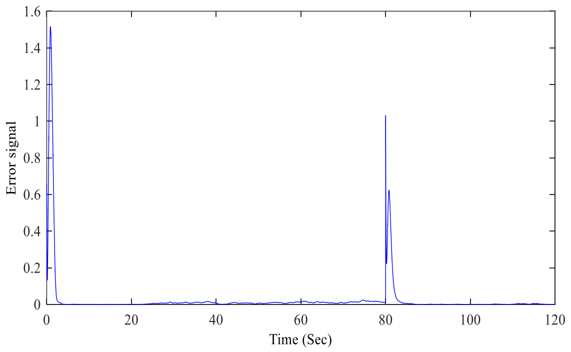

Figure 10. Output error signal after optimization

TABLE 2. Used ranges of FOPID parameters of the controller in a hybrid power system

Parameters	Lower bound	Upper bound
K_D	0	1
K_E	0	1
K_{PI}	0	40
K_{PD}	0	40
λ	0.01	0.99
μ	0.01	0.99

Tent chaotic function: This function is described by Equation (16) and resulted in a chaotic sequence in the interval $(0,1)$.

$$\begin{cases} \dfrac{1}{x_n} & x_n < 0.7 \\ \dfrac{1-x_n}{0.3} & Otherwise \end{cases}, \tag{16}$$

which are replaced in Equation (3). GAPSO algorithm is similar to Zhang et al. [19]. The objective function is also described in section (4).

Table 3 briefly compares the performance of different algorithm in control parameters. As clear from the table, the best optimum value is obtained by PEPSO. In addition since the aim of optimization system design is often to achieve an optimal value in an appropriate time, PEPSO shows an acceptable and optimum value in less iteration and time. The corresponding and desired error of Table 2 is obtained in minimum time by the PEPSO illustrated in Figure 11. As it is clear from the figure, PEPSO was achieved to a best optimal solution in all iterations. Also it was achieved the best solution in initial iterations, with a large difference in resulted value, than the others. Therefore, the result showed a better performance of this algorithm to the optimal value.

From the tabulated results in Table 4, the performance of PEPSO against the other algorithm could be observed. As it is clear from the table, PEPSO has the best performance in normal conditions of hybrid power system. The next is Logistic, Tent, PSO, and GAPSO. Equation (17) shows the calculation equation of performance ratio. In this equation k represents the algorithm. $Performance - Ratio_k$ is the performance difference percentage of the other algorithms in comparison with PEPSO.

$$Performance - Ratio_k = \frac{(ISE, ISDCO, J_{\min})_{PEPSO} - (ISE, ISDCO, J_{\min})_k}{(ISE, ISDCO, J_{\min})_{PEPSO}}, \\ k \in (GAPSO, LogisticPSO, TentPSO, PSO) \tag{17}$$

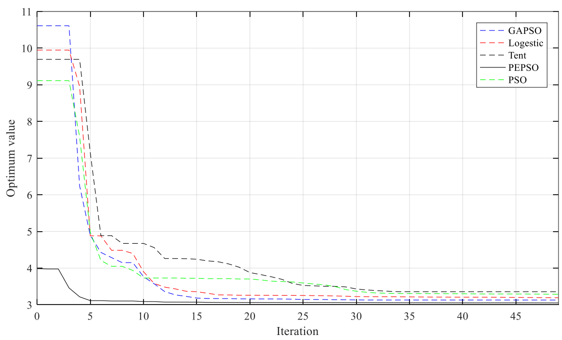

Figure 11. Optimum value achievement for 5 algorithms. This figure shows that PEPSO has been resulted to an optimal solution with significant difference than the other algorithms (GAPSO [19], Logistic [15], Tent [15], and PSO [16]) in initial iterations and the best solution

TABLE 3. Comparison of the results for 5 algorithms in the optimal solution and running time

Parameters	Lower bound	Upper bound	Parameters	Lower bound	Upper bound	Parameters	Lower bound	Upper bound
K_D	0	1	K_{PI}	0	40	λ	0.01	0.99
K_E	0	1	K_{PD}	0	40	μ	0.01	0.99

Algorithm Type	K_E	K_D	K_{PI}	K_{PD}	λ	μ	J_{min}	Iteration	Running Time
PSO	0.2312	0.0493	8.3117	3.3303	0.1372	-0.9900	3.2856	12	9016.898475
Chaotic PSO (Logistic map)	0.1906	0.1413	40.0000	14.2068	0.0100	-0.9900	3.1284	11	8935.013240
PEPSO	0.1993	0.0032	6.3430	3.2717	0.1762	-0.9713	3.0498	6	6987.518315
Chaotic PSO (Tent map)	0.2352	0.0959	12.0500	5.7828	0.0351	-0.9531	3.1895	11	7884.394049
GA-PSO	0.9023	0.0000	1.5141	0.8001	0.6528	-0.9900	3.3548	21	7018.75095

TABLE 4. Performance ratio of GAPSO, Logistic PSO, Tent PSO, PSO versus PEPSO

Algorithm	ISE	ISDCO	J	Performance decrease of ISE	Performance decrease of ISDCO	Performance decrease % of J
PEPSO	1.0195	2.0303	3.0498	-	-	-
GAPSO	1.2390	2.1158	3.3548	-21.5	-4.21	-10.00
Logistic PSO	1.0828	2.0456	3.1284	-5.84	-0.75	-2.58
Tent PSO	1.1039	2.0856	3.1895	-8.28	-2.72	-4.58
PSO	1.0258	2.2598	3.2856	-0.61	-11.30	-7.73

5. 1. Robustness Analysis In this section, we studied the parameter estimation of PEPSO and the other algorithms at the worst state of the hybrid power system. This is done by disconnecting FESS, BESS, DEG and check the controller operation. To test the robustness:

1. The corresponding performance error measures by each algorithm from nominal values against disconnecting different energy storage component of the hybrid power system are investigated in Table 5. The percentage change of ISE, ISDCO, J from its nominal value is calculated by each algorithm according to Equation (18). In this equation k represents the opened element and i represents the algorithm. $Performance-decreases_{k_i}$ shows the performance decrease of each algorithm in opened element states of the hybrid power system in comparison with normal conditions.

$$Performance-Decrease_{k_i} = \frac{(ISE, ISDCO, J_{\min})_{Nominal_i} - (ISE, ISDCO, J_{\min})_{k_i}}{(ISE, ISDCO, J_{\min})_{Nominal_i}}, \tag{18}$$
$$k \in (DEG, FESS, BESS), i \in (GAPSO, LogisticPSO, TentPSO, PSO)$$

Results represent the best performance with PEPSO. This also shows a better robustness investigation by PEPSO in comparison with the other algorithm. The next rank belongs to Tent for ISE, GAPSO for ISDCO and J. So from the viewpoint of the best achieved ISE, ISDCO and J, PEPSO outperforms all other optimization algorithm.

2. The parameter of the transfer function of the maximum power storing/producing component should be modified to consider the worst state. This component is UC [16]. So a 30 and 50% increase and decrease of the UC transfer function parameters are given to the system to test the robustness. Deviation and control signal in accordance with 30% and 50% variation in parameters of the UC transfer function are shown in Figures 12 and 13, respectively. The control signal representations which have been achieved by using different algorithms in Figures 12 and 13 show system behavior at the worst. Comparable results can be seen in Table 6. This table shows that by a 30% increase in gain and time constant, PEPSO has the best performance to achieve the minimum error. This algorithm also shows acceptable performance by a 50% increase in achieving to ISDCO (Integral of Squired Deviation of Controller Output) and total (ISDCO+ISE) error. Optimal performance of the algorithm by a 50% decrease in gain and time constant in ISE (Integral of Squired Error) is investigated from the table. In some cases, in which the other algorithm including Logistic and Tent PSO has better performance, the output error achieved by PEPSO is so close to the other optimization algorithms. Then, it could be deducted that the best total error reduction refers to PEPSO.

TABLE 5. Performance of each algorithm in error calculation when a part of the hybrid system (DEG, FESS, BESS) is opened

Algorithm	REMOVED Element	Performance measure			Performance Decrease		
		ISE	ISDCO	J	ISE	ISDCO	J
PEPSO	Nominal	1.0195	2.0303	3.0498	-	-	-
	DEG	1.1528	2.1052	3.2580	13.07	3.69	6.83
	FESS	1.2213	2.2451	3.4664	19.79	10.58	13.66
	BESS	1.1498	2.1100	3.2598	12.78	3.92	6.89
GAPSO	Nominal	1.2390	2.1158	3.3548	-	-	-
	DEG	1.4989	2.2211	3.7218	20.98	4.97	10.93
	FESS	1.6089	2.3485	3.9574	29.85	11.00	17.96
	BESS	1.5520	2.1992	3.7512	25.26	3.94	11.81
Logistic PSO	Nominal	1.0828	2.0456	3.1284	-	-	-
	DEG	1.2550	2.2675	3.5225	15.90	10.84	12.59
	FESS	1.3509	2.3715	3.6924	24.75	15.93	18.03
	BESS	1.2565	2.2783	3.5348	16.01	11.37	12.99
Tent PSO	Nominal	1.1039	2.0856	3.1895	-	-	-
	DEG	1.2695	2.3589	3.6284	15.00	13.1	13.76
	FESS	1.3307	2.4552	3.7859	20.54	17.72	18.70
	BESS	1.2915	2.3931	3.6846	16.99	14.74	15.52
PSO	Nominal	1.0258	2.2598	3.2856	-	-	-
	DEG	1.2633	2.4521	3.7154	23.15	8.51	13.08
	FESS	1.3000	2.5112	3.8112	26.73	11.12	16.00
	BESS	1.2397	2.4954	3.7351	20.85	10.42	13.68

TABLE 6. Robustness test against 30 and 50% variations of the transfer function UC parameters for different algorithm -optimal values is shown by green color

Algorithm	Parameter	30% decrease	50% decrease	30% increase	50% increase
PEPSO	ISE	1.5434	3.7594	1.0604	1.0237
	ISDCO	50.8295	243.3733	11.2241	23.9957
	Total	52.3729	247.1327	12.2845	25.0194
Chaotic PSO (Tent map)	ISE	1.5396	3.7713	1.2024	1.0265
	ISDCO	51.7938	243.365	11.3193	24.0153
	Total	53.3334	247.1363	12.5217	25.0318
Chaotic PSO (Logistic map)	ISE	1.5654	3.8831	1.4056	1.0242
	ISDCO	50.868	243.3124	11.6293	24.0210
	Total	52.4334	247.1955	13.0349	25.0452
GA-PSO	ISE	1.6989	3.7558	1.1153	1.2174
	ISDCO	52.0777	246.5943	11.2824	24.1207
	Total	53.7766	250.3501	12.3977	25.3381

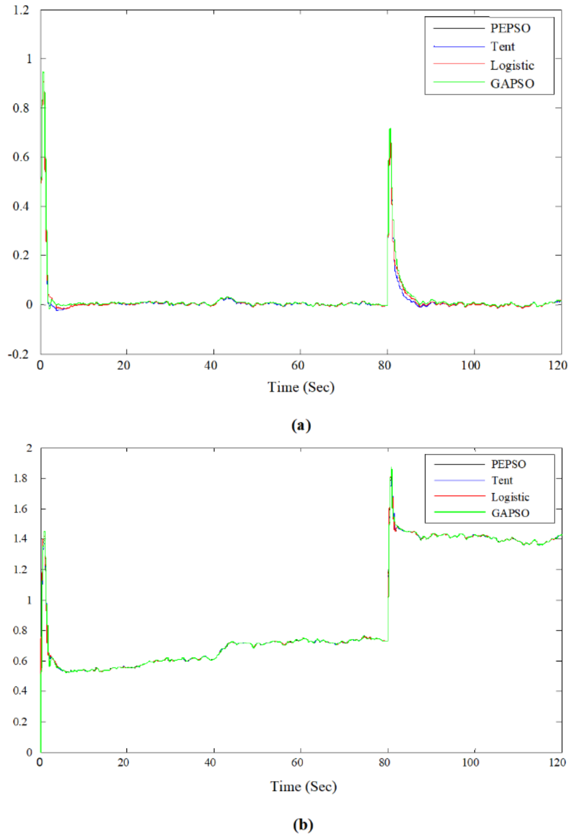

Figure 12. (a) Deviation signal generated by 30% increase in UC transfer function parameters. (b) Control signal generated by 30% increase in UC transfer function parameters

6. DISCUSSION

We used PEPSO algorithm in comparison with the other algorithm including GAPSO, Logistic PSO, Tent PSO, PSO. This is done to estimate the parameter of a FOPID controller to test the performance of PEPSO in a realistic example. Control system output and error are shown in Figures 8 and 9, respectively in section (4). PEPSO has achieved the optimal value in less iteration according to Figure 11. This is due to the refining step before applying particles to modified PSO algorithm. Robustness analysis has also been studied in section (5). This is done to show the best algorithm in parameters estimation of the controller in worst state. So, we are trying to improve convergence time by omitting functionless particles. This is done at a refinery step called President Election. But there are some challenges for every algorithm to execute. Giving intelligence to the particles for this algorithm is one of them. After researches and tests, center of gravity method had been chosen. But there may be existed another method for refinery step to improve this algorithm. Another challenge is President Election system which is different in every country. For this problem, the main common principle which exists in most countries has been selected. But every country principle could be test as an individual.

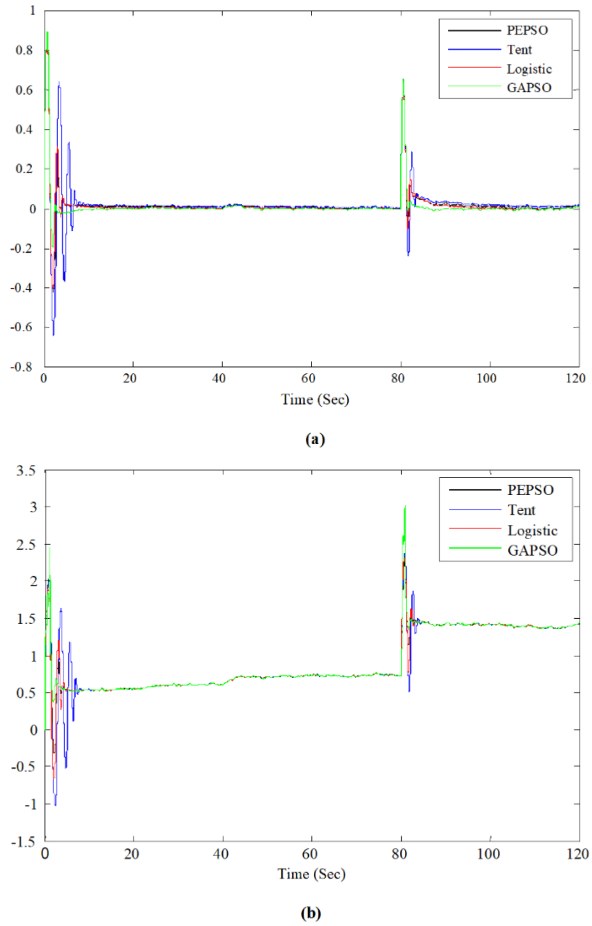

Figure 13. (a) Deviation signal generated by 50% increase in UC transfer function parameters. (b) Control signal generated by 50% increase in UC transfer function parameters

7. CONCLUSION

A new algorithm called PEPSO was proposed in this paper. It is inspired from president election procedure. After an introduction of how this algorithm works, performances of a variety of several heuristic algorithms including PSO, GAPSO, LOGISTIC PSO and TENT PSO on hybrid power systems, as a practical example, was discussed and compared with PEPSO. At first basic PSO was explained and then PEPSO introduced. After that fractional order fuzzy logic PID controller which was used as the heart of hybrid system was studied, objective function is the next part. Results was discussed and studied at last. It has been observed that among the algorithms, PEPSO algorithm had a better convergence than the others. By using this proposed algorithm, the iterations significantly reduced which is the main feature of this algorithm. In this new approach, original refining particles on the whole population was performed. So by avoiding functionless particles and using several search spaces in order not to getting stuck in local minima and a faster convergence, a better optimal solution was resulted. To test the robustness, the parameter of the transfer function of the maximum power storing/producing component was studied. Results in this part have also shown a better performance of PEPSO among all used hybrid algorithms in this paper.

Some future works could be done to improve the algorithm or implementation on controller like using adaptive fuzzy optimal controller design by referring to fuzzy adaptive decentralized optimal control for strict feedback nonlinear large-scale systems, fuzzy adaptive output feedback optimal control design for strict-feedback nonlinear systems, and the other similar approaches. So, combining the other algorithms such as genetic with this proposed approach in optimization problem of the hybrid power systems is the testable that can be conducive for further improvements.

8. REFERENCES

1. Gao, H., Kwong, S., Yang, J., and Cao, J., "Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation", ***Information Sciences***, Vol. 250, (2013), 82-112. DOI: 10.1016/j.ins.2016.07.017.
2. Soufi, Y., Bechouat, M., and Kahla, S., "Fuzzy-PSO controller design for maximum powerpoint tracking in photovoltaic system", ***International Journal of Hydrogen Energy***, Vol. 42, No. 13, (2017), 8680-8688. DOI: 10.1016/j.ijhydene.2016.07.212.
3. Khanna, V., Das, B.K., Bisht, D., Vandana, and Singh, P.K., "A three diode model for industrial solar cells and estimation of solar cell parameters using PSO algorithm", ***Renewable Energy***, Vol. 78, (2019), 105-113. DOI: 10.1016/j.renene.2014.12.072.
4. Yahyazadeh, M. and Rezaeeye, H., "Optimal Placement and Sizing of Distributed Generation Using Wale Optimization Algorithm Considering Voltage Stability and Voltage Profile Improvement, Power Loss and Investment Cost Reducing", ***Iranian Journal of Science and Technology, Transactions of Electrical Engineering***, Vol. 44, (2020), 227-236. DOI: 10.1007/s40998-019-00224-4.
5. Gholami Dehbalaeea, M.R., Shaeisia, G.H., and Valizadeh M., "A Proposed Improved Hybrid Hill Climbing Algorithm with the Capability of Local Search for Solving the Nonlinear Economic Load Dispatch Problem", ***International Journal of Engineering, Transactions A: Basics***, Vol. 33, No. 4, (2020), 575-585. DOI: 10.5829/ije.2020.33.04a.09.
6. Park, J-B., Jeong, Y-W., Shin, J-R., and Lee, K.Y., "An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems", ***IEEE Transactions on Power Systems***, Vol. 25, NO., (2010), 156-166. DOI: 10.1109/TPWRS.2009.2030293.
7. Madoliat, R., Khanmirza, E., and Pourfard, A., "Application of PSO and Cultural Algorithms forTransient Analysis of Natural Gas Pipeline", ***Journal of Petroleum Science and Engineering***, Vol. 149, (2017), 504-514. DOI: 10.31590/ejosat.717872.
8. Akbarpour H., Karimi G., and Sadeghzadeh A., "Discrete Multi Objective Particle Swarm Optimization Algorithm for FPGA Placement (Research Note)", ***International Journal of Engineering, Transactions C: Aspects***, Vol. 28, No. 3, (2015), 410-418. DOI: 10.5829/idosi.ije.2015.28.03c.10.
9. Jam S., Shahbahrami A., and Ziyabari, S.H.S., "Parallel Implementation of Particle Swarm Optimization Variants Using Graphics Processing Unit Platform", ***International Journal of Engineering, Transactions A: Basics***, Vol. 30, No. 1, (2017), 48-56. DOI: 10.5829/idosi.ije.2017.30.01a.07.
10. Sancaktar I., Tuna B., and Ulutas M., "Inverse kinematics application on medical robot using adapted PSO method", ***Engineering Science and Technology, an International Journal,*** Vol. 21, No. 5, (2018), 1006-1010. DOI: 10.1016/j.jestch.2018.06.011.
11. Abdelshafy, A.M., Hassan, H., and Jurasz, J., "Optimal design of a grid-connected desalination plant powered by renewable energy resources using a hybrid PSO–GWO approach", ***Energy Conversion and Management***, Vol. 173, (2018), 331-347. DOI: 10.1016/j.enconman.2018.07.083.
12. Jana, B., Mitra, S., and Acharyaa, S., "Repository and Mutation based Particle Swarm Optimization (RMPSO): A new PSO variant applied to reconstruction of gene regulatory network", ***Applied Soft Computing***, Vol. 74, (2019), 330-355. DOI: 10.1016/j.asoc.2018.09.027.
13. Anescu, G. and Paul Ulmeanu, A., "A No Speeds and Coefficients PSO approach to reliability optimization problems", ***Computers & Industrial Engineering***, Vol. 120, (2018), 31-41. DOI: 10.1016/j.cie.2018.04.020.
14. Liu P. and Liu J., "Multi-leader PSO (MLPSO): A new PSO variant for solving global optimization problems", ***Applied Soft Computing***, Vol. 61, (2017), 256-263. DOI: 10.1016/j.asoc.2017.08.022.
15. Alatas B., Akin E., and Bedri Ozer A., "Chaos Embedded Particle Swarm Optimization Algorithms", ***Chaos Solitons and fractals***, Vol. 40, No. 4, (2009), 1715-1734. DOI: 10.1016/j.chaos.2007.09.063.
16. Pan I. and Das S., "Fractional Order Fuzzy Control of Hybrid Power System with Renewable Generation Using Chaotic PSO", ***ISA Transaction***, Vol. 62, (2016), 19-29. DOI: 10.1016/j.isatra.2015.03.003.
17. Monje, C.A., Chen, Y.Q., Vinagre, B.M., Xue, D., and Feliu, V., Fractional-order systems and controls: fundamental and applications, Advanced Industrial Control Series, Springer-Verlag, (2010).
18. Das, S., Pan, I., and Das, Sh., "Performance Comparison of Optimal Fractional order Hybrid Fuzzy PID Controllers For Handling Oscillatory Fractional Order Processes with Dead Time", ***ISA Transactions***, Vol. 52, No. 4, (2013), 550-566. DOI: 10.1016/j.isatra.2013.03.004.
19. Zhang, Q., Ogren, R.M., and Kong, S., "A comparative study of biodiesel engine performance optimization using enhanced hybrid PSO–GA and basic GA", ***Applied Energy***, Vol. 165, (2016), 676-684. DOI: 10.1016/j.apenergy.2015.12.044.

Persian Abstract

چکیده

بهینه سازی ازدحام ذرات از زمان پیدایش یک الگوریتم متداول محبوب و رایج بوده است. با این حال، برخی از مشکلات مانند همگرایی زودرس، توانایی کاوش ضعیف و تعداد تکرار زیاد با ماهیت این الگوریتم همراه بوده است. بنابراین، در این مقاله طبقه‌بندی جدیدی برای ذرات پیشنهاد می‌گردد تا با روشی بهتر بتوان آن‌ها را سازماندهی نمود. روش جدید که از انتخابات رئیس جمهوری الهام گرفته شده است، بهینه‌سازی ازدحام ذرات انتخابات رئیس جمهور (PEPSO) نامیده می‌شود. این الگوریتم سعی دارد ذرات مفید را انتخاب کرده و در مراحل اولیه الگوریتم، ذرات بدون عملکرد را حذف نماید و علاوه بر این، اثرات تمام ذرات تولید شده را برای دستیابی به یک همگرایی سریع در نظر می‌گیرد. برخی مقدمات نیز برای رهایی از همگرایی زودرس انجام می‌شود. بمنظور اعتبارسنجی کاربرد بهینه‌سازی ازدحام ذرات پیشنهادی، برای یافتن پارامترهای کنترل کننده برای یک سیستم قدرت ترکیبی، نتایج با الگوریتم‌های دیگر شامل GAPSO، Logistic PSO، Tent PSO و PSO مقایسه می‌شود. نتایج نشان می‌دهند که PEPSO حتی در بدترین شرایط در یافتن پارامترهای کنترل‌کننده واکنش و دقت بهتری دارد.