A Hybrid Particle Swarm Optimization Algorithm for the Economic Dispatch Problem

Dinu Calin Secui
Department of Energy Engineering, University of Oradea, Oradea, Romania
Email: csecui@uoradea.ro

Received: March 2014 Revised: June 2014 Accepted: August 2014

ABSTRACT
This article proposes a hybrid global-local algorithm - Hybrid Particle Swarm Optimization (HPSO) - applied to solve the Economic Dispatch (ED) problem. The HPSO algorithm combines the classical Particle Swarm Optimization (PSO) with the Conjugate Gradient (CG) non-linear optimizing method, included in the optimizing tool within MathCAD commercial software product. The global optimizer is the PSO algorithm, and the local one is the CG method. Two variants including the CG within the PSO, which are analyzed, called HPSO-RC (randomly controlled) and HPSO-RU (randomly uncontrolled). Both PSO and CG methods are easy to implement and together help reaching the best solution. The HPSO algorithm's ability to avoid premature convergence and provide a stabile solution is tested on three systems consisting of 6, 13 and 38 thermal generating units. The HPSO algorithm's efficiency in solving the ED problem is shown through a comparison with several other recently published algorithms.

KEYWORDS: Economic Dispatch, Transmission Losses, Particle Swarm Optimization, Hybridization, Optimizing Tool.

1. INTRODUCTION

Economic Dispatch (ED) is a fundamental optimizing problem in planning, operating and controlling the electric power system. The main purpose of ED is to calculate the power outputs of all generating units of a given system as to have minimum fuel cost on the overall system, and meet the equality and inequality restrictions demanded. In the simplified form of the mathematic model of the ED, the equality restriction is given by a single relation of the overall system's power balance, and the inequality restrictions refer to keeping the power of the generating units in its operating limits.
The overall fuel cost is the sum of all fuel costs of each generating unit. Conventionally the fuel cost for each generator is defined by a single quadratic function with or without the valve-point effects. Taking into consideration the valve-point effects determines the cost's non-smooth and non-convex characteristics for the generating units. Also, it takes into consideration the transmission losses, ramp rate limits and prohibited operating zones of the units. These characteristics determine a nonlinear and non-continuous mathematical model of optimization.
So far many optimizing methods, classic or meta-heuristic, have been used to solve the ED problem. The classical mathematical methods including non-linear programming [1], quadratic programming [2], and dynamic programming [3] have difficulties in finding the global optimal solution due to the generating units' non-linear and non-convex characteristics. As a consequence, a series of meta-heuristic search and optimizing algorithms have been developed to find the global and quasi-global solutions for the ED problem. Among them are the genetic algorithms [4, 5], Tabu search algorithms [6], Hopfield neural networks [7], evolutionary algorithms [8, 9], differential evolution [10], fuzzy systems [11], harmony search [12, 13], ant colony optimization [14], chaotic optimization algorithm [15], artificial bee colony [16], versions of the PSO algorithm [17-19] and hybrid artificial intelligence [20-25].
The hybrid methods combine either various meta-heuristic algorithms among them or classic optimizing techniques with meta-heuristic algorithms to enforce the hybrid's capacity to explore and exploit the solutions' space, and in the end, to provide only high-quality stable solutions. The recent publications show hybrid algorithms used in solving the ED problem, combining: PSO with the Sequential Quadratic Programming (SQP) [21], genetic algorithm with SQP technique [22, 23], PSO with tabu search based algorithm [24].
To solve continuous optimization problems several hybrid algorithms are suggested: Fesanghary and Ardehali (2009) combined the harmony search (HS) algorithm with SQP method [13]; Qteish-Hamdan and

Chen et al. (2007) enhanced PSO with conjugate gradient [26, 27]; Kayhan et al. (2010) presented the PSO algorithm with a spreadsheet “solver” [28]; Ayvaz et al. (2009) associated the HS algorithm with a spreadsheet “solver” [29]. The “solver” is an optimizing module that is part of various commercial spreadsheet products (such as Microsoft Excel®, Lotus 1-2-3® etc.), with the ability to solve linear and non-linear optimization problems.

Based on this principle, this article suggests a hybrid algorithm to solve the ED problem which combines the classical PSO technique with the CG method comprised in the optimizing tool within MathCAD® commercial software product. Also, an improved variant to include the CG method within the PSO algorithm is suggested, called HPSO-RC. HPSO-RC is compared with another variant to include CG within PSO (HPSO-RU) and described in other studies [13, 28]. The PSO algorithm is the main optimizer and the MathCAD optimizing tool provides a fine-tuning of the solutions given by the PSO algorithm. The two variants of HPSO method (HPSO-RC and HPSO-RU) are applied to study three test systems and then compared with several other published methods from solutions’ stability and quality point of view.

2. ED PROBLEM FORMULATION

We consider a power system containing n generating units, each unit having its own generated power P_j, j=1,2,...n, and with the solution vector $P=[P_1, P_2,\ldots,P_j,\ldots,P_n]^T$. The fuel cost $F_j(P_j)$, in \$/h, for each generator j, is represented by a quadratic polynomial function such as:

$$F_j(P_j) = c_j P_j^2 + b_j P_j + a_j \tag{1}$$

If the valve-point effects are taken into consideration, then the cost function for unit j includes also a sine factor [5, 8, 21]:

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

Where a_j, b_j and c_j are fuel cost coefficients of generator j, and e_j and f_j are the coefficients of generator j reflecting the valve-point effects. P_j represents the output power of generator j, in MW.

The solution for the ED problem consists in determining the P_j powers of the generating units, so that the total fuel cost of the entire system is minimal, respecting the restriction of power balance on the overall system and the inequality restrictions for each unit j. The objective function is:

$$\min F = \sum_{j=1}^{n} F_j(P_j) \tag{3}$$

The ED problem constraints can be expressed using the inequality and equality relations (4)-(9) [10]:

i) Minimum and maximum real power operating limits:

$$P_{j,min} \le P_j \le P_{j,max},\ j=1,2,\ldots,n \tag{4}$$

Where $P_{j,min}$ and $P_{j,max}$ represent the minimum and the maximum operating limits of a generator j.

ii) Generator ramp-rate limits:

$$P_j \le P^0_j + UR_j, \text{ if output power increases} \tag{5}$$

$$P_j \ge P^0_j - DR_j, \text{ if output power decreases} \tag{6}$$

Where P^0_j is the previous hour output power of unit j. DR_j and UR_j are the down-ramp and up-ramp limits of the j unit (in MW/time-period).

Relations (4)-(6) can also be expressed by:

$$PO_{j,min} \le P_j \le PO_{j,max} \tag{7}$$

where $PO_{j,min}$= max($P_{j,min}$, $P^0_j - DR_j$) and

$PO_{j,max}$= min($P_{j,max}$, $P^0_j + UR_j$).

iii) Generator’s prohibited operating zones:

$$\begin{cases} P_{j,min} \le P_j \le P^l_{j,1} \\ P^u_{j,z-1} \le P_j \le P^l_{j,z} \qquad , z=2,3,\ldots, NZ_j \\ P^u_{j,NZ_j} \le P_j \le P_{j,max} \end{cases} \tag{8}$$

Where NZ_j is the number of prohibited zones of unit j. $P^l_{j,z}$ and $P^u_{j,z}$ are the lower and upper boundary of the z is the prohibited operating zone for the unit j.

iv) Real power balance constraint:

$$P_G - P_L - P_D = 0 \tag{9}$$

Where P_D is the load demand in the system, in MW. P_L represents the transmission loss, in MW.

The transmission losses P_L at the entire system level are quadratic functions in relation to variables P_j and they are calculated using constant B coefficient formula:

$$P_L = \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} \tag{10}$$

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.

The total generated power (P_G) by the n units is:

$$P_G = \sum_{j=1}^{n} P_j \tag{11}$$

In solving the ED problem relation (9) is satisfied with a certain tolerance. This tolerance (TOL_M) is calculated using the best solution obtained with one method M, with relation:

$$TOL_M = P_G - P_L - P_D \tag{12}$$

3. SOLUTION OF ED PROBLEM WITH HPSO

3.1. Classical Particle Swarm Optimization

Particle swarm optimization algorithm is a population-based optimization technique, which models the social behavior of birds flocking or fish schooling. The PSO method is used to solve non-linear and non-continuous optimization problems, and was introduced by Kennedy and Eberhart (1995) [30]. Since then more PSO techniques have been developed, but in this paper references will be made only to the classical PSO algorithm [31, 32].

To search for a solution in an *n*-dimensional space, PSO uses a population of *N* particles. Particle *i* of population *N*, at iteration *k*, will have the solutions vector $X_i^k = (x_{i1}^k, x_{i2}^k, ..., x_{ij}^k, ..., x_{in}^k)$. During the optimization process, the passing from solution (X_i^k) to solution (X_i^{k+1}) is accomplished using the particles velocity described by the vector $V_i^k=(v_{i1}^k, v_{i2}^k, ..., v_{ij}^k, ..., v_{in}^k)$, according to the following relation:

$$X_i^{k+1} = X_i^k + V_i^{k+1}, \; i=1, 2, ..., N \tag{13}$$

The updated particle velocity in the next iteration ($k+1$) is given by:

$$V_i^{k+1} = w^k \cdot V_i^k + c_1 \cdot r_1 \cdot \left(Pbest_i^k - X_i^k\right) + c_2 \cdot r_2 \cdot \left(Gbest^k - X_i^k\right) \tag{14}$$

Where V_i^k, V_i^{k+1} represent the velocity vector of particle *i* at iteration *k*, respectively $k+1$;

X_i^k, X_i^{k+1} represent the solution vector of particle *i* at iteration *k*, respectively $k+1$;

$Pbest_i^k$ represents the best solution vector of particle *i*, until iteration *k*;

$Gbest^k$ represents the vector corresponding to the best solution of the swarm, until iteration *k*;

c_1 and c_2 are coefficients corresponding to cognitive and social behaviour;

r_1 and r_2 are random numbers between 0 and 1.

The inertia weight factor *w* has a linear decrease from w_{max} to w_{min} and is usual determined by relation:

$$w^k = w_{max} - (w_{max} - w_{min})k/k_{max} \tag{15}$$

where w_{max} and w_{min} represent the maximum and minimum of the inertia *w*; *k* is the current number of iterations, and k_{max} is the maximum number of iterations from the optimizing process.

3.2. HPSO implementation in ED problem

HPSO is a global-local hybrid optimization algorithm. The global exploring of the solutions' space is provided by the classical PSO, and the local exploiting is done using a non-heuristic algorithm available in the optimizing tool included in the MathCAD commercial software product. MathCAD optimizing tool provides the conjugate gradient method for optimizing the non-linear problems with constraints. For this paper the conjugate gradient (CG) method was used as a local optimizer.

Three issues arise in using the CG method: (i) implementing this method to function as the local optimizer with constraints in solving ED; (ii) integrating it in the PSO global optimizer; (iii) PSO global optimizer implementation.

3.2.1. CG(P) local optimizer implementation

The implementation of the CG method in MathCAD to perform a local search (named *CG*(*P*)) is described in the following MathCAD sequence:

{*P* is the start solution vector for *CG*(*P*) and *CG*(*P*) is the local optimizer}

$F(P) \leftarrow$ the objective function defined by relations (1) and (3) or (2) and (3) (16)

Given {reserved word in MathCAD used before the inequality and equality constraints block}

$P_{min} \le P \le P_{max}$ {operating restrictions (4)} (17)

$P^0 - DR \le P \le P^0 + UR$ {ramp-rate restrictions (5) and (6)} (18)

$(P_{min} \le P \le P^l_1) \vee (P^u_1 \le P \le P_{max})$ {Implementing the restrictions for a single prohibition zone} (19)

$\sum_{j=1}^{n} P_j - P_L(P) - P_D = 0$ {Power balance relation (9)} (20)

$CG(P) := minimize(F,P)$ {performs the actual minimization} (21)

Within the MathCAD module defined by relations (16)-(21) the following notations were used: $[P_{min}]_n$ and $[P_{max}]_n$ are the column vectors, of size *n*, representing the minimum and maximum values of the generating units; $[P^0]$, $[DR]$ and $[UR]$ are the column vectors, of size *n*, representing the previous output powers, down-ramp and up-ramp limits of the units; $[P^l_z]_n$ and $[P^u_z]_n$ are the column vectors representing the lower and upper limits for a certain prohibition zone *z*. If the generating units have more prohibition zones, then these zones are included in the "*Given*" block using the "$\vee$" operator ("logic OR" symbol). For example, if each unit has two prohibition zones, relation (19) is rewritten like: $(P_{min} \le P \le P^l_1) \vee (P^u_1 \le P \le P^l_2) \vee (P^u_2 \le P \le P_{max})$. It is noticed that in order to apply the restrictions (17)-(20) the MathCAD software features are used, that allows writing the relations in vector form (merged). In the case where the generating units have a different number of prohibition zones, then the restrictions can be written for each unit *j* (for example, in case of a single prohibition zone for each unit *j*, relations (19) are written as follows: $(P_{j,min} \le P_j \le P^l_{j,1}) \vee (P^u_{j,1} \le P_j \le P_{j,max})$, $j=1,2,..,n$).

Launching the $CG(P)$ local search is done inside the PSO algorithm using the n-dimensional vector P corresponding to particle i. The P vector is one solution generated by the PSO algorithm and used as the start solution for the MathCAD function *minimize(F,P)*. The MathCAD function - *minimize(F,P)* – performs the actual minimization and returns one local minimum which satisfies the constraints from *Given block*. This local minimum, put in $CG(P)$, is then saved in the variable called *PLocal* ($PLocal \leftarrow CG(P)$), for future computation.

3.2.2. Including the CG(P) local optimizer in PSO

The global and local search processes integration strategy is important in order to perform an efficient computation and get a high quality solution. The literature contains more integration strategies of the global-local search. In [29] and [33] two of these strategies are described. The first strategy applies the global search algorithm to determine as good a solution. The returned solution following the global search process is then used as a start solution for the local search process. In the second strategy, the global search algorithm runs simultaneously with the local search algorithm. Thus, at each iteration k, the global search returns a start solution for the local search. The optimizing process is longer than for the previous case, but it can provide better quality solutions [33]. Still, the experiments preformed by many authors on different hybrids [13, 28, 29] show that a relatively small number for calling the local search is sufficient to get a high quality solution and computational efficiency. In the mentioned papers the call of the local search is done randomly with a probability Pc found between 0.01 and 0.1.

This paper uses the second strategy, applied in two variants:

(i) the first variant, called randomly uncontrolled is identical with the one used in other papers [13, 28]. Thus, the local search $CG(P)$ is randomly launched when the following condition is met:

$$rnd(1) \le P_c \tag{22}$$

This variant of including the $CG(P)$ in the PSO global search is called HPSO-RU. Applying the HPSO-RU causes some particles (solutions) generated by the PSO which never been accepted as start solutions for the $CG(P)$, while other particles to be accepted for a greater number of times. This unbalance can leads to a disadvantage for the HPSO-RU, because a part of the search space is not exploited by the local search $CG(P)$.

(ii) The second variant, called randomly controlled, is suggested by the author in order to improve the HPSO hybrid. This variant of including $CG(P)$ in PSO is called HPSO-RC. The HPSO-RC variant creates a relative uniform distribution of the number of $CG(P)$ launches for each particle. Therefore the total number of times when the $CG(P)$ local optimizer is launched is limited to the $[\mathrm{trunc}(k_{max}\cdot P_c\cdot\alpha)+1,\ \mathrm{trunc}(k_{max}\cdot P_c\cdot\beta)+1]$ interval. The "$trunc(x)$" function is the integer part of the real number x. The scalar quantities α and β control the maximum and minimum number of times that the $CG(P)$ local optimizer is launched.

The distribution process of the $CG(P)$ launches is random, but is controlled through the following conditions set:

$$(rnd(1) \le P_c) \text{ and } (N_i \le k\cdot P_c\cdot\beta) \tag{23}$$

$$(rnd(1) > P_c) \text{ and } (N_i \le k\cdot P_c\cdot\alpha) \tag{24}$$

where $rnd(1)$ is a random number between 0 and 1; N_i is the number of times which one particle i is chosen as a starting solution when the $CG(P)$ was launched, through the entire optimizing process, $N_i\in[\mathrm{trunc}(k_{max}\cdot P_c\cdot\alpha)+1,\ \mathrm{trunc}(k_{max}\cdot P_c\cdot\beta)+1]$.

The condition ($N_i \le k\cdot P_c\cdot\beta$) limits the number of times that the CG(P) is launched for particle i to the maximum value $\mathrm{trunc}(k_{max}\cdot P_c\cdot\beta)+1$, with a goal to lower the computing time. The condition ($N_i \le k\cdot P_c\cdot\alpha$) ensures a minimum number of $CG(P)$ launches for each particle i.

As long as one of the relations (23) or (24) are fulfilled, $CG(P)$ returns a solution (named *PLocal*) which is compared with the starting solution of particle i, defined by (P_i) vector. The comparison is done by the evaluating of the objective function using relation (3), as follows: If $F(PLocal)<F(P_i)$ then the starting solution corresponding to particle i is updated ($P_i \leftarrow PLocal$), otherwise P_i solution remains unchanged. Also, both $Pbest_i$ and *Gbest* are updated:

if $F(P_i)<F(Pbest_i)$ then $Pbest_i\leftarrow P_i$, else it remains unchanged (25)

if $F(P_i)<F(Gbest)$ then $Gbest\leftarrow P_i$, else it remains unchanged (26)

3.2.3. PSO global optimizer implementation

The variant in which the global-local optimization algorithm is applied in order to solve the ED problem is given in the following steps:

Step 1: Set the HPSO parameters. For the HPSO algorithm the following parameters must be set: the maximum number of iterations (k_{max}), the number of particles (N), coefficients c_1 and c_2, quantities w_{min}, w_{max}.

Step 2: Swarm initialization. The $P_i=[P_{i1},P_{i2},\ldots,P_{ij},\ldots,P_{in}]$ swarm's particles are randomly initialized together with their velocity $V_i=[v_{i1},v_{i2},\ldots,v_{ij},\ldots v_{in}]$, i=1,2,...,N. Initializing the solution satisfying the relations (7)-(9) is done as follows:

2.1 To ensure that relation (7) is respected, the following relation is used:

$$P_{ij} = PO_{j,min} + rnd(1)(PO_{j,max} - PO_{j,min}),\ j=1,2,\ldots,n \quad (27)$$

2.2 If the ED problem is solved without the relations (5) and (6), then relation (27) is changed as follows: $PO_{j,min}$ is replaced with $P_{j,min}$, and $PO_{j,max}$ with $P_{j,max}$;

2.3 In order to satisfy the relation (8) it proceeds as follows: if P_{ij} is randomly generated according to relation (27) and belongs to prohibition zone z (having the limits $P^l_{j,z}$ and $P^u_{j,z}$), then P_{ij} is set with the closest limit ($P^l_{j,z}$ or $P^u_{j,z}$);

2.4 To ensure that the relation (9) is respected, the equality constraints handle mechanism (ECHM) similar with the one proposed by [17] is used;

2.5 For each generating unit j, the velocities from [$v_{j,min}$, $v_{j,max}$] interval are randomly initialized. The minimum ($v_{j,min}$) and maximum ($v_{j,max}$) limits of each generating unit's velocity are given by:

$$v_{j,max} = \gamma(P_{j,max} - P_{j,min}) \text{and} v_{j,min} = - v_{j,max} \quad (28)$$

In this paper the γ factor was considered equal to 1/8 (γ =1/8).

Step 3: Initialization of $Pbest^0$ and $Gbest^0$. For k=0 iteration $Pbest^0_i=P^0_i$, i=1,2,…,N. The best value returned by the objective function (F^0) of all solutions $Pbest^0_i$ determines $Gbest^0$: $Gbest^0=Pbest^0_g$, where $g=\arg\min(F(Pbest_i^0))$, i=1,2,…,N.

Step 4: Update the particles velocity. The particles velocity (V_i^{k+1}) for the next iteration (k+1) is given by relation (14). For any dimension j the v_{ij}^{k+1} velocity must find itself in the [$v_{j,min}$, $v_{j,max}$] interval:

$$v_{ij}^{k+1} = \max(v_{j,min}, \min(v_{j,max}, v_{ij}^{k+1})) \quad (29)$$

Step 5: Updates the particles position. The particles position for the next iteration (P_{ij}^{k+1}) is updated based on relation (13). It is checked if the particles' position satisfies the relation (7):

$$P_{ij}^{k+1} = \max(PO_{j,min}, \min(PO_{j,max}, P_{ij}^{k+1})) \quad (30)$$

Step 6: Obtaining an improved feasible solution. A feasible solution can be obtained using the equality constraints handling mechanism proposed by [17]. In order to obtain an improved feasible solution repeat Step 6 for a previously set number of times (S_{preset}) for the last iterations (k_{preset}), $k > k_{preset}$.

Step 7: Evaluate the objective function: The objective function F is assessed for each particle i and for each iteration k using relation (3). Also, $Pbest_i$ and Gbest are updated using relations (25) and (26).

Step 8: Launch the CG(P) local search: The solution resulted in Step 6 is considered the starting solution when launching the CG(P) local search. The actual way of including the CG(P) local optimizer within PSO is described in paragraph 3.2.2 by implementing the HPSO-RU or HPSO-RC variant.

Step 9**:** *Stopping the process.* The criterion used in the paper to stop the computation process is given by reaching the maximum number (k_{max}) of iterations set. When the criterion is met then the computation process stops with *Gbest* as the best solution. Otherwise the process is resumed from Step 4.

The flow chart showing the HPSO-RU and HPSO-RC variants of the HPSO algorithm is shown in Figure 1.

4. SIMULATION RESULTS AND COMPARISON

The proposed method's efficiency is shown through the study of three different test systems consisting of 6, 13 and 38 generating units. All case studies were implemented in MathCAD, on a personal computer having a 2.40 GHz processor and 512 MB of RAM. The analyzed test systems are: Test system 1: 6-unit system, with consideration of the power losses; Test system 2: 13-unit system with valve point effects, without power losses; Test system 3: 38-unit system, without power losses.

The solution's quality is evaluated through 100 trials. For each trial the values of the following items are kept: best total fuel cost F (B), average total fuel cost F (A), worst total fuel cost F (W) and standard deviation (SD). For each system studied, used the parameters (c_1, c_2, N, k_{max}, S_{preset}, k_{preset}) were determined by performing experimental trials. The values k_{preset} are fixed in the [$2{\cdot}k_{max}/3$, $4{\cdot}k_{max}/5$] range. For the systems studied, the parameters used in the HPSO algorithm were set to the values presented in Table 1.

Also, for each case the saved total cost is shown, being calculated as the difference between the average total cost F obtained using method M and the average total cost F obtained using HPSO-RC (Cost saving = (Average cost FM - Average cost FHPSO-RC×8760)). If there is no data on the Average cost F, then the Best cost F item is used (Cost saving = (Best cost FM - Best cost FHPSO-RC×8760)).

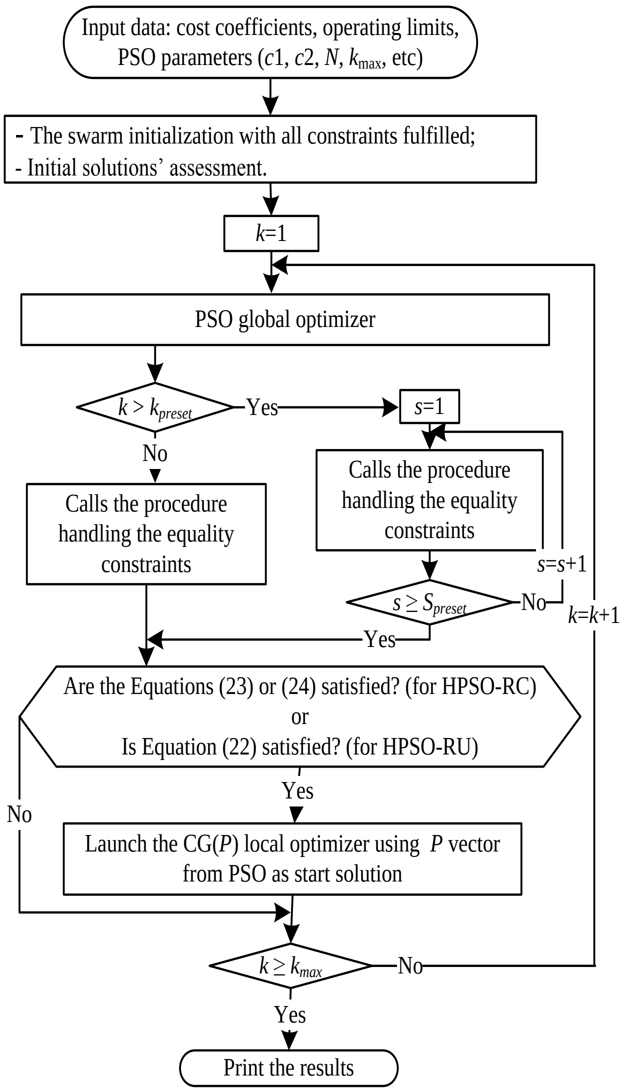

Fig. 1. Flow chart with the HPSO-RC and HPSO-RU approach

Table 1. The values of the parameters used in HPSO algorithm, for the tested systems

System	c_1	c_2	w_{min}	w_{max}	N	k_{max}	S_{preset}	k_{preset}	P_c	α	β
6-unit	1.5	2.5	0.3	0.9	20	200	20	150	0.009	1	1.2
13-unit	1.5	1.5	0.3	0.9	250	600	20	400	0.012	1.4	1.7
38-unit	1.5	1.5	0.3	0.9	15	300	10	240	0.012	1.2	1.3

4.1. Test system 1: 6-unit with losses

A six-unit test system that takes into consideration the transmission losses, ramp rate limits and prohibited operating zones of the units is studied in solving the ED problem with the proposed method. The tested system data related to the cost coefficients (a, b, c), power operating limits, ramp-rate limits, prohibited operating zones of the units, and also the loss coefficient B are taken from [34], and corrected after [35] regarding the B00 factor. The cost characteristics Fj(Pj), j=1,...,6 of the six generating units are increasing functions. The load demand is PD = 1263 MW.

Solution's quality and convergence. The best solution obtained using the proposed HPSO method (applied in the HPSO-RU and HPSO-RC variants) and the CG method, is shown in Table 2. In Table 2 is noticed that the tolerance for satisfying the power balance is very small ($TOL_{\text{HPSO-RC}}$=-0.5×10^{-10} MW) and the solution's stability is very good ($SD_{\text{HPSO-RC}}$=5.0456×10^{-9}\$/h).

Table 2. The best solution obtained using CG, HPSO-RU and HPSO-RC methods (6 units, P_D = 1263MW, 100 trials)

Output	Method CG	Method HPSO-RU	Method HPSO-RC
P_1 (MW)	447.5038599864	447.5038616562	447.5036991964
P_2 (MW)	173.3181636760	173.3184203342	173.3182615588
P_3 (MW)	263.4629619995	263.4631345189	263.4628678097
P_4 (MW)	139.0652980908	139.0651676873	139.0651245208
P_5 (MW)	165.4732174094	165.4735178788	165.4733230366
P_6 (MW)	87.1347391141	87.1341448368	87.1349671158
P_G (MW)	1275.9582402762	1275.9582469122	1275.9582432381
P_L (MW)	12.9582402762	12.95824691223	12.95824323815
P_D (MW)	1263	1263	1263
TOL_M (MW)	5.8×10^{-14}	-3.0×10^{-11}	-0.5×10^{-10}
Best cost F ($/h)	15449.8995248664	15449.8995248703	15449.8995248657
Average cost F ($/h)	15512.2461518378	15449.8997328390	15449.8995248754
Worst cost F ($/h)	15644.2923306296	15449.9021141893	15449.8995248855
SD ($/h)	49.0625	$3.3995\cdot10^{-4}$	$5.0456\cdot10^{-9}$

M = {CG, HPSO-RU, HPSO-RC}.

The HPSO-RC method's convergence characteristic is displayed in Figure 2, for five independent simulations, starting from different random points. It can be noticed that after approximately 125 iterations the optimization process stabilizes itself.

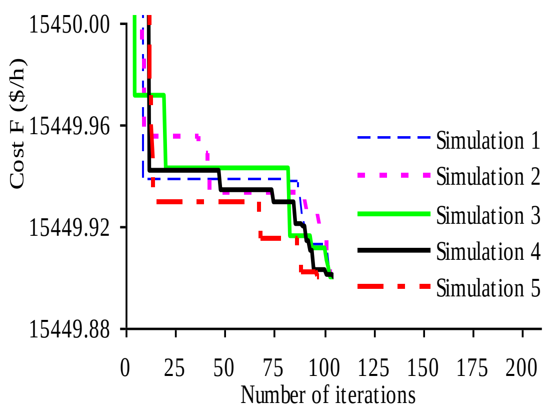

Fig. 2. The convergence characteristic for the HPSO-RC method, 6-units

Robustness. In order to study the HPSO-RU and HPSO-RC methods' robustness, 100 independent trials were performed. The best cost *F*, obtained for each trial, is displayed in the graphic from Figure 3.

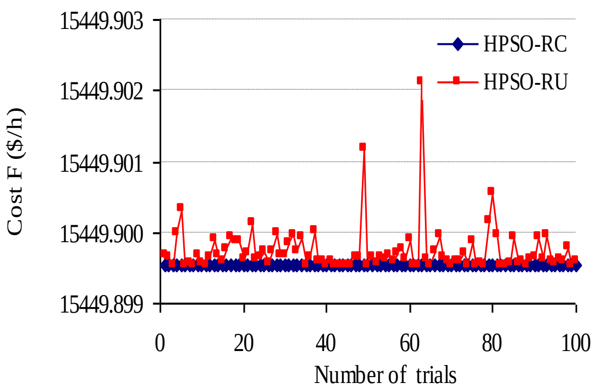

Fig. 3. Best total cost *F* obtained with HPSO-RU and HPSO-RC, 6 units (100 trials)

Following the values for the B, A, W and SD items, as also the cost's variation *F*, Figure 3 shows a very good stability for the obtained solutions using HPSP-RC method, as well as HPSO-RU. Still, the HPSO-RC method presents a better stability than HPSO-RU, because the standard deviation is smaller.

Comparing HPSO-RC with other methods: Table 3 shows a comparison between the results reached with the HPSO-RC method and the results obtained with other eleven methods relatively recently published in literature. The set of methods presented in Table 3 for the system with 6 unit is M={EHM(Effortless Hybrid Method), PSO(Particle Swarm Optimization), HHS(Hybrid swarm intelligence-based HS algorithm), MPSO(Modified PSO), DSPSO-TSA(Distributed Sobol PSO with Tabu Search Algorithm), MSFL (Modified Shuffled Frog Leaping), DHS(Differential Harmony Search), IABC(Incremental Artificial Bee Colony), IABC-LS (IABC-with Local Search), CQGSO(Continuous Quick Group Search Optimizer)}.

Following the results from Table 3, Section 3, it is noticed that $TOL_{HPSO\text{-}RC} > TOL_M$ (or $Dif\_TOL>0$) for each method *M*. This means that the suggested HPSO-RC method was applied in more detrimental conditions than the methods specified in Table 3.

Table 3. Comparing the results obtained with different optimization techniques (6 units, $P_D = 1263$ MW)

Method	EHM [36]	PSO [34]	HHS [12]	MPSO [37]	DSPSO-TSA [24]
1. Result described according to the indicated references					
P_1 (MW)	449.1546	447.497	449.9094	447.506	439.2935
P_2 (MW)	173.0613	173.3221	172.7347	176.178	187.7876
P_3 (MW)	266.0092	263.4745	262.9643	261.95	261.026
P_4 (MW)	127.1203	139.0594	136.03	137.771	129.4973
P_5 (MW)	174.2603	165.4761	166.967	165.448	171.7101
P_6 (MW)	85.8777	87.128	86.8778	86.613	86.1648
Best cost F_M (\$/h)	15441.5974	15450.0000	15442.8313	15443.29	15441.5700
Average F_M (\$/h)	-	15454.0000	15446.7142	15449.14	15443.8400
Worst cost F_M (\$/h)	-	15492.0000	-	15486.82	15446.2200
SD_M (\$/h)	-	-	1.8275	-	0.3700
Number of trials	-	50	-	50	100
CPU time (s)	0.32	14.89	0.94	1.46	1.07
2. The computation of the power loss (P_L), generated power(P_G) and TOL_M tolerance based on the best solution described in the indicated references					
P_G (MW)	1275.4834	1275.9571	1275.4832	1275.4660	1275.4793
P_L (MW)	13.25804122	12.95837787	13.01669093	12.96407447	13.1481124
TOL_M (MW)	-0.7746412239	-0.0012778743	-0.5334909305	-0.49807446659	-0.6688123993
3. The best results obtained with the proposed method HPSO-RC, imposing the tolerance $TOL_{HPSO\text{-}RC} \geq TOL_M$ (100 trials)					
P_1 (MW)	447.34306783	447.50262437	447.39321103	447.40044769	447.36503875
P_2 (MW)	173.19917082	173.32241616	173.23637580	173.24167404	173.21556357
P_3 (MW)	263.33819757	263.44869535	263.37678657	263.38259998	263.35516814
P_4 (MW)	138.93380687	139.06504138	138.97471326	138.98079521	138.95186600
P_5 (MW)	165.34799012	165.47534315	165.38699826	165.39279326	165.36516463
P_6 (MW)	87.00693997	87.14278020	87.04672814	87.05257863	87.02417026
P_G (MW)	1275.16917318	1275.95690061	1275.41481306	1275.45088881	1275.27697135
P_L (MW)	12.94381440	12.95817828	12.94830399	12.94896327	12.94578374
$TOL_{HPSO\text{-}RC}$ (MW)	-0.774641220	-0.001277670	-0.533490930	-0.49807446	-0.66881239
Dif_TOL* (MW)	3.9×10^{-9}	2.1×10^{-7}	0.5×10^{-9}	6.6×10^{-9}	9.3×10^{-9}
Best cost *F* (\$/h)	**15439.41090815**	**15449.8822264**	**15442.6758751**	**15443.1553987**	**15440.8437168**
Average cost *F* (\$/h)	15439.41090825	15449.8822272	15442.6758751	15443.1553987	15440.8437168
Worst cost *F* (\$/h)	15439.41090828	15449.8822273	15442.6758751	15443.1553987	15440.8437168
SD (\$/h)	1.1830×10^{-8}	8.5220×10^{-8}	6.8110×10^{-9}	5.1580×10^{-9}	6.4750×10^{-9}
Cost saving (\$/yr)	19153.6686	36071.6893	35375.7258	52425.1073	26247.4406

* $Dif\_TOL = TOL_{HPSO\text{-}RC} - TOL_M$; TOL_M, $TOL_{HPSO\text{-}RC}$ - the tolerance for a method *M*, respectively method HPSO-RC; M={EHM, PSO, HHS, MPSO, DSPSO-TSA}; "-"data not available.

Table 3 (continuation).

Method	MSFL [38]	DHS [39]	IABC, IABC-LS [16]	CQGSO [40]
1. Result described according to the indicated references				
P_1 (MW)	445.0140	447.5285	451.5204	263.9079 (447.076669*)
P_2 (MW)	175.5156	173.2791	172.1750	173.2418
P_3 (MW)	264.2614	263.4772	258.4186	263.9079
P_4 (MW)	137.3012	139.0291	140.6441	139.0529
P_5 (MW)	162.7899	165.4864	162.0797	165.6013
P_6 (MW)	90.4992	87.1587	90.3415	86.5357
Best cost F_M ($/h)	15442.5911	15449.8996	15441.108	15442.6608
Average F_M ($/h)	15447.60	15449.9264	15441.108	15442.6630
Worst cost F_M ($/h)	15460.29	15449.9884	15441.108	15442.6614
SD_M ($/h)	4.07	$2.04 \cdot 10^{-2}$	-	-
Number of trials	50	-	30	50
CPU time (s)	-	0.01	0.018	8.22
2. The computation of the power loss (P_L), generated power(P_G) and TOL_M tolerance based on the best solution described in the indicated references				
P_G (MW)	1275.3813	1275.959	1275.1793	1275.416269
P_L (MW)	12.943445394	12.95903723189	12.87289931601	12.95134380327
TOL_M (MW)	-0.562145394	-0.00003723189	-0.69359931601	-0.53507480327
3. The best results obtained with the proposed method HPSO-RC, imposing the tolerance $TOL_{HPSO\text{-}RC} \geq TOL_M$ (100 trials)				
P_1 (MW)	447.3874051568	447.5040042922	447.3596969738	447.3918605427
P_2 (MW)	173.2319625357	173.3184968419	173.2117812789	173.2364257675
P_3 (MW)	263.3721930956	263.4632164590	263.3519570314	263.3770681267
P_4 (MW)	138.9699553911	139.0649861658	138.9468373698	138.9746869190
P_5 (MW)	165.3823068826	165.4732061267	165.3612138951	165.3866767256
P_6 (MW)	87.0418002652	87.1342990084	87.0202523905	87.0464769449
P_G (MW)	1275.3856233270	1275.9582088940	1275.2517389395	1275.4131950264
P_L (MW)	12.9477687210	12.9582461259	12.9453382555	12.9482698296
$TOL_{HPSO\text{-}RC}$ (MW)	-0.5621453940	-0.0000372319	-0.6935993160	-0.5350748032
*Dif_TOL*** (MW)	6.6×10^{-13}	3.2×10^{-12}	1.0×10^{-11}	7.0×10^{-11}
Best cost F ($/h)	15442.2879088525	15449.8990207023	15440.5081254302	15442.6544301909
Average cost F ($/h)	15442.2879088598	15449.8990207101	15440.5081254381	15442.6544301986
Worst cost F ($/h)	15442.2879088721	15449.8990207191	15440.5081254493	15442.6544302094
SD ($/h)	4.7229×10^{-9}	4.7042×10^{-9}	4.7230×10^{-9}	4.5385×10^{-9}
Cost saving ($/yr)*	4621.52	23.82	521.89	7.46

For unit 1 power P_1 was recalculated using the solution presented in [40]; M={MSFL, DHS, IABC, IABC-LS, CQGSO}

In relation to item B (Best cost *F*) the HPSO-RC method returns better values than the *M* methods presented in Table 3. In all eleven cases HPSO-RC returns very close values for the B, A, W items and SD has very small values (Table 3, section “3. ”). This shows a very good stability of the obtained solutions with this method. Also, the W item obtained with this HPSO-RC method ($W_{HPSO\text{-}RC}$) is smaller than the B item obtained with any other method *M* (B_M) from Table 3 ($W_{HPSO\text{-}RC} < B_M$). All these elements show that the HPSO-RC method is superior to those presented in Table 3, being able to find high quality and very stable solutions. Applying in addition a statistic validation test to establish if “the results obtained with the HPSO-RC method are significantly different from the results obtained with a method *M* ” is useless. Table 3 shows that the HPSO-RC method returns better results than other PSO variants (PSO, MPSO, DSPSO-TSA) or others recent optimization techniques (EHM, HHS, MSFL, DHS, IABC).

The distribution of CG(P) launches. In the HPSO-RC variant the number of launches for the *CG*(*P*) local optimizer, for a particle *i*, is $N_i \in \{2 \text{ or } 3\}$, i=1,2,..20, and for HPSO-RU is $N_i \in \{0,1,2,3 \text{ or } 4\}$ (or even more).

The launches distribution for each particle *i*, in HPSO-RC case respectively HPSO-RU, is shown in Figure 4 (for one simulation).

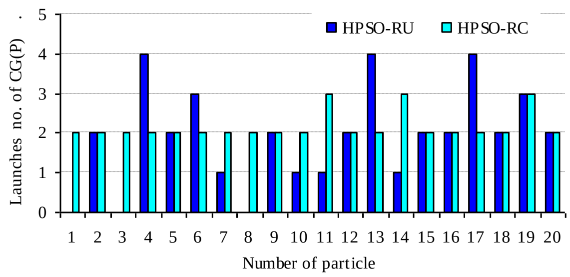

Fig. 4. The *CG*(*P*) launches distribution for the particles, for HPSO-RC and HPSO-RU variants, 6-units

4.2. Test system 2: 13-unit with valve point effects, without losses

The HPSO method (in the HPSO-RC and HPSO-RU variants) and the CG method are applied to solve the ED problem for a 13 thermal generating units system with valve-point loading effects, which neglect the transmission line losses (P_L=0). The system's data (*a*, *b*, *c*, *e*, *f* coefficients and limits of generated powers) are taken from [8]. Solving the ED problem is done taking into consideration the total power demand set to P_D=2520 MW.

Solution's quality and convergence. The best solutions obtained with HPSO-RC, HPSO-RU and CG are shown in Table 4. Figure 5 shows the Cost *F* objective function's variation throughout the optimizing process, using HPSO-RC and HPSO-RU.

Table 4. Best solutions obtained with HPSO-RC, HPSO-RU and CG, (13 units, P_D = 2520 MW, 100 trials).

Output (MW)	CG	HPSO-RU	HPSO-RC
P_1	631.6450735432	628.31853071645	628.31853071788
P_2	296.9354094981	299.19930034162	299.19930034061
P_3	355.0823922167	299.19930034035	299.19930034158
P_4	173.4542527145	159.73310011369	159.73310011288
P_5	174.5584794134	159.73310011241	159.73310011193
P_6	116.3270570710	159.73310011284	159.73310011317
P_7	119.1744178533	159.73310011346	159.73310011416
P_8	157.1229982003	159.73310011308	159.73310011346
P_9	161.754769301	159.73310011170	159.73310011261
P_{10}	119.1494499975	77.39991254180	77.39991253868
P_{11}	86.5511186421	77.39991253647	77.39991254142
P_{12}	67.6778601017	87.68453031849	87.68453030058
P_{13}	60.5667214468	92.39991252762	92.39991254103
TOL_M	$-4.001\cdot10^{-10}$	$-2.000\cdot10^{-11}$	$-1.046\cdot10^{-11}$
Best *F* ($/h)	24986.6951888434	24169.9176968388	24169.9176968257

M={CG, HPSO-RU, HPSO-RC}.

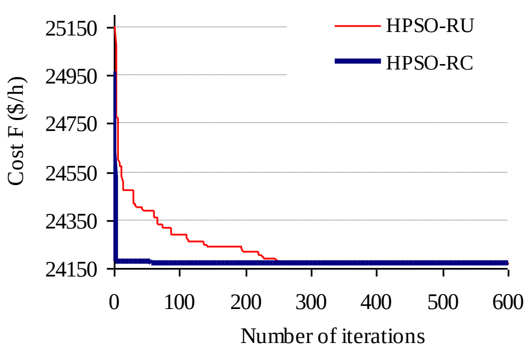

Fig. 5. Cost *F* variation with the number of iterations, for HPSO-RC and HPSO-RU, 13-units

Robustness. Figure 6 shows the best cost obtained with HPSO-RC and HPSO-RU, for 100 trials. The HPSO-RC hybrid has a very good stability, while HPSO-RU displays large variations of the best cost *F*. In order to clearly see the best cost *F* variation (for HPSO-RC) an extra display axis (axis right) is used.

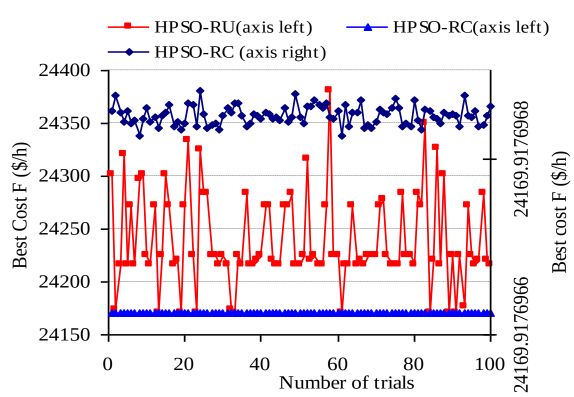

Fig. 6. The best total cost *F* obtained with HPSO-RU and HSO-RC, 13-units (100 trials)

Comparing HPSO-RC with other methods. The results obtained with HPSO-RC, HPSO-RU and CG were compared with the results from other reported published algorithms, like: IGA_MU (Improved Genetic Algorithm with Multiplier Updating), GA(Genetic Algorithm), TS(Tabu Search), DE(Differential Evolution), PSO, FAPSO(Fuzzy Adaptive PSO), FAPSO-NM(FAPSO-with Nelder–Mead), HS(Harmony Search), HHS(Hybrid HS), ACO(Ant Colony Optimization), PSO-SQP, EP-SQP(Evolutionary Programming-SQP), HGA(Hybrid GA), UHGA(Uniform HGA), TSA, DSPSO-TSA, CPSO(Chaotic PSO), CPSO-SQP, CASO(Chaotic Ant Swarm Optimization), FCASO-SQP(Fuzzy adaptive CASO), SDE(Shuffled Differential Evolution), AGA(Atavistic GA), SA-PSO(Simulated Annealing-PSO). According with items (B, A, W and SD) it is noticed that the HPSO-RC algorithm behavior is better than most of the algorithms shown in Table 5, and is just as good as HHS [13], DE [10] and SDE [43] algorithms

.

Table 5. HPSO-RC and HPSO-RU results comparison with other published algorithms (13 units, $P_D = 2520$MW)

Methods	Best *F* ($/h)	Average *F* ($/h)	Worst *F* ($/h)	SD ($/h)	Cost saving ($/year)
IGA_MU [4]	24169.9790	24385.4113	24754.1450	-	1887724.0
GA [6]	24183.31	24225.24	24293.37	40.10	484623.4
TS [6]	24178.65	24201.46	24305.81	29.50	276310.6
PSO [6]	24171.64	24194.01	24242.57	20.77	211048.6
DE [10]	24169.9177	-	24169.9180	$4.45 \cdot 10^{-5}$	0.0
PSO [11]	24262.73	24271.9231	24277.81	-	893567.3
FAPSO-NM [11]	24169.92	24170.0017	24170.5	-	735.9
FAPSO [11]	24170.93	24173.0069	24176.4	-	27061.4
HS [13]	24208.7	24323.2	24503.7	-	1342753.0
HHS [13]	24169.9	24169.9	24169.9	-	-
ACO [14]	24174.39	24211.09	24243.90	21.10	360669.4
PSO-SQP [21]	24261.05	-	-	-	798319.0
EP-SQP [21]	24266.44	-	-	-	845535.4
HGA [22]	24169.92	-	-	-	20.2
UHGA [23]	24172.25	-	-	-	20431.0
DSPSO-TSA [24]	24169.923	24173.137	24230.803	7.72	28201.1
TSA [24]	24171.211	24184.055	24392.203	41	123842.8
GA [24]	24170.804	24188.394	24567.974	59.53	161852.4
PSO [24]	24170.167	24184.849	24377.890	38.86	130798.2
CPSO-SQP [41]	24190.97	-	-	-	184418.2
CPSO [41]	24211.56	-	-	-	364786.6
CASO [42]	24212.93	-	-	-	376787.8
FCASO-SQP [42]	24190.63	-	-	-	181439.8
SDE [43]	24169.92	-	-	-	20.2
CG	24986.6951888434	25595.21564211	25914.43223867	190.01	12485610.0
HPSO-RU	24169.9176968388	24238.82964995	24381.42822641	43.66	603668.7
HPSO-RC	24169.9176968257	24169.91769684	24169.91769687	$1.07 \cdot 10^{-8}$	-

“-”data was not available

It is needed to be mentioned that the HHS, DE and SDE algorithms either have a standard deviation (SD) higher than $1.07 \cdot 10^{-8}$ \$/h (obtained by HPSO-RC), or the SD value was not specified by the authors. Also, for the case of the second system that has non-smooth cost characteristics (due to the valve-point), the suggested HPSO-RC method returns high quality and very stable solutions.

4.3. Test system 3: 38-unit, without losses

The HPSO method (in the HPSO-RC and HPSO-RU variants) and the CG method are applied to solve the ED problem for a 38 - units system, which neglect the transmission line losses ($P_L = 0$). The system's data (*a*, *b*, *c* coefficients and limits of generated powers) are taken from [44]. Solving the ED problem is done taking into consideration the total power demand set to P_D=6000 MW.

Solution quality and convergence: The best solutions obtained with HPSO-RC, HPSO-RU are shown in Table 6. Figure 7 shows the Cost F objective function's variation throughout the optimizing process, using HPSO-RC and HPSO-RU.

Robustness:. Figure 8 shows the best cost obtained with HPSO-RC and HPSO-RU, for 100 trials. The HPSO-RC and HPSO-RU hybrids have a very good stability of the best cost *F*.

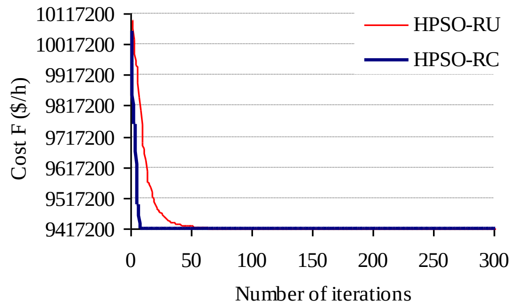

Fig. 7. Cost *F* variation with the number of iterations, for HPSO-RC and HPSO-RU, 38-units

Table 6. Best solutions obtained with HPSO-RU and HPSO-RC (38 units, P_D = 6000 MW, 100 trials)

Output (MW)	HPSO-RU	HPSO-RC	Output (MW)	HPSO-RU	HPSO-RC
P_1	426.60710722392	426.6071875130	P_{20}	271.99999999997	272.0000000000
P_2	426.60629288079	426.6064711888	P_{21}	271.99999999987	272.0000000000
P_3	429.66384437984	429.6625414693	P_{22}	260.00000000000	260.0000000000
P_4	429.66273581737	429.6636893856	P_{23}	130.64783039764	130.6482892479
P_5	429.66393496646	429.6621565653	P_{24}	10.00000000000	10.0000000000
P_6	429.66133628059	429.6626430260	P_{25}	113.30557364301	113.3049018997
P_7	429.66254855354	429.6631893912	P_{26}	88.06658658316	88.0672068581
P_8	429.66487700758	429.6611685814	P_{27}	37.50484809206	37.5050725873
P_9	114.00000000001	114.0000000000	P_{28}	20.00000000000	20.0000000000
P_{10}	114.00000000000	114.0000000000	P_{29}	20.00000000000	20.0000000000
P_{11}	119.76839216473	119.7697318471	P_{30}	20.00000000000	20.0000000000
P_{12}	127.07258266083	127.0731107675	P_{31}	19.99999999999	20.0000000000
P_{13}	110.00000000000	110.0000000000	P_{32}	20.00000000000	20.0000000000
P_{14}	89.99999999999	90.0000000000	P_{33}	24.99999999999	25.0000000000
P_{15}	82.00000000000	82.0000000000	P_{34}	17.99999999999	18.0000000000
P_{16}	119.99999999997	120.0000000000	P_{35}	8.00000000000	8.0000000000
P_{17}	159.59797697212	159.5979562405	P_{36}	25.00000000000	25.0000000000
P_{18}	65.00000000007	65.0000000000	P_{37}	21.78168428913	21.7823343641
P_{19}	65.00000000009	65.0000000000	P_{38}	21.06184808142	21.0623490616
			Best *F* (\$/h)	9417235.78639226	9417235.78639142
			TOL_M (MW)	$-5.871 \cdot 10^{-9}$	$-5.747 \cdot 10^{-9}$

Comparing HPSO-RC with other methods: The results obtained with HPSO-RC, HPSO-RU and CG were compared with the results from other reported published algorithms (shown in Table 7), as like: HS variants (HS, HHS [13]), PSO variants (New_PSO, PSO_TVAC (PSO_Time Varying Acceleration Coefficients), SPSO (Simple PSO), PSO_Crazy [19]), biogeography-based optimization methods (BBO, DE/BBO [20]) or Multi-strategy Ensemble Biogeography - Based Optimization with migration operator (MsEBBO/mig), mutation operator (MsEBBO/mut) and sinusoidal migration model (MsEBBO/sin) [25]. Also, it is possible to count the number of trials NT (out of a total of 100 trials) placed

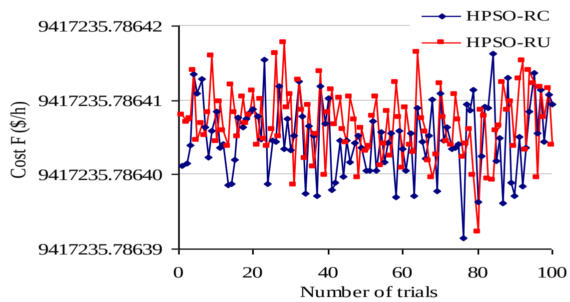

Fig. 8. The best total cost *F* obtained with HPSO-RU and HSO-RC, 38-units (100 trials).

in the "range of cost", when *N* and k_{max} are fixed at the values of column 4, Table 8.

According to items B, A, W and SD it is noticed that the HPSO-RC algorithm's behavior is better than the algorithms shown in Table 7 and as good as MsEBBO algorithm [25]. Also, in Table 7 it is noticed that tolerance $TOL_{HPSO-RC}$ is smaller for HPSO-RC, compared with most methods ($|TOL_{HPSO-RC}|<|TOL_M|$, M={New_PSO, PSO_TVAC, SPSO, PSO_Crazy, HS, HHS, MsEBBO/mig, MsEBBO/mut, MsEBBO/sin}).

Table 7. HPSO-RC and HPSO-RU results comparison with other published algorithms (38 units, P_D = 6000 MW)

Methods	Best *F* ($/h)	Average *F* ($/h)	Worst *F* ($/h)	SD ($/h)	Cost saving ($/year)
HS[13]	9419960	9421056	9427466	-	33,465,071.1
HHS[13]	9417325	9417336	9417466	-	877,871.1
New_PSO [19]	9516448.312	-	-	-	869,101,724.3
PSO_TVAC[19]	9500448.307	-	-	-	728,941,680.5
SPSO [19]	9543984.777	-	-	-	1,110,321,158.1
PSO_Crazy [19]	9520024.601	-	-	-	900,430,016.3
BBO[20]	9417633.637644372	-	-	-	3,485,176.9
DE/BBO[20]	9417235.786391673	-	-	-	≈0
MsEBBO/mig [25]	9417235.7758	9417238.6310	9417267.1572	7.6573	24,918.6
MsEBBO/mut [25]	9417236.4186	9417238.9941	9417243.4263	1.4829	28,099.4
MsEBBO/sin [25]	9417235.7772	9417248.6242	9417278.7014	11.0169	112,459.1
MsEBBO [25]	9417235.7758*	9417235.7781	9417235.7790	0.0032	-72.7
CG	9417266.43845436	9497066.65611085	9820791.14660850	90072	699,318,418.6
HPSO-RU	9417235.78639226	9417235.78640690	9417235.78641767	$4.59\cdot10^{-6}$	≈0
HPSO-RC	9417235.78639142	9417235.78640492	9417235.78641629	$4.73\cdot10^{-6}$	-

The solution obtained by MsEBBO [25] determines a cost of B_{MsEBBO}=9417235.78639251 $/h; "-"data not available;
* TOL_M was calculated based on the solution presented in the indicated references

4.4. Computational efficiency

The computational efficiency is measured through the CPU time. The CPU time is influenced by several factors like: convergence tolerance (controls the number of iterations in applying the conjugate gradient method), constraint tolerance (controls the way in which the inequality and equality constraints are met), as also the personal computer's configuration. The CPU time is displayed in the shape of a range of values, being obtained with the HPSO-RC method. The lower limit corresponds to *N* and k_{max} values from column 4, Table 8 and the upper limit corresponds to *N* and k_{max} values presented in column 5 of Table 8.

For the analyzed systems it is possible to establish a limited "range of cost", where most cost F values can be located (out of a total of 100 trials). "Range of cost" can be determined using the relation (31):

$$8760\times (F[P] - F[Gbest]) \le \Delta F_{admitted} \quad (31)$$

Where $\Delta F_{admitted}$ is the admitted fuel cost increase for the solution [*P*] as opposed to solution [*Gbest*](it is assumed that $\Delta F_{admitted}$ is a neglected cost at the analyzed system level); [*Gbest*] is the best solution obtained with the HPSO-RC method, and *F*([*Gbest*]) is the best fuel cost.

Also, it is possible to count the number of trials NT (out of a total of 100 trials) placed in the "range of cost", when *N* and k_{max} are fixed at the values of column 4, Table 8.

Table 8. CPU time for the analyzed systems (100 trials)

Analysis System	$\Delta F_{admitted}$ ($/year)	Range of cost ($/hour)	Lower limit of CPU time (s)	Upper limit of CPU time (s)	NT
6-unit system	250	(15449.8995248657 - 15449.9280636785)	0.13 (N = 4, k_{max} = 10)	12.33 (N = 20, k_{max} = 200)	100 of 100
13-unit system	500	(24169.9176968257 - 24169.9747744513)	116.17 (N =140, k_{max} =300)	373.28 (N = 250, k_{max} = 600)	92 of 100
38-unit system	750	(9417235.78639142 - 9417235.87200786)	9.11 (N = 12, k_{max} = 100)	26.34 (N = 15, k_{max} = 300)	100 of 100

From Table 8, for the systems with 6-units and 38-units, the CPU time is good, comparable with the one obtained through other techniques suggested by other authors and shown in Table 3, respectively Table 7. It is important to be mentioned that in both cases all 100 trials were placed in the specified cost range, $\Delta F_{admitted}$ being assumed at 250 $/year, respectively 750 $/year. In case of the 13 units system, for the cost F values to be included in the specified “range of cost”, the computation time is relatively high.

The assumed values for $\Delta F_{admitted}$ have allowed decreasing the number of particles and iterations to the following values: *N*=4 and k_{max}=10 (for 6-units), *N*=140 and k_{max}=300 (for 13-units) and *N*=12 and k_{max}=100 (for 38-units). The higher upper limits of the CPU time for the three systems (12.33s, 373.28s and 26.34s) are due to the fact that very small tolerance ($TOL_{HPSO\text{-}RC}<10^{-9}$MW) were imposed to meet the inequality and equality constraints. Also, a greater number of particles (*N*) and iterations (k_{max}) were imposed, in order to ensure the solution’s stability (*N* and k_{max} are shown for each analyzed system).

4.5. Comparing the HPSO-RC and HPSO-RU variants

Comparing the HPSO-RC and HPSO-RU variants of the HPSO hybrid is done using the “t-test” statistic test, taking into consideration the significance level. The null hypothesis H_0 is: between the two variants results there are no statistically significant differences, and the alternate hypothesis (H_1) is the opposite of the H_0 hypothesis. The test was performed for all three studied systems. The outcome of applying the test is shown in Table 9. For all systems comparisons resulted in statistically significant differences, HPSO-RC being superior to HPSO-RU. Also, it must be mentioned that for the 13 units systems, the saved cost in case the HPSO-RC variant is important comparing to HPSO-RU, being $6.03 \cdot 10^5$ $/year.

Table 9. Testing the HPSO-RC and HPSO-RU variants

Method	Test System	t_{value}	Significant
HPSO-RC vs. HPSO-RU	6 units	6.117*	Yes
	13 units	15.783*	Yes
	38 units	3.011*	Yes

$p < .05$; t_{value} – statistic value for “t-test”.

4.6. The last iterations strengthening effect

As described in paragraph 3.2.3. (step 6), strengthening the last *k* iterations refers to applying for a preset number of times (S_{preset}) the equality constraints handling mechanism. The strengthening effect of the last iterations is studied for the three test systems. Table 10 shows the A and SD items (during the 100 trials) for four values of the S_{preset} parameter (S_{preset} ={0, 10, 20, 25}). Also, Table 10 shows the results after applying the “t test” statistic test between S_{preset} = 25 and the three cases in which S_{preset}={0, 10 and 20}.

Table 10. The S_{preset} parameter's effect in HPSO-RC (for 100 trials)

Test system	S_{preset}	Average F (\$/h)	SD (\$/h)	t_{value}	Significant
6-units	0	15449.8997804823	$3.5771 \cdot 10^{-4}$	7.14*	Yes (0 vs.25[a])
	10	15449.8995248769	$5.0623 \cdot 10^{-9}$	2.27*	Yes (10 vs.25)
	20	15449.8995248754	$5.0456 \cdot 10^{-9}$	0.25	No (20 vs. 25)
	25	15449.8995248753	$4.7644 \cdot 10^{-9}$	-	-
13-units	0	24169.9243927141	$2.8743 \cdot 10^{-3}$	23.29*	Yes (0 vs.25[a])
	10	24169.9176968806	$1.0604 \cdot 10^{-7}$	3.08*	Yes (10 vs.25)
	20	24169.9176968491	$1.0700 \cdot 10^{-8}$	1.01	No (20 vs. 25)
	25	24169.9176968476	$9.7710 \cdot 10^{-9}$	-	-
38-units	0	9417235.78640492	$4.7331 \cdot 10^{-6}$	5.75*	Yes (0 vs.25[a])
	10	9417235.78640214	$3.6796 \cdot 10^{-6}$	1.39	No (10 vs. 25)
	20	9417235.78640202	$3.8951 \cdot 10^{-6}$	1.12	No (20 vs.25)
	25	9417235.78640140	$3.8665 \cdot 10^{-6}$	-	-

[a] For example, 0 vs. 25 means that the statistic case $S_{preset} = 0$ is compared with $S_{preset} = 25$; * p < .05.

The significance level is 5%. The test indicates significant statistic differences between 0 vs. 25 (for all analyzed systems) and 10 vs. 25 (for 6-units and 13-units) and not significant statistic differences between 20 vs. 25 (for all analyzed systems). This means that for $S_{preset} \geq 20$ (for 6-units and 13-units) and $S_{preset} \geq 10$ (for 38-units) the solutions obtained don't significantly statistically differ between them. The values of S_{preset} parameter are presented in Table 1. From the mathematical point of view there is a significant difference between the different cases (especially between 0 vs. 25) for all analyzed systems. Also, an improvement for the B, A, W and SD items is noticed once the S_{preset} value is increased.

5. CONCLUSION

The separate use of the PSO and CG methods in solving the ED problem has the disadvantage that the obtained solutions are not stable. Additionally, taking into consideration the valve-point effects gives a premature convergence towards a local minimum. In this paper, to solve the ED problem with and without valve-point effects, a hybrid (HPSO) that successfully combines the PSO technique with the CG method available in MathCAD commercial software product's optimizing tool is used. Using MathCAD's conjugate gradient method is simple and doesn't require the development of a special module for the partial derivatives computation. Integrating the CG local search inside the PSO algorithm can be easily performed with the HPSO-RC and HPSO-RU variants.

The HPSO-RC and HPSO-RU variants have a similar behavior for the systems with smooth cost characteristics (the 6-units and 38-units systems). A slight advantage is shown by HPSO-RC. Still for the system with non-smooth cost characteristics HPSO-RC has the ability to reach better results than HPSO-RU.

For the analyzed systems that have smooth cost characteristics the HPSO-RC hybrid is superior to the methods taken from literature as far as the solution's quality and stability is involved, having a good computational efficiency. For the system with non-smooth characteristics (13-units) the HPSO-RC hybrid is capable of obtaining better or just as good results as other described techniques.

The main feature of the HPSO-RC hybrid is its ability to reach high quality and very stable solutions, having a good computation time for the systems with smooth cost characteristics.

REFERENCES

[1] J. Nanda, L. Hari and M. L. Kothari, "**Economic emission load dispatch with line flow constraints using a classical technique**", *IEE Proceedings Generation, Transmission and Distribution*, Vol. 141, No.1, pp. 1-10, 1994.

[2] L. Papageorgiou and E. Fraga, "**A mixed integer quadratic programming formulation for the economic dispatch of generators with prohibited operating zones**", *Electric Power System Research*, Vol.77, No.10, pp. 1292–1296, 2007.

[3] D. Travers, R. Kaye, "**Dynamic dispatch by constructive dynamic programming**", *IEEE Transaction on Power System*, Vol.13, No.1, pp. 72-78, 1998.

[4] C.L. Chiang, "**Genetic-based algorithm for power economic load dispatch**", *IET Generation, Transmission and Distribution*, Vol.1, No.2, pp. 261–269, 2007.

[5] D.C. Walters, G.B. Sheble, "**Genetic algorithm solution of economic dispatch with valve-point loadings**", *IEEE Transaction on Power System*, Vol.8, No.3, pp. 1325–1331, 1993.

[6] W. Sa-ngiamvibool, S. Pothiya and I. Ngamroo, "**Multiple tabu search algorithm for economic dispatch problem considering valve-point effects**", *Electric Power and Energy Systems*, Vol.33, No.4, pp. 846–854, 2011.

[7] C.T. Sun, C.T. Lin, "**New approach with a Hopfield modeling framework to economic dispatch**", *IEEE Transaction on Power System*, Vol.15, No.2, pp. 541-545, 2000.

[8] N. Sinha, R. Chakrabarti and P.K. Chattopadhyay, "**Evolutionary programming techniques for economic load dispatch**", *IEEE Transactions on Evolutionary Computation*, Vol.7, No.1, pp. 83–94, 2003.

[9] A. Pereira-Neto, C. Unsihuay and O.R. Saavedra, "**Efficient evolutionary strategy optimisation procedure to solve the nonconvex economic dispatch problem with generator constraints**", *IEE Proceedings Generation, Transmission and Distribution*, Vol.152, No.5 pp. 653–660, 2005.

[10] N. Noman, H. Iba, "**Differential evolution for economic load dispatch problems**", *Electric Power System Research*, Vol.78, No.3, pp. 1322–1331, 2008.

[11] T. Niknam, "**A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem**", *Applied Energy*, Vol. 87, No.1 pp. 327-339, 2010.

[12] R.V. Pandi, B.K. Panigrahi, R.C. Bansal, S.Das and A. Mohapatra, "**Economic load dispatch using hybrid swarm intelligence based harmony search algorithm**", *Electric Power Components and Systems*, Vol.39, No.8, pp. 751–767, 2011.

[13] M. Fesanghary, M.M. Ardehali, "**A novel meta-heuristic optimization methodology for solving various types of economic dispatch problem**", *Energy*, Vol.34, No.6, pp. 757-766, 2009.

[14] S. Pothiya, I. Ngamroo and W. Kongprawechnon, "**Ant colony optimisation for economic dispatch problem with non-smooth cost functions**", Electrical Power and Energy Systems, Vol.32, No.5, pp. 478–487, 2010.

[15] J. Cai, X. Ma, L. Li and H. Peng, "**Chaotic particle swarm optimization for economic dispatch considering the generator constraints**", *Energy Conversion and Management*, Vol.48, No.2, pp. 645–653, 2007.

[16] S. Özyön, D. Aydin, "**Incremental artificial bee colony with local search to economic dispatch problem with ramp rate limits and prohibited operating zones**", *Energy Conversion and Management*, Vol. 65, No.1, pp. 397-407, 2013.

[17] J.B. Park, Y.W. Jeong, J.R. Shin and K. Y Lee, "**An improved particle swarm optimization for nonconvex economic dispatch problems**", *IEEE Transaction on Power System*, Vol. 25, No.1,pp. 156–166, 2010.

[18] N. M. Jothi Swaroopan1 , P. Somasundaram, "**A Novel Combined Economic and Emission Dispatch Control by Hybrid Particle Swarm Optimization Technique",** *Majlesi Journal of Electrical Engineering*, Vol.4, No.2, 2010.

[19] K.T. Chaturvedi, M. Pandit and L. Srivastava, "**Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch**", *Electrical Power and Energy Systems*, Vol.31, No.6, pp. 249-257, 2009.

[20] A. Bhattacharya , P.K. Chattopadhyay, "**Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch**", *IEEE Transactions On Power Systems*, Vol.25, No.4, pp. 1955-1964, 2010.

[21] T.A.A. Victoire , A.E. Jeyakumar, "**Hybrid PSO–SQP for economic dispatch with valve-point effect**", *Electric Power System Research*, Vol.71, No.1, pp. 51–59, 2004.

[22] D.K. He, F.L. Wang and Z.Z. Mao, "**A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect**", *Electrical Power and Energy Systems*, Vol. 30, No.1, pp. 31-38, 2008.

[23] D.K. He, F.L. Wang and Z.Z. Mao, "**Hybrid genetic algorithm for economic dispatch with valve-point effect**", *Electric Power System Research*, Vol.78, No.4, pp. 626–633, 2008.

[24] S. Khamsawang , S. Jiriwibhakorn, "**DSPSO-TSA for economic dispatch problem with nonsmooth and noncontinuous cost functions**", *Energy Conversion and Management*, Vol.51, No.4, pp. 365–375, 2010.

[25] G. Xiong, D. Shi and X. Duan, "**Multi-strategy ensemble biogeography-based optimization for economic dispatch problems**", *Applied Energy*, Vol. 111, pp. 801-811, 2013.

[26] A. Qteish , M. Hamdan, "**Hybrid particle swarm and conjugate gradient optimization algorithm**", *Advances in Swarm Intelligence (Lecture Notes in Computer Science)*, Part 1, pp. 582-588, 2010.

[27] S. Chen, T. Mei, M. Luo and X. Yang, "**Identification of nonlinear system based on a new hybrid gradient-based PSO algorithm**", *in Proc. 2007 International Conference on Information Acquisition*, ICIA, Seogwipo-si, pp. 265-268.

[28] A.H. Kayhan, H. Ceylan and M.T. Ayvaz, Gurarslan, G. "**PSOLVER: A new hybrid particle swarm optimization algorithm for solving continuous optimization problems**", *Expert Systems with Applications*, Vol.37, No.10, pp. 6798-6808, 2010.

[29] M.T. Ayvaz, A.H. Kayhan, H. Ceylan and G. Gurarslan, "**Hybridizing the harmony search algorithm with a spreadsheet "solver" for solving continuous engineering optimization problems**", *Engineering Optimization*, Vol.41, No.12, pp. 1119-1144, 2009.

[30] J. Kennedy, R.C. Eberhart, "**Particle swarm optimization**", *in Proc. 1995 of the IEEE international conference on neural networks*, Perth, Australia. Piscataway, NJ: IEEE, pp. 1942–1948.

[31] Y. Shi and R.C. Eberhart, "**A modified particle swarm optimizer**", *in Proc. 1998 of the IEEE conference on evolutionary computation*, Anchorage, Piscataway, NJ: IEEE Press, pp. 69–73.

[32] Y. Shi , R.C. Eberhart, "**Empirical study of particle swarm optimization**", *in Proc. 1999 of the IEEE congress on evolutionary computation*,

Piscataway, NJ. Piscataway, NJ: IEEE Press, pp. 1945–1950.

[33] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan and Y. Alizadeh, "**Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems**", *Computer Methods in Applied Mechanics and Engineering*, Vol. 197(33-40), pp. 3080–3091, 2008.

[34] Z-L. Gaing, "**Particle swarm optimization to solving the economic dispatch considering the generator constraints**", *IEEE Transaction on Power System*, Vol.18, No.3, pp. 1187–1195, 2003.

[35] Z-L. Gaing, "**Closure to "Discussion of "Particle swarm optimization to solving the economic dispatch considering the generator constraints""**, *IEEE Transactions on Power Systems*, Vol.19, No.4, pp. 2122–2123, 2004.

[36] M. Pourakbari-Kasmaei , M. Rashidi-Nejad, "**An effortless hybrid method to solve economic load dispatch problem in power systems**", *Energy Conversion and Management*, Vol.52(8-9), pp. 2854-2860, 2011.

[37] A.Y. Saber, S. Chakraborty, S.M. Abdur Razzak and T. Senjyu, "**Optimization of economic load dispatch of higher order general cost polynomials and its sensitivity using modified particle swarm optimization**", *Electric Power Systems Research*, Vol.79, No.1, pp. 98-106, 2009.

[38] E. Sayedi, M.M. Farsangi, M. Barati and K.Y. Lee, "**A modified shuffled frog leaping algorithm for nonconvex economic dispatch problem**", *in Proc. 2012 IEEE Power and Energy Society General Meeting*, San Diego, CA, pp. 1-8.

[39] L. Wang, L.P. Li, "**An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems**", *Electrical Power and Energy Systems*, Vol. 44, No.1, pp. 832-843, 2013.

[40] M. Moradi-Dalvand, B. Mohammadi-Ivatloo, A. Najafi and A. Rabiee, "**Continuous quick group search optimizer for solving non-convex economic dispatch problems**", *Electric Power Systems Research*, Vol. 93, pp. 93-105, 2012.

[41] J. Cai, Q. Li, L. Li, H. Peng and Y. Yang, "**A hybrid CPSO–SQP method for economic dispatch considering the valve-point effects**", *Energy Conversion and Management*, Vol.53, No.1, pp. 175-181, 2012.

[42] J. Cai, Q. Li, L. Li, H. Peng and Y. Yang, "**A hybrid FCASO-SQP method for solving the economic dispatch problems with valve-point effects**", *Energy*, Vol. 38, No.1, pp. 346–353, 2012.

[43] A.S. Reddy, K. Vaisakh, "**Shuffled differential evolution for large scale economic dispatch**", Electric *Power Systems Research*, Vol. 96, pp. 237-245, 2013.

[44] M. Sydulu, "**A very fast and effective noniterative ''Lamda Logic Based'' algorithm for economic dispatch of thermal units**", *in Proc. 1999 IEEE region 10 conference TENCON*, Cheju Island, Vol.2, pp. 1434–1437.