An Artificial Immune System for Solving Dynamic Economic Power Dispatch Problems

Victoria S. Aragón[1], Carlos A. Coello Coello[2] and Mario G. Leguizamón[1]

[1] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional***
Universidad Nacional de San Luis - Ej. de Los Andes 950
San Luis (5700), ARGENTINA
[2] CINVESTAV-IPN (Evolutionary Computation Group)†
Departamento de Computación - Av. IPN No. 2508, Col. San Pedro Zacatenco
México D.F. 07300, MÉXICO

Abstract. In this paper, we propose an artificial immune system called IA_DED, which stands for *Immune Algorithm Dynamic Economic Dispatch*. It is designed for solving the Dynamic Economic Dispatch (DED) problem. Our approach considers the DED problem as a dynamic problem whose constraints change over time. IA_DED considers the activation process that T cells suffer in order to find partial solutions. The proposed approach is validated using several DED problems taken from the specialized literature. Our results are compared with respect to those obtained by other approaches taken from the specialized literature. We also provide some statistical analysis in order to determine the sensitivity of the performance of our proposed approach to its parameters.

Keywords: Artificial immune systems, dynamic economic dispatch problem, metaheuristics

1 Introduction

Electricity generation is the process by which electrical power is generated from other sources of energy. In other words, the generation of electrical energy is done by transforming some other type of energy (chemical combustion, nuclear fission, kinetic energy of flowing water and wind, solar photo-voltaic and geothermal power, among others) into electrical energy. This transformation takes place at a power station by electromechanical generators. It constitutes the first step of the electrical supply system. Then, electrical energy is transmitted and distributed to consumers by means of specialized systems.

The demand for electrical energy from a city, region or country has a variation throughout the day. This variation depends on many factors, such as: types of existing industries in the area and shifts performed on their production, weather (extremes of heat or cold), type of appliances that are most frequently used, type

*** LIDIC is financed by Universidad Nacional de San Luis. The first and third authors acknowledges support from PROICO 31018.
† The second author acknowledges support from CONACyT project no. 1920.

of water heaters installed at homes, the season of the year and the time of day at which the energy demands are considered, among others. The generation of electrical energy should respond to the demand curve; that is, if energy demand is increased, power supply must also increase and vice versa. In the Dynamic Economic Dispatch (DED) problem a sequence of load demands has to be met by minimizing the production cost while some constraints are met.

On the other hand, any time-dependent problem can be considered as a dynamic problem. Such problems can change the objective function, the constraints or both. A change over a constraint exists when the problem conditions change (for instance, how much energy has to produce the system at one point). So, in this paper, the DED problems are considered as dynamic problems whose load demands constraint change over time in a random fashion.

The aim of this study is to assess the performance of the proposed algorithm which is designed to solve the DED problem. The proposed algorithm is able to minimize the production cost as well as the time invested to find it. Considering T load demands, the problem is regarded as a sequence of T problems. But, each problem (at time i): 1) depends on the solution produced for the previous problem (at time $i-1$) and 2) conditions its successor (at time $i+1$).

2 Problem Formulation

In the DED problem the main aim is to minimize the total production cost (TC) associated with N dispatch units for a time period:

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

where TC is the fuel cost over the whole dispatch period, $\sum_{i=1}^{N} F_i(P_i^t)$ is the fuel cost for the t^{th} interval, $P^t = (P_1^t, P_2^t, \ldots, P_N^t)$ is the power output of each unit at time t, T is the number of intervals in the period, N is the number of generators or units in the system, P_i^t is the power of the i^{th} unit at time t (in MW) and F_i is the fuel cost for the i^{th} unit (in \$/h).

The simplest fuel cost function (i.e., smooth) can be expressed as a single quadratic function: $F_i(P_i^t) = a_i(P_i^t)^2 + b_i P_i^t + c_i$, where a_i , b_i and c_i are the fuel consumption cost coefficients of the i^{th} unit. But, if the valve-point effects are taken into account, the fuel cost function becomes non-smooth and the i^{th} unit is expressed as the sum of a quadratic and a sinusoidal function in the form: $F_i(P_i^t) = a_i(P_i^t)^2 + b_i P_i^t + c_i + \mid e_i sin(f_i(P_{min_i} - P_i^t)) \mid$, where e_i and f_i are the fuel cost coefficients of the i^{th} unit with valve-point effects.

The minimization of TC is subject to:

1. Power Balance Constraint: the power generated has to be equal to the power demand required. It is defined as: $\sum_{i=1}^{N} P_i^t - P_D^t - P_L^t = 0$, where $t = 1, 2, \ldots, T$. P_D^t is the power demand at time t, and P_L^t is the transmission power loss at time t (in MW). This value considers the transmission loss due to the geographical distribution of the power stations. In this

paper, to calculate this value we use Kron's formula which represents the losses as a function of the output level of the system generators and it uses some B-matrix loss coefficients. The general form of the loss formula using B-coefficients is: $P_L^t = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i^t B_{ij} P_j^t + \sum_{i=1} B0_i P_i^t + B00$. If transmission power loss is not considered, $P_L^t = 0$.

2. Operating Limit Constraints: units have physical limits regarding the minimum and maximum power they can generate: $P_{min_i} \leq P_i^t \leq P_{max_i}$, where P_{min_i} and P_{max_i} are the minimum and maximum power output of the i^{th} unit in MW, respectively.
3. 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: $P_j^t - P_j^{(t-1)} \leq UR_j$ if $P_j^t > P_j^{(t-1)}$ and $P_j^{(t-1)} - P_j^t \leq DR_j$ if $P_j^t < P_j^{(t-1)}$, where $P_j^{(t-1)}$ is the output power of j^{th} unit at a previous hour and UR_j and DR_j are the ramp-up and ramp-down limits of the j^{th} unit in MW, respectively. Due to ramp-rate constraints, equation from item 2. is replaced by: $max(P_{min_j}^t, P_j^{(t-1)} - DR_j) \leq P_j^t$ and $P_j^t \leq min(P_{max_j}^t, P_j^{(t-1)} + UR_j)$ such that
$$\begin{cases} P_{min_j}^t = max(P_{min_j}, P_j^{(t-1)} - DR_j) \\ P_{max_j}^t = min(P_{max_j}, P_j^{(t-1)} + UR_j) \end{cases} \tag{2}$$
4. Prohibited Operating Zones: they restrict the operation of the units due to steam valve operation conditions or to vibrations in the shaft bearing. Thus, a unit with prohibited operating zones has a discontinuous input-output power generation characteristic which gives rise to additional constraints on the unit operating range. They are: $P_{min_i} \leq P_i^t \leq PZ_{i,1}^L$ or $PZ_{i,k-1}^U \leq P_i^t \leq PZ_{i,k}^L$ or $PZ_{i,n_1}^U \leq P_i^t \leq P_{max_i}$, $k = 2, 3, \ldots, n_i$, where n_i is the number of prohibited zones of the i^{th} unit, k is the index of the prohibited operating zones of the i^{th} unit. $PZ_{i,k}^L$ and $PZ_{i,k}^U$ are the lower and upper bounds of the k^{th} prohibited operating zones of unit i.

3 Literature Review

Artificial Intelligence (AI) techniques are appropiate to solve the DED problem because this is a real-world problem with several particular features that make it difficult to solve, since its nonlinear search space is nonsmooth, discontinuous and non-differentiable. In fact, if valve-point effects or prohibited zones are considered, then we are dealing with a nonconvex problem [18].

This section aims to highlight how the DED problem has been tackled using different AI techniques, rather than providing a comprehensive description of each of them. These methods include: evolutionary algorithms [18], differential evolution [7], particle swarm optimization [17], Harmony Search [13] and Artificial Immune Systems [5]. Additional techniques have been reported in [13] and [4]. Other iterative methods are reported in [6] and [12] which minimize T subproblems instead of an NT problem.

4 Our Proposed Algorithm

We propose here an adaptation from a previous algorithm presented in [1]. It is an artificial immune system designed to solve DED problems. It is based on the activation process that T cells suffer. This process is divided in two stages: proliferation and differentiation [11]. The proposed approach is called IA_DED (Immune Algorithm for Dynamic Economic Dispatch problem). It works on a cell population. Each cell is activated in order to find partial feasible solutions. Special receptors present on the cells surface, called T cell receptors (TCR) are used to represent the decision variables of the problem. In this case, each variable is a real value and it represents the output power of a thermal unit, so a TCR has N variables, for an N-unit power system.

The algorithm works in the following way (see Algorithm 1). First, the TCRs are randomly initialized within the limits of the units with real values (Step 1) (interval 1). Then, violation rate and objective function value are calculated for each cell (Step 2). Note that only if a cell is feasible, its objective function value is calculated. Next, the following steps are repeated T times (i.e. for each interval)(Step 4 to 23): while a predetermined number of objective function evaluations had not been reached and 5×10^7 iterations had not been performed, the cells are activated according to their feasibility (Step 6). Then, the best solution at time t is recorded (Step 9). The time (interval) is increased (Step 10) and new operational limits are updated according to Eq. (2) (Steps 11-14). Those units whose power outputs fall outside the new operational limits are replaced by random values from the new valid limits (Steps 15-21). Since the power outputs could change, the TPs (TP is the total power generated by a TCR) are updated and the cells are re-evaluated according to the new power demand (Step 23). Finally, (Step 25) the final solution is the union of the solutions found at times 1, 2 to T ($BEST$).

The proliferation process clones N times each cell and the differentiation process changes these clones so that they acquire specialized functional properties. The differentiation process to be applied depends on the feasibility cell.

- **Differentiation for feasible cells**
  Based on a probability P_a, each unit exchanges part of its output power with another unit from the same cell. The idea is to take a value (called d) from a unit (say i) and add it to another unit (say j). The i^{th} and j^{th} units are modified according to: $cell.TCR_i = cell.TCR_i - d$ and $cell.TCR_j = cell.TCR_j + d$, where $d = U(0, P_c * min(cell.TCR_i - P^t_{min_i}, P^t_{max_j} - cell.TCR_j))$, $U(w_1, w_2)$ refers to a real random number with a uniform distribution in the range (w_1,w_2) and P_c is a change factor ($P_c \in [0, 1]$). The best from among the clones and the original cell passes to the next iteration.
- **Differentiation for infeasible cells**
  The number of decision variables to be changed is determined by a random number $U(1, N)$. Each variable to be changed is chosen in a random way and it is modified according to: $cell.TCR_i^{'} = cell.TCR_i \pm m$,

where $cell.TCR_i$ and $cell.TCR_i^{'}$ are the original and the mutated decision variables, respectively. $m = U(0,1) * (cell.ECV + cell.ICS)$ (ECV is the equality constraint violation for TCR and ICS is the inequality constraints sum for TCR). In a random way, it is decided 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 ($cell.TCR_i^{'} = U(cell.TCR_i, P^t_{max_i})$ if m should be added or $cell.TCR_i^{'} = U(P^t_{min_i}, cell.TCR_i)$, otherwise). If the clone is feasible, then the differentiation process stops. Otherwise, the process is applied to the clone instead of the infeasible original cell. This methodology is repeated until N differentiations have been applied or a feasible clone had been reached.

ECV is calculated as at time t, for each cell j, its ECV_j is calculated as $ECV_j = \mid \sum_{i=1}^{N} TCR_i^t - P_D^t - TCR_L^t \mid$, where TCR_i^t, P_D^t and TCR_L^t are the output power for unit i, the load demand and the loss transmission, respectively. This rate indicates how far is the generated power from the demanded power. Thus, if $ECV_j > 0$ then the generated power by cell j is larger than the demanded power and if $ECV < 0$, the power generated by cell j is lower than the required power. ICS is calculated as $\sum_{i=1}^{N} \sum_{j=1}^{n_i} poz(TCR_i, i, j)$

$$poz(p,i,j) = \begin{cases} \min(p - PZ_{i,j}^L, PZ_{ij}^U - p) \; if p \in [PZ_{i,j}^L, PZ_{ij}^U] \\ 0 \qquad\qquad\qquad\qquad\qquad\quad otherwise \end{cases}$$

where n_i is the number of prohibited operating zones and $[PZ_{i,j}^L, PZ_{ij}^U]$ is the j^{th} prohibited range for the i^{th} unit. A cell is considered as feasible if: 1) $ECV = 0$ for problems without transmission network loss and $0 \leq ECV < \epsilon$ for problems with transmission loss and 2) $ICS = 0$ for problems which consider prohibited operating zones.

5 Numerical Experiments

The proposed algorithm was tested on five 24-h dynamic power systems (T=24). The first example is a 6-unit system. Its data and daily load demands were taken from [16]. The second system has 10 thermal units (10-unit system). The data and daily load demands for this problem were taken from [8]. An extension from this is the 30-unit system. It has the same cost characteristics from the last one. The load pattern is taken as three times the value which is considered in the 10 unit system for a 24hrs time period. The fourth power system has 15 generating units (15-unit system). The data and daily load demands for this problem were taken from [10]. The last test case is a 54-unit system [8]. The detailed data of this system were taken from [14], [15] and [10].

The algorithm was implemented in Java (v. 1.6.0_24) under Linux (UBUNTU 12.04) on a Pentium IV personal Computer while the experiments were performed on an Intel Q9550 Quad Core processor running at 2.83GHz and with 4GB DDR3 1333Mz in RAM. For each problem, 100 independent runs were performed.

Algorithm 1 IA_DED Algorithm

$C \leftarrow$ Initialize_Population();
Evaluate(C);
for $t \leftarrow 1$ **to** T **do**
  $top \leftarrow 0$;
  while A number of evaluations has not been reached **and** $top < 5 * 10^7$ **do**
    Activation_Process(C);
    $top++$;
  end while
  $best_t \leftarrow$ Search_best_at_Population(C);
  $t++$;
  for $j \leftarrow 1$ **to** N **do**
    $P^t_{min_j} = max(P_{min_j}, best_{t-1} - DR_j)$
    $P^t_{max_j} = min(P_{max_j}, best_{t-1} + UR_j)$
  end for
  for $i \leftarrow 1$ **to** $\mid C \mid$ **do**
    for $j \leftarrow 1$ **to** N **do**
      if $cell_i.TCR_j \notin [P^t_{min_j}, P^t_{max_j}]$ **then**
        $cell_i.TCR_j \leftarrow U(P^t_{min_j}, P^t_{max_j})$
      end if
    end for
  end for
  Update_output_power(C);
  Evaluate(C);
end for
$BEST \leftarrow (best_1, best_2, \ldots, best_T)$;

5.1 Statistical Analysis

The parameters required by IA_DED are: population size (C), maximum number of objective function evaluations, change factor (P_c), differentiation probability (P_a) and tolerance factor (ϵ). This last parameter was set to 0.9 for all the test problems that consider transmission losses. To analyze the effect of C, P_c and P_a on IA_DED's behavior, we tested it with different parameters settings. As part of this process, some preliminary experiments were performed to discard some parameter values. Hence, the selected parameter levels were the following: a) population size (C): 5, 10 and 20 cells, b) probability P_c: 0.1, 0.5 and 0.9 and c) probability P_a: 0.01 and 0.1.

As the results do not follow a normal distribution, we applied the Kruskal-Wallis test, to perform ANOVA and then the Turkey method in order to determine the experimental conditions for which significant differences exist. After the statistical analysis of the results obtained by our proposed, for the five test problems, we can infer the following general conclusions. For the 15-unit system, there are no significant differences when C is fixed and the probabilities vary. However, the median values improve with a small change factor. For the 6-unit system, when C is increased, better results are obtained and they have significant differences. Increasing the change factor from 0.1 to 0.5 and 0.9 improves the results with significant differences. For the 10-unit system, increasing the change factor from 0.1 to 0.5 and 0.9 improves the results with significant differences. When $C = 5$ or $C = 10$, increasing P_c from 0.5 to 0.9, also improves the results. In general, best median values are obtained with the highest probability set for the application of the differentiation operator. For the 30-unit system, increasing the change factor improves the results with significant differences. Contrary to the previous case, the best median values are obtained with the lowest probability established for the application of the differentiation operator. Considering the 54-unit system, for $C = 5$, increasing the change factor from 0.1 to 0.5 and 0.9 produces better results and they present significant differences. For $C = 10$ or $C = 20$, increasing the probabilities produces better results.

5.2 Comparison of Results and Discussion

Table 1 provides the most relevant features of the problems previously described as well as the maximum number of function evaluations performed by IA_DED.

Table 1. Test Problems Features

Problem	Objective	P_L	POZ	Total Load Demand (MW)	MaxEv	C	P_c	P_a
6-unit system	smooth	Yes	Yes	25954	2000	20	0.5	0.1
10-unit system	non-smooth	No	No	40108	5000	10	0.9	0.1
15-unit system	smooth	Yes	No	60981	30000	20	0.9	0.1
30-unit system	non-smooth	No	No	120324	50000	5	0.9	0.1
54-unit system	non-smooth	No	Yes	111600	40000	5	0.9	0.01

Several methods are selected to be compared with our proposed algorithm, according to the chosen test cases. Our comparison of results is presented in Table 2. It shows the best, mean, worst, standard deviation as well as the running times obtained by the approaches, when available (integer costs are shown but they are not rounded up). For all the test problems, IA_DED found feasible solutions in all the runs performed, considering the parameters settings given in Table 1, except for the 10-unit system where feasible solutions were found in 86% of the runs. The running times are compared in an indirect manner, to give a rough idea of the computational costs of the different algorithms considered in our comparative study.

Analyzing Table 2, for the 6-unit system, IA_DED exceeds by \$104 the cost found by SAMF [3], but our approach obtained this best total fuel cost just in 0.924 seconds while SAMF [3] required 1.965 seconds. For the 10-unit system, IA_DED exceeds by \$1397 the cost found by EBSO [9], but this approach reports a running time of 0.205 minutes, i.e., 12.3 seconds. The other approaches took times measured in minutes to find feasible solutions, whereas our proposed approach took only 2.552 seconds.Considering the 15-unit system IA_DED outperformed all considered approaches. It finds a solution whose total fuel cost is \$759302 in 2.660 seconds. Thus, our proposed approach found the best solution requiring the lowest running time. However, the Brent-Method [6] found an acceptable solution in only 0.53 seconds. For the 30-unit system, IA_DED obtained a best total fuel cost of \$3056592, outperforming all the approaches with respect to which it was compared, except for EBSO [9]. However, EBSO produced a solution which required 0.95 minutes (57 seconds) to obtain a solution which is only 0.08% cheaper than the one produced by IA_DED, but it required 634%

more time than IA_DED. For the 54-unit system, IA_DED outperformed all the other approaches with respect to which it was compared, in terms of the total fuel cost. IA_DED just required 13.169 seconds to find this solution, whose cost is $1717901. In this case, OCD [12], found a feasible solution which is 3% more expensive than the one produced by IA_DED but it produced it in only 0.132 seconds. It is worth noting that the methods considered in this paper, which subdivide the whole dispatch into T periods such as the Brent Method [6], SAMF [3, 12], and IA_DED, are able to find high-quality solutions in seconds rather than minutes.

6 Conclusions and Future Work

This paper presented an algorithm inspired on the T-Cells of the immune system, IA_DED, which was used to solve dynamic economic dispatch problems. IA_DED is able to handle the different types of constraints that are involved in this type of problem: power balance constraints with and without transmission loss, operating limit constraints, ramp rate limit constraints and prohibited operating zones. Additionally, it can handle both smooth and non-smooth functions.

At the beginning, the search performed by IA_DED 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, a different differentiation operator is applied. This operator modifies two decision variables at a time, it decreases the power in one unit, and it selects other unit to generate the power that has been taken.

Our proposed approach was validated with five test problems having different features. Comparisons were provided with respect to several approaches that have been reported in the specialized literature. Our proposed approach produced competitive results in all cases, being able to outperform some of the other approaches when running times are considered. The best performance of our proposed algorithm is observed in the largest systems with which it was tested. Besides, best results are obtained when the highest change factor probability is used. As part of our future work, we are interested in testing the algorithm with even larger systems and we intend to incorporate renewable energy resources.

References

1. Victoria S. Aragón and Susana C. Esquivel. Immune algorithm for solving the dynamic economic dispatch problem. In *XXII Congreso Argentino de Ciencias de la Computación*. Universidad Nacional de San Luis, 2016.
2. R. Arul, G. Ravi, and S. Velusami. Chaotic self-adaptive differential harmony search algorithm based dynamic economic dispatch. *International Journal of Electrical Power & Energy System*, 50:85–96, 2013.
3. S. Ganesan and S. Subramani. Dynamic Economic Dispatch Based on Simple Algorithm. *International Journal of Computer and Electrical Engineering*, 3(2):1793–8163, 2011.

Table 2. Comparison of results. The best values are shown in **boldface**. The last column indicates the running time (s ≡ seconds and m ≡ minutes). - denotes that the value was not available.

Problem/ Algorithm	Best($)	Mean($)	Worst($)	Std.	Time
6-unit system					
SAMF [3]	**313363**	-	-	-	1.965s
Brent Method[6]	313405	-	-	-	0.078s
BPSO-DE [16]	314025	314144	314351	-	21.89s
IA_DED	313467	**313497**	**313534**	14.58	0.924s
10-unit system					
EBSO [9]	**1017147**	**1017526**	**1017891**	147.01	0.205m
ICA [8]	1018467	1019291	1021795	-	-
CSADHS [2]	1018681	1018718	1018760	-	2.72m
ICPSO [17]	1019072	1020027	-	-	0.467m
HHS [13]	1019091	-	-	-	12.233m
CDE method3 [7]	1019123	1020870	1023115	-	0.32m
DHS [2]	1019952	1020025	1020107	-	3.34m
AIS [5]	1021980	1023156	1024973	-	19.01m
IA_DED	1018544	1020193	1022064	764.04	2.552s
15-unit system					
SAMF [3]	759406	-	-	-	2.951s
NPAHS [10]	759603	759779	759988	-	250.0s
CSADHS [2]	759689	759766	**759845**	-	3.36m
SGHS[10]	759897	760118	760343	-	303.3s
Brent Method[6]	760287	-	-	-	0.53s
IA_DED	**759302**	**759542**	760125	149.59	2.660s
30-unit system					
HHS [13]	3057313	-	-	-	27.65m
ICPSO [17]	3064497	3071588	-	-	1.03m
CDE method3 [7]	3083930	3090542	-	-	0.67m
EBSO [9]	**3054001**	**3054697**	**3055944**	-	0.95m
IA_DED	3056592	3060513	3064397	1545.83	7.756s
54-unit system					
OCD [12]	1772724	-	-	-	0.132s
ICA [8]	1807081	1809664	1811388	-	-
IA_DED	**1717901**	**1718127**	**1718411**	108.08	13.169s

4. H. Hardiansyah. A modified ABC algorithm for solving non-convex dynamic economic dispatch problems. *International Journal of Electrical and Computer Engineering (IJECE)*, 6(6):2621–2628, December 2016.
5. S. Hemamalini and S. Simon. Dynamic economic dispatch using artificial immune system for units with valve-point effect. *International Journal of Electrical Power & Energy System*, 33:868–874, 2011.
6. M. Sydulu K. Chandram, N. Subrahmany. Brent method for dynamic economic dispatch with transmission losses. *Iranian Journal Electrical and Computer Engineering*, 8(1):16–22, 2009.
7. Youlin Lu, Jianzhong Zhou, Hui Qin, Ying Wang, and Yongchuan Zhang. Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects. *Engineering Applications of Artificial Intelligence*, 24(2):378 – 387, 2011.
8. Behnam Mohammadi-ivatloo, Abbas Rabiee, Alireza Soroudi, and Mehdi Ehsan. Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch. *Energy*, 44(1):228 – 240, 2012. Integration and Energy System Engineering, European Symposium on Computer-Aided Process Engineering 2011.
9. T. Niknam and F. Golestaneh. Enhanced bee swarm optimization algorithm for dynamic economic dispatch. *IEEE Systems Journal*, 7(4):754–762, Dec 2013.
10. Qun Niu, Hongyun Zhang, Kang Li, and George W. Irwin. An efficient harmony search with new pitch adjustment for dynamic economic dispatch. *Energy*, 65:2543, 2014.
11. Leandro Nunes de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, New York, 2002.
12. Abbas Rabiee, Behnam Mohammadi-Ivatloo, and Mohammad Moradi-Dalvand. Fast Dynamic Economic Power Dispatch Problems Solution Via Optimality Condition Decomposition. *IEEE Transactions on Power Systems*, 29(2):982–983, 2014.
13. Pandi V. Ravikumar and Bijaya Ketan Panigrahi. Dynamic economic load dispatch using hybrid swarm intelligence based harmony search algorithm. *Expert Systems with Applications*, 38:8509 – 8514, 2011.
14. M. Shahidehpour. Available from: motor.ece.iit.edu/data/SCUC_118test.xls[accessed 2015].
15. M. Shahidehpour. Available from: motor.ece.iit.edu/data/118_nonsmooth.xls [accessed 2015].
16. K. Vaisakh, P. Praveena, S. Rama Mohana Rao, and Kala Meah. Solving dynamic economic dispatch problem with security constraints using bacterial foraging PSO-DE algorithm. *International Journal of Electrical Power & Energy Systems*, 39(1):56–67, 2012.
17. Wang Ying, Zhou Jianzhong, Qin Hui, and Youlin Lu. Improved chaotic particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects. *Energy Convers Manage*, 51:2893 – 900, 2010.
18. Forhad Zaman, Saber M. Elsayed, Tapabrata Ray, and Ruhul A. Sarker. Configuring two-algorithm-based evolutionary approach for solving dynamic economic dispatch problems. *Engineering Applications of Artificial Intelligence*, 53:105–125, 2016.