DEMOCRATIC PSO ALGORITHM FOR THE ECONOMIC DISPATCH PROBLEM WITH VALVE-POINT EFFECTS

SECUI D.C., DUMITRESCU D., BUNDA S.
University of Oradea, Universităţii no.1, Oradea,
csecui@uoradea.ro

Abstract - This paper presents an optimization algorithm called Democratic Particle Swarm Optimization (DPSO), aiming to solve the economic dispatch problem. The DPSO algorithm is applied in the original version and also in a new version in which it is endowed with the chaotic Sine map (DPSO-Sine algorithm). The performance of DPSO and DPSO-Sine algorithms is tested on two systems having 13, and respectively 40 generating units. The results show that DPSO and DPSO-Sine have better performances than PSO algorithm and few other optimization techniques used to solve the economic dispatch problem.

Keywords: economic dispatch, Democratic PSO algorithm, valve-point effects.

1. INTRODUCTION

The economic dispatch (EcD) is a problem that concerns the specialists in the field and represents a topic of interest in optimizing the power systems. The EcD problem involves the determination of the generating units powers from a given power system, so that the fuel cost of whole system to be minimal, under the conditions in which the required restrictions given to the generating units and the system are met. The solving of the EcD problem is an approach to reduce the costs (by optimal scheduling of operating the whole generating units' assembly of the system) without making investments in new capacities of electricity production. A key part of the optimization model for EcD problem is the input-output characteristic (or the cost-power characteristic) of generating units. This characteristic was modeled by polynomial functions of 1, 2 or 3 degree or polynomial functions to which a sinusoidal term is added (as taking into account the valve-point loading effect). The consideration of the valve-point loading effect determines that the form of cost-power characteristics to be non-convex, non-smooth, generating local minimums in multiple solutions area. Thus, the mathematical model of EcD problem is nonlinear and requires the use of adequate solving techniques.

Meanwhile, the EcD problem was mainly solved using traditional methods or methods based on artificial intelligence. The classical methods (linear programming [1], nonlinear programming [2, 3]) have a number of shortcomings related to the differentiability of the functions involved in model, the type of objective function (convex/non-convex) and the type of constraints (linear/nonlinear). The methods based on artificial intelligence are not sensitive to the form of the objective function or constraints. Furthermore, the results show that these methods have been successfully applied for the optimization of technical systems [4, 5, 6, 7]. Some of the methods based on artificial intelligence applied to EcD problem solving are: Evolutionary programming (EP) [8], Genetic algorithm (GA) [9], simulated annealing (SA) [10], Particle Swarm Optimization (PSO) [4, 11], Differential Evolution (DE) [12, 13], Harmony Search (HS) [5], Ant Colony Optimization (ACO) [14], Civilized Swarm Optimization (CSO) [15], Artificial Bee Colony (ABC) [16, 17, 18], Bacterial Foraging Optimization (BFO) [6], Biogeography-based optimization (BBO) [19, 20], Quick group search (QGS) [7], Cuckoo Search algorithm (CSA) [21], Teaching learning based optimization (TLBO) [22].

One of the commonly used algorithms for solving the EcD problem is the PSO algorithm. This algorithm was applied in the original form [4, 15, 23], in various modified versions [24, 25] or by hybridization with other algorithms [9, 26, 27, 28, 29]. A recent version of PSO is the Democratic PSO algorithm, which was applied for structural optimization with frequency constraints [30]. In [30] there is shown that the Democratic PSO algorithm gains better results than the original PSO or other algorithms (such as genetic algorithm, charged system search algorithm). In this paper the Democratic PSO algorithm in its original form and endowed with a chaotic map (Sine map) is used for solving EcD problem.

The main contributions brought in this paper are: (i) the use and implementation of Democratic PSO algorithm for solving EcD problem; (ii) the investigation of the Democratic PSO algorithm behavior in case of introducing certain chaotic components (generated by chaotic Sine map) in update relation of the solutions.

The performances of Democratic PSO algorithm are tested on two systems with different characteristics and compared with other techniques/optimizing algorithms in terms of the solutions quality. Also, the Democratic PSO algorithm in its original form (using few statistic items) is compared with Democratic PSO algorithm endowed with Sine chaotic map.

2. FORMULATION OF THE EcD PROBLEM

The EcD problem follows up the determination of powers generated by the thermal units of a system such as the total cost of fuel to be minimal in terms of fulfilling certain required constraints. We consider a power system having n generating units. The generated powers (P_j, j=1,2,..,n) by each unit represents the problem variables and are represented by [P] vector in the form of $[P]=[P_1,P_2,..P_j,..,P_n]$. The $F[P]$ objective function is defined as following [8]:

$$\min F[P]=\sum_{j=1}^{n}F_j(P_j) \qquad (1)$$

where, $F[P]$ is the fuel cost at whole system level; $F_j(P_j)$ is the fuel cost, in \$/h, for the i-th unit.
Traditionally, the fuel cost per i unit is modeled by the relation (2) [8, 26]:

$$F_j(P_j)=c_jP_j^2+b_jP_j+a_j+\left|e_j\sin(f_j(P_{j,\min}-P_j))\right| \qquad (2)$$

where P_j is the real power of the jth generator; $P_{j,min}$ is the minimum real power of the jth unit; a_j, b_j and c_j are the fuel cost coefficients of unit j, and e_j and f_j are the coefficients of unit j in respect of the valve-point effect;
The $F_j(P_j)$ cost, defined by (2) relation, is modeled through a 2-nd degree polynomial function to which is added a sinusoidal term as considering the valve-point effect.

The constraining relations imposed to the optimizing model are listed below [23]:
i) The generating unit operates within the minimum and maximum generating capacity. These restrictions are linear, of inequality and expressed as follows:

$$P_{j,min}\le P_j\le P_{j,max},\ j=1,2,\dots,n \qquad (3)$$

where $P_{j,min}$ and $P_{j,max}$ are the minimum and maximum limits of j unit power.
ii) The generating units may have certain restricted operating areas in order to avoid the vibrations. These restricted areas are called prohibition zones and their consideration imposes the below restrictions:

$$\begin{cases}P_{j,\min}\le P_j\le P_{j,1}^{m}\\ P_{j,zp-1}^{M}\le P_j\le P_{j,zp}^{m} \quad ,\ zp=2,3,\dots,NZ_j\\ P_{j,NZ_j}^{M}\le P_j\le P_{j,\max}\end{cases} \qquad (4)$$

where NZ_j represents the number of prohibition zones for j unit. $P_{j,zp}^{m}$ and $P_{j,zp}^{M}$ are the minimum and maximum limits of zp prohibition zone for j unit.
iii) In order to ensure the real power balance only one restriction of equality at the entire system is used. This restriction shows that the power generated in the system is equal to the demanded power plus the power losses. The power balance constraint is defined by the equation:

$$P_{Gen}-P_{Loss}-P_{Demand}=0 \qquad (5)$$

where, P_{Demand} is the power demand of the system, in MW; P_L represents the total losses in the system, in MW; P_{Gen} is the total generated power in the system by the n units.

The total power losses can be calculated using the constant B coefficient formula [23]:

$$P_{Loss}=\sum_{i=1}^{n}\sum_{j=1}^{n}P_iB_{ij}P_j+\sum_{i=1}^{n}B_{0i}P_i+B_{00} \qquad (6)$$

where B_{ij} is an element of the loss coefficient matrix of size nxn, B_{0i} is i element of the loss coefficient vector of size n and B_{00} is the loss coefficient constant.

3. THE DEMOCRATIC PSO (DPSO) ALGORITHM

The Democratic PSO algorithm is another variant of the PSO algorithm, which was recently developed by Kaveh and Zolghadr [30] to strengthen the PSO exploration ability. The DPSO algorithm has basically the same implementing steps as PSO algorithm, but the update equation of solutions is modified by adding an supplementary term. This term reflects the "democratic" effect of particles swarm upon the movement (in solutions space) of a certain i particle. DPSO algorithm is based on a population of particles or individuals (called swarm) that cooperate among themselves to obtain the optimal solution. A particle is a solution to the problem, and its quality is assessed through F objective function.
We consider an n-dimensional searching space and a N particles population. A particle i is represented by two elements: the i particle position in n-dimensional space as represented by $[X_i]=[x_{1i}, x_{2i},\dots,x_{ji},\dots x_{ni}]$, i=1,2,…,N and the velocity of the particle represented by $[V_i]=[v_{1i}, v_{2i},\dots,v_{ji},\dots v_{ni}]$, i=1,2,…,N vector. In DPSO algorithm, the i particle movement from a $[X_i(k)]$ position (corresponding to k iteration) to a different $[X_i(k+1)]$ position (corresponding to k+1 iteration) involves the $[X_i(k)]$ and $[V_i(k)]$ vectors updating using the equations:

$$x_{ij}(k+1)=x_{ij}(k)+v_{ij}(k+1) \qquad (7)$$

$$v_{ij}(k+1)=\omega\cdot v_{ij}(k)+c_1\cdot r_1(PLocal_{ij}(k)-x_{ij}(k))+ c_2\cdot r_2(PGlobal_j(k)-x_{ij}(k))+c_3\cdot r_3\cdot d_{ij}(k) \qquad (8)$$

where, $x_{ij}(k)$ and $x_{ij}(k+1)$ represent the j component position of i particle for k iteration, k+1 respectively;
$v_{ij}(k)$ and $v_{ij}(k+1)$ represents the j component velocity of i particle for k iteration, k+1 respectively; $PLocal_i(k)$ is the best solution of the i particle founded until k iteration ($PLocal_{ij}(k)$ is the jth component of $PLocal_i(k)$ solution); $PGlobal(k)$ is the best solution gained by swarm until k iteration; $PGlobal_j(k)$ is j component of $PGlobal(k)$ solution; ω is inertia weight factor.
A way of calculating it is $\omega=\omega_{\max}-(\omega_{\max}-\omega_{\min})k/k_{max}$.
where, $\omega_{\min}$ and $\omega_{\max}$ are the minimum and maximum values of ω factor; k_{max} is the maximum number of iterations ; k is the current number of iterations.
c_1, c_2 are cognitive and social acceleration coefficients; c_3 is a factor used in order to control the $d_{ij}(k)$ "democratic" component from relation (8); r_1, r_2, r_3 are numbers

uniformly distributed in (0,1) range; $d_{ij}(k)$ is j component of $D_i(k)$ democratic vector of i particle.

The $D_i(k)$ vector reflects "the democratic effect" of the particles from swarm upon i particle and is calculated by relation (9) [30]:

$$D_i(k) = \sum_{p=1}^{N} Q_{ip}(X_p(k) - X_i(k)) \quad (9)$$

$$Q_{ip} = E_{ip}\frac{F_{Best}}{F_p} \Bigg/ \sum_{q=1}^{N} E_{iq}\frac{F_{Best}}{F_q} \quad (10)$$

$$E_{ip} = \begin{cases} 1 & if \ \dfrac{F_p - F_i}{F_{Worst} - F_{Best}} > rand \vee (F_p < F_i) \\ 0 & else \end{cases} \quad (11)$$

where, $X_p(k)$, $X_i(k)$ are the associated solutions to p and i particles, at k iteration; F_{Best}, F_{Worst} represent the best value, respectively the worst value of F objective function (obtained till k iteration); F_p, F_i are the values of objective function adequate to p and i solutions; Q_{ip} is a proportionality factor determined by relation (10); E is a binary parameter which is calculated using relation (11).

The values of i particle (V_i, i=1,2,..N) for j unit are comprised between the ($v_{j,min}$) minimum and ($v_{j,max}$) maximum limits, which can be assessed by the relations: $v_{j,max}=(P_{j,max}-P_{j,min})\cdot\beta$ and $v_{j,min}=-v_{ji,max}$, j=1,2,…,n. β factor was considered between 0.1 and 0.25.

The relation (8) shows that the PSO Democratic algorithm contains an additional ($c_3 \cdot r_3 \cdot d_{ij}(k)$) term to PSO algorithm. The introduction of this term can improve the PSO algorithm performance in two ways [30]:

(i) the particles in the swarm can receive additional information about the regions with potential in the search space of solutions;

(ii) a few of the low performant particles are allowed to influence the movement of the swarm in order to improve the exploring capacities of the algorithm.

4. IMPLEMENTING THE DPSO ALGORITHM FOR THE EcD PROBLEM

The structure of each i particle is represented by a n dimensional $[P_i]=[P_{1i}, P_{2i},…,P_{ji},…,P_{ni}]$ vector, its components being the real powers of the generating units. The application of PSO algorithm involves the following steps:

Step 1. The introducing the input data of the problem and the setting of parameters for DPSO algorithm;

Step 2. The initializing of population:

2.1 The initializing of k iterations counter (k=0);

2.2 The initializing of the population by random setting of the initial position and velocity of swarm particles:

$$P_{ji}(0)=P_{j,min}+\text{rnd}(1)(P_{j,max}-P_{j,min}),\ i=1,2,..N,\ j=1,2,,n \quad (12)$$

$$v_{ji}(0)=v_{j,min}+\text{rnd}(1)(v_{j,max}-v_{j,min}),\ i=1,2,..N,\ j=1,2,\dots,n \quad (13)$$

where, $P_{j,min}$, $P_{j,max}$ are the minimum and maximum limits of j unit; $v_{j,min}$, $v_{j,max}$ are the minimum and maximum limits of velocity for j unit; $P_{ji}(0)$ and $v_{ji}(0)$ are the values of powers and velocity for i particles and j unit, at k=0 iteration; rnd(1) is a number uniformly distributed in (0,1) range;

2.3 The evaluation of $F_i(0)=F[X_i(0)]$ objective function for each initial solution $X_i(0)$, i=1,2,..N;

2.4 The identification of the best $PLocal_i(0)$, i=1,2,..N particle and the best $PGlobal(0)$ global particle, at k=0 iteration;

Step 3. The update of velocity and particles position.

3.1 The determination of velocity and of the particles position is realized at each k iteration using (7) and (8) relations;

3.2 Verifies if the velocity and the particles position is between the minimum and maximum limits. Verifies if each generating unit satisfies (3)-(5) restrictions;

Step 4. Updates the $PLocal_i(k+1)$ and $PGlobal(k+1)$ vectors if the value of objective functions was improved in relation to the previous iteration;

Step 5. Stopping process. If $k<k_{max}$, then $k=k+1$ and go to step 3; Otherwise go to step 6;

Step 6: Print the X_{best} best solution and the adequate value of $F_{best}=F(X_{best})$ objective function.

5. CHAOTIC DPSO ALGORITHM

The DPSO algorithm shown in section 4 utilizes random numbers of r_1, r_2, r_3 to determine the velocity $v_{ij}(k)$ from equation (8). In this section is inserted chaos in updating equation (8) of the velocity ($v_{ij}(k)$). Thus, random numbers r_1, r_2, r_3 from equation (8) are replaced with chaotic sequences (cr_1, cr_2 and cr_3) generated by Sine chaotic map [31]. The chaotic sequences generated by Sine map is based on recurrence equations:

$$\begin{aligned} cr_{1(h+1)} &= (a/4)\cdot\sin(\pi\cdot cr_{1(h)}), \\ cr_{2(h+1)} &= (a/4)\cdot\sin(\pi\cdot cr_{2(h)}), \\ cr_{3(h+1)} &= (a/4)\cdot\sin(\pi\cdot cr_{3(h)}),\ h=1,2,3,\dots \end{aligned} \quad (14)$$

The a parameter varies in (0,4] interval and the chaotic sequences ($cr_{1(h)}$, $cr_{2(h)}$ and $cr_{3(h)}$, h=1,2,3,…) varies in (0,1) interval. In this paper was chosen for the parameter a the standard value of (a=4). In scientific literature it shows that introductions of chaotic maps in metaheuristic algorithms may lead to its performance improvement [31, 32, 33]. In this paper, insertion of the chaotic sequences in DPSO algorithm aims the same purpose (performance enhancement of DPSO algorithm). We note that algorithm resulted by insertion of the chaotic sequences in the DPSO algorithm will be called DPSO-Sine. The steps for implementation of DPSO-Sine algorithm are similar to the ones presented in section 4.

6. CASE STUDIES

The DPSO and DPSO-Sine algorithms have been applied to analyze two systems consisting of 13, respectively 40 generating units. The first system is studied in two cases (considering for demanded power the values consecrated in scientific literature: P_{Demand}=1800 MW and P_{Demand}=2520 MW) and for the second system, a single case is considered (the demanded power being P_{Demand}=10500 MW). For each case were made 100 independent trials and were calculated the following statistics items: best total fuel cost *F* (B), average total fuel cost *F* (A), worst total fuel cost *F* (W) and standard deviation (SD).

The (N, k_{max}) parameters of DPSO and DPSO-Sine algorithms were selected by conducting several experimental tests for each case studied and for the (c_1, c_2, c_3) parameters were utilized the recommended values from [30]: c_1=2, c_2=2 and c_3=4. The DPSO and DPSO-Sine algorithms were implemented in Mathcad, utilizing a PC with Intel i5 processor having the following characteristics: 2.2 GHz CPU and 4 GB of RAM.

6.1 Test system 1 (with 13 units)

The first two case study (C_1 and C_2) analyzes a system having 13 generating units with valve point loading. The C_1 case study utilize a demanded power of P_{Demand}=1800MW and the second one (C_2), a demanded power of P_{Demand}=2520 MW. We mention that both values of the demanded power are frequently used in comparison of different algorithms for EcD problem solving. The input data related to cost coefficients of the generating units and power operating limits are taken from [8].

For the C_1 (1800 MW) and C_2 (2520 MW) cases, the best solutions obtained by applying of the DPSO and DPSO-Sine algorithms are showed in Table 1. Also, statistics items B, A, W and SD resulted by utilizing the DPSO and DPSO-Sine algorithms are presented in Table 2 (for C_2 case) and Table 3 (for C_3 case). Table 2 and Table 3 shows the obtained results, for the 13 unit system (C_2 and C_3 cases), by a variety of optimization techniques, such as: PSO [26, 28], PSO varieties [4, 13, 24, 27, 28, 29], PSO hybrids [11, 26, 27, 29, 34, 38] and other techniques (HS [5], BFO [6], GA [9], SA [10], ACO [14], ABC [17]).

Analyzing the results from Table 2 and Table 3 we can observe the following:

(i) for C_1 and C_2 cases the DPSO and DPSO-Sine algorithms obtains better qualities solutions than the algorithms mentioned in Table 2 and Table 3;

(ii) The DPSO-Sine is more performant (considering the B, A, W items) than PSO algorithm [9, 26, 28], than the algorithms derived from PSO (NewPSO [4], DPSO [13], IPSO [24], CPSO [27], CLPSO [29], NPSO [35]), then the hybrid PSO (PSO-SQP [26], CPSO-SQP [27], SQP-CLPSO [29], FCASO-SQP [34]), or than the other optimization techniques (HS [5], BFO [6], IFEP [8], NDS [10], ACO [14], MABC [17], TLBO [22], CASO [34]);

(iii) the DPSO-Sine algorithm is more performant than DPSO considering the B, A, W, and SD items. This shows that chaotic map Sine enhances the performance of the original DPSO algorithm.

Table 1 The best solutions obtained through DPSO and DPSO-Sine algorithms for 13-units, cases C_1 (P_{Demand}=1800 MW) and C_2 (P_{Demand}=2520 MW)

Algorithm Output	DPSO (C_1: 1800 MW)	DPSO-Sine (C_1: 1800 MW)	DPSO (C_2: 2520 MW)	DPSO-Sine (C_2: 2520 MW)
P_1 (MW)	628.31858	628.31826	628.30838	628.31731
P_2 (MW)	149.42636	223.86847	299.17433	299.18466
P_3 (MW)	223.08707	148.98806	299.19167	299.19001
P_4 (MW)	60.00000	60.00000	159.72715	159.72896
P_5 (MW)	109.84407	109.42881	159.73204	159.73119
P_6 (MW)	109.85780	109.83620	159.66598	159.71873
P_7 (MW)	109.82246	109.83059	159.72789	159.73234
P_8 (MW)	109.74806	109.86588	159.72585	159.73118
P_9 (MW)	109.89559	109.86372	159.72820	159.72754
P_{10} (MW)	40.00000	40.00000	77.36594	77.39745
P_{11} (MW)	40.00000	40.00000	77.37821	77.39769
P_{12} (MW)	55.00000	55.00000	87.90567	92.39801
P_{13} (MW)	55.00000	55.00000	92.36869	87.74492
B ($/h)	17964.555	17964.372	24170.232	24170.015

Table 2 Comparison between DPSO and DPSO-Sine algorithms and other optimization techniques (the case C_1: 13-units, P_{Demand}=1800 MW)

Algorithms/ Items	B ($/h)	A ($/h)	W ($/h)	SD ($/h)
NewPSO [4]	18120.594	18227.052	18427.631	-
PSO [28]	18030.72	18205.9247	18401.35	-
PSO [26]	18030.72	18205.78	-	-
IPSO [24]	17998.44	18176.95	-	-
IFEP [8]	17994.07	18127.06	18267.42	-
EP-SQP [26]	17991.03	18106.93	-	-
NDS [10]	17976.9512	17976.9512	17976.9512	0.0000
DPSO [13]	17976.31	18084.99	18310.43	-
SA [10]	-	18299.2550	-	123.8335
NPSO[35]	17976.015	-	-	-
HDE [12]	17975.73	18134.80	-	-
BFO [6]	17974.48	17997.12	18018.75	-
SQP-CLPSO [29]	17973.12	18005.05	18069.35	23.81023
CLPSO [29]	17970.67	18019.41	18071.57	22.67055
PSO-SQP [26]	17969.93	18029.99	-	-
CASO [34]	17965.15	18022.04	-	-
DPSO	17964.555	17975.687	17995.552	5.727
DPSO-Sine	17964.372	17973.049	17978.919	2.571

Table 3 Comparison between DPSO and DPSO-Sine algorithms and other optimization techniques (the case C_2: 13-units, P_{Demand} =2520 MW)

Items / Algorithms	B ($/h)	A ($/h)	W ($/h)	SD ($/h)
EP-SQP [26]	24266.44	-	-	-
PSO [28]	24262.73	24271.9231	24277.81	-
PSO-SQP [26]	24261.05	-	-	-
CASO [34]	24212.93	-	-	-
CPSO [27]	24211.56	-	-	-
MABC [17]	24208.8330	-	-	-
HS [5]	24208.7	24323.2	24503.7	-
TLBO [22]	24197	-	-	-
CPSO-SQP [27]	24190.97	-	-	-
FCASO-SQP [34]	24190.63	-	-	-
ACO [14]	24174.39	24211.09	24243.90	21.10
SA-PSO [11]	24171.395	-	-	-
GA–DE–PS [38]	24171.3467	-	-	-
TSA [9]	24171.211	24184.055	24392.203	41
GA [9]	24170.804	24188.394	24567.974	59.53
DPSO	24170.232	24173.968	24178.347	2.027
DPSO-Sine	24170.015	24172.885	24176.515	1.994

"-"data not available

The calculus times obtained by DPSO and DPSO-Sine algorithms have closed values, being approximately equals to 17 s.

6.2 Test system 2 (with 40 units)

The C_3 case study analyses a system with large sizes, having 40 generator units and its demanded power is P_{Demand}=10500 MW. The calculus data for this system are given in [8]. The best solutions obtained by the DPSO and DPSO-Sine algorithms are presented in Table 4. In Table 5 is made a comparison between the DPSO, respectively DPSO-Sine algorithms and other optimization techniques presented in literature.

Table 4 The best solutions obtained through DPSO and DPSO-Sine algorithms for 40-units, case C_3 (10500MW)

Output (MW)	DPSO	DPSO-Sine
P_1	110.85132	110.82092
P_2	110.83000	110.84145
P_3	97.39919	97.40097
P_4	179.74177	179.79721
P_5	88.10939	96.19795
P_6	139.99627	140.00000
P_7	259.63279	259.63430
P_8	284.63455	284.82380
P_9	284.64857	284.63962
P_{10}	130.00000	130.00000
P_{11}	168.80553	168.79623
P_{12}	94.00000	94.00000
P_{13}	214.76362	214.76721
P_{14}	394.28005	394.28113
P_{15}	394.28152	304.52193
P_{16}	304.53204	394.27870
P_{17}	489.27858	489.28249
P_{18}	489.31819	489.31098
P_{19}	511.27943	511.30607
P20	511.28544	511.27842
P21	523.27943	523.27701
P22	523.28939	523.32843
P_{23}	523.30836	523.28013
P_{24}	523.28214	523.27644
P_{25}	523.28419	523.27811
P_{26}	523.29463	523.29853
P_{27}	10.00000	10.00000
P_{28}	10.00000	10.00000
P_{29}	10.00000	10.00000
P_{30}	95.97791	88.06416
P_{31}	190.00000	190.00000
P_{32}	190.00000	190.00000
P_{33}	190.00000	190.00000
P_{34}	165.32995	164.93686
P_{35}	200.00000	200.00000
P_{36}	200.00000	200.00000
P_{37}	110.00000	110.00000
P_{38}	110.00000	110.00000
P_{39}	110.00000	110.00000
P_{40}	511.28577	511.28093
B ($/h)	121424.1275	121424.0947

Analyzing the results from Table 5 we can find that:
(i) the DPSO and DPSO-Sine algorithms are superior (considering B, A, W and SD items) to other optimization techniques utilized for solving the EcD problem (exception is the DE algorithm [36]);
(ii) the DPSO and DPSO-Sine algorithms are more performant than PSO algorithm [14, 15, 26], than diverse PSO varieties (QPSO [25], CPSO-SQP [27], FCASO-SQP [34]) and than other optimization techniques presented in Table 5 (ACO [14], GA [14], CSO [15], ABC [18], BBO [19], EP [26], DE [37],);
(iii) the DPSO-Sine algorithm is more performant than the DPSO algorithm in relation with B, A and W items.

The calculus times obtained by DPSO and DPSO-Sine algorithms have closed values, being approximately equals to 123 s.

Table 5 Comparison between DPSO and DPSO-Sine algorithms and other optimization techniques (the C_3case: 40-units, P_{Demand}=10500 MW)

Algorithm/ Items	B ($/h)	A ($/h)	W ($/h)	SD ($/h)
GSO [7]	124265.3984	124609.1799	125204.4753	-
PSO [26]	123930.45	124154.49	-	-
EP [26]	122624.35	123382.00	-	-

Table 5 - continous

Algorithm/ Items	B ($/h)	A ($/h)	W ($/h)	SD ($/h)
PSO [15]	122588.5093	123544.8853	124733.6795	-
GA [14]	121996.40	122919.77	123807.97	320.31
MILP [39]	121986	-	-	-
DE [37]	121840	-	-	-
PSO [14]	121800.13	121899.57	122000.80	84.21
ST-HDE [12]	121698.51	122304.30	-	-
CTLBO [40]	121553.83	121790.23	122116.18	150
ACO [14]	121532.41	121606.45	121679.64	45.58
SAHS [41]	121516.94	121694.49	121900.42	113.75
ABC [42]	121515.1	124827.34	-	-
ABC [18]	121479.6467	121984.24	122137.42	-
BBO [19]	121479.5029	121512.0576	121688.6634	-
CSO [15]	121461.6707	121936.1926	122844.5391	-
CPSO-SQP [27]	121458.54	122028.16	-	-
FCASO-SQP [34]	121456.98	122026.21	-	-
DE [36]	121442.2682	121448.8196	121457.2746	-
ABC [16]	121441.03	121995.82	122123.77	-
ABCLogistic [42]	121440.2	123314.12	-	-
BBO [20]	121426.9530	121508.0325	121688.6634	-
CSA [21]	121425.61	-	-	-
QPSO [25]	121424.6399	121586.9412	121994.0267	114.080
DPSO	121424.127	121491.889	121597.205	35.842
DPSO-Sine	121424.094	121459.909	121508.002	21.097

7. CONCLUSIONS

This paper utilizes the Democratic PSO (DPSO) algorithm to solve the economic dispatch problem, considering different operating constraints of the power generating units and of the power system. The DPSO algorithm is applied in original form, but also in a new form in which it is endowed with the chaotic Sine map. Including the Sine map has the purpose to increase original DPSO algorithm performance. The DPSO and DPSO-Sine algorithms have been tested on two systems in which the generating units are modeled taking in consideration the valve-point effect. The mathematical optimization model is nonlinear, having a higher complexity due to the number of variable involved, the type of equality and inequality restrictions and the fact that the objective function is non-convex.

The results obtained by DPSO and DPSO-Sine algorithms shows that these are more performant (considering B, A, W items) than PSO algorithm, but also than other mentioned techniques in this article (GA, ACO, HS, TSA, EP, GSO, ABC). Also, the algorithm endowed with the Sine map (DPSO-Sine) has a better behavior than DPSO (considering B, A, W items). This shows that including the chaotic maps in the metaheuristic algorithms may increase their performances by obtaining better quality solutions.

8. REFERENCES

[1]. Parikh J, Chattopadhyay D. *A multi-area linear programming approach for analysis of economic operation of the Indian power system*. IEEE Transaction on Power Systems 1996; 11(1):52–8.

[2]. Wood A, Wollenberg BF, *Power Generation Operation and Control*, John Wiley&Sons, New York, 1984.

[3]. Fan JY, Zhang L. *Real-time economic dispatch with line flow and emission constraints using quadratic programming*. IEEE Transaction on Power Systems 1998; 13(2):320–325.

[4]. Chaturvedi KT, Pandit M, Srivastava L. *Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch*. International Journal of Electrical Power & Energy Systems 2009; 31(6):249-257.

[5]. Fesanghary M, Ardehali MM. *A novel metaheuristic optimization methodology for solving various types of economic dispatch problem*. Energy, 2009; 34 (6): 757-766.
[6]. Vijay R. *Intelligent Bacterial Foraging Optimization Technique to Economic Load Dispatch Problem*. International Journal of Soft Computing and Engineering 2012; 2(2): 55-59.
[7]. Moradi-Dalvand M, Mohammadi-Ivatloo B, Najafi A, Rabiee A. *Continuous quick group search optimizer for solving non-convex economic dispatch problems*. Electric Power System Research, 2012; 93: 93–105.
[8]. Sinha N, Chakrabarti R, Chattopadhyay PK. *Evolutionary programming techniques for economic load dispatch*. IEEE Transaction on Evolutionary Computation 2003; 7(1): 83–94.
[9]. Khamsawang S, Jiriwibhakorn S. *DSPSO-TSA for economic dispatch problem with nonsmooth and noncontinuous cost functions*. Energy Conversion and Management 2010; 51(2): 365–375.
[10]. Lin W-M, Gow H-J, Tsai M-T. *Combining of Direct Search and Signal-to-Noise Ratio for economic dispatch optimization*. Energy Conversion and Management 2011; 52(1): 487-493.
[11]. Kuo CC. *A novel coding scheme for practical economic dispatch by modified particle swarm approach*. IEEE Transaction on Power Systems 2008; 23(4): 1825-1835.
[12]. Wang S-K, Chiou J-P, Liu C-W. *Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm*. IET Generation, Transmission & Distribution 2007; 1(5):793–803.
[13]. Kumar R, Sharma D, Sadu A. *A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch*. International Journal of Electrical Power and Energy Systems 2011; 33(1): 115-123.
[14]. Pothiya S, Ngamroo I, Kongprawechnon W. *Ant colony optimisation for economic dispatch problem with non-smooth cost functions*. International Journal of Electrical Power and Energy Systems 2010; 32 (5): 478–87.
[15]. Selvakumar AI, K. Thanushkodi K. *Optimization using civilized swarm: solution to economic dispatch with multiple minima*. Electric Power System Research 2009; 79(1): 8–16.
[16]. Hemamalini S, Simon SP. *Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions*. Electric Power Components and Systems 2010; 38(7):786–803.
[17]. Hardiansyah. *Solving Economic Dispatch Problem with Valve-Point Effect using a Modified ABC Algorithm*. International Journal of Electrical and Computer Engineering, 2013; 3(3): 377-385.
[18]. Labbi Y, Attous Db, Mahdad B. *Artificial bee colony optimization for economic dispatch with valve point effect*, Front Energy 2014, 8(4): 449–458.
[19] Bhattacharya A, Chattopadhyay PK. *Solving complex economic load dispatch problems using biogeography-based optimization*, Expert Systems with Applications 2010; 37(5): 3605–3615.
[20]. Bhattacharya A, Chattopadhyay PK. *Biogeography-based optimization for different economic load dispatch problems*. IEEE Transaction on Power Systems 2010; 25(2): 1064–1077.
[21]. Basu M, Chowdhury A. *Cuckoo search algorithm for economic dispatch*. Energy 2013; 60: 99-108.
[22]. Banerjee S, Maity D, Chanda CK. *Teaching learning based optimization for economic load dispatch problem considering valve point loading effect*. International Journal of Electrical Power and Energy Systems, 2015; 73:456–464.
[23]. Gaing Z-L. *Particle swarm optimization to solving the economic dispatch considering the generator constraints*. IEEE Transaction on Power Systems 2003; 18(3): 1187–1195.
[24]. Abdullah MN, Bakar AHA, Rahim NA, Jamian JJ, Aman MM. *Economic Dispatch with Valve Point Effect using Iteration Particle Swarm Optimization*. 47th International Universities Power Engineering Conference (UPEC), London, UK, pp. 1-6, Sept. 2012.
[25]. Hosseinnezhad V, Rafiee M, Ahmadian M, Ameli MT. *Species-based Quantum Particle Swarm Optimization for economic load dispatch*. International Journal of Electrical Power and Energy Systems 2014; 63: 311–322.
[26]. Victoire TAA, Jeyakumar AE. *Hybrid PSO–SQP for economic dispatch with valve-point effect*. Electric Power System Research 2004; 71(1): 51–59.
[27]. Cai J, Li Q, Li L, Peng H, Yang Y. *A hybrid CPSO–SQP method for economic dispatch considering the valve-point effects*. Energy Conversion and Management 2012; 53(1): 175-181.
[28]. Niknam T. *A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem*. Applied Energy 2010; 87(1): 327-39.
[29]. Wang Y, Li B, Yuan B. *Hybrid of comprehensive learning particle swarm optimization and SQP algorithm for large scale economic load dispatch optimization of power system*. Science China-Information Sciences, 2010; 53(8): 1566–1573.
[30]. Kaveh A, Zolghadr A. *Democratic PSO for truss layout and size optimization with frequency*. Computers and Structures 2014; 130: 10–21.
[31]. Gandomi AH, Yang XS. *Chaotic bat algorithm*. Journal of Computational Science 2014; 5(2): 224–232.
[32]. Ghasemi M, Ghavidel S , Akbari E , Vahed AA. *Solving non-linear, non-smooth and non-convex optimal power flow problems using chaotic invasive weed optimization algorithms based on chaos*. Energy 2014; (73): 340-353
[33]. Arul R., Velusami S. and Ravi G. *Solving economic load dispatch problems using chaotic self adaptive differential harmony search algorithm*. International Transaction on Electrical Energy Systems; 2015, 25(5): 845–858.
[34]. Cai J, Li Q, Li L, Peng H, Yang Y. *A hybrid FCASO-SQP method for solving the economic dispatch problems with valve-point effects*. Energy 2012; 38(1): 346–353.
[35]. Tsai MT, Yen CW. *The influence of carbon dioxide trading scheme on economic dispatch of generators*. Applied Energy 2011; 88(12): 4811–6.
[36]. Basu M. *Improved differential evolution for economic dispatch*. International Journal of Electrical Power and Energy Systems 2014, 63: 855–861.
[37]. Basu M. *Economic environmental dispatch using multi-objective differential evolution*. Applied Soft Computing 2011; 11(2): 2845–2853.
[38]. Mahdad B, Srairi K. *Solving practical economic dispatch using hybrid GA–DE–PS method*, International Journal of System Assurance Engineering and Management 2014; 5(3): 391–398
[39]. Fraga ES, Yang L, Papageorgiou LG. *On the modelling of valve point loadings for power electricity dispatch*. Applied Energy 2012; 91: 301–303.
[40]. He X, Rao Y, Huang J. *A novel algorithm for economic load dispatch of power Systems*. Neurocomputing 2016, 171(1): 1454–1461.
[41]. Niu Q, Zhang H, Wang X, Li K, Irwin GW. *A hybrid harmony search with arithmetic crossover operation for economic dispatch.* International Journal of Electrical Power and Energy Systems 2014; 62: 237–257.
[42]. Shayeghi H, Ghasemi A. *A modified artificial bee colony based on chaos theory for solving non-convex emission/economic dispatch*, Energy Conversion and Management 2014; 79(3): 344–354.