Solving Economic Load Dispatch Problem by Natural Computing Intelligent Systems

Richard Gonçalves, Carolina Almeida, Josiel Kuk, Myriam Delgado

Abstract—This paper presents Intelligent Systems based on Natural Computing for solving economic load dispatch problems. The proposed methods use three different natural computing techniques: Cultural Algorithms, Artificial Immune Systems and Fuzzy Inference Systems. The base for the methods is a real-coded Immune System centred on the clonal selection principle. Two Cultural variations are presented, where the operators of the Immune System are guided by knowledge accumulated through the evolutionary process. One of the cultural variations uses a Fuzzy Inference System to decide the knowledge to be applied. Three test problems are used to validate the proposed methods which are compared with state-of-the-art algorithms.


I. INTRODUCTION

Economic Load Dispatch (ELD) is one of the most important problems to be solved in the operation and planning of a power system [12]. The objective of the economic dispatch problem is to minimize the total fuel cost of thermal power plants subjected to the operating constraints. In this paper, the input-output characteristics of generating units are inherently nonlinear and highly non-smooth because of valve-point loadings. The practical ELD problem with valve-point effects is addressed as a hard optimization problem which normally renders classical optimization methods ineffective.

In the absence of exact methods, natural computing systems [1] have grown in popularity for this practical engineering problem. Specially the evolutionary methods such as Artificial Immune Systems [2], Genetic Algorithms [25] [3], Particle Swarm [29] [12] [27] [28] and Differential Evolution [23] [26]

Hybrid methods that are defined in this paper as systems in which two or more methodologies are joined to enhance the final model have attracted the attention of many researchers in the last few years. In this paper we combine an Artificial Immune Systems (AIS) with Cultural Algorithms (CA) and a Fuzzy Inference System (FIS) in order to solve the Economic Load Dispatch. Some of such hybrid approaches are supposed to be robust enough to tackle practical hard optimization problems, such as the instances of the ELD considered. The aim is to evaluate how the proposed approaches (Hybrid Intelligent Systems based on Natural Computing) solve the problem considering always the same set of parameters.

This paper is organized as described below. In Section II the Economic Load Dispatch Problem is briefly described. Section III introduces the basic concepts of the Natural Computing Systems considered here. The proposed approaches are detailed in Section IV while Section V describes the experiments and the results achieved. Finally, Section VI presents some conclusions and future works.

II. ECONOMIC LOAD DISPATCH

The primary objective of the economic load dispatch problem is to determine the optimal quantity of energy that must be generated by each unit to meet the required load demand at minimum operating cost while satisfying system equality and inequality constraints [12].

The objective function can be formulated as:

\[
\text{Minimize } F = \sum_{j=1}^{n} F_j(P_j)
\]

subject to

\[
PD = \sum_{j=1}^{n} P_j \text{ and } P_j^{\text{min}} \leq P_j \leq P_j^{\text{max}}
\]

where \(F_j(P_j)\) is the fuel cost function of the \(j\)th generator (in $/hr), \(P_j\) is the power output of the \(j\)th unit, \(n\) is the number of generating units in the system, \(PD\) is the total power demand, \(P_j^{\text{min}}\) and \(P_j^{\text{max}}\) are, respectively, the minimum and maximum power outputs of the \(j\)th unit. The equality constraint of Equation 1 is called power balance constraint while the inequality constraints are called operational constraints.

According to [12] the fuel cost function, considering valve-point effects, of each generator is given by:

\[
F_j(P_j) = a_j P_j^2 + b_j P_j + c_j + |e_j \sin(f_j (P_j^{\text{min}} - P_j))| \]

where \(a_j, b_j, c_j, e_j\) and \(f_j\) are the fuel cost coefficients of the \(j\)th unit with valve point effects.

III. NATURAL COMPUTING INTELLIGENT SYSTEMS

The next subsections present the basic concepts of the three Natural Computing techniques used in the proposed approaches: Cultural Algorithms, Artificial Immune Systems and Fuzzy Inference Systems.

A. Cultural Algorithms

Cultural algorithms (CA) are techniques derived from the cultural evolutionary process [20]. CA add domain knowledge to evolutionary computation methods and assume that such knowledge can be extracted from individuals of the population during the evolutionary process and subsequently used to guide
B. Artificial Immune Systems

The natural immune system is highly distributed, highly adaptive, self-organizing in nature, besides it maintains a memory of past encounters and has the ability to continually learn about new encounters [4]. Artificial Immune Systems inherit these features, so they can be considered useful, flexible, and powerful computational tools. According to [7], the most used biological metaphors are the negative selection, the positive selection, the immune networks, and the clonal selection principle. The last one (particularly in the Clonalag algorithm) will be briefly described since it is the basis of the AIS proposed here.

The clonal selection principle establishes the idea that only those cells that recognize the antigens are selected to proliferate. The selected cells are subject to an affinity maturation process, which improves their affinity to some specific antigens [7]. The most important algorithm based on this principle is Clonalag [15]. The Clonalag algorithm works with a population of candidate solutions (B cells or antibodies), composed of a subset of memory cells (generally the best ones) and a subset of other good individuals. At each generation some of the best individuals of the population are selected, based on their affinity measures. The selected individuals are cloned, giving rise to a temporary population of clones. The clones are subjected to a hypermutation operator, whose rate is proportional (or inversely proportional) to the affinity between the antibody (the solution) and the antigen (the problem to be solved). From this process a maturated antibody population is generated. Some of the individuals of this temporary population are selected to be memory cells or be part of the next population. This whole process is repeated until a termination condition is achieved [15].

In this paper we are particularly interested in Immune Algorithms inspired by the Clonal Selection Principle, such as the Clonalag [15] and opt-IMMALG [16]. This interest is justified by the recent proof that this class of algorithm (under certain conditions) is capable of finding the global optimum of an optimization problem [17].

C. Fuzzy Inference Systems

The fuzzy sets theory was introduced by Zadeh [9]. Fuzzy sets appeared as powerful math tools to deal with imprecise or incomplete information. The basic structure of a fuzzy inference system has three conceptual components [10]. Rules Base: contains the description of the fuzzy rules of the system; Data Base: defines the membership functions used in the fuzzy rules; Reasoning Mechanism: realizes the inference process to obtain the output or conclusion based on the known rules and the facts representing the inputs. Fuzzy systems design depends on the specification of various elements (including the number and type of the fuzzy rules, the membership functions parameters, semantic of the rules that participate in the approximated reasoning and the operators of the inference mechanism) to obtain the output from the input data [11]. There are two different ways to determine these parameters: based on the expert knowledge or by means of automatic procedures such as the genetic fuzzy systems [10]. In this paper, a Fuzzy Inference System is designed by an expert to implement the main influence function of a Cultural Algorithm.

IV. THE PROPOSED APPROACHES

A. Proposed Artificial Immune System (IS)

The Immune System (IS) proposed in this paper to solve the Economic Load Dispatch represents each antibody as a valid combination of power outputs (encoded as real numbers) for the generator units. The affinity of an antibody with the antigen is given by:

\[ off_i = (\text{Max}_{gen} C_i - C_i)/ (\text{Max}_{gen} C_i - \text{Min}_{gen} C_i) \]  

where \( C_i \) is the total fuel cost of a particular configuration (represented by the ith antibody), \( \text{Min}_{gen} C_i \) and \( \text{Max}_{gen} C_i \) are the maximum and minimum fuel cost of an antibody in generation \( gen \), respectively.

The steps of the implemented Immune System are the same as those presented in Pseudo-Code 2 (see Section IV-B) with the following modifications: steps 3, 4.8, 4.9 and 4.10 are eliminated of the algorithm and step 4.2 is substituted by “Apply a Hypermutation to the Clones”. The proposed IS can be described as follows.

Each antibody is composed of \( n \) components (power generators) that are initialized at the first generation as \( \text{ant}_{i,j} = P_{i}^{\text{min}} + U(0, 1) \times (P_{i}^{\text{max}} - P_{i}^{\text{min}}) \), where \( \text{ant}_{i,j} \) is jth component of the ith antibody (i.e., the power output of the jth unit generator), \( U(0, 1) \) is a random variable sampled from an uniform distribution in the interval \([0, 1] \), \( P_{i}^{\text{min}} \) and \( P_{i}^{\text{max}} \) are, respectively, the minimum and maximum power outputs of the jth unit. This initialization procedure guarantees that no antibody violates the operational constraints of the generator units.

After the evaluation of the initial generation, the Immune System enters its main loop (which represents the affinity maturation stage of the algorithm). This loop is repeated until a termination criterion is satisfied. The termination criterion adopted in this work is a maximum number of generations.
The affinity maturation process begins by cloning the antibodies of the past generation using the static cloning operator [13].

The hypermutation operators are subsequently applied. The implemented algorithm makes use of two hypermutation operators: an Adaptive Gaussian Hypermutation (AGH) and an Adaptive Cauchy Hypermutation (ACH). These hypermutation operators are given by Equation 4.

\[
ant_{i,j} = ant_{i,j} + \gamma \ast (C_i / MinC_{gen}) \ast R(0,1) \ast (P_{j,\max}^{\text{max}} - P_{j,\min}^{\text{min}})
\]

where \(ant_{i,j}\) is the \(j\)th component of the \(i\)th antibody, \(\gamma\) is an adaptive multiplier given by \(\gamma = e^{-gen/\varphi}\) (where \(\varphi\) is a parameter that dictates the decrease speed of the multiplier), \(P_{j,\max}^{\text{max}}\) and \(P_{j,\min}^{\text{min}}\) are the maximum and minimum limits of the \(j\)th generator unit, respectively, \(R(0,1)\) is a random number sampled according to a Gaussian or a Cauchy distribution depending on the hypermutation type (AGH or ACH). In both cases the distributions have mean equal to zero and variance equal to one.

The term \(C_i / MinC_{gen}\) makes the mutation more intensive in antibodies with a high fuel cost (low affinity) and smooth in antibodies with low fuel cost (high affinity). According to [18] Cauchy random numbers allow relatively coarse-grained steps, while Gaussian random numbers produce fine-grained steps, which in theory is a good balance. The multiplier tries to make the search in the beginning of the evolution intense and smooth at the end of the evolution.

The number of mutations applied to each antibody is given by \(M(i) = \frac{1}{P_{aff}^i} \ast n\), where \(M(i)\) is the number of mutations applied to the \(i\)th antibody, \(P_{aff}^i\) is the affinity of the \(i\)th antibody, \(\rho\) is a parameter that regulates the number of mutations and \(n\) is the number of generators.

After hypermutation a quasi-Simplex method is applied as a local search procedure to the best antibodies of the hypermutated clones.

At this point the constraints can be violated, which could cause infeasible antibodies. To avoid such violation a repair process is applied to each clone in order to guarantee that the generated antibodies are feasible. So instead of penalizing infeasible antibodies we repair them. The implemented procedure is shown in Pseudo-Code 1.

After the application of the repair procedure each hypermutated antibody is evaluated accordingly to Equation 3.

The next step in the implemented algorithm is the application of the aging operator. In this work the static pure aging operator [16] [14] is used. This aging operator eliminates old antibodies in order to maintain the diversity of the population and to avoid the premature convergence. In this operator an antibody is allowed to survive for at most \(T_B\) generations, after this period it is assumed that this antibody corresponds to a local optima and must be eliminated from the population. A clone inherits the age of its parent and is assigned an age equal to zero when it is successfully hypermutated (i.e., when the hypermutation improves the affinity of the antibody).

Finally, the last step of the affinity maturation process is the selection of the antibodies that will compose the next population. The scheme used is a \((\mu + \lambda)\)-Selection operator [16] that is applied to parents and hypermutated clones that survived after the aging operator.

B. Proposed Cultural Immune System (CIS)

Our second proposed approach uses the previously described Immune System-based algorithm as the population space of a Cultural Algorithm. Pseudo-Code 2 summarizes the general steps of the proposed algorithm.

Pseudo-Code 1: Repair Procedure

```
Repeat for each component \(j\) of an antibody \(i\)
  If \(ant_{i,j} < P_{j,\min}^{\text{min}}\) then \(ant_{i,j} = P_{j,\min}^{\text{min}}\)
  If \(ant_{i,j} > P_{j,\max}^{\text{max}}\) then \(ant_{i,j} = P_{j,\max}^{\text{max}}\)
End repeat
While \(\sum_{i=1}^{n} P_i \neq PD\)
  Randomly select a component \(j\)
  If \(\sum_{i=1}^{n} P_i < PD\)
    Add an amount to \(ant_{i,j}\) that doesn't violate the operational constraint and minimize the power balance violation
  Else
    Subtract an amount from \(ant_{i,j}\) that doesn't violate the operational constraint and minimize the power balance violation
End While
```

Pseudo-Code 2: Proposed Cultural Immune System

The Cultural Immune System (CIS) can be considered as an extension of the previous algorithm where the Belief Space and the communication protocols are added to improve the performance of the original Immune System. The belief space is used to extract information from the antibodies’ population and uses this knowledge as a guide to generate new antibodies during the hypermutation operators through the influence functions, i.e., the hypermutation operators are replaced by the influence functions. The communication protocols dictates which antibodies will be considered during the update of the belief space (acceptance function) and the probability of a knowledge stored in the belief space to influence a
hypermutation operator (main influence function). Follows a brief description of the Communication Protocols and the Belief Space.

1) Communication Protocols: Acceptance and Influence Functions: In this work a dynamic acceptance function [19] [20] is used. This acceptance function is given by:

\[ n_{Accepted} = popSize \times (\text{accept}_{perc} + \frac{\text{accept}_{perc}}{\text{gen}}) \] (5)

where \( n_{Accepted} \) are the number of antibodies of the population that will be used to update the knowledge sources, \( popSize \) is the number of antibodies in the population, \( \text{accept}_{perc} \) is a parameter that determines the percentage of accepted antibodies and \( \text{gen} \) is the current generation. This acceptance function allows more antibodies to contribute to the update during the beginning of the evolution (when there is little accumulated knowledge) and less antibodies at the end of the evolution (when most of the knowledge have already been acquired).

The main influence function is responsible for choosing (using a Roulette Wheel method [21]) the knowledge source that will influence the hypermutation operators. At the beginning, all the knowledge sources have the same probability to be applied (0.25). During the evolutionary process, the probability of a hypermutation operator being influenced by the \( k \)th knowledge source (\( \text{prob}_{KS} \)) is given by Equation 6.

\[ \text{prob}_{KS} = 0.1 + 0.6 \times \left( \frac{\text{ant}_{gbKS}}{\text{popSize}} \right) \] (6)

where \( \text{ant}_{gbKS} \) is the number of antibodies that were generated by an influence of the \( k \)th knowledge and \( \text{popSize} \) is the size of the population. Equation 6 favors the application of knowledge sources that are capable of maintaining their generated antibodies in the population and guarantees that each knowledge source has at least a 10% chance of being applied.

2) Belief Space: The Belief Space of a Cultural Algorithm is composed by its knowledge sources. The following knowledge sources are utilized in this work:

Situation Knowledge The Situation Knowledge stores the \( \text{best} \) antibodies found during the evolutionary process [14]. These antibodies are used as leaders to influence the hypermutation operators. This influence is similar to the hypermutation operation (Equation 4) but the term \( \text{ant}_{i,j} \) on the right side is substituted by the term \( \text{best}_{k,j} \), where \( \text{best}_{k,j} \) is the \( j \)th component of the \( k \)th best antibody stored in the Situation Knowledge, and \( k \) is an index randomly selected among the best antibodies.

Normative Knowledge The Normative Knowledge contains the intervals for the power outputs of the generator units where good solutions have been found, and is used to move the outputs of the new solutions towards those intervals. The intervals of the Normative Knowledge are initialized with the lower and upper bounds of the output of the generator units. The update of the normative knowledge can reduce or expand the intervals stored on it as described in [14] and [21].

The influence of the Normative Knowledge is as follows:

\[
\begin{align*}
\text{ant}_{i,j} &= \begin{cases} 
\text{ant}_{i,j} + \gamma \times C_i / \text{Min}_{C_{gen}} \times |R(0,1) \times \Delta|, & \text{if } \text{ant}_{i,j} < \text{NLo}_j \\
\text{ant}_{i,j} - \gamma \times C_i / \text{Min}_{C_{gen}} \times |R(0,1) \times \Delta|, & \text{if } \text{ant}_{i,j} > \text{NU}_j \\
\text{ant}_{i,j} + \gamma \times C_i / \text{Min}_{C_{gen}} \times |R(0,1) \times \Delta|, & \text{otherwise}
\end{cases}
\end{align*}
\] (7)

where \( \Delta = (\text{NU}_j - \text{NLo}_j) \), and \( \text{NU}_j \) and \( \text{NLo}_j \) are, respectively, the upper and lower bounds of the normative interval associated with the \( j \)th component, \( |x| \) is the absolute value of \( x \) and the other terms are as defined in Equation 4. This influence function is adaptive: it is intensive when the normative interval is large (the good interval is very uncertain) and it is smooth when the normative interval is small (the good interval is known).

Historical Knowledge This knowledge source was introduced in Cultural Algorithms as a mean to adapt to environmental changes [22]. It stores a list of the best antibodies found before the last \( HS \) environmental changes. It also stores the average direction and the average distance (size) of the changes for each component between the environmental changes. In this work, since there is no environmental change, this knowledge is adapted and it's triggered when the algorithm is trapped at a local optimum (there is no change in the best antibody found during the last \( p \) generations). The influence function of the Historical Knowledge used in this work is:

\[
\text{ant}_{i,j} = \begin{cases} 
\text{ant}_{i,j} + \gamma \times C_i / \text{Min}_{C_{gen}} \times |\text{AvD}_{t_j}|, & \text{if } \text{AvD}_{t_j} \geq 0 \\
\text{ant}_{i,j} - \gamma \times C_i / \text{Min}_{C_{gen}} \times |\text{AvD}_{t_j}|, & \text{if } \text{AvD}_{t_j} < 0
\end{cases}
\] (8)

where \( \text{AvD}_{t_j} \) is the average distance change in the \( j \)th component, \( \text{AvD}_{t_j} \) is the average direction of the change in the \( j \)th component (both are given by Equation 9) and the other terms are as previously defined.

\[
\begin{align*}
\text{AvD}_{t_j} &= \frac{\sum_{i=1}^{\text{HS}} |H_{B_{i+1,j}} - H_{B_{i,j}}|}{(\text{HS} - 1)} \\
\text{AvD}_{t_j} &= \sum_{i=1}^{\text{HS}-1} \text{Sign}(H_{B_{i+1,j}} - H_{B_{i,j}})
\end{align*}
\] (9)

where \( \text{HS} \) is the number of antibodies stored in historical knowledge, \( H_{B_{i,j}} \) is the \( j \)th component of the antibody associated with the \( i \)th environmental change and \( \text{Sign}(x) \) is a function that returns +1 if \( x \) is positive and -1 otherwise.

This influence tries to increment the \( j \)th component of the antibody submitted to hypermutation if in average the \( j \)th component of the new best antibody was greater than or equal to the \( j \)th component of the previous best antibody and tries to decrement this component otherwise. In either case the hypermutation is proportional to the average distance observed between changes (so if new best antibodies are found far away from the previous ones the hypermutation will be intensive, and it will be smooth if they are found near the previous ones).

Topographical Knowledge The Topographical Knowledge is used to create a map of the fitness landscape of the problem during the evolutionary process [21] [20]. It consists of a set
of regions and the best individual found on each region. It also stores a sorted list of the \( r \) best regions (which is sorted according to its best antibody). A region is represented as a node in a binary tree and stores the upper and lower bounds for each component and the best antibody found so far in that region.

The influence function of the Topographical Knowledge is described by Equation 10.

\[
ant_{i,j} = ant_{i,j} + \gamma \frac{C_i}{\text{Min}C_{\text{gen}}} \times R(0, 1) \times (R_{k,j}^{\text{upper}} - R_{k,j}^{\text{lower}})
\]

where \( R_{k,j}^{\text{upper}} \) and \( R_{k,j}^{\text{lower}} \) are the upper and lower bounds for the \( j \)th component in the \( k \)th region which is randomly selected according to the affinity of the best antibody of each region if a uniform random number in the interval \([0, 1]\) is less than \( p_{\text{Elite}} \) (probability of the best regions be chosen more often) and randomly selected independently of the affinity otherwise. The other terms of the equation are as previously defined. This influence function tends to explore good regions of the search space.

C. Proposed Fuzzy Cultural Immune System (FCIS)

The difference between the Cultural Immune System (CIS) and the Fuzzy Cultural Immune System (FCIS) resides in the implementation of the main influence function: in CIS Equation 6 is used to determine the probability of applying each knowledge source while in FCIS four fuzzy inference systems (FIS) are used to the same purpose (one for each knowledge source).

Each FIS has three input linguistic variables (age, diversity and quality) and one output linguistic variable (the probability of applying its knowledge source). All linguistic variables use trapezoidal membership functions. The age (initial, intermediary or final) represents the evolutionary stage of the population. The diversity (low, average and high) measures the diversity of the individuals in the antibody population. The quality (poor, average and good) measures the average affinity of the antibodies influenced by the knowledge source associated with this FIS. The probability is defined as low, average and high and is defuzzified using a centroid method.

A Mandani reasoning mechanism is used in all fuzzy inference systems. Each rule base is composed of 27 rules defined by a specialist and uses the algebraic product for the aggregation of the antecedents and rules are aggregated by the maximum.

V. EXPERIMENTS AND RESULTS

Three test cases were used to validate the proposed Natural Intelligent Systems: 13 generators with a load demand of 1800 MW [23], 13 generators with a load demand of 2520 MW and 40 generators with a load demand of 10500 MW [23].

The algorithms were executed 50 independent times with the following set of parameters (when applicable): \( \text{popSize} = 100 \), termination criterion = 3000 generations, \( \gamma = 4 \), \( \varphi = 40 \), \( \rho = 4 \), \( \tau_3 = 100 \), \( \text{pbest} = 10 \), \( J_S = 10 \), \( \rho = 200 \) and \( \text{acceptPer} = 0.2 \), which were empirically determined. It is important to point out that such parameters are maintained fixed for all the experiments reported in this section.

Although the great number of parameters, the proposed Immune Systems are little influenced by them. This fact was observed during the experiments realized to find this particular good set of parameters (the adjustment tests were done using only the first test case).

In order to validate our methodologies, we compared the proposed natural intelligent systems with other state-of-the-art approaches. Table I summarizes the results obtained for the three test cases.

<table>
<thead>
<tr>
<th>Method</th>
<th>Min Cost</th>
<th>Mean Cost</th>
<th>Max Cost</th>
<th>St. Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS</td>
<td>17972.90</td>
<td>17976.14</td>
<td>17980.84</td>
<td>5.94</td>
</tr>
<tr>
<td>CIS</td>
<td>17964.68</td>
<td>17974.28</td>
<td>17982.23</td>
<td>4.81</td>
</tr>
<tr>
<td>FCIS</td>
<td>17964.37</td>
<td>17974.50</td>
<td>17982.81</td>
<td>3.19</td>
</tr>
<tr>
<td>DEC-SQP [23]</td>
<td>17963.94</td>
<td>17973.13</td>
<td>17984.81</td>
<td>1.97</td>
</tr>
<tr>
<td>IGA [25]</td>
<td>18063.38</td>
<td>18096.40</td>
<td>18293.47</td>
<td>45.79</td>
</tr>
<tr>
<td>MPSO [27]</td>
<td>17973.34</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

For the test case of 13 generators with a load demand of 1800 MW the table shows that the proposed methods achieved similar results which are close to best results reported in the literature.

Table I also presents the comparison of the results obtained by our algorithms and other methods reported in the literature for the test case with 13 generators and a load demand of 2520 MW. For this test case the SDE algorithm achieved better results and all proposed algorithms achieved similar results.

The comparison for the test case with 40 generators and load demand of 10500 MW shows that the proposed CIS was able to attain the better value for the minimum fuel cost, while FCIS was able to attain better values for the average fuel cost and maximum fuel cost. For this instance all proposed methods achieved better results in all statistics than the methods reported in the literature.

In order to better understand the behavior of the proposed methods we calculated the 95% confidence interval for the mean fuel cost value using a BC, bootstrap method [30] with \( 10^5 \) re-samples. The intervals found are as follows:

- 13 generators and load demand of 1800MW: IS = (17974.89; 17977.70), CIS = (17973.30; 17975.55) and FCIS = (17973.85; 17975.34);
- 13 generators and load demand of 2520MW: IS = (24185.86; 24202.36), CIS = (24184.07; 24198.72) and FCIS = (24181.35; 24197.51);
• 40 generators and load demand of 105000MW: IS = (121712.85; 121759.35), CIS = (121684.24; 121750.43) and FCIS = (121632.84; 121696.65).

These confidence intervals show that the proposed methods are comparable, except for the case of 40 generators where there is no intersection between the confidence intervals of IS and FCIS (where FCIS is better than IS). Also, these intervals can be used by other researchers for future comparison [31].

VI. CONCLUSIONS AND FUTURE RESEARCHES

Our proposed natural intelligent systems achieved results comparable to those reported in the literature for the economic load dispatch problem even with no parameters optimization for each instance. For the instance with 40 generators and a load demand of 105000 MW the results compared favorably with state-of-the-art algorithms, improving the best reported minimum cost, maximum cost, mean cost and standard deviation of the costs. Because of these facts we assume that our methods can be considered among the state-of-the-art algorithms for the economic load dispatch problem.

Confidence intervals for the mean fuel cost were reported, permitting future comparison with other methods.

In future works we intend to analyze the behavior of the proposed algorithms in other variations of the economic load dispatch problem, for example, in the economic/environmental load dispatch problem and the economic load dispatch problem with security constraints.

ACKNOWLEDGMENT

Myriam Delgado acknowledges the CNPq grant 307735/2008-7 and Fundação Araucária project 233/07 - 8331 for her partial financial support. Carolina Almeida would like to thank the CNPq and Josiel Kuk acknowledges Capes for their financial support.

REFERENCES