

# Sensing Cloud Optimization applied to a non-convex constrained economical dispatch

P. M. Fonte<sup>a,c</sup>

<sup>c</sup>Department of Electrical Engineering and Automation  
 Instituto Superior de Engenharia de Lisboa  
 Rua Conselheiro Emídio Navarro, Lisboa  
 pfonte@deea.isel.pt

Cláudio Monteiro<sup>c</sup>, F.P. Maciel Barbosa<sup>b,c</sup>

<sup>c</sup>Universidade do Porto, Faculdade de Engenharia  
<sup>b</sup>INESC TEC PORTO  
 Rua Dr. Roberto Frias, Porto  
 cdm@fe.up.pt, fmb@fe.up.pt

**Abstract** – In this paper it is intended to solve an Economical Dispatch (ED) problem with a new tool, named Sensing Cloud Optimization (SCO). It is a technique based on clouds of particles which allow a dynamic change in search space. It has appropriate heuristic characteristic to solve not convex, not differentiable and highly constrained optimisation problems. It is provided with a statistical analysis which determines the cloud's dimension with dynamic adjustments in search space in order to accelerate the convergence and to avoid to get trapped in local minima. Two case studies are presented in which SCO demonstrated good performances reaching lower cost values where compared with other techniques.

**Index Terms** – Cloud of particles, Optimization, Economic Dispatch, Non-convex cost functions

## I. INTRODUCTION

The Economical Dispatch (ED) problem is an important issue in the power system operation. Fundamentally it is intended to evaluate the value that each on-line unit should generate with the lowest cost respecting the technical and load demand constrains. The ED uses as a basis the Unit Commitment solution, excluding from the optimization the generation units that are off. It is common to adopt approaches that merge the problem of Unit Commitment with the problem of Economic Dispatch using Integer Mixed [1].

The Dynamic Economical Dispatch (DED) problem is an extension of the conventional ED since it takes into account the variation of demand over time(1):

$$\min \sum_t^T \sum_{i=1}^{N_G} F_{i,t}(P_{i,t}) \quad (1)$$

Where  $F_{i,t}$  is the cost function of each unit  $P_i$  at interval  $t$ , and  $N_G$  represents the number of on-line units [2]. Over the past decade, many methods have been developed to solve the economical dispatch problem. There are the traditional methods such as Gradient, Lagrangean function, Lambda-iteration method, Dynamic Programming, Newton's method, Linear Programming and Interior Point method, among others [3]. However, some generation cost functions of recent thermal units are not continues, not convex, neither differentiable due to valve-point loading effect. Thus, the ED problem becomes a non-convex problem with constrains, which cannot be solved directly by some of the traditional

mathematical methods. Dynamic programming can solve this kind of problems, but can suffer with the dimension and the time needed to solve it [4-6]. On the other hand, as the traditional optimization packages used in power systems only supports convex functions, non-convex functions are manipulated so that they can be solved.

In parallel, several heuristic methods were proposed to solve this kind of problems such as Genetic Algorithms (GA), Simulated Annealing (SA), Taboo Search (TS), Evolutionary Programming (EP), Evolutionary Strategies (ES), Particle Swarm Optimization (PSO), Bacteria Foraging Optimization (BFO)[3],[7], Ant Colony Optimization (ACO), Artificial Neural Networks approach with Hopfield Networks, Artificial Bee Colony (ABC)[7] and hybrid artificial intelligence methods [3], [10] and [7-9]. From the base algorithms several improved approaches and hybrid were proposed, as Improved Taboo Search [4], Fast Evolutionary Programming and Improved Fast Evolutionary Programming [11], Improved Particle Swarm Optimization [10] and hybrids as PSO with Evolutionary programming [12], PSO with crossover operations [13] and Fast Evolutionary Programming with Swarm Direction [2].

From the tests done up to now, the proposed algorithm [14] demonstrate good convergence velocity and precise results. As well as, dealing easily with large quantity of variables, as can be seen by the problems analysed and which will be presented later.

## II. PROBLEM DESCRIPTION

Traditional ED problems characterise the cost function as quadratic, but due to valve point effects, an absolute term is added to the quadratic cost function, characterised by (2)[15],

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + |e_i \sin(f_i(P_{i\min} - P_i))| \quad (2)$$

where  $a_i$ ,  $b_i$ ,  $c_i$ ,  $e_i$  and  $f_i$  are the function coefficients for  $i^{\text{th}}$  unit while  $P_{i\min}$  is the minimum power constrain. Beyond this, some thermal units, as combined cycle, operate with different kind of fuels and multiple cost curves, resulting an "hybrid cost function" represented by several piecewise functions reflecting the effects of fuel type changes.

In an Electric Power System, the total production must equal the load demand  $D$  plus power transmission losses  $P_L$  (3). The losses are function of unit's power outputs and can be represented by the loss coefficients matrix  $\mathbf{B}$  (4).

$$\sum_{i=1}^{N_G} P_i = P_L + D \quad (3)$$

$$P_L = \sum_{i=1}^{N_G} \sum_{j=1}^{N_G} P_i B_{ij} P_j + \sum_{i=1}^{N_G} B_{oi} P_i + B_{oo} \quad (4)$$

Simultaneously several technical operation constrains must be satisfied such as minimum and maximum power generation and ramp limits. The increasing ramp limit is expressed by  $UR_i$ , and  $DR_i$  represents the decreasing ramp limit. Together with minimum  $P_i^{\max}$  and maximum limits  $P_i^{\min}$ , results (5), where  $P_i$  represents the power output of  $i^{\text{th}}$  unit at a given time interval, and  $P_i^o$  the power at previous time interval,

$$\max(P_i^{\min}, P_i^o - DR_i) \leq P_i \leq \min(P_i^{\max}, P_i^o + UR_i) \quad (5)$$

In some cases, the total operation range of a generating unit is not always available due to physical operation limitations, as steam valve operation, vibrations in the shaft bearings, among others, resulting in prohibited operating zones. In practical operation, the units must avoid operating in steady state in the prohibited zones as (6), where the indices  $l$  and  $u$  represents respectively the lower and upper bounds of prohibited operating zones and  $z = 2,3..n_z$  the number of prohibited zones of  $i^{\text{th}}$  unit[2].

$$P_i \in \begin{cases} P_i^{\min} \leq P_i \leq P_{i,1}^l \\ P_{i,z-1}^u \leq P_i \leq P_{i,z}^l \\ P_{i,n_z}^u \leq P_i \leq P_i^{\max} \end{cases} \quad (6)$$

### III. ALGORITHM DESCRIPTION

The SCO algorithm proposed by [14] is based on a stochastic technique based in populations with parallel search using groups of particles, without presenting evolutionary strategies. It has appropriate heuristic characteristic to solve not convex, not differentiable and highly constrained optimization problems.

It is characterized by two distinct steps, the cloud particles fitness evaluation and a statistical analysis. The first one determines the cloud's direction and the second one determines its dimension with dynamic adjustments of the search space in order to accelerate the convergence and avoid to get trapped in local minima.

The concept of central particle was introduced, which represents the optimal value, being the remaining particles of cloud spread around, according with a Gaussian distribution. These particles will act as sensors "to fill" the search space and give "signals" to the central particle moves into the best direction. An adaptive adjustment of search space is performed by two inverted sigmoid functions controlled by the coefficient of determination as demonstrated in point *F*.

#### A. Central particle generation

SCO starts creating a central particle  $X_q$  at a random position in the search space, normalized by the *min-max* method, as in (7).

$$X_{q(i)} = Rnd(0,1) \cdot (x_{\max(i)} - x_{\min(i)}) + x_{\min(i)} \quad (7)$$

With  $i = 1..N$ , where  $N$  represents the problem dimension and  $x_{\max}$  and  $x_{\min}$  the maximum and minimum values of search space.

#### B. Cloud generation

A cloud with  $j=1..M$  particles with  $N$  dimensions is created, following a Gaussian distribution, with mean centered in the central particle and variance  $\sigma^2$ . In the first iteration, this value is defined by the user and in the followings is defined by a self-adaptive process.

#### C. Particles evaluation

After the cloud generation, the fitness value of each particles are calculated and the minimum fitness value of the cloud,  $f(x_{b(j)})$  as well as its position  $x_{b(j)}$  are registered.

The originality contained in this methodology is based in how the fitness values are addressed.

#### D. Quadratic regression coefficients determination

This step is an attempt to program the least square method to estimate the quadratic regression coefficients which best fit the particles of the cloud. The added value of using particles is the fact that even if the search space is not continuous it is always possible the determination of regression coefficients from the fitness cloud values. The use of a second order polynomial has also the added value that always indicate a local or global minimum within the search space if the gradient is negative, hence a convex function, which can give some information for where the central particle should be moved. Depending on the search space, however, the polynomial may result on a concave function with a central particle progressing toward the maximum. A possible solution will be shown ahead in this paper.

#### E. Cloud statistical parameters determination

To understand if the central particle is far or close to the fitness function minimum, the Euclidean distance between each particle and the central particle,  $X_q$ , is calculated by (8).

$$\varphi_{(j)} = \left\| X_{q(j)} - x_{(j)} \right\| \quad (8)$$

Fig. 1 represents the graph with fitness values in function of the distance  $\varphi_{(j)}$ . It can be verified that the points closest to central particle have lowest fitness, while those farther away have worse values. It means that the central particle is above a minimum and the search should be more refined and the cloud should be focused in the search space near the central particle. The opposite case can be observed in fig. 2 where the cloud particles have no sufficient information and decision capacity to determine if the central particle is far away or close a minimum.

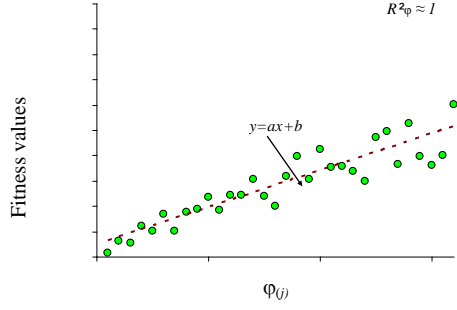


Fig. 1 Evaluation of  $\varphi_{(j)}$  parameter near optimal solution situation

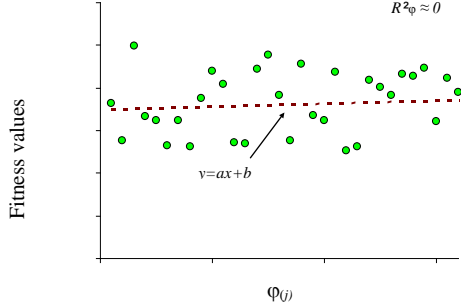


Fig. 2 Evaluation of  $\varphi_{(j)}$  parameter in distant optimal solution situation

This situation can be evaluated calculating the determination coefficient  $R_\varphi^2$  of a first-order function which best fit the values, as presented in fig. 1 and 2. At same time the slope is calculated and normalized by (9), where  $\sigma_{\varphi_{(j)}}$  represents the distance's standard deviation between  $X_{q(i)}$  and the cloud particles  $X_{(j)}$  and  $\sigma_{fitness_{(j)}}$  the standard deviation of fitness values.

$$\nabla \varphi = \frac{df(\varphi_{(j)})}{d\varphi_{(j)}} \left( \frac{\sigma_{\varphi_{(j)}}}{\sigma_{fitness_{(j)}}} \right) \quad (9)$$

The slope determination gives additional information, if it is high it means that cloud is above a narrow valley, otherwise the cloud is above a wide valley.

$$\Delta \varphi = \frac{1}{1 + h \cdot \nabla \varphi} \quad (10)$$

In (10) the  $h$  parameter can be used to accelerate the search however, if the value is too high the search can be faster but not so accurate.

#### F. New cloud generation

As mentioned in section D., if the quadratic function (parabola) defined by the cloud particles fitness values, results in a convex function and if the new central particle moves to the parabola vertex, the objective function trend to be lower or at least equal to the value obtained in previous iteration. However it is necessary to evaluate the quality of new central particle trend point. This evaluation may be obtained calculating the determination coefficient of the quadratic

function  $R_{q(i)}^2$ . If the determination coefficient is low, the confidence of quadratic function is low and the vertex may not provide reliable information about a supposed lower fitness value in the search space. According to this considerations, it is proposed that the position of new central particle  $X_{q(i)}$  must result from an weighted value between the value given by the parabola vertex,  $t_{p(i)}$ , and the position of particle with the best fitness value  $x_{b(i)}$ , trough (11).

$$X_{q(i)} = R_{q(i)}^2 \left( t_{p(i)} \right) + \left( 1 - R_{q(i)}^2 \right) x_{b(i)} \quad (11)$$

If in the limit  $R_{q(i)}^2$  is equal to 0 the central particle will move toward  $x_{b(i)}$ , on the other hand, if it is equal to 1 it is concluded that the parabola's vertex indicates a minimum inside the search space bounded by the limits of each dimension of cloud.

Depending on the cost function to be optimized and the search space covered by the cloud, the parabola can be concave indicating a maximum instead a minimum. In this case, it is necessary to calculate the trend point in order to ensure that the central particle continues to move toward a minimum. It is done by calculating the roots of the concave function, subtracted by the best fitness, as indicated in (12).

$$a_{0(i)} + a_{1(i)}x_{(i)} + a_{2(i)}x_{(i)}^2 - f(x_{b(i)}) = 0 \quad (12)$$

Then the trend point value closest  $X_{b(i)}$  will be chosen(11).

$$t_{p(i)} = \min \left\{ \left| x_{1(i)} - x_{b(i)} \right|, \left| x_{2(i)} - x_{b(i)} \right| \right\} \quad (13)$$

To prevent a premature convergence into a local optimum and, at the same time, ensure a rapid search of optimum, the cloud particles variance of Gaussian distribution must be dynamic adapting along the search space. After the definition of new central particle position, a new cloud is created, centred on the central particle and variance, to each dimension, calculated by(14), where  $k$  is the iteration number.

$$\sigma_{(i)}^{2(k+1)} = \sigma_{(i)}^{2(k)} F_{s1}^{(k)} F_{s2(i)}^{(k)} \quad (14)$$

The changing in the cloud variance is done by (15) and (16) which represent inverted sigmoid functions, with shape coefficients  $t_c$ ,  $t_s$ ,  $t_m$  and  $t_n$ .

$$F_{s1} = 1 + \frac{\Delta \varphi - 1}{e^{\left( \frac{-8(R_\varphi^2 - t_c)}{t_s} \right)}} \quad (15)$$

According fig.1 for values of  $R_\varphi^2$  near 1, the central particle is near an optimal solution, so the algorithm should narrow the search space for the next iteration by  $F_{s1}$ (15).

$$F_{s2(i)} = 1 - \frac{(K-1)}{1 + e^{(-t_m \cdot R_{q(i)}^2 + t_n)}} \quad (16)$$

On the other hand, if the determination coefficient  $R_{q(i)}^2$  has very low values, it means that the search space delimited by the cloud is quite flat and the cloud search space should be increased by a  $K$  value. On the contrary, if  $R_{q(i)}^2$  is approximately 1, low or no variation are made to the search space.

At the end, the variance of each dimension is given by (14) allowing the cloud search space increase or decrease in a dynamic and automatic way.

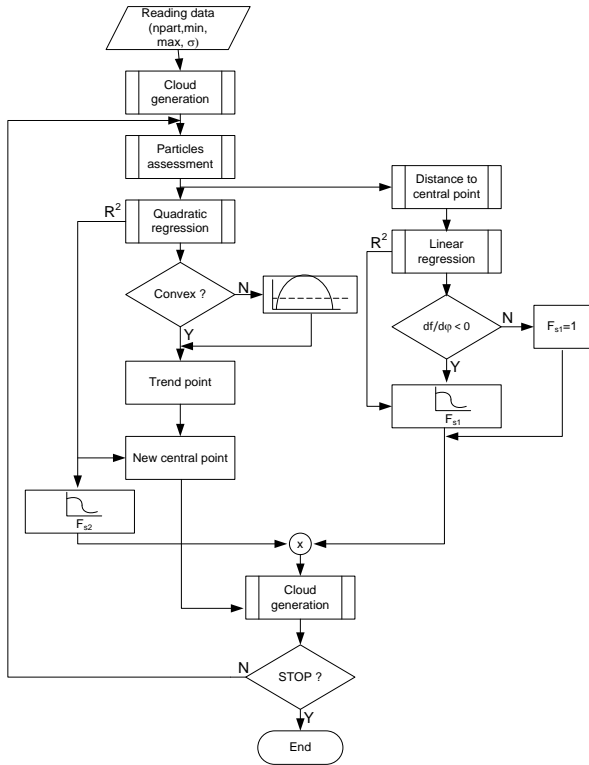


Fig. 3 SCO's flowchart

#### IV. ALGORITHM APPLIED TO ECONOMIC DISPATCH

Step 1 Create randomly a central particle  $P_{q(i)}$ , with  $i=1..N_G$  dimensions, representing each generation unit according to its limits, as in(17),

$$P_{q(i)} = rand(0,1) \cdot (P_i^{\max} - P_i^{\min}) + P_i^{\min} \quad (17)$$

if there are starting values  $P_{q(i)} = P_i^o$ . Create the remaining cloud  $[N_G \times N_P]$  with  $j=1..N_P$  particles and  $i=1..N_G$  dimensions normally distributed, centered in  $P_{q(i)}$  and with standard deviation  $\sigma_{(i)}^{(k=1)}$  (18).

$$P_{(i,j)}^{(k)} \sim N\left(P_{q(i)}^{(k)}, \sigma_{(i)}^{(k)}\right) \quad (18)$$

Step 2 To each particle  $P_{(j)}^{(k)}$ , calculate the transmission loss  $P_L^{(k)}$ , by (4), evaluate each particle with (19) and retain the best fitness value  $P_{(j)_{BEST}}^{(k)}$  and its position.

$$f_{(j)}^{(k)} = \sum_{i=1}^{N_G} F_i(P_{(i,j)}^{(k)}) + q \cdot \left( \sum_{i=1}^{N_G} P_{(i,j)}^{(k)} - D - P_L^{(k)} \right)^2 \quad (19)$$

Step 3 Calculate the second order function regression coefficients to each dimension  $i$ ,  $[\beta_{0(i)} \beta_{1(i)} \beta_{2(i)}]$  and the coefficient of determination  $R_{q(i)}^{2(k)}$ .

Step 4 Verify the convexity of the parabolas. If they are convex,  $t_{p(i)}^{(k)} = -b_i / 2a_i$ , if not, calculate  $t_{p(i)}^{(k)}$  by (13).

Step 5 Generate new central particle by (20) and verify if satisfies the constrains (5) and (6). If the solution violate the constrains must be modified to satisfy (5) and (6).

$$P_{q(i)}^{(k+1)} = t_{p(i)}^{(k)} \cdot R_{q(i)}^2 + (1 - R_{q(i)}^2) \cdot P_{(i)_{BEST}}^{(k)} \quad (20)$$

As the remaining particles act as sensors and they are not candidates to solution do not need to satisfy all constrains.

Step 6 Calculate the Euclidean distance from each particle  $j$  to the central particle, by (21).

$$d_{(j)}^{(k)} = \|P_q^{(k)} - P_{(j)}^{(k)}\| \quad (21)$$

Step 7 Calculate the linear regression coefficients  $[\alpha_{(0,i)}, \alpha_{(1,i)}]$  and the coefficient of determination  $R_{\varphi(i)}^{2(k)}$ .

Step 8 Calculate the new standard deviation for each dimension  $i$  of new cloud by (14).

Step 9 If  $k = it_{max}$  go to step 10, otherwise,  $k=k+1$  and go to step 1.

Step 10 The central particle that generates the latest best fitness is the optimal generation value for each unit and minimum total generation cost.

#### V. NUMERICAL EXAMPLES AND RESULTS

To verify the feasibility of the proposed method, 2 cases were studied to demonstrate the capacity of the algorithm to reach the optimal values as well as, the capacity to solve high constrained problems with crescent dimension. The proposed algorithm was implemented in *Matlab*® (R2010a) and executed on a Core (i2) 1,59 GHz processor.

### A. First case study

The first test system consisted of 15 thermal units with prohibited operation zones and ramp limits, feeding a load demand of 2630 MW. The network consisted in 30-buses network characterized by the loss coefficients matrix **B**. The units had no valve-point effects and the initial values of each unit were defined by  $P_i^o$ . All parameters used in this case study are described in [2] and [4], (100 particles and 100 generations in [2] and 100 particles and 200 generations in [4]).

To simulate this case, the central particle as well as, each individual particle of the cloud have 15 dimensions ( $P_1...P_{15}$ ), one for each generation unit. Depending on the number of particles  $N_p$  the cloud dimension should be  $[15 \times N_p]$ .

As heuristic methods may not converge to exactly the same solution at each run owing to their stochastic behaviour, their performances could not be judge by the results of a single trial. Due to that, all cases were performed 50 trials keeping the average, maximum and minimum reached values. The number of iterations was limited to 100 as in [4].

Excluding the central point, the remaining particles act as "sensors" of the search space and are used to calculate the first and second order polynomials, there is a minimum number of particles necessary to describe the curve fitting. On the other hand, a large number will make the algorithm slower due to the necessary of evaluating each particle. After some experiences, the particles number was set to 50. In first case study the parameter  $h$  of (10) was 1, on the second one  $h = 10$ . In (15)  $t_c$  was set to 0,5 and  $t_s$  to 1, in (16) the values of  $K$ ,  $t_m$  and  $t_n$  were set, respectively to 1,01, 50 and 5. The penalty factor  $q$  of (19) was set to one after several experiments.

The obtained results were compared with Fast Evolutionary Programming (FEP), Improved Fast Evolutionary Programming (IFEP) and Swarm Direction Fast Evolutionary Programming (SFEP) proposed by [2] and Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) proposed by [4]. Although in [2] and [4] the unit's parameters were the same, the values of  $P_i^o$  were different. This fact is enough to produce quite different results as can be seen in Table I. These differences were due to the ramps limits associated with start values  $P_i^o$  leading to different production portfolios. Fig. 4 shows the total costs obtained by [2] and [4] and the values obtained using SCO. In both situations is clear the improvement of SCO over all other methods presenting the lowest values for all indicators as well as the lowest average value after 50 trials.

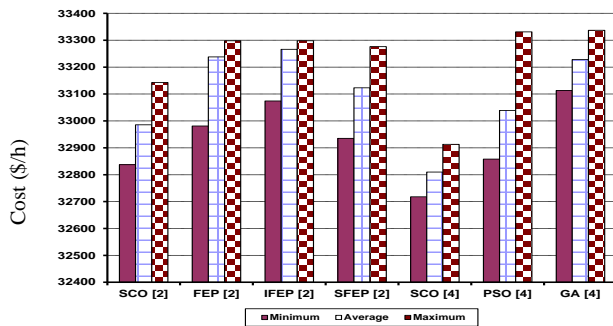


Fig. 4 Comparative minimum, maximum and average solutions

TABLE I RESULTS OBTAINED (15-UNITS 2630 MW) (BEST INDIVIDUAL)

Power output (MW)	Method						
	SCO [2]	FEP [2]	IFEP [2]	SFEP [2]	SCO [4]	PSO [4]	GA [4]
P <sub>G1</sub>	455,0	389,9	333,8	343,7	455,0	439,1	415,3
P <sub>G2</sub>	455,0	450,4	343,4	346,5	380,0	408,0	359,7
P <sub>G3</sub>	130,0	89,8	85,3	127,9	130,0	119,6	104,4
P <sub>G4</sub>	130,0	55,9	119,9	96,4	130,0	130,0	74,9
P <sub>G5</sub>	355,2	272,6	457,0	426,1	170,0	151,1	380,3
P <sub>G6</sub>	288,0	455,1	403,9	413,9	460,0	460,0	426,8
P <sub>G7</sub>	353,9	457,9	464,1	452,3	430,0	425,6	341,3
P <sub>G8</sub>	162,0	109,7	102,9	117,5	60,0	98,6	124,8
P <sub>G9</sub>	25,0	90,5	103,8	42,5	51,3	113,5	133,1
P <sub>G10</sub>	90,7	53,9	102,9	100,9	160,0	101,1	89,3
P <sub>G11</sub>	80,0	79,9	28,3	79,1	78,4	33,9	60,1
P <sub>G12</sub>	76,6	50,8	24,1	26,7	80,0	80,0	50,0
P <sub>G13</sub>	27,4	54,9	50,0	36,1	31,7	25,0	38,8
P <sub>G14</sub>	17,7	23,4	31,5	43,6	20,4	41,4	41,9
P <sub>G15</sub>	19,1	26,2	21,6	15,0	22,6	35,6	22,6
P <sub>T</sub> (MW)	2665,71	2668,2	2662,4	2663,4	2659,4	2662,4	2668,4
P <sub>L</sub> (MW)	35,71	38,387	32,431	38,278	29,4	32,43	38,278
Cost (\$/h)	<b>32837</b>	32981	33074	32935	<b>32718</b>	32858	33113

Fig. 5 shows the convergence speed of SCO to first case study with setting similar to [2].

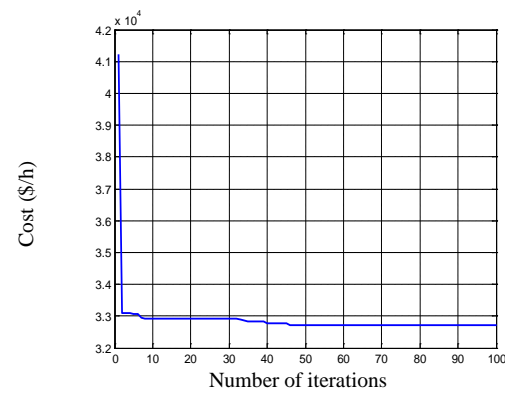


Fig. 5 Convergence behaviours of SCO for a 15-units problem

Some additional tests were done with the setting of [4]. Increasing the number of iterations to 500, even lower cost value was obtained (32708\$/h) with total production of 2659,58 MW and 29,58 MW of losses after 324 iterations. On the other hand, with 100 particles and 200 iterations, the value of (32703\$/h) was reached. These results are in line with those obtained in [13] with PSO with Chaotic Sequences (CSPSO), PSO with Crossover Operation (COPSO) and PSO with both Chaotic sequences and Crossover operation (CCPSO).

### B. Second case study

To investigate the behaviour of a larger power system, a case with 40 units was tested without ramp limits, prohibited zones, and losses. In addition the units had valve-point effect, which transform the cost function on a non-convex neither differentiable function. Due to the problem's complexity the number of particles was increased to 100 and maximum iterations to 500.

The convergence velocity is shown in fig. 6 revealing good velocity converging in less than 500 iterations. The obtained values after 50 runs were compared with several other heuristic optimization techniques such as PSO[6,12], Classical Evolutionary Programming (CEP), PSO Embebbed in CEP (PSO\_CEP)[10], Improved Fast Evolutionary programming (IFEP), Evolutionary Strategy Optimization (ESO), PSO Local

Random Search (PSO\_LRS), New PSO (NPSO), NPSO\_LRS[15] and CTPSO[11].

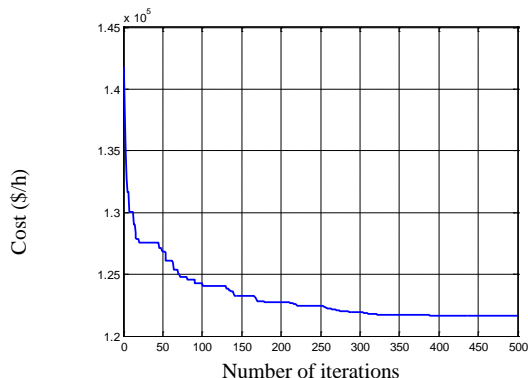


Fig. 6 Convergence behaviours of SCO for a 40-units problem

The numerical values of the best value obtained for each technique are shown in table II, while the comparison between the minimum, average and maximum are shown in fig. 7. The unit’s production values are not indicated due to the large amount of data.

TABLE II RESULTS OBTAINED (40-UNITS 10500 MW) (BEST INDIVIDUAL)

Method	SCO	CEP	PSO	PSO_EP	PSO	IFEP
Cost (\$/h)	<b>121632</b>	125420	126849	123670	121751	122624
Method	ESO	PSO-LRS	NPSO	NPSO-LRS	CTPSO	
Cost (\$/h)	122122	122036	121704	121664	121695	

## VI. CONCLUSIONS

This paper presents a new optimization algorithm which shows to be capable to solve high constrained ED problems, as well as better results concerning cost values than evolutionary programming methods, classical Particle Swarm Optimization and some evolutionary and hybrid techniques with evolutionary problems and PSO. Next steps will be the application of this algorithm to high dimensional and more constrained problems, with multi-fuels units and valve-point effects. As conclusion, this paper shows the algorithm capacity to solve ED problems with both convex and non-convex cost functions. These results were obtained simulating case studies presented in biography. In the future to the algorithm should be tested in real environment to compare it with commercial optimization packages and by this way to assess the algorithm capacity to be used in industrial environmental.

## REFERENCES

- [1] João P. S. Catalão, (2012) Electric Power Systems: Advanced Forecasting Techniques and Optimal Generation Scheduling. CRC Press. ISBN1439893942.
- [2] Z.-L. Gaing, “Dynamic Economic Dispatch Solution Using Fast Evolutionary Programming with Swarm Direction,” in *Electrical Engineering*, 2009, no. 1, pp. 1538-1544.
- [3] J. Hazra and A. Sinha, “Application of soft computing methods for Economic Dispatch in Power Systems,” *Engineering*, no. 2, pp. 538-543, 2009.
- [4] Z.-L. Gaing, “Particle swarm optimization to solving the economic dispatch considering the generator constraints,” *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1187-1195, Aug. 2003.
- [5] Whei-Min Lin; Fu-Sheng Cheng; Ming-Tong Tsay; , “An improved tabu search for economic dispatch with multiple minima ,” *Power Systems, IEEE Transactions on* , vol.17, no.1, pp.108-112, Feb 2002.
- [6] Selvakumar and K. Thanushkodi, “A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems,” *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 42-51, Feb. 2007.
- [7] Serapião, “Fundamental de Optimization by Swarm Intelligence: an Overview”, *Fundamentos de Otimização por Inteligência de Enxames: uma Visão Geral*, Revista Contole & Automação, Vol. 20, no 3, Jul-Set, 2009 (in portuguese)
- [8] Mahdad and K. Srairi, “Fuzzy controlled parallel PSO to solving large practical economic dispatch,” *IECON 2010-36th Annual Conference of the IEEE Industrial Electronics Society*, pp. 2695–2701, Nov. 2010.
- [9] J. Xu, S. Tan, and S. K. Panda, “Optimization of economic load dispatch for a microgrid using evolutionary computation,” *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3192–3197, Nov. 2011.
- [10] K. Y. Lee, “An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems,” *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 156-166, Feb. 2010.
- [11] Sinha, N.; Chakrabarti, R.; Chattopadhyay, P.K.; , "Evolutionary programming techniques for economic load dispatch," *Evolutionary Computation, IEEE Transactions on* , vol.7, no.1, pp. 83- 94, Feb 2003.
- [12] Sinha, N; Purkayastha, B.; , "PSO embedded evolutionary programming technique for nonconvex economic load dispatch," *Power Systems Conference and Exposition, 2004. IEEE PES* , vol., no., pp. 66-71 vol.1, 10-13 Oct. 2004.
- [13] Jong-Bae Park; Yun-Won Jeong; Joong-Rin Shin; Lee, K.Y., "An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems," *Power Systems, IEEE Transactions on* , vol.25, no.1, pp.156-166, Feb. 2010
- [14] Monteiro, C; Barbosa F.M., Fonte, P.M., “SCO – Sensing Cloud Optimization – A Novel Optimization Technique” (in review)
- [15] J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, “A Particle Swarm Optimization for Economic Dispatch with Nonsmooth Cost Functions,” *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 34-42, Feb. 2005.

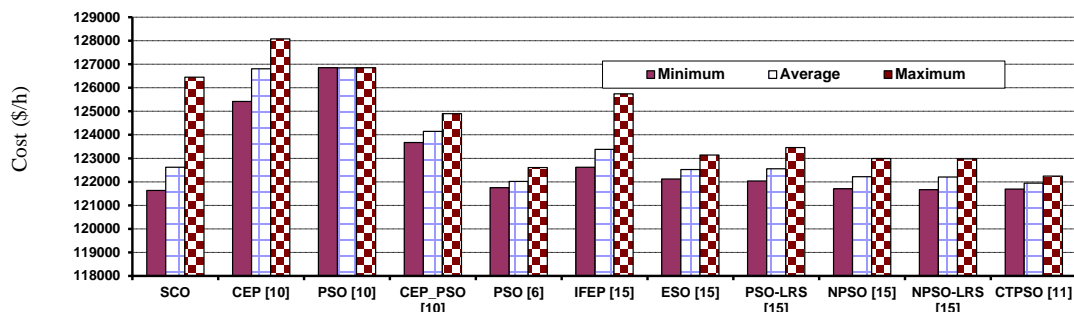


Fig. 7 Comparative minimum, maximum and average solutions for 40 units with valve points