Information Sciences 295 (2015) 609–632

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An immune algorithm with power redistribution for solving economic dispatch problems

V.S. Aragón [a,*], S.C. Esquivel [a], C.A. Coello Coello [b,c]

[a] LIDIC Research Group, Universidad Nacional de San Luis, Ej. de los Andes 950, 5700 San Luis, Argentina
[b] CINVESTAV-IPN (Evolutionary Computation Group), Computer Science Department, Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07300, Mexico
[c] UMI 3175 CNRS at CINVESTAV-IPN, Mexico

ARTICLE INFO

Article history:
Received 21 May 2014
Received in revised form 31 August 2014
Accepted 6 October 2014
Available online 22 October 2014

Keywords:
Artificial immune system
Economic dispatch problem
Metaheuristic

ABSTRACT

In this paper, we present an algorithm inspired on the T-Cell model of the immune system (i.e., an artificial immune system), which is used to solve economic dispatch problems. The proposed approach is called IA_EDP, which stands for *Immune Algorithm for Economic Dispatch Problem*, and it uses two versions of a redistribution power operator which tries to keep feasible the solutions that it finds. The proposed approach is validated using eight problems taken from the specialized literature. Our results are compared with respect to those obtained by several other approaches. We also perform some statistical analysis in order to determine the sensitivity of our proposed approach to its parameters.

1. Introduction

Power engineering is a subfield of electrical engineering that deals with the generation, transmission, distribution, and utilization of electric power. This is a network of interconnected components which converts different forms of energy to electrical energy. The four subsystems that compose a modern power system are: the generation subsystem, the transmission subsystem, the distribution subsystem, and the utilization subsystem. In the first one, the power plant produces the electricity. The second one transmits electricity to the load centers. The distribution subsystem continues to transmit the power to the customers. The utilization system is concerned with the different uses of electrical energy such as light, refrigeration, heating, air conditioning, domestic devices (e.g., TV sets, personal computers, microwave ovens, etc.), and water pumps, among many others.

During the generation of electrical power, another energy (e.g., hydraulic) is transformed into electricity. This transformation process may include the use of chemical, photo-voltaic, and electromechanical energy.

The fuel cost and the efficiency of the power station determine the operating costs of generating electrical energy. Thus, the economic dispatch problem (EDP) has become a very important task in the operation and planning of power systems. Its main objective is to optimize the generation of electricity from among the available units, such that the total generation cost is minimized whilst the constraints considered by the system are satisfied.

Classical methods have been proposed to solve EDP, but they suffer from some limitations (for instance, the objective functions and the constraints must be differentiable). On the other hand, modern heuristic algorithms have proved to be able to deal with nonlinear optimization problems, e.g., EDPs.

* Corresponding author at: CONICET, Argentina.
E-mail addresses: vsaragon@unsl.edu.ar (V.S. Aragón), esquivel@unsl.edu.ar (S.C. Esquivel), ccoello@cs.cinvestav.mx (C.A. Coello Coello).

http://dx.doi.org/10.1016/j.ins.2014.10.026
0020-0255/

In this paper, we propose an algorithm to solve EDPs which is inspired on the T cells from the immune system. Once the algorithm has found a feasible solution, it applies two redistribution power operators in order to improve the original solution with the aim of keeping such a solution feasible at a low computational cost.

The remainder of this paper is organized as follows. Section 2 defines different variations of the economic dispatch problem. Section 3 provides a short review of some of the approaches which were used to solve EDPs. In Section 4, we describe our proposed algorithm. In Section 5, we present the test problems used to validate our proposed approach and a statistical analysis of the impact of the parameters settings on the performance of our proposed IA_EDP. In Section 6, we present our results and we discuss and compare them with respect to other approaches. Finally, in Section 7, we present our conclusions and some possible paths for future research.

2. Problem formulation

The objective of economic dispatch problem (EDP) is to minimize the total generation cost of a power system while satisfying several constraints associated to the system, such as load demands, ramp rate limits, maximum and minimum limits, and prohibited operating zones. The objective function type (smooth or non smooth) and the constraints which are considered in the problem will determine how hard is to solve the problem.

2.1. Objective function

The mathematical formulation of the total fuel cost function (TC) is given by:
minimize

$$TC = \sum_{i=1}^{N} F_i(P_i) \tag{1}$$

where N is the number of generating units in the system, P_i is the power of ith unit (in MW) and F_i is the total fuel cost for the ith unit (in $/h).

The fuel cost function characteristics define if it is smooth[1] or non-smooth:

An EDP with a smooth cost function: it represents the simplest cost function. It can be expressed as a single quadratic function:

$$Fi(P_i) = a_i P_i^2 + b_i P_i + c_i \tag{2}$$

where $a_i,\ b_i$ and c_i are the fuel consumption cost coefficients of the ith unit.

EDP with a non-smooth cost function: it includes multiple non-differentiable points in order to represent the valve-points loading effects[2] that are present in EDP. It can be expressed as a quadratic and a sinusoidal function:

$$Fi(P_i) = a_i P_i^2 + b_i P_i + c_i + |e_i \sin(f_i(P_{min_i} - P_i))| \tag{3}$$

with e_i and f_i being the fuel cost coefficients of the ith unit with valve-point effects.

2.2. Constraints

As we mentioned before, the schedule has to minimize the total production cost and involves the satisfaction of both equality and inequality constraints.

1. Power Balance Constraint: the power generated has to be equal to the power demand required. It is defined as:

$$\sum_{i=1}^{N} P_i = P_D \tag{4}$$

2. Operating Limit Constraints: thermal units have physical limits about the minimum and maximum power that can generate:

$$P_{min_i} \leqslant P_i \leqslant P_{max_i} \tag{5}$$

where P_{min_i} and P_{max_i} are the minimum and maximum power output of the ith unit, respectively.

3. Power Balance with Transmission Loss: some power systems include the transmission network loss, thus Eq. (4) is replaced by:

---

[1] In mathematical analysis, a function that has derivatives of all orders is called *smooth*.

[2] This is a phenomenon that occurs in power plants that usually have multiple valves to control the power output of the units. When steam admission valves are first opened in thermal units, a sudden increase in losses is observed which leads to ripples in the cost function curve [29].

Table 1
Test problems characteristics.

Problem	Thermal units	P_L	Objective function	Ramp limits	Prohibited zones	P_D (MW)
SYS_3U_a	3	No	Smooth	No	No	850.0
SYS_3U_b	3	No	Non smooth	No	No	850.0
SYS_6U	6	Yes	Smooth	Yes	Yes	1263.0
SYS_13U	13	No	Non smooth	No	No	1800.0
SYS_15U	15	Yes	Smooth	Yes	Yes	2630.0
SYS_18U	18	No	Smooth	No	No	365.0
SYS_20U	20	Yes	Smooth	No	No	2500.0
SYS_40U	40	No	Non smooth	No	No	10500.0

$$\sum_{i=1}^{N} P_i = P_D + P_L \tag{6}$$

The P_L value is calculated with a function of unit power outputs that uses a loss coefficients matrix B, a vector B0 and a value B00:

$$PL = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1} B0_i P_i + B00 \tag{7}$$

4. Ramp Rate Limits: they restrict the operating range of all on-line units. Such limits indicate how quickly the unit's output can be changed:

$$\max\left(P_{min_j}, P_j^0 - DR_j\right) \leqslant P_j \leqslant \min\left(P_{max_j}, P_j^0 + UR_j\right) \tag{8}$$

where P_j^0 is the previous output power of the jth unit(in MW) and, UR_j and DR_j are the up-ramp and down-ramp limits of the jth unit (in MW/h), respectively.

5. Prohibited Operating Zones: they restrict the operation of the units due to steam valve operation conditions or to vibrations in the shaft bearing:

$$\begin{cases} P_{min_i} \leqslant P_i \leqslant P_{i,1}^l \\ P_{i,j1}^u \leqslant P_i \leqslant P_{i,j}^l, \quad j = 2, 3, \ldots, nj \\ P_{i,nj}^u \leqslant P_i \leqslant P_{max_i} \end{cases} \tag{9}$$

where nj is the number of prohibited zones of the ith unit, $P_{i,j}^l$ and $P_{i,j}^u$ are the lower and upper bounds of the jth prohibited zone.

For each problem, we need the following information:

1. Number of thermal units (N).
2. Cost for each thermal unit.
3. Operating Limit Constraints.
4. Power Demand (P_D).
5. Maximum number of objective function evaluations.
6. Ramp Rate Limits (if applicable).
7. Prohibited Operating Zones (if applicable).

Table 2
Parameter values.

Problem	Pop size	Evaluations	Probability	ϵ
SYS_3U_a	1	1000	0.8	–
SYS_18U	1	40,000	0.8	–
SYS_3U_b	20	1500	0.7	–
SYS_13U	1	25,000	0.7	–
SYS_40U	1	24,000	0.8	–
SYS_6U	10	3000	0.4	0.1
SYS_15U	20	6000/20,000	0.8	0.1
SYS_20U	5	20,000	0.9	0.1

Table 3
Approaches with respect to which we compared our proposed IA_EDP. OFE = Objective Function Evaluations. N.A. = Not Available.

Problem	Approach	OFE
SYS_3U_a	NM [56]	N.A.
	IEP [38]	N.A.
	MPSO [37]	N.A.
	IPSO [45]	3000
	ModPSO [48]	N.A.
	fast-CPSO [5]	3000
SYS_3U_b	GA [52]	10,000
	EP [57]	1500
	IEP [38]	Not found
	TM [25]	–
	MPSO [37]	–
	IPSO [45]	6000
	DE [19]	–
	fast-CPSO [5]	6000
SYS_6U	PSO [15]	20,000
	GA [15]	20,000
	NPSO_LRS [43]	20,000
	AIS [34]	–
	MTS [40]	100,000
	DE [30]	36,000
	ICA-PSO [51]	20,000
	SA-PSO [22]	20,000
	BBO [3]	50,000
	IHS [33]	100,000
	DSPSO-TSA [18]	4200
	DHS [53]	3000

3. Literature review

This section is aimed to review some of the most representative approaches used to solve the EDP. Our aim is really to highlight how the EDP has been tackled using different methods, rather than providing a comprehensive description of each of them.

Over the last years, several methods have been proposed to solve the EDP. They can be divided in three main groups: classical, based on artificial intelligence (AI) and hybrid methods.

Classical techniques become very cumbersome when dealing with complex dispatch problems, and they are limited due to their lack of robustness and efficiency in a number of practical applications. Examples of classical methods include Lagrangian relaxation (LR) [32] and dynamic programming (DP) [24,41].

Even when AI methods based on optimization techniques do not guarantee, in general, finding the global optimal solution, they can normally produce feasible sub-optimal solutions in a reasonably short computational time, which is the reason why they are widely used to solve the EDP. AI methods include simulated annealing (SA) [55], genetic algorithms (GAs) [52,6,11,7,2,14,49,27,26], particle swarm optimization (PSO) [15,37,45,43,42,48,35,51,5], evolutionary programming (EP) [57,38,44,16], tabu search (TS) [39,40,20], differential evolution (DE) [30,19,54,1], harmony search (HS) [9,33], artificial immune systems [34,12] and neural networks [36,23].

Finally, some researchers have reported the use of hybrid approaches, such as a combination of a genetic algorithm and the Taguchi method [47], a fuzzy adaptive hybrid particle swarm optimization algorithm [28], a hybrid multi-agent based particle swarm optimization algorithm [21], a combination of chaotic differential evolution and quadratic programming [8] a particle swarm optimizer hybridized with simulated annealing [22,17], a differential evolution algorithm with biogeography-based optimization [4], a differential evolution approach hybridized with a cultural algorithm [10], a Shuffle Frog Learning algorithm hybridized with SA [29], a bacterial foraging PSO-DE algorithm [50] and a differential evolution approach hybridized with a harmony search algorithm [53], among others.

4. Our proposed algorithm

In this paper, an adaptive immune system model based on the immune responses mediated by the T cells is presented. T cells belong to a group of white blood cells known as lymphocytes. They play a central role in cell-mediated immunity. They present special receptors on their cell surface called T cell receptors (TCR[3]) [31].

[3] TCRs are responsible for recognizing antigens bound to major histocompatibility complex (MHC) molecules.

Table 4
Approaches with respect to which we compared our proposed IA_EDP. OFE = Objective Function Evaluations. N.A. = Not Available.

Problem	Approach	OFE
SYS_13U	ICA-PSO [51]	40,000
	CDE_SQP [8]	180,000
	DE [30]	130,000
	DECDM [10]	25,000
	opt-aiNET [12]	180,000
	Z-opt-aiNET [12]	180,000
	TSARGA [47]	50,000
	HMAPSO [21]	N.A.
	MDE [1]	280,000
	SOMA [13]	25,000
	CSOMA [13]	25,000
	DHS [53]	60,000
SYS_15U	PSO [15]	20,000
	GA [15]	20,000
	AIS [34]	N.A.
	DE [30]	45,000
	CCPSO [35]	30,000
	MTS [40]	100,000
	MDE [1]	160,000
	DSPSO-TSA [18]	6000
	SA-PSO [22]	20,000
	ICA-PSO [51]	40,000
SYS_18U	ICA-PSO [51]	40,000
SYS_20U	Lambda-iteration method [46]	N.A.
	Hopfield neural network [46]	N.A.
SYS_40U	IFEP [16]	24,000
	CDE_SQP [8]	180,000
	DE [30]	240,000
	DECDM [10]	25,000
	CCPSO [35]	30,000
	EDA/DE [54]	40,000
	ARCGA [2]	N.A.
	BBO [3]	50,000
	DE/BBO [4]	80,000
	TSARGA [47]	25,000
	HMAPSO [21]	N.A.
	FAPSO-NM [28]	60,000
	SOMA and CSOMA [13]	25,000
	DHS [53]	24,000
	ICA-PSO [51]	70,000

Table 5
Results obtained by our proposed IA_EDP.

Problem	Best	Worst	Mean	Median	Std. dev.
SYS_3U_a	8194.3561	8194.3972	8194.3617	8194.3586	0.0080
SYS_18U	25429.0192	25429.0234	25429.0202	25429.0200	8.3093E−4
SYS_3U_b	8220.9337	8245.1847	8224.5114	8221.4482	5.0812
SYS_13U	17961.4331	18052.3155	17980.1898	17973.3475	21.6666
SYS_40U	121436.9729	121648.4401	122492.7018	121648.4401	182.5274
SYS_6U	15442.9369	15449.0294	15444.0361	15443.7217	1.04109
SYS_15U	32698.2018	32823.7790	32750.2176	32752.8928	9.2989
SYS_20U	62466.8044	62528.9870	62487.5109	62484.8616	12.0380

The model considers some processes that T cells suffer. These are proliferation (to clone a cell) and differentiation (to change the clones so that they acquire specialized functional properties); this is the so-called activation process.

IA_EDP (Immune Algorithm for Economic Dispatch Problem) is an algorithm inspired on the activation process, which is proposed to solve the EDP. IA_EDP operates on one population which is composed of a set of T cells.

For each cell, the following information is kept:

1. *TCR*: it identifies the decision variables of the problem ($TCR \in \Re^N$). Each thermal unit is represented by one decision variable.

2. *objective*: objective function value for TCR, $(TC(TCR))$.
3. *prolif*: it is the number of clones that will be assigned to the cell, it is N for all problems.
4. *differ*: it is the number of decision variables that will be changed when the differentiation process takes place (if applicable).
5. *TP*: it is the power generated by TCR $\left(\sum_{i=1}^{N} TCR_i\right)$.
6. P_L: it is the transmission loss for TCR (if the problem does not consider transmission loss, then $P_L = 0$).
7. *ECV*: it is the equality constraint violation for TCR $(|TP - P_D - P_L|)$. If $ECV > 0$, then the power generated is bigger than the demanded power, and if $ECV < 0$ then the power generated is lower than the required power.
8. *ICS*: it is the inequality constraints sum, $\sum_{i=1}^{nj} poz(TCR_i, i)$
$$poz(p,i) = \begin{cases} \min(p - PZ_{ll_i}, PZ_{ul_i} - p) & \text{if } p \in [PZ_{ll_i}, PZ_{ul_i}] \\ 0 & \text{otherwise} \end{cases}$$
where nj is the number of prohibited operating zones and $[PZ_{ll_i}, PZ_{ul_i}]$ is the prohibited range for the ith thermal unit.
9. feasible: it indicates if the cell is feasible or not. A cell is considered as feasible if: (1) $ECV = 0$ for problems without transmission network loss and $0 \leqslant ECV < \epsilon$ for problems with transmission loss. This means that if a solution generates less than the demanded power, then it is considered as infeasible $(ECV < 0)$ and (2) $ICS = 0$ for problems which consider prohibited operating zones.

4.1. Differentiation processes

Each type of cell, feasible or infeasible, has its own differentiation process.

4.1.1. Differentiation for feasible cells

In a random way, one of these processes is applied:

redistribution_decrease process: the idea is to take a value (called d) from one unit (say i) and distribute it (or a part of it) among other units (variables). ith unit is modified according to:

$$cell.TCR_i = cell.TCR_i - d \tag{10}$$

where $d = U(0, \min(cell.TCR_i - ll_i, max)), U(w_1, w_2)$ refers to a random number with a uniform distribution in the range $(w_1, w_2), ll_i$ is the lower limit of i and max is the maximum power that can be generated by the other units according to their current outputs (i.e. max $= \sum_{n=1 \wedge n \neq i}^{N} ul_n - cell.TCR_n$, where ul_n is the upper limit of n).
d was designed to avoid: (1) that the ith unit falls below its lower limit and (2) to take from the ith unit more power of what other units can generate. Next, d has to be distributed among the other units. First, the process tries to increase one unit, called k (which is chosen either in a random way or by taking into account its cost, such that less expensive units are preferred), with d. If this is not possible because the kth unit cannot generate that power (i.e., $cell.TCR_k + d > ul_k$, where ul_k is the upper limit of k), then we assign to the kth unit its upper limits. The remaining power is distributed among other units following the same idea.

redistribution_increase process: the idea here is to increase with a certain value (called j) to one unit (say i) and to take off this same value (or a part of it) from the other units. The ith unit is modified according to:

$$cell.TCR_i = cell.TCR_i + j \tag{11}$$

where $j = U(0, \min(ul_i - cell.TCR_i, \min)),\ U(w_1, w_2)$ refers to a random number with a uniform distribution in the range $(w_1, w_2),\ ul_i$ is the upper limit of i and min is the minimum power that can be generated by the other units according to their current outputs. j was designed to avoid: (1) that the ith unit exceeds its upper limit and (2) that other units fall below their lower limits. Now, j has to be removed from the other units. First, the process tries to decrease one unit, called k (which is chosen either in a random way or by taking into account its cost, in such a way that units more expensive are preferred), with j. If this is not possible because the kth unit falls below its lower limit (i.e., $cell.TCR_k - j < ll_k$, where ll_k is the lower limit of k), we assign to the kth unit its lower limit. The remaining power is removed from the other units following the same idea.

It is worth emphasizing that these operators only preserve the feasibility of solutions by taking into account the power balance constraints, but without considering transmission losses, operation limit constraints or prohibited operation zones.

4.1.2. Differentiation for infeasible cells

For infeasible cells, the number of decision variables to be changed is determined by their differentiation level. This level is calculated as a random value between 1 and the number of units into the system. Each variable to be changed is chosen in a random way and it is modified according to Eq. (12):

$$cell.TCR'_i = cell.TCR_i \pm m \tag{12}$$

where $cell.TCR_i$ and $cell.TCR'_i$ are the original and the mutated decision variables, respectively. $m = U(0,1) * |cell.ECV + cell.ICS|$. $U(0,1)$ refers to a random number with a uniform distribution in the range (0,1). In a

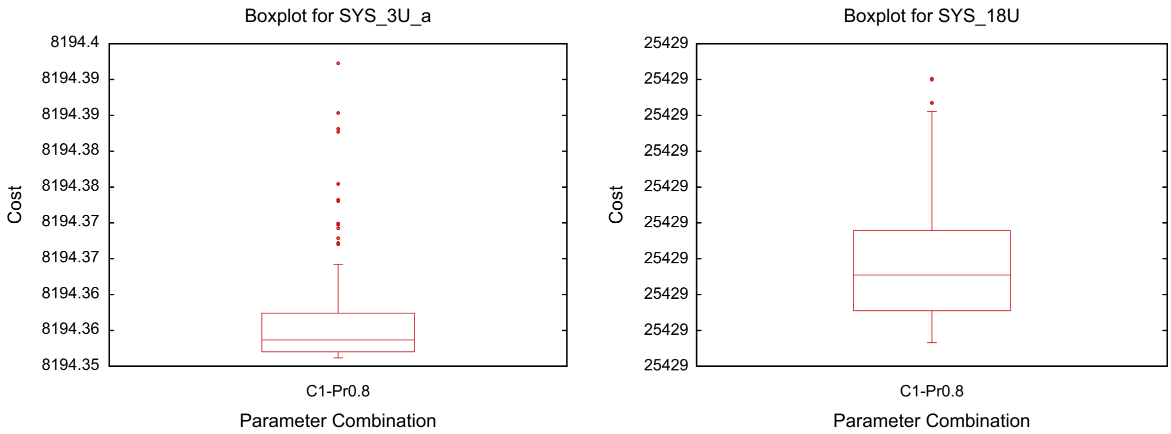

Fig. 1. Box plots for the test problems with the best parameters combination.

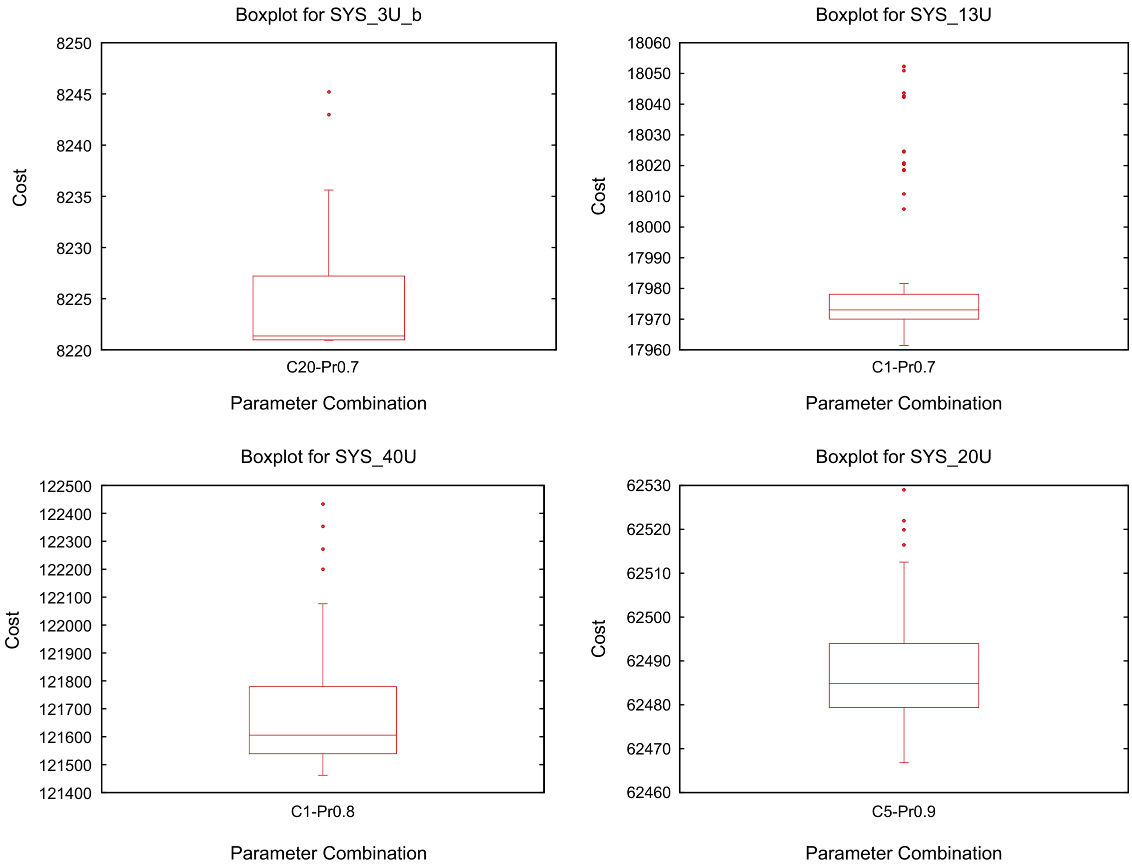

Fig. 2. Box plots for the test problems with the best parameters combination.

random way, it decides if m will be added or subtracted to $cell.TCR_i$. If the procedure cannot find a TCR_i' in the allowable range, then a random number with a uniform distribution is assigned to it ($cell.TCR_i' = U(cell.TCR_i, ul_i)$ if m should be added or $cell.TCR_i' = U(ll_i, cell.TCR_i)$, otherwise. ul_i and ll_i are the upper and lower limits of i, respectively).

The algorithm works in the following way (see Algorithm 1). First, the TCRs are randomly initialized within the limits of the units (Step 1). Then, *ECV* and *ICS* are calculated for each cell (Step 2). Only if a cell is feasible, its objective function value is calculated (Step 3). Next, while a predetermined number of objective function evaluations had not been reached (Steps 4–6) the cells are proliferated and differentiated considering if they are feasible or infeasible. Finally, statistics are calculated (Step 8).

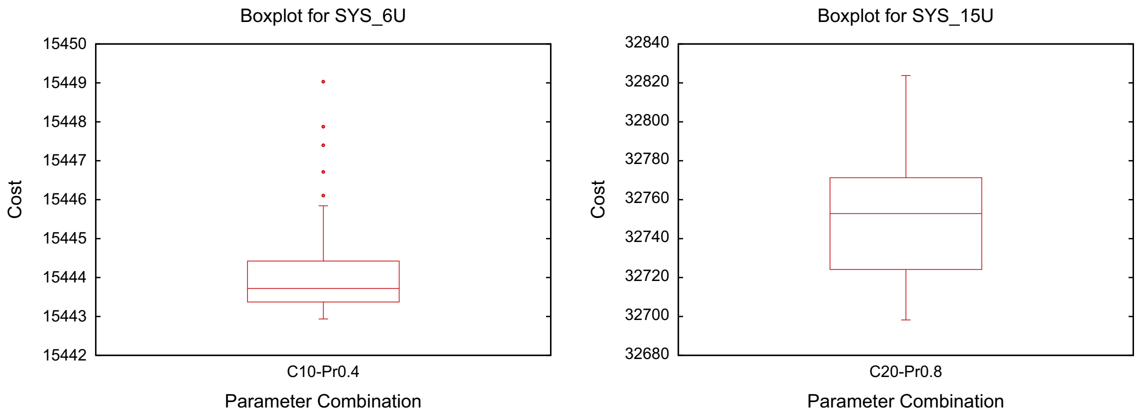

Fig. 3. Box plots for the test problems with the best parameters combination.

Table 6
IA_EDP's performance compared to that of the other approaches. *a*/*b* means *a* = number of approaches which are outperformed by IA_EDP, *b* = number of approaches which report the measure. – indicates any approaches that reported the measure.

Problem	Best	Worst	Mean	Time (s)
SYS_3U_a	6/6	–	–	1/2
SYS_18U	1/1	1/1	1/1	1/1
SYS_3U_b	8/8	–	–	3/5
SYS_13	8/12	5/12	4/12	5/6
SYS_40U	4/16	4/16	2/16	10/11
SYS_6U	12/12	8/11	9/11	3/6
SYS_15U	10/10	4/10	5/10	7/8
SYS_20U	2/2	–	–	–

Algorithm 1. IA_EDP Algorithm.

```
1: Initialize_Population();
2: Evaluate_Constraints();
3: Evaluate_Objective_Function();
4: while A predetermined number of evaluations has not been reached do
5:    Proliferation_Population();
6:    Differentiation_Population();
7: end while
8: Statistics();
```

5. Validation

5.1. Test problems

To validate our proposed approach, we tested its performance with eight test problems. Appendix A includes a detailed description of each of them while Table 1 provides their most relevant characteristics. IA_EDP was implemented in Java (version 1.6.0_24) and the experiments were performed in an Intel Q9550 Quad Core processor running at 2.83 GHz and with 4 GB DDR3 1333Mz in RAM.[4]

The required parameters for our algorithm are the following: size of population, number of objective function evaluations, and probability of application of the redistribution operators. To statistically analyze the effect of the first and third parameters on IA_EDP's behavior, we tested it with different parameters settings. Some preliminary experiments were performed to discard some values for the population size parameter. Hence, the selected parameter levels were:

- Population size (C) has four levels: 1, 5, 10 and 20 cells.
- Probability has ten levels: 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9.

[4] The source code of our proposed approach can be downloaded from http://www.lidic.unsl.edu.ar/node/442.

Table 7
Comparison of results on problems with a smooth objective function which do not consider transmission loss, rate ramp limits and prohibited zones. The best values are shown in **boldface**.

Problem/Algorithm	Best	Worst	Mean	Std.	Time (s)
SYS_3U_a					
NM	**8194.3561**		–	–	–
IEP	**8194.3561**		–	–	–
MPSO	**8194.3561**		–	–	–
IPSO	**8194.3561**		–	–	0.42
ModPSO	8194.4000	–	–	–	–
fast-CPSO	**8194.3561**	–	–	–	0.01
IA_EDP	**8194.3561**	**8194.3972**	**8194.3617**	0.0080	0.14
SYS_18U					
ICA-PSO	25430.16	25462.34	25440.89	–	18.585
IA_EDP	**25429.0192**	**25429.0234**	**25429.0202**	8.3093E−4	1.212

Table 8
Comparison of results on problems with a non-smooth objective function which do not consider transmission loss, rate ramp limits and prohibited zones. The best values are shown in **boldface**.

Problem/Algorithm	Best	Worst	Mean	Std.	Time (s)
SYS_3U_b					
GA	8237.60	–	–	–	16
EP	8234.07	–	–	–	0.09
IEP	8234.09	–	–	–	–
TM	8234.07	–	–	–	–
MPSO	8234.07	–	–	–	–
IPSO	8234.07	–	–	–	0.5
DE	8234.07	–	–	–	0.30
fast-CPSO	8234.07	–	–	–	0.02
IA_EDP	**8220.9337**	**8245.1847**	**8224.5114**	5.0812	0.188
SYS_13U					
CDE_SQP	17963.84	18152.69	17986.20	14.21	18.34
DE	17963.83	17975.36	17965.48	–	1.05
DECDM	17961.9440	18061.4110	17974.6869	20.3066	12.6
opt-aiNET	18095.4417	18345.6257	18202.8244	68.8738	–
Z-opt-aiNET	17977.0905	18266.1573	18168.6791	51.4270	–
TSARGA	17963.94	18089.61	17974.31	3.18	17.69
HMAPSO	17969.31	1799.31	17969.31	–	–
MDE	17960.39	17969.09	17967.19	–	–
SOMA	17967.4219	18017.6161	17985.3242	20.6772	–
CSOMA	**17960.3661**	17970.8323	17967.8708	0.8858	–
DHS	**17960.3661**	**17968.3610**	**17961.1226**	1.92	0.12
ICA-PSO	17960.37	17978.14	17967.94	–	9.984
IA_EDP	17961.4331	18052.3155	17980.1898	21.6666	0.876
SYS_40U					
IFEP	122624.35	125740.00	123382.00	–	1167.35
CDE_SQP	121741.9793	122839.2941	122295.1278	386.1809	14.26
DE	121416.29	121431.47	121422.72	–	–
DECDM	121423.4013	121696.9868	121526.7330	54.8617	44.3
CCPSO	121403.5362	121535.4934	121445.3269	32.4898	19.3
EDA/DE	121412.50	121517.80	121460.70	26.29	–
ARCGA	121410.1038	121536.8745	121462.1502	–	15.67
BBO	121426.66	121688.66	121508.03	–	–
DE/BBO	121420.89	121420.90	121420.90	–	60.00
TSARGA	121463.07	124296.54	122928.31	315.18	696.01
HMAPSO	121586.90	121586.90	121586.90	–	–
FAPSO-NM	121418.3	121419.8	121418.803	–	40
SOMA	121418.7856	121508.3757	121449.8796	26.8385	–
CSOMA	121414.6978	121417.8045	121415.0479	0.5598	–
DHS	**121403.5355**	**121417.2274**	**121410.5967**	4.80	1.32
ICA-PSO	121413.20	121453.56	121428.14	–	139.92
IA_EDP	121436.9729	121648.4401	122492.7018	182.5274	1.092

Thus, we have 40 parameters settings for eight problems. They are identified as C⟨size⟩–Pr⟨Prob⟩, where C and Pr indicate the population size and the probability, respectively. For each problem, 100 independent runs were performed.

To determine which parameter produces results with significant differences, we performed an analysis of variance (ANOVA) taking into account the objective function value attained by IA_EDP from each run of all the experiments performed. Thus, the hypotheses were the following:

Null Hypothesis: there is no significant difference among the means of the objective function values. If there are differences, they are due to random effects.
Alternative Hypothesis: there is a combination of factor values for which the means of the objective function values are significantly different and these differences are not due to random effects.

Results were tested with the Kolmorogov–Smirnov test to determine if they follow a normal distribution. Since that was not the case, then the Kruskal–Wallis test was applied. This test can establish whether there is a difference between two or more groups but does not say which are the specific groups that present differences. Therefore, a multiple comparison test was applied in order to know which groups are different.

This analysis proved the Null Hypothesis for several combinations of parameters. However, the Alternative Hypothesis was also proved.

In the second phase of the statistical analysis, we adopted the box plot method to visualize the distribution of the objective function values for each power system. This allowed us to determine the robustness of our proposed algorithm with respect to its parameters. Figs. B.4, B.5, B.6 and B.7 show in the x-axis the parameter combinations and the y-axis indicates the objective function values for each problem.

Thus, these two phases are meant to answer the following questions:

1. Is this probability responsible for causing the significant differences in the results?
2. Is the population size responsible for causing the significant differences in the results?
3. What are the parameters that cause greater dispersion of the results?

After this statistical analysis,[5] we can infer the following general conclusions:

- When the population size is fixed, varying the probability of application of the redistribution operator does not produce results with significant differences in most cases, for SYS_3U_a, SYS_3U_b, SYS_6U, SYS_13U, SYS_15U, SYS_18U, SYS_20U, and SYS_40U. From a total of 1440 cases, 1401 (97.29%) do not present significant differences when the population size remains fixed and the probability is increased.
- When the population size grows and the probability is fixed, the results show significant differences in SYS_3U_a, SYS_20U, SYS_6U, SYS_15U, SYS_13U, SYS_40U and SYS_18U. From a total of 480 cases, 302 (62.91%) present significant differences when the probability is fixed and the population size grows.
- Considering SYS_3U_a, even when the results for $C = 20$ present more spread than for the other population sizes, the standard deviation is less than 1. For SYS_3U_b, SYS_6U, SYS_15U and SYS_20U, the results show more spread with $C = 1$ than with any of the other population sizes. Regarding SYS_13U, even for $C = 1$, we get a few outliers, and the results show a lower median than that obtained with the other population sizes. For SYS_18U, we can see that the higher the population size, the higher the spread. For SYS_40U, we can see that the higher the population size, the higher the median.

5.2. Parameters for our proposed IA_EDP

Taking into account the previous statistical analysis, we selected a set of parameter values (see Table 2) to compare our results with those produced by other approaches. From these approaches, we adopted the lowest number of objective function evaluations (see Tables 3,4) to run our proposed IA_EDP.

6. Comparison of results and discussion

As we mentioned before, for each test problem, we performed 100 independent runs. Table 5 shows: the best, worst, mean, median and standard deviation obtained by IA_EDP. Only four decimal digits are shown due to space restrictions. Figs. 1–3 show the box plots corresponding to the best parameters settings for each problem. For all the test problems, our proposed IA_EDP found feasible solutions in all the runs performed.

Problems with a smooth objective function which do not consider transmission loss, rate ramp limits or prohibited zones, i.e., SYS_3U_a and SYS_18U, do not seem to be a challenge for our proposed IA_EDP. For this sort of problem, the standard deviations obtained by our proposed IA_EDP are lower than 1. Additionally, the problem dimensionality does not seem to

[5] For answering question 1, 1440 cases were analyzed. In this case, let $|prob|$ and $|C|$ be the number of levels for the parameters probability and population size, respectively. Thus, we have: $\binom{|prob|}{2} * |C| * |Problems| = \binom{10}{2} * 4 * 8 = 1440$. For answering question 2, 480 cases were analyzed. In this case, $|prob| * \binom{|C|}{2} * |Problems| = 10 * \binom{4}{2} * 8 = 480$.

Table 9
Comparison of results on problems with a smooth objective function which consider transmission loss, rate ramp limits and prohibited zones. [1] and [2] show the results obtained by our proposed IA_EDP when performing 6000 and 20,000 objective function evaluations, respectively. The best values are shown in **boldface**.

Problem/Algorithm	Best	Worst	Mean	Std.	Time (s)
SYS_6U					
PSO	15450.00	15492.00	15454.00	14.86	–
GA	15459.00	15469.00	15469.00	41.58	–
NPSO_LRS	15450.00	15452.00	15450.50	–	
AIS	15448.00	15472.00	15459.70	–	6.25
MTS	15450.06	15453.64	15451.17	1.29	0.93
DE	15449.77	15449.87	15449.78	–	0.03
IHS	15444.302	–	15449.865	4.5312	–
DSPSO-TSA	15441.57	15446.22	15443.84	1.07	0.37
BBO	15443.096	15443.096	**15443.0964**	–	–
DHS	15449.8996	15449.9884	15449.9264	2.04E−2	0.01
ICA-PSO	15443.24	**15444.33**	15443.97		
SA-PSO	15,447	15,455	15,447	2.528	7.58
IA_EDP	**15442.9369**	15449.0294	15444.0361	1.04109	0.796
SYS_15U					
PSO	32858.00	33331.00	33105.00	26.59	–
GA	33113.00	33337.00	33228.00	49.31	–
AIS	32854.00	32892.00	32873.25	–	10.81
DE	32588.865	32641.419	32609.851	–	1.16
CCPSO	32704.4514	**32704.4514**	**32704.4514**	0.0000	16.2
MTS	32716.87	32796.15	32767.21	3.65	17.51
MDE	32704.9	32711.5	32708.1	–	–
DSPSO-TSA	32715.06	32730.39	32724.63	2.30	8.40
SA-PSO	32708.00	32789.00	32732.00	18.025	12.79
IA_EDP[1]	32712.6325	32920.7045	32817.7285	43.3935	1.652
IA_EDP[2]	**32698.2018**	32823.7790	32750.2176	29.2989	1.628

Table 10
Comparison of results on a problem with a smooth objective function which considers transmission loss but not rate ramp limits and prohibited zones. The best values are shown in **boldface**.

Problem/Algorithm	Best	Worst	Mean	Std.	Time (s)
SYS_20U					
IA_EDP	**62466.8044**	**62528.9870**	**62487.5109**	12.0380	1.928

Table 11
Comparison of results on SYS_3U_a. The best values are shown in **boldface**.

Unit	IEP	MPSO	IPSO	Fast-CPSO	IA_EDP
1	393.170	393.170	393.170	393.170	393.2785
2	334.603	334.604	334.604	334.604	334.5255
3	122.227	122.226	122.226	122.226	122.1959
TP	850.0	850.0	850.0	850.0	850.0
TC	**8194.3561**	**8194.3561**	**8194.3561**	**8194.3561**	**8194.3561**

Table 12
Comparison of results on SYS_3U_b. The best values are shown in **boldface**.

Unit	MPSO	IPSO	DE	fast-CPSO	IA_EDP
1	300.27	300.27	300.27	300.27	349.4791
2	400.00	400.00	400.00	400.00	400.0
3	149.73	149.73	149.73	149.73	100.5208
TP	850.0	850.0	850.0	850.0	850.0
TC	8234.07	8234.07	8234.07	8234.07	**8220.9337**

affect the performance of our proposed approach either. Problems with a non-smooth objective function which do not consider transmission loss, rate ramp limits or prohibited zones, SYS_3U_b, SYS_13U and SYS_40U, seem to be more difficult for our proposed IA_EDP. In these cases, the standard deviations increase with the dimensionality of the problem, which has a negative impact on its performance.

Table 13
Comparison of results on SYS_6U. The best values are shown in **boldface**.

Unit	AIS	DSPSO-TSA	ICA-PSO	SA-PSO	IA_EDP
1	458.2904	439.2935	447.09	446.71	446.6761
2	168.0518	187.7876	173.15	173.01	172.2169
3	262.5175	261.0260	263.90	265.00	264.1762
4	139.0604	129.4973	139.05	139.00	143.6750
5	178.3936	171.7101	165.63	165.23	161.3429
6	69.3416	86.1648	86.64	86.78	87.2039
TP	1275.655	1275.514	–	1275.7	1275.2910
PL	13.1997	13.0421	–	–	12.2903
ECV	−0.5447	−0.5281	–	–	6.8281E−4
TC	15,448	15441.57	15443.24	15,447	**15442.9369**

Table 14
Comparison of results on SYS_13U. The best values are shown in **boldface**.

Unit	MDE	CSOMA	DHS	ICA-PSO	IA_EDP
1	628.318	628.3185	628.3185	628.32	628.3066
2	149.594	149.5997	149.5995	149.6	149.5246
3	222.758	222.7491	222.7491	222.75	223.1148
4	109.865	109.8666	109.8666	109.86	109.8754
5	109.864	109.8665	109.8666	109.86	109.8489
6	109.866	109.8665	109.8666	60.0000	60.0
7	109.865	109.8665	109.8666	109.87	109.8319
8	60.000	60.0000	60.0000	109.87	109.8434
9	109.866	109.8666	109.8666	109.87	109.8049
10	40.000	40.0000	40.0000	40	40.0000
11	40.000	40.0000	40.0000	40	40.0000
12	55.000	55.0000	55.0000	55	55.0
13	55.000	55.0000	55.0000	55	55.0
TP	1800.0	1800.0	1800.0	1800.0	1800.0
TC	17960.39	**17960.3661**	**17960.3661**	17960.37	17961.4331

Table 15
Comparison of results on SYS_15U. The best values are shown in **boldface**.

Unit	DE	AIS	CCPSO	DSPSO-TSA	IA_EDP
1	454.997	441.159	455.000	453.627	455.0
2	419.997	409.587	380.000	379.895	379.9999
3	129.997	117.298	130.000	129.482	130.0
4	129.998	131.258	130.000	129.923	129.9999
5	269.917	151.011	170.000	168.956	169.9999
6	459.990	466.258	460.000	45.9907	459.9999
7	429.995	423.368	430.000	42.9971	429.9999
8	60.007	99.948	71.7526	103.673	67.9628
9	25.001	110.684	58.9090	34.909	65.7269
10	63.111	100.229	160.000	154.593	156.3294
11	79.973	32.057	80.000	79.559	80.0
12	79.983	78.815	80.000	79.388	79.9999
13	25.001	23.568	25.000	25.487	25.0000
14	15.001	40.258	15.000	15.952	15.0
15	15.000	36.906	15.000	15.640	15.0000
TP	2657.966	2662.04	2660.6616	2660.96	2660.0191
PL	27.975	32.4075	30.6616	30.9520	30.0187
ECV	−0.007	−0.0035	−9.2E−14	0.01	4.5334E−4
TC	32588.87	32854.00	32,704	32715.06	**32698.2018**

For problems with a smooth objective function which consider transmission loss, rate ramp limits and prohibited zones, SYS_6U_a and SYS_15U, the standard deviations also increase with the problem dimensionality, and this has a negative effect on its performance. However, in these cases, the effect is not as negative as in the previous type of problems.

For the only problem with a smooth objective function which considers transmission loss but not rate ramp limits or prohibited zones, SYS_20U, the standard deviation is lower than 13.

Table 6 summarizes the performance of our proposed IA_EDP with respect to that of the other methods. As shown in Table 6, considering the best cost found, IA_EDP outperforms all other approaches in five cases: SYS_3U_a, SYS_18U,

Table 16
Comparison of results on SYS_18U. The best values are shown in **boldface**.

Unit	ICA-PSO	IA_EDP
1	15.0	15.0
2	45.0	45.0
3	25.0	25.0
4	25.0	25.0
5	25.0	25.0
6	4.13	4.3355
7	4.13	4.3340
8	12.28	12.28
9	12.28	12.28
10	12.28	12.28
11	12.28	12.28
12	24.0	22.8664
13	3.0	3.0
14	34.04	34.4361
15	35.35	35.7172
16	37.0	36.5985
17	36.23	36.5919
18	3.0	3.0
TP	365.0	365.0
TC	25430.16	**25429.0192**

Table 17
Comparison of results on SYS_20U. The best values are shown in **boldface**.

Unit	Lambda-iteration method	Hopfield neural network	IA_EDP
1	512.7805	512.7804	498.3856
2	169.1033	169.1035	194.5007
3	126.8898	126.8897	109.7942
4	102.8657	102.8656	100.0175
5	113.6836	113.6836	118.2894
6	73.5710	73.5709	73.8652
7	115.2878	115.2876	122.2779
8	116.3994	116.3994	119.3704
9	100.4062	100.4063	99.2393
10	106.0267	106.0267	97.9034
11	150.2394	150.2395	146.9011
12	292.7648	292.7647	298.0860
13	119.1154	119.1155	116.1543
14	30.8340	30.8342	35.6257
15	115.8057	115.8056	112.5822
16	36.2545	36.2545	36.3446
17	66.8590	66.8590	67.1374
18	87.9720	87.9720	91.2890
19	100.8033	100.8033	95.9706
20	54.3050	54.3050	59.7995
TP	2591.9670	2591.9669	2593.5349
PL	91.9670	91.9669	93.5348
ECV	−0.21	−0.21	9.7909E−5
TC	62456.6391	62456.6341	**62466.8044**

SYS_3U_b, SYS_6U and SYS_15U. Taking into account the running times, IA_EDP requires less than one second to find solutions with an acceptable quality for SYS_3U_a, SYS_3U_b, SYS_13U, SYS_6U. Additionally, it requires less than 2 s for the remaining problems.

Tables 7–10 show: the best, worst, mean, standard deviation and times in seconds of each of the considered approaches in our comparative study, including our proposed IA_EDP. Tables 11–18 show the output power for each unit, TP (total power), TC (total cost), PL (transmission loss) and ECV (violation equality constraint), if applicable. More than twenty methods are compared with respect to IA_EDP. It is worth noticing that the running time of each algorithm is affected by both the hardware environment and the software environment. That is the reason why the main comparison criterion that we adopted for assessing efficiency was the number of objective function evaluations performed by each approach. For having a fair comparison of the running times of all the algorithms considered in our study, they should all be run in the same software and hardware environment (something that was not possible in our case, since we do not have the source code of several of

Table 18
Comparison of results on SYS_40U. The best values are shown in **boldface**.

Unit	ARCGA	EDA/DE	CCPSO	CSOMA	DHS	IA_EDP
1	110.8252	111.1110	110.7998	110.8016	110.7998	111.1104
2	113.9112	110.8299	110.7999	110.8068	110.7998	110.7733
3	97.4000	97.4122	97.3999	97.4007	97.3999	97.3741
4	179.7331	179.7443	179.7331	179.7333	179.7331	179.7578
5	88.6454	88.1510	87.7999	87.8180	87.7999	96.9999
6	140.0000	139.9959	140.0000	139.9997	140.0000	139.9999
7	259.6000	259.6065	259.5997	259.6010	259.5997	259.6075
8	284.6000	284.6045	284.5997	284.6000	284.5997	284.5951
9	284.6000	284.6149	284.5997	284.6005	284.5997	284.8914
10	130.0000	130.0002	130.0000	130.0003	130.0000	130.0
11	168.7985	168.8029	94.0000	168.7999	94.0000	94.0000
12	168.7994	94.0000	94.0000	168.7999	94.0000	168.6781
13	214.7600	214.7591	214.7598	214.7599	214.7598	214.7054
14	394.2800	394.2716	394.2794	394.2794	394.2794	394.2123
15	304.5200	304.5206	394.2794	304.5196	394.2794	304.4392
16	394.2800	394.2834	394.2794	394.2794	394.2794	394.0673
17	489.2798	489.2912	489.2794	489.2796	489.2794	489.3697
18	489.2800	489.2877	489.2794	489.2795	489.2794	489.3156
19	511.2806	511.2977	511.2794	511.2794	511.2794	511.2529
20	511.2800	511.2791	511.2794	511.2796	511.2794	511.1218
21	523.2803	523.2958	523.2794	523.2797	523.2794	523.2877
22	523.2800	523.2849	523.2794	523.2798	523.2794	523.2790
23	523.2800	523.2856	523.2794	523.2801	523.2794	523.2297
24	523.2800	523.2979	523.2794	523.2795	523.2794	523.2785
25	523.2800	523.2799	523.2794	523.2797	523.2794	523.2692
26	523.2801	523.2910	523.2794	523.2799	523.2794	523.2633
27	10.0000	10.0064	10.00	10.0004	10.0000	10.0000
28	10.0000	10.0018	10.00	10.0004	10.0000	10.0000
29	10.0000	10.0000	10.00	10.0003	10.0000	10.0000
30	88.7611	96.2132	87.8000	92.7158	87.7999	88.0000
31	190.0000	189.9996	190.0000	189.9998	190.0000	190.0
32	190.0000	189.9998	190.0000	189.9998	190.0000	190.0
33	190.0000	189.9981	190.0000	189.9998	190.0000	190.0
34	164.8000	164.9126	164.7998	164.8014	164.7998	164.8390
35	164.8000	199.9941	194.3976	164.8015	200.0000	199.9999
36	164.8054	200.0000	200.0000	164.8051	194.3978	199.9999
37	110.0000	109.9988	110.0000	109.9998	110.0000	109.9999
38	110.0000	109.9994	110.0000	109.9998	110.0000	109.9999
39	110.0000	109.9974	110.0000	109.9996	110.0000	109.9999
40	511.2800	511.2800	511.2794	511.2797	511.2794	511.2805
TP	10500.0	10500.0	10500.0	10500.0	10500.0	10500.0
TC	121410.10	121412.50	**121403.54**	121414.70	**121403.54**	121436.9729

them). Clearly, in our case, the emphasis is to identify which approach requires the lowest number of objective function evaluations to find solutions of a certain acceptable quality.

However, the running times are also compared in an indirect manner, to give at least a rough idea of the complexities of the different algorithms considered in our comparative study. Only for SYS_18U, SYS_15U and SYS_20U IA_EDP found the best cost in the lowest time. For SYS_3U_a, IA_EDP spent 0.14 s to find the best solution while fast-PSO just required 0.01 s. For SYS_3U_b, IA_EDP spent 0.186 s more than fast-PSO but the solution that it found was better. For SYS_13U and SYS_EDP, only IA_EDP is outperformed by DHS.

Table 7 shows IA_EDP's behavior for SYS_3U_a. In this case, IA_EDP obtained the same results as the other approaches, while performing only one third of the objective function evaluations performed by them. In SYS_18U, IA_EDP outperformed ICA-PSO in all the measures considered.

As indicated before, problems with a non-smooth objective function which do not consider transmission loss, rate ramp limits and prohibited zones (see Table 8) are harder for our proposed IA_EDP, although in the case of SYS_3U_b, our proposed approach obtained the best results. In SYS_40U, the best approach was DHS, which performed the same number of objective function evaluations as our proposed IA_EDP but was able to obtain better results. However, our proposed approach was able to outperform IFEP, which also performed the same number of objective function evaluations. In this case, ten approaches found better solutions than our proposed IA_EDP: DE, DECDM, CCPSO, EDA/DE, ARCGA, BBO, DE/BBO, FAPSO-NM, SOMA, ICA-PSO. But, all of them required more objective function evaluations.

Table 9 shows our comparison of results for the two problems with a smooth objective function which consider transmission loss, rate ramp limits and prohibited zones. In SYS_6U, our proposed IA_EDP obtained the best solution. DPSO-TSA had apparently found better solutions than our approach, but as indicated in Table 13, the solution that it found is infeasible because ECV is lower than zero. DHS is the only approach which performed the same number of objective function evaluations as our proposed IA_EDP. Although, our proposed approach obtained better values than DHS. The other algorithms required more objective function evaluations than our proposed IA_EDP and could not outperform it with respect to the best solution found. In SYS_15U, our proposed IA_EDP also obtained the best solution. Regarding worst and mean values, CCPSO outperformed our proposed approach, but it required 1000 additional objective function evaluations. Furthermore, the best solution obtained by CCPSO is infeasible (see Table 15). In the case of DE, it apparently found a better solution than our proposed IA_EDP. Even though, as seen in Table 15, such solution is infeasible, because ECV is lower than zero. Since DSPSO-TSA only performed 6000 objective function evaluations, we ran our proposed IA_EDP for this same number of evaluations and we were able to outperform DSPSO-TSA with respect to the best solution found.

It is worth mentioning that the running time that we report for 6000 objective function evaluations is bigger than the one indicated for 20,000 objective function evaluations, which seems to be a mistake from our side. However, that is not the case. What happens is that our proposed IA_EDP spends more time looking for feasible solutions in the first case, because the redistribution power operators do not preserve the feasibility of prohibited zones. Therefore, the solutions generated by these operators were infeasible and, since our stopping criterion is the number of objective function evaluations, the algorithm simply keeps running, which explains the slightly higher running time produced in this case.

As it is shown in Table 17, the solutions found by the Lambda-iteration method and a Hopfield neural network are infeasible. Therefore, our proposed IA_EDP obtained the best results.

7. Conclusions and future work

This paper presented an algorithm inspired on the T-Cell model of the immune system, called IA_EDP, which was used to solve economic dispatch problems. The purpose of IA_EDP is to optimize a set of cells. Each of them contains decision variables which, in this case, represent the output power that has to be generated by each thermal unit from a power system. IA_EDP is able to handle the five types of constraints that are involved in an economic dispatch problem: power balance constraint with and without transmission loss, operating limit constraints, ramp rate limit constraint and prohibited operating zones.

At the beginning, the search performed by IA_EDP is based on a simple differentiation operator which takes an infeasible solution and modifies some of its decision variables by taking into account their constraint violation. Once the algorithm finds a feasible solution, redistribution power operators are applied. These operators aim to keep feasible the solutions that have been found so far, and only consider the balance power constraint. The two versions of this operator, which are included in IA_EDP are: (1) to decrease the power in one unit, and to select other units to generate the power that has been taken, and (2) to increase the power in one unit and to decrease the power from other units, so that the sum of the output power does not exceed the demand power.

Our proposed approach was validated with eight test problems having different characteristics and comparisons were provided with respect to several approaches that have been reported in the specialized literature. Our results indicated that our proposed approach produced competitive results in most cases, being able to outperform the other approaches while performing (at least in some cases), the same or a lower number of objective function evaluations than the other approaches.

As part of our future work, we are interested in redesigning the redistribution operators in order to maintain the solutions' feasibility when a problem involves prohibited operating zones. Additionally, we would like to analyze if it is possible to reduce more the number of objective function evaluations performed by our proposed approach.

Acknowledgements

The third author acknowledges support from CONACYT Project No. 221551.

Table A.19
Data for SYS_3U_a. P_{min} and P_{max} are expressed in MW. a, b and c are expressed in \$/MW^2, \$/MW and \$, respectively.

Unit	P_{min}	P_{max}	a	b	c
1	150	600	0.001562	7.92	561
2	100	400	0.001940	7.85	310
3	50	200	0.004820	7.97	78

Table A.20
Data for SYS_3U_b. P_{min} and P_{max} are expressed in MW. a, b and c are expressed in \$/MW^2, \$/MW and \$, respectively.

Unit	P_{min}	P_{max}	a	b	c	e	f
1	100	600	0.001562	7.92	561	300	0.0315
2	100	400	0.001940	7.85	310	200	0.042
3	50	200	0.004820	7.97	78	150	0.063

Table A.21
Data for SYS_6U. P_{min} and P_{max} are expressed in MW. a, b and c are expressed in \$/MW^2, \$/MW and \$, respectively. UR_i and DR_i are expressed in MW/h.

Unit	P_{min}	P_{max}	a	b	c	P_i^0	UR_i	DR_i	Prohibited zones
1	100	500	0.0070	7.0	240	440	80	120	[210,240][350,380]
2	50	200	0.0095	10.0	200	170	50	90	[90,110][140,160]
3	80	300	0.0090	8.5	220	200	65	100	[150,170][210,240]
4	50	150	0.0090	11.0	200	150	50	90	[80,90][110,120]
5	50	200	0.0080	10.5	220	190	50	90	[90,110][140,150]
6	50	120	0.0075	12.0	190	110	50	90	[75,85][100,105]

Table A.22
B's loss coefficients matrix for SYS_6U.

B00 =	0.056					
B0 =	$10^{-3}*$					
	−0.3908	−0.1297	0.7047	0.0591	0.2161	−0.6635
B =	$10^{-4}*$					
	1.7	1.2	0.7	−0.1	−0.5	−0.2
	1.2	1.4	0.9	0.1	−0.6	−0.1
	0.7	0.9	3.1	0.0	−1.0	−0.6
	−0.1	0.1	0.0	0.24	−0.6	−0.8
	−0.5	−0.6	−0.1	−0.6	12.9	−0.2
	−0.2	−0.1	−0.6	−0.8	−0.2	15.0

Table A.23
Data for SYS_13U. P_{min} and P_{max} are expressed in MW. a, b and c are expressed in \$/MW^2, \$/MW and \$, respectively.

Unit	P_{min}	P_{max}	a	b	c	e	f
1	0	680	0.00028	8.10	550	300	35
2	0	360	0.00056	8.10	309	200	42
3	0	360	0.00056	8.10	307	150	42
4	60	180	0.00324	7.74	240	150	63
5	60	180	0.00324	7.74	240	150	63
6	60	180	0.00324	7.74	240	150	63
7	60	180	0.00324	7.74	240	150	63
8	60	180	0.00324	7.74	240	150	63
9	60	180	0.00324	7.74	240	150	63
10	40	120	0.00284	8.60	126	100	84
11	40	120	0.00284	8.60	126	100	84
12	55	120	0.00284	8.60	126	100	84
13	55	120	0.00284	8.60	126	100	84

Table A.24
Data for SYS_15U. P_{min} and P_{max} are expressed in MW. a, b and c are expressed in \$/MW^2, \$/MW and \$, respectively. UR_i and DR_i are expressed in MW/h.

Unit	P_{min}	P_{max}	a	b	c	P_i^0	UR_i	DR_i	Prohibited zones
1	150	455	0.000299	10.1	671	400	80	120	–
2	150	455	0.000183	10.2	574	300	80	120	[185,225] [305,335][420,450]
3	20	130	0.001126	8.8	374	105	130	130	–
4	20	130	0.001126	8.8	374	100	130	130	–
5	150	470	0.000205	10.4	461	90	80	120	[180,200] [305,335][390,420]
6	135	460	0.000301	10.1	630	400	80	120	[230,255] [365,395][430,455]
7	135	465	0.000364	9.8	548	350	80	120	–
8	60	300	0.000338	11.2	227	95	65	100	–
9	25	162	0.000807	11.2	173	105	60	100	–
10	25	160	0.001203	10.7	175	110	60	100	–
11	20	80	0.003586	10.2	186	60	80	80	–
12	20	80	0.005513	9.9	230	40	80	80	[30,40] [55,65]
13	25	85	0.000371	13.1	225	30	80	80	–
14	15	55	0.001929	12.1	309	20	55	55	–
15	15	55	0.004447	12.4	323	20	55	55	–

Table A.25
B's loss coefficients matrix for SYS_15U.

B00 =	0.0055														
B0 =	$10^{-3}*$														
	−0.1	−0.2	2.8	−0.1	0.1	−0.3	−0.2	−0.2	0.6	3.9	−1.7	0	−3.2	6.7	−6.4
B =	$10^{-5}*$														
	1.4	1.2	0.7	−0.1	−0.3	−0.1	−0.1	−0.1	−0.3	−0.5	−0.3	−0.2	0.4	0.3	−0.1
	1.2	1.5	1.3	0	−0.5	−0.2	0	0.1	−0.2	−0.4	−0.4	0	0.4	1.0	−0.2
	0.7	1.3	7.6	−0.1	−1.3	−0.9	−0.1	0	−0.8	−1.2	−1.7	0	−2.6	11.1	−2.8
	−0.1	0	−0.1	3.4	−0.7	−0.4	1.1	5.0	2.9	3.2	−1.1	0	0.1	0.1	−2.6
	−0.3	−0.5	−1.3	−0.7	9.0	1.4	−0.3	−1.2	−1.0	−1.3	0.7	−0.2	−0.2	−2.4	−0.3
	−0.1	−0.2	−0.9	−0.4	1.4	1.6	0	−0.6	−0.5	−0.8	1.1	−0.1	−0.2	−1.7	0.3
	−0.1	0	−0.1	1.1	−0.3	0	1.5	1.7	1.5	0.9	−0.5	0.7	0	−0.2	−0.8
	−0.1	0.1	0	5.0	−1.2	−0.6	1.7	16.8	8.2	7.9	−2.3	−3.6	0.1	0.5	−7.8
	−0.3	−0.2	−0.8	2.9	−1.0	−0.5	1.5	8.2	12.9	11.6	−2.1	−2.5	0.7	−1.2	−7.2
	−0.5	−0.4	−1.2	3.2	−1.3	−0.8	0.9	7.9	11.6	20.0	−2.7	−3.4	0.9	−1.1	−8.8
	−0.3	−0.4	−1.7	−1.1	0.7	1.1	−0.5	−2.3	−2.1	−2.7	14.0	0.1	0.4	−3.8	16.8
	−0.2	0	0	0	−0.2	−0.1	0.7	−3.6	−2.5	−3.4	0.1	5.4	−0.1	−0.4	2.8
	0.4	0.4	−2.6	0.1	−0.2	−0.2	0	0.1	0.7	0.9	0.4	−0.1	10.3	−10.1	2.8
	0.3	1.0	11.1	0.1	−2.4	−1.7	−0.2	0.5	−1.2	−1.1	−3.8	−0.4	−10.1	57.8	−9.4
	−0.1	−0.2	−2.8	−2.6	−0.3	0.3	−0.8	−7.8	−7.2	−8.8	16.8	2.8	2.8	−9.4	128.3

Table A.26
Data for SYS_18U. P_{min} and P_{max} are expressed in MW. a, b and c are expressed in \$/MW^2, \$/MW and \$, respectively.

Unit	P_{min}	P_{max}	a	b	c
1	7.0	15.0	0.602842	22.45526	85.74158
2	7.0	45.0	0.602842	22.45526	85.74158
3	13.0	25.0	0.214263	22.52789	108.9837
4	16.0	25.0	0.077837	26.75263	49.06263
5	16.0	25.0	0.077837	26.75263	49.06263
6	3.0	14.75	0.734763	80.39345	677.73
7	3.0	14.75	0.734763	80.39345	677.73
8	3.0	12.28	0.514474	13.19474	44.39
9	3.0	12.28	0.514474	13.19474	44.39
10	3.0	12.28	0.514474	13.19474	44.39
11	3.0	12.28	0.514474	13.19474	44.39
12	3.0	24.0	0.657079	56.70947	574.9603
13	3.0	16.2	1.236474	84.67579	820.3776
14	3.0	36.2	0.394571	59.59026	603.0237
15	3.0	45.0	0.420789	56.70947	567.9363
16	3.0	37.0	0.420789	55965	567.9363
17	3.0	45.0	0.420789	55965	567.9363
18	3.0	16.2	1.236474	84.67579	820.3776

Table A.27
Data for SYS_20U. P_{min} and P_{max} are expressed in MW. a, b and c are expressed in \$/MW^2, \$/MW and \$, respectively.

Unit	P_{min}	P_{max}	a	b	c
1	150	600	0.00068	18.19	1000
2	50	200	0.00071	19.26	970
3	50	200	0.0065	19.8	600
4	50	200	0.005	19.1	700
5	50	160	0.00738	18.1	420
6	20	100	0.00612	19.26	360
7	25	125	0.0079	17.14	490
8	50	150	0.00813	18.92	660
9	50	200	0.00522	18.27	765
10	30	150	0.00573	18.92	770
11	100	300	0.0048	16.69	800
12	150	500	0.0031	16.76	970
13	40	160	0.0085	17.36	900
14	20	130	0.00511	18.7	700
15	25	185	0.00398	18.7	450
16	20	80	0.0712	14.26	370
17	30	85	0.0089	19.14	480
18	30	120	0.00713	18.92	680
19	40	120	0.00622	18.47	700
20	30	100	0.00773	19.79	850

Table A.28
B's loss coefficients matrix for SYS_20U.

$B = 10^{-4} *$

8.7	0.43	−4.61	0.36	0.32	−0.66	0.96	−1.6	0.8	−0.1	3.6	0.64	0.79	2.1	1.7	0.8	−3.2	0.7	0.48	−0.7
0.43	8.3	−0.97	0.22	0.75	−0.28	5.04	1.7	0.54	7.2	−0.28	0.98	−0.46	1.3	0.8	−0.2	0.52	−1.7	0.8	0.2
−4.61	−0.97	9.0	−2.0	0.63	3.0	1.7	−4.3	3.1	−2.0	0.7	−0.77	0.93	4.6	−0.3	4.2	0.38	0.7	−2.0	3.6
0.36	0.22	−2.0	5.3	0.47	2.62	−1.96	2.1	0.67	1.8	−0.45	0.92	2.4	7.6	−0.2	0.7	−1.0	0.86	1.6	0.87
0.32	0.75	0.63	0.47	8.6	−0.8	0.37	0.72	−0.9	0.69	1.8	4.3	−2.8	−0.7	2.3	3.6	0.8	0.2	−3.0	0.5
−0.66	−0.28	3.0	2.62	−0.8	11.8	−4.9	0.3	3.0	−3.0	0.4	0.78	6.4	2.6	−0.2	2.1	−0.4	2.3	1.6	−2.1
0.96	5.04	1.7	−1.96	0.37	−4.9	8.24	−0.9	5.9	−0.6	8.5	−0.83	7.2	4.8	−0.9	−0.1	1.3	0.7	1.9	1.3
−1.6	1.7	−4.3	2.1	0.72	0.3	−0.9	1.2	−0.96	0.56	1.6	0.8	−0.4	0.23	0.75	−0.56	0.8	−0.3	5.3	0.8
0.8	0.54	3.1	0.67	−0.9	3.0	5.9	−0.96	0.93	−0.3	6.5	2.3	2.6	0.58	−0.1	0.23	−0.3	1.5	0.74	0.7
−0.1	7.2	−2.0	1.8	0.69	−3.0	−0.6	0.56	−0.3	0.99	−6.6	3.9	2.3	−0.3	2.8	−0.8	0.38	1.9	0.47	−0.26
3.6	−0.28	0.7	−0.45	1.8	0.4	8.5	1.6	6.5	−6.6	10.7	5.3	−0.6	0.7	1.9	−2.6	0.93	−0.6	3.8	−1.5
0.64	0.98	−0.77	0.92	4.3	0.78	−0.83	0.8	2.3	3.9	5.3	8.0	0.9	2.1	−0.7	5.7	5.4	1.5	0.7	0.1
0.79	−0.46	0.93	2.4	−2.8	6.4	7.2	−0.4	2.6	2.3	−0.6	0.9	11.0	0.87	−1.0	3.6	0.46	−0.9	0.6	1.5
2.1	1.3	4.6	7.6	−0.7	2.6	4.8	0.23	0.58	−0.3	0.7	2.1	0.87	3.8	0.5	−0.7	1.9	2.3	−0.97	0.9
1.7	0.8	−0.3	−0.2	2.3	−0.2	−0.9	0.75	−0.1	2.8	1.9	−0.7	−1.0	0.5	11.0	1.9	−0.8	2.6	2.3	−0.1
0.8	−0.2	4.2	0.7	3.6	2.1	−0.1	−0.56	0.23	−0.8	−2.6	5.7	3.6	−0.7	1.9	10.8	2.5	−1.8	0.9	−2.6
−3.2	0.52	0.38	−1.0	0.8	−0.4	1.3	0.8	−0.3	0.38	0.93	5.4	0.46	1.9	−0.8	2.5	8.7	4.2	−0.3	0.68
0.7	−1.7	0.7	0.86	0.2	2.3	0.76	−0.3	1.5	1.9	−0.6	1.5	−0.9	2.3	2.6	−1.8	4.2	2.2	0.16	−0.3
0.48	0.8	−2.0	1.6	−3.0	1.6	1.9	5.3	0.74	0.47	3.8	0.7	0.6	−0.97	2.3	0.9	−0.3	0.16	7.6	0.69
−0.7	0.2	3.6	0.87	0.5	−2.1	1.3	0.8	0.7	−0.26	−1.5	0.1	1.5	0.9	−0.1	−2.6	0.68	−0.3	0.69	7.0

Table A.29
Data for case SYS_40U. P_{min} and P_{max} are expressed in MW. a, b and c are expressed in $/MW^2, $/MW and $, respectively.

Unit	P_{min}	P_{max}	a	b	c	e	f
1	36	114	0.00690	6.73	94705	100	0.084
2	36	114	0.00690	6.73	94705	100	0.084
3	60	120	0.02028	7.07	309.54	100	0.084
4	80	190	0.00942	8.18	369.03	150	0.063
5	47	97	0.0114	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	287.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.60	455.76	200	0.042
10	130	300	0.00605	12.9	722.82	200	0.042
11	94	375	0.00515	12.9	635.20	200	0.042
12	94	375	0.00569	12.8	654.69	200	0.042
13	125	500	0.00421	12.5	913.40	300	0.035
14	125	500	0.00752	8.84	1760.40	300	0.035
15	125	500	0.00708	9.15	1728.30	300	0.035
16	125	500	0.00708	9.15	1728.30	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	649.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.10	801.32	300	0.035
26	254	550	0.00277	7.10	801.32	300	0.035
27	10	150	0.52124	3.33	1055.10	120	0.077
28	10	150	0.52124	3.33	1055.10	120	0.077
29	10	150	0.52124	3.33	1055.10	120	0.077
30	47	97	0.01140	5.35	148.89	120	0.077
31	60	190	0.00160	6.43	222.92	150	0.063
32	60	190	0.00160	6.43	222.92	150	0.063
33	60	190	0.00160	6.43	222.92	150	0.063
34	90	200	0.0001	8.95	107.87	200	0.042
35	90	200	0.0001	8.62	116.58	200	0.042
36	90	200	0.0001	8.62	116.58	200	0.042
37	25	110	0.0161	5.88	307.45	80	0.098
38	25	110	0.0161	5.88	307.45	80	0.098
39	25	110	0.0161	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035

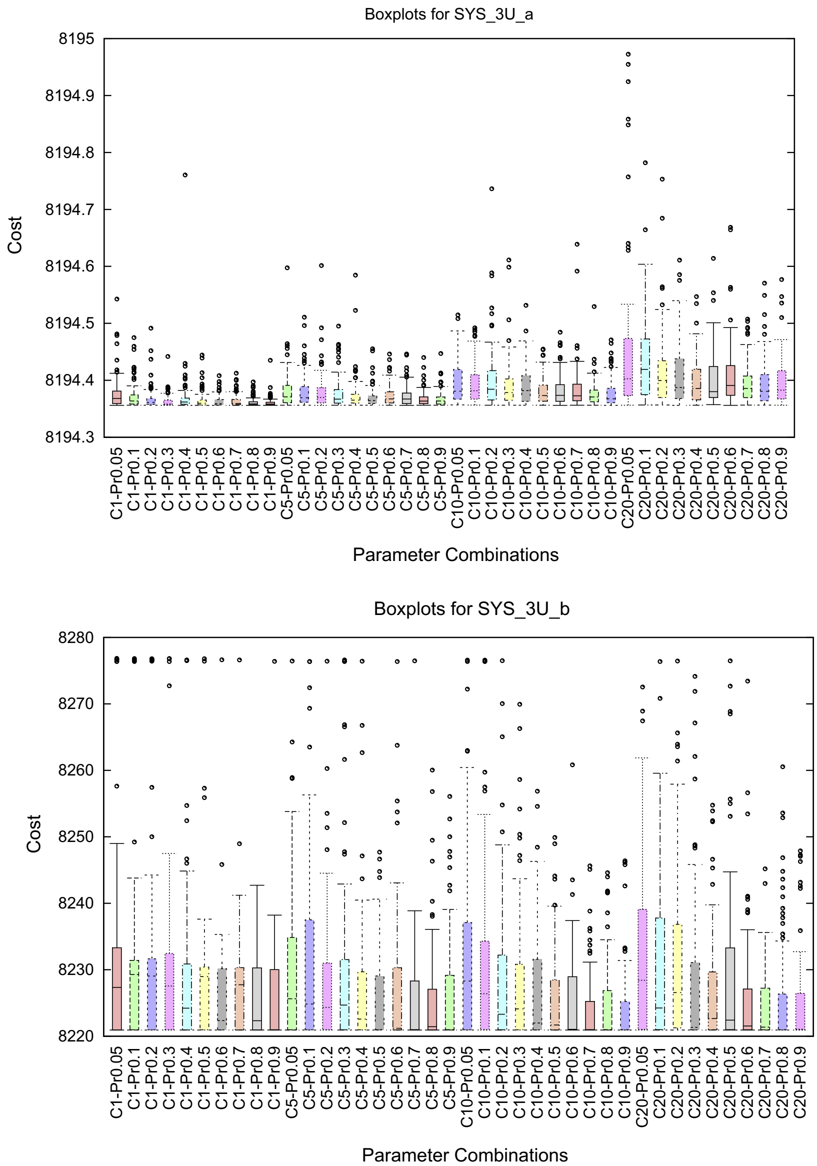

Fig. B.4. Box plots for test problems SYS_3U_a and SYS_3U_b.

Appendix A. Description of the test problems adopted

A.1. Data for SYS_3U_a

This problem comprises three generating units with a smooth cost function. Its data is given in Table A.19. The total load demand of the system is 850.0 MW.

A.2. Data for SYS_3U_b

This system comprises three generating units with quadratic cost functions together with the effects of valve-point loadings (a non-smooth cost function). Its data is given in Table A.20. The total load demand of the system is 850.0 MW.

A.3. Data for SYS_6U

The system contains six thermal units, 26 buses, and 46 transmission lines. The load demand is 1263.0 MW. The data for this problem is given in Table A.21. The loss coefficients are provided in Table A.22.

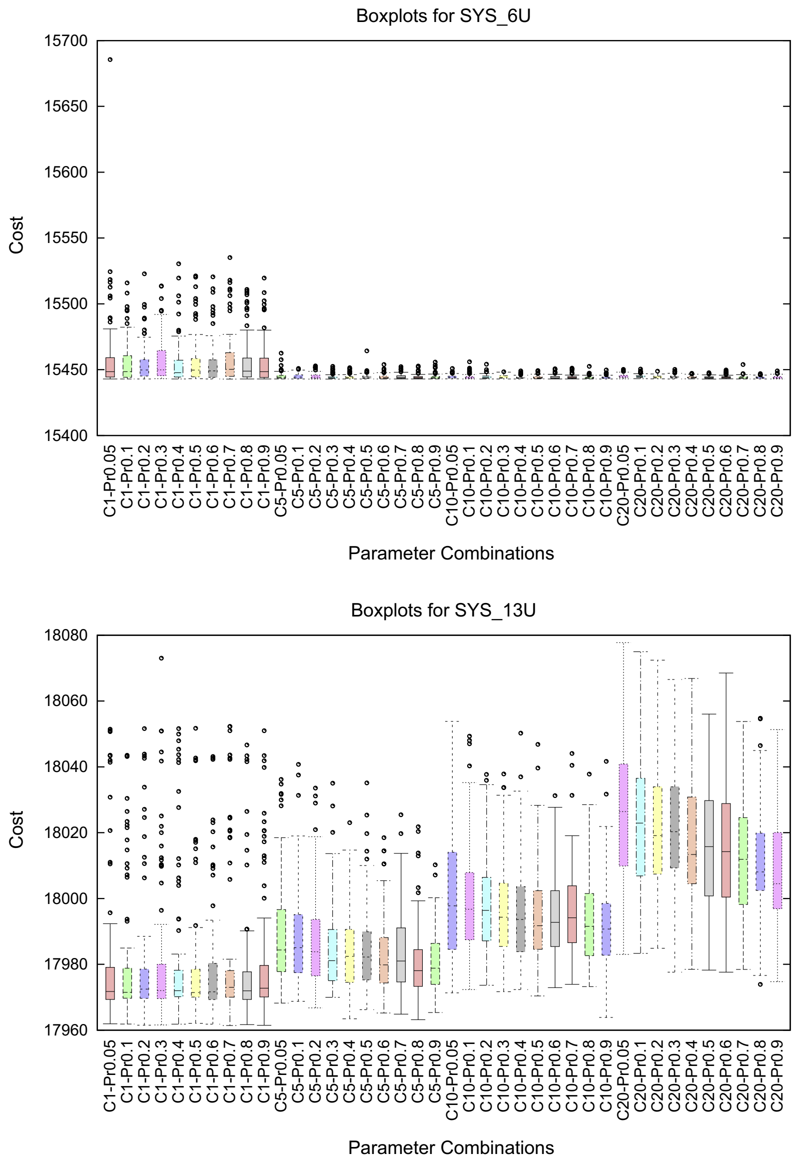

Fig. B.5. Box plots for test problems SYS_6U and SYS_13U.

A.4. Data for SYS_13U

This power system has 13 generating units. The load demand of the system is 1800.0 MW. The data for this problem is given in Table A.23.

A.5. Data for SYS_15U

This power system has 15 generating units, where four units have prohibited operating zones. The load demand is 2630.0 MW. The data for this problem is given in Table A.24. The loss coefficients are provided in Table A.25.

A.6. Data for SYS_18U

This system comprises 18 generating units with quadratic (convex) cost functions, 52 buses and 66 branches. The data is given in Table A.26. The total load demand of the system is 365.0 MW.

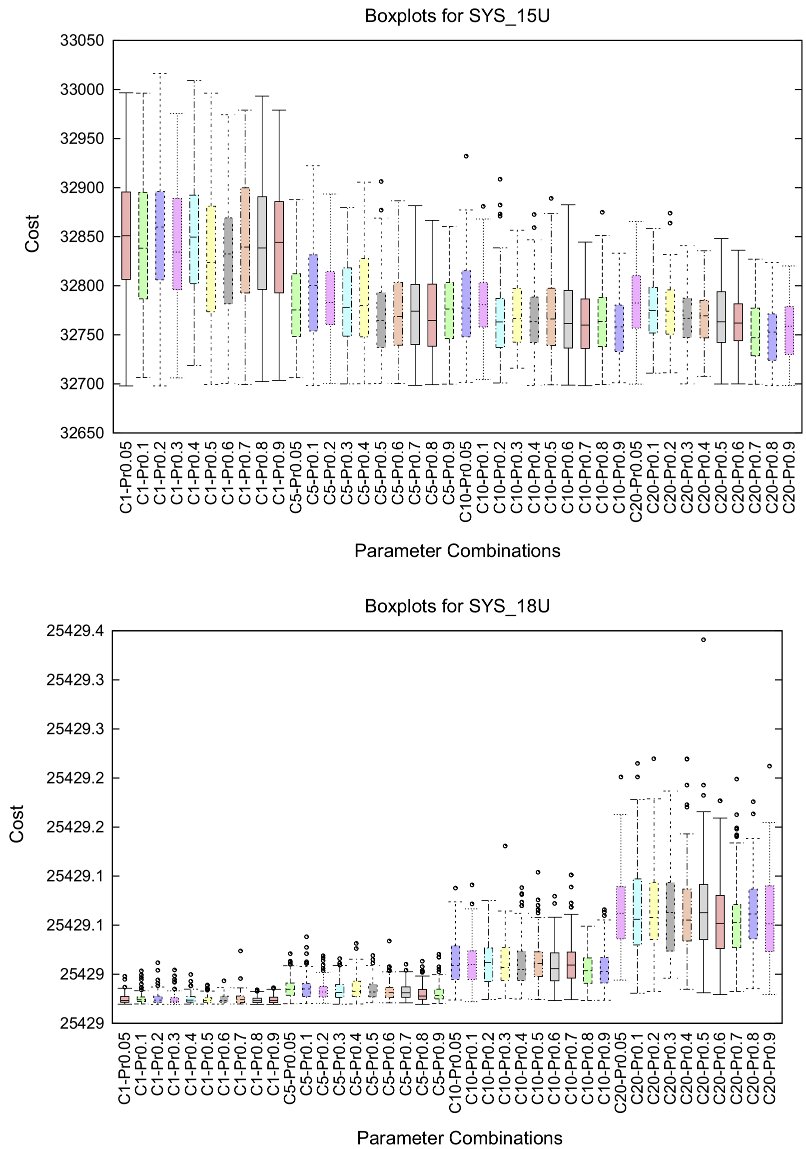

Fig. B.6. Box plots for test problems SYS_15U and SYS_18U.

A.7. Data for SYS_20U

In this power system there are 20 generating units, and the total load demand of the system is 2500.0 MW. The data is given in Table A.27. In this case, B0 and B00 present zero values. The B matrix of the transmission line loss coefficient is given in Table A.28.

A.8. Data for SYS_40U

In this power system there are 40 generating units, and the total load demand of the system is 10500.0 MW. The data is given in Table A.29.

Appendix B. Box plots

See Figs. B.4, B.5, B.6 and B.7.

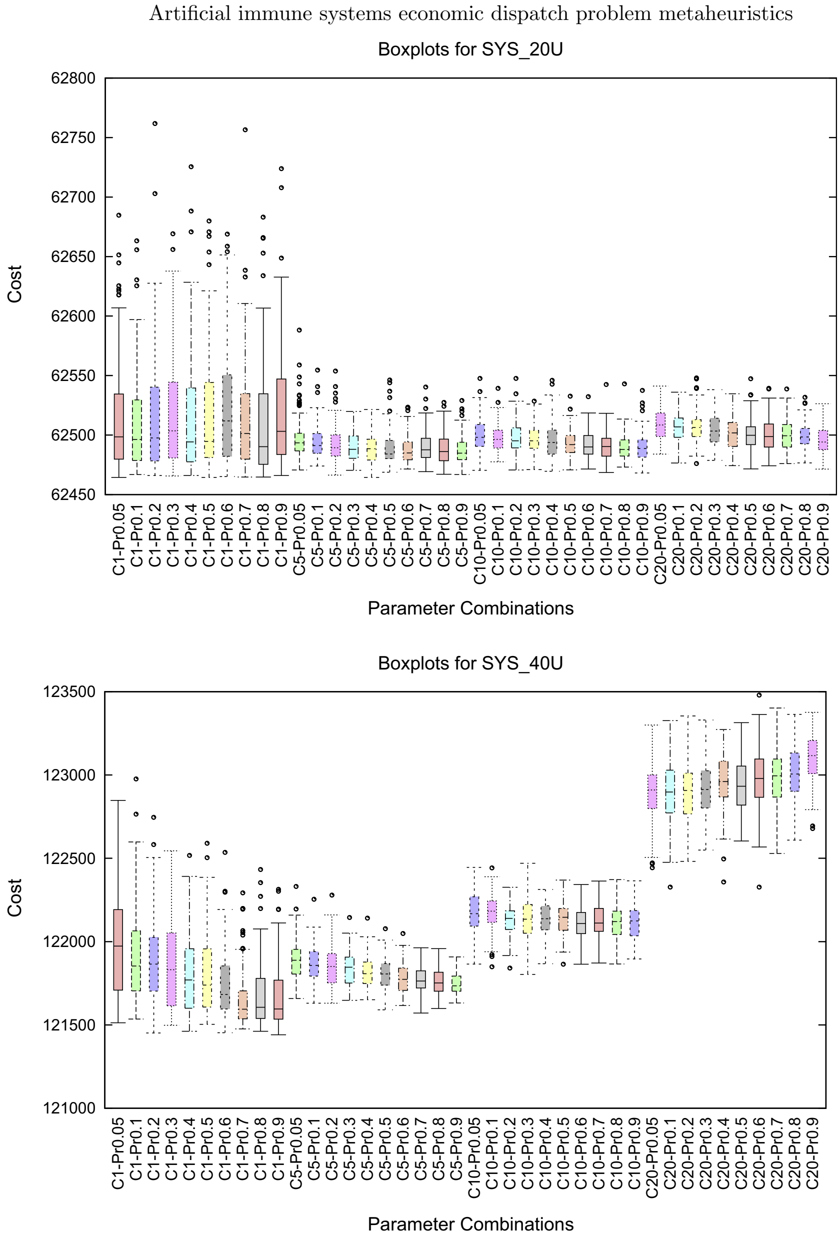

Fig. B.7. Box plots for test problems SYS_20U and SYS_40U.

References

[1] N. Amjady, H. Sharifzadeh, Solution of non-convex economic dispatch problem considering valve loading effect by a new modified differential evolution algorithm, Int. J. Electr. Power Energy Syst. 32 (8) (2010) 893–903.

[2] Nima Amjady, Hadi Nasiri-Rad, Solution of nonconvex and nonsmooth economic dispatch by a new adaptive real coded genetic algorithm, Expert Syst. Appl. 37 (7) (2010) 5239–5245.

[3] A. Bhattacharya, P.K. Chattopadhyay, Biogeography-based optimization for different economic load dispatch problems, IEEE Trans. Power Syst. 25 (2) (2010) 1064–1077.

[4] A. Bhattacharya, P.K. Chattopadhyay, Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch, IEEE Trans. Power Syst. 25 (4) (2010) 1955–1964.

[5] Leticia Cecilia Cagnina, Susana Cecilia Esquivel, Carlos A. Coello Coello, A fast particle swarm algorithm for solving smooth and non-smooth economic dispatch problems, Eng. Optimiz. 43 (5) (2011) 485–505.

[6] Po-Hung Chen, Hong-Chan Chang, Large-scale economic dispatch by genetic algorithm, IEEE Trans. Power Syst. 10 (4) (1995) 1919–1926.

[7] C.-L. Chiang, Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels, IEEE Trans. Power Syst. 20 (4) (2005) 1690–1699.

[8] L.D.S. Coelho, V.C. Mariani, Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect, IEEE Trans. Power Syst. 21 (2006) 989–996.
[9] L.d.S. Coelho, V.C. Mariani, An improved harmony search algorithm for power economic load dispatch, Energy Convers. Manage. 50 (10) (2009) 2522–2526.
[10] L.d.S. Coelho, R.C.T. Souza, V.C. Mariani, Improved differential evolution approach based on cultural algorithm and diversity measure applied to solve economic load dispatch problems, Math. Comput. Simul. 79 (10) (2009) 3136–3147.
[11] I.G. Damousis, A.G. Bakirtzis, P.S. Dokopoulos, Network-constrained economic dispatch using real-coded genetic algorithm, IEEE Trans. Power Syst. 18 (1) (2003) 198–205.
[12] Leandro dos Santos Coelho, Viviana Cocco Mariani, Chaotic artificial immune approach applied to economic dispatch of electric energy using thermal units, Chaos Solitons Fract. 40 (5) (2009) 2376–2383.
[13] Leandro dos Santos Coelho, Viviana Cocco Mariani, An efficient cultural self-organizing migrating strategy for economic dispatch optimization with valve-point effect, Energy Convers. Manage. 51 (12) (2010) 2580–2587.
[14] Attia A. El-Fergany, Solution of economic load dispatch problem with smooth and non-smooth fuel cost functions including line losses using genetic algorithm, Int. J. Comput. Electr. Eng. 3 (5) (2011) 706–710.
[15] Z.L. Gaing, Particle swarm optimization to solving the economic dispatch considering the generator constraints, IEEE Trans. Power Syst. 18 (3) (2003) 1187–1195.
[16] T. Jayabarathi, K. Jayaprakash, D.N. Jeyakumar, T. Raghunathan, Evolutionary programming techniques for different kinds of economic dispatch problems, Electr. Power Syst. Res. 73 (2) (2005) 169–176.
[17] V. Karthikeyan, S. Senthilkumar, V.J. Vijayalakshmi, A new approach to the solution of economic dispatch using particle swarm optimization with simulated annealing, Int. J. Comput. Sci. Appl. (IJCSA) 3 (3) (2013) 37–49.
[18] S. Khamsawang, S. Jiriwibhakorn, DSPSO-TSA for economic dispatch problem with nonsmooth and noncontinuous cost functions, Energy Convers. Manage. 51 (2) (2010) 365–375.
[19] S. Khamsawang, P. Wannakarn, S. Pothiya, S. Jiriwibhakorn, Solving the economic dispatch problem by using differential evolution, in: 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2009), vol. 1, 2009, pp. 212–215.
[20] C. Kumar, T. Alwarsamy, Solution of economic dispatch problem using differential evolution algorithm, Int. J. Soft Comput. Eng. (IJSCE) 1 (6) (2012) 236–241.
[21] R. Kumar, D. Sharma, A. Sadu, A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch, Int. J. Electr. Power Energy Syst. 33 (1) (2011) 115–123.
[22] Cheng-Chien Kuo, A novel coding scheme for practical economic dispatch by modified particle swarm approach, IEEE Trans. Power Syst. 23 (4) (2008) 1825–1835.
[23] J.H. Park, K.Y. Lee, A. Sode-Yome, Adaptive hopfield neural networks for economic load dispatch, IEEE Trans. Power Syst. 13 (1998) 519–525.
[24] Z.-X. Liang, J.D. Glover, A zoom feature for a dynamic programming solution to economic dispatch including transmission losses, IEEE Trans. Power Syst. 7 (2) (1992) 544–550.
[25] Derong Liu, Ying Cai, Taguchi method for solving the economic dispatch problem with nonsmooth cost functions, IEEE Trans. Power Syst. 20 (4) (2005) 2006–2014.
[26] Ravinder Singh Maan, Om Prakash Mahela, Mukesh Gupta, Economic load dispatch optimization of six interconnected generating units using particle swarm optimization, IOSR J. Electr. Electron. Eng. (IOSR-JEEE) 6 (2) (2013) 21–27.
[27] B. Mohammadi-Ivatloo, A. Rabiee, A. Soroudi, M. Ehsan, Iteration {PSO} with time varying acceleration coefficients for solving non-convex economic dispatch problems, Int. J. Electr. Power Energy Syst. 42 (1) (2012) 508–516.
[28] Taher Niknam, A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem, Appl. Energy 87 (1) (2010) 327–339.
[29] Taher Niknam, Mohammad Rasoul Narimani, Rasoul Azizipanah-Abarghooee, A new hybrid algorithm for optimal power flow considering prohibited zones and valve point effect, Energy Convers. Manage. 58 (2012) 197–206.
[30] N. Noman, H. Iba, Differential evolution for economic load dispatch problems, Electr. Power Syst. Res. 78 (8) (2008) 1322–1331.
[31] Leandro Nunes de Castro, Jonathan Timmis, Artificial Immune Systems: A New Computational Intelligence Approach, Springer-Verlag, New York, 2002.
[32] P. Oliveira, S. McKee, C. Coles, Lagrangian relaxation and its application to the unit-commitment-economic-dispatch problem, IMA J. Manage. Math. 4 (3) (1992) 261–272.
[33] V.R. Pandi, K.B. Panigrahi, M.K. Mallick, A. Abraham, S. Das, Improved harmony search for economic power dispatch, in: Ninth International Conference on Hybrid Intelligent Systems, 2009. HIS '09, vol. 3, 2009, pp. 403–408.
[34] B.K. Panigrahi, S.R. Yadav, S. Agrawal, M.K. Tiwari, A clonal algorithm to solve economic load dispatch, Electr. Power Syst. Res. 77 (10) (2007) 1381–1389.
[35] J.-B. Park, Y.-W. Jeong, J.-R. Shin, K.Y. Lee, An improved particle swarm optimization for nonconvex economic dispatch problems, IEEE Trans. Power Syst. 25 (1) (2010) 156–166.
[36] J.H. Park, Y.S. Kim, I.K. Eom, K.Y. Lee, Economic load dispatch for piecewise quadratic cost function using hopfield neural network, IEEE Trans. Power Syst. 8 (3) (1993) 1030–1038.
[37] Jong-Bae Park, Ki-Song Lee, Joong-Rin Shin, K.Y. Lee, A particle swarm optimization for economic dispatch with nonsmooth cost functions, IEEE Trans. Power Syst. 20 (1) (2005) 34–42.
[38] Y.M. Park, J.R. Won, J.B. Park, A new approach to economic load dispatch based on improved evolutionary programming, Eng. Intell. Syst. Elect. Eng. Commun. 6 (2) (1998) 103–110.
[39] S. Pothiya, I. Ngamroo, W. Kongprawechnon, Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints, Energy Convers. Manage. 49 (4) (2008) 506–516.
[40] Saravuth Pothiya, Issarachai Ngamroo, Waree Kongprawechnon, Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints, Energy Convers. Manage. 49 (4) (2008) 506–516.
[41] D.W. Ross, S. Kim, Dynamic economic dispatch of generation, IEEE Trans. Power Appar. Syst. PAS-99 (6) (2002) 2060–2067.
[42] A. Immanuel Selvakumar, K. Thanushkodi, Anti-predatory particle swarm optimization: Solution to nonconvex economic dispatch problems, Electr. Power Syst. Res. 78 (1) (2008) 2–10.
[43] A.I. Selvakumar, K. Thanushkodi, A new particle swarm optimization solution to nonconvex economic dispatch problems, IEEE Trans. Power Syst. 22 (1) (2007) 42–51.
[44] N. Sinha, R. Chakrabarti, P.K. Chattopadhyay, Evolutionary programming techniques for economic load dispatch, IEEE Trans. Evol. Comput. 7 (1) (2003) 83–94.
[45] P. Sriyanyong, Y.H. Song, P.J. Turner, Particle swarm optimisation for operational planning: unit commitment and economic dispatch, in: Keshav P. Dahal, Kay Chen Tan, Peter I. Cowling (Eds.), Evolutionary Scheduling, Studies in Computational Intelligence, vol. 49, Springer, Berlin, Heidelberg, 2007, pp. 313–347.
[46] Ching-Tzong Su, Chien-Tung Lin, New approach with a hopfield modeling framework to economic dispatch, IEEE Trans. Power Syst. 15 (2) (2000) 541–545.
[47] P. Subbaraj, R. Rengaraj, S. Salivahanan, Enhancement of self-adaptive real-coded genetic algorithm using Taguchi method for economic dispatch problem, Appl. Soft Comput. J. 11 (1) (2011) 83–92.

[48] S. Siva Subramani, P. Raja Rajeswari, A modified particle swarm optimization for economic dispatch problems with non-smooth cost functions, Int. J. Soft Comput. 3 (4) (2008) 326–332.
[49] Hafiz Tehzeeb-Ul-Hassan, Raheel Zafar, Syed Ali Mohsin, Omer Lateef, Reduction in power transmission loss using fully informed particle swarm optimization, Int. J. Electr. Power Energy Syst. 43 (1) (2012) 364–368.
[50] K. Vaisakh, P. Praveena, S. Rama Mohana Rao, Kala Meah, Solving dynamic economic dispatch problem with security constraints using bacterial foraging PSO-DE algorithm, Int. J. Electr. Power Energy Syst. 39 (1) (2012) 56–67.
[51] J.G. Vlachogiannis, K.Y. Lee, Economic load dispatch a comparative study on heuristic optimization techniques with an improved coordinated aggregation-based PSO, IEEE Trans. Power Syst. 24 (2) (2009) 991–1001.
[52] D.C. Walters, G.B. Sheble, Genetic algorithm solution of economic dispatch with valve point loading, IEEE Trans. Power Syst. 8 (3) (1993) 1325–1332.
[53] Ling Wang, Ling po Li, An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems, Int. J. Electr. Power Energy Syst. 44 (1) (2013) 832–843.
[54] Y. Wang, B. Li, T. Weise, Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems, Inf. Sci. 180 (12) (2010) 2405–2420.
[55] K.P. Wong, C.C. Fung, Simulated annealing based economic-dispatch algorithm, IEE Proc.–C Gener. Transm. Distrib. 140 (6) (1993) 509–515.
[56] Allen J. Wood, Bruce F. Wollenberg, Power Generation, Operation, and Control, second ed., Wiley-Interscience, 1984.
[57] Hong-Tzer Yang, Pai-Chuan Yang, Ching-Lien Huang, Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions, IEEE Trans. Power Syst. 11 (1) (1996) 112–118.