We are IntechOpen, the world’s leading publisher of Open Access books
Built by scientists, for scientists

3,900
Open access books available

116,000
International authors and editors

120M
Downloads

154
Countries delivered to

TOP 1%
Our authors are among the most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™
Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com
Multi-Objective Optimization Techniques to Solve the Economic Emission Load Dispatch Problem Using Various Heuristic and Metaheuristic Algorithms

Jorge de Almeida Brito Júnior, Marcus Vinicius Alves Nunes, Manoel Henrique Reis Nascimento, Jandecy Cabral Leite, Jorge Laureano Moya Rodriguez, Carlos Alberto Oliveira de Freitas, Milton Fonseca Júnior, Edson Farias de Oliveira, David Barbosa de Alencar, Nadime Mustafa Moraes, Tirso Lorenzo Reyes Carvajal and Haroldo Melo de Oliveira

Abstract

The main objective of thermoelectric power plants is to meet the power demand with the lowest fuel cost and emission levels of pollutant and greenhouse gas emissions, considering the operational restrictions of the power plant. Optimization techniques have been widely used to solve engineering problems as in this case with the objective of minimizing the cost and the pollution damages. Heuristic and metaheuristic algorithms have been extensively studied and used to successfully solve this multi-objective problem. This chapter, several optimization techniques (simulated annealing, ant lion, dragonfly, NSGA II, and differential evolution) are analyzed and their application to economic-emission load dispatch (EELD) is also discussed. In addition, a comparison of all approaches and its results are offered through a case study.

Keywords: economic-emission load dispatch, optimization techniques, heuristic, metaheuristic algorithms, power plants
1. Introduction

The main problem of economic-emission load dispatch (EELD) is to reduce the emission level and total cost of generation at the same time accomplishing the demand for electricity from the power plant. Thermal power plants are among the maximum significant sources of contamination by sulfur dioxide (SO$_2$), carbon dioxide (CO$_2$), and nitrogen oxides (NOx), which create atmospheric pollution [1].

Some authors developed three approaches to solve the EELD problem [2–4]. The first one is using a single objective, considering emissions and pollution as restrictions with permissible limits [5]; the second one combines cost and emission functions into a single objective function with different weights, where cost and emission are minimized simultaneously [2]; and the third one uses the separated cost and emission functions in a multi-objective optimization [1, 6].

The solution of EELD problem is to minimize the total cost of fuel consumption and carbon emissions [7], considering power demand and operational restrictions [8]. Several techniques such as particle swarm optimization [9, 10], linear programming [11, 12], ant colony optimization [13], biogeography-based optimization [14], genetic algorithms (GA) [15, 16], Tabu search algorithm [17], simulated annealing (SA) [1], neural networks [18], differential evolution (DE) [19], harmony search algorithm [20], Lagrange functions [7], and others [19] have been used to fix the problem of EELD. In spite that all of them have been used, few are used with cost and emission functions in a multi-objective optimization.

This chapter, six multi-objective optimization approaches to solve the EELD problem are going to be presented, and a brief comparison among them is done.

In addition, the “shutdown” of the most inefficient generators is included in all multi-objective optimization approaches used [1, 8, 21].

2. Materials and methods

To solve a problem of EELD, two important objectives in an electrical thermal power system must be considered; they are environmental and economy impacts [22].

The parameters of objective functions are determined by curve fitting techniques based on tests of engine performance.

The multi-objective optimization problem is defined as:

\[
\text{minimize}_{P} \quad [F_1(P), F_2(P)]
\]

where \(F_1(P)\) and \(F_2(P)\) are the objective functions to be minimized over the set of permissible decision vector \(P\), as follow in the next Subsections 2.1 and 2.2.
2.1. Cost function ($F_1$)

The fuel cost is considered as an essential criterion for economics analysis in thermal power plants. The cost function of each generator can be assumed to be approximated by a quadratic function of generator power output $P_i$ [8, 22]:

$$F_1(P_i) = \sum_{i=1}^{n} \left( a_i + b_i P_i + c_i P_i^2 \right) \text{ \$/h}$$

where $a_i$, $b_i$, and $c_i$ are the fuel cost coefficients of the $i$th unit generating, $n$ the number of generators, and $P_i$ the active power of each generator (Table 2).

2.2. Economic emission function ($F_2$)

Emissions can be represented by a function that links emissions with power generated by each unit [23]. The emission function in ton/h, which normally represents the emission of SO$_2$ and NOx, is a function of the power output of the generator, and it can be expressed as follows [1]:

$$F_2(P_i) = \sum_{i=1}^{n} \left( d_i + e_i P_i + f_i P_i^2 \right) \text{ \text{kg/h}}$$

where $d_i$, $e_i$, and $f_i$ are emission coefficients.

2.3. Economical load dispatch constrains

The restrictions used in the problem were of three types as follows.

2.3.1. Equality power balance constrain

The real power of each generator is limited by the lower and upper limits. The following equation is the equality restriction of power balance [24, 25]:

$$\sum_{i=1}^{n} P_i - P^D - P^L = 0$$

where $P_i$ is the output power of each $i$ generator, $P^D$ is the load demand, and $P^L$ are transmission losses; in other words, the total power generation has to meet the total demand $P^D$ and the actual power losses in transmission lines $P^L$, i.e.

$$\sum_{i=1}^{n} P_i = P^D + P^L$$

The calculation of power losses $P^L$ involves the solution of the load flow problem, which has equality constraints in the active and reactive power on each bar as follows [26]:

$$P^L = \sum_{i=1}^{n} B_i P_i^2$$
A simplification is applied to model the transmission losses, setting them as a function of the generator output through Kron’s loss coefficient derivatives of the Kron formula for losses [21]:

\[ P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00}, \]  

(7)

where \( B_{ij}, B_{0i}, \) and \( B_{00} \) are the energy loss coefficients in the transmission network and \( n \) is the number of generators. A reasonable accuracy can be obtained when the actual operating conditions are close to the base case, where the \( B \) coefficients were obtained [26].

2.3.2. Production capacity constraint

The power capacity total generated from each generator is restricted by the lower limit and by the upper limit, so the constrain is [1]

\[ P_{\text{min},i} \leq P_i \leq P_{\text{max},i}, \]  

(8)

where \( P_i \) is the output power of the \( i \) generator, \( P_{\text{min},i} \) is the minimal power of the \( i \) generator, and \( P_{\text{max},i} \) the maximal power of the \( i \) generator.

2.3.3. Fuel delivery constraint

At each time interval, the amount of fuel supplied to all units must be less than or equal to the fuel supplied by the seller, i.e., the fuel delivered to each unit in each interval should be within its lower limit \( F_{\text{min},i} \) and its upper limit \( F_{\text{max},i} \) so that [21].

\[ F_{\text{min},i} \leq F_{im} \leq F_{\text{max},i}, \quad i \in N, \quad m \in M, \]  

(9)

where \( F_{im} \) is the fuel supplied to the engine \( i \) at the interval \( m \), \( F_{\text{min},i} \) is the minimum amount of fuel supplied to \( i \) generator, and \( F_{\text{max},i} \) is the maximum amount of fuel supplied to \( i \) generator.

3. Multi-objective optimization techniques to solve EELD

There are a lot of multi-objective optimization approaches that can be used to solve the EELD problem [19], but implementations need to be made to solve EELD problem. On this chapter six metaheuristic algorithm methods that already have the implementation are going to be presented.

3.1. Simulated annealing (SA)

SA is a powerful optimization technique which exploits the resemblance between a minimization process and the cooling of molten metal [1].

The physical annealing process is simulated in the SA technique for the determination of global or near-global optimum solutions for optimization problems. In this algorithm a parameter \( T \), called temperature, is defined [27].
On **Figure 1**, the simulated annealing algorithm is shown.

Starting from a high temperature, a molten metal is cooled slowly until it is solidified at a low temperature. The iteration number in the SA technique is analogous to the temperature level. In each iteration, a candidate solution is generated. If this solution is a better solution, it will be accepted and used to generate yet another candidate solution. If it is a deteriorated solution, the solution will be accepted when its probability of acceptance \( Pr(\Delta) \) as given in Eq. (10) is greater than a randomly generated number between 0 and 1 [27]:

\[
Pr(\Delta) = \exp\left(-\frac{\Delta}{T}\right)
\]

where \( \Delta \) is the amount of deterioration between the new and the current solutions and \( T \) is the temperature at which the new solution is generated. Accepting deteriorated solutions in the above manner enables the SA solution to “jump” out of the local optimum solution points and to seek the global optimum solution [1]. The last accepted candidate solution is then taken as the starting solution for the generation of candidate solutions in the next iteration. The reduction of the temperature in successive iterations is governed by the following geometric function:

\[
T_v = r^{(v-1)}T_0
\]

where \( v \) is the iteration number and \( r \) is temperature reduction factor. \( T_v \) is the initial temperature; its value can be set arbitrarily or estimated [29]. A multi-objective simulated annealing optimization to fix a generation dispatch problem is analyzed in [30], and used too in [1], but
with the implementation of turning off the most inefficient generators, and this information will be used to compare with the new methods presented in this chapter.

3.2. NSGA II

The Non-dominated Sorting Genetic Algorithm II (NSGA II) has been one of the algorithms most used to solve the multi-objective optimization (MOO) problems in general and EELD problem in particular [31–33]. It was proposed by Deb et al. in 2002 [34]. NSGA II uses a faster selection, an elitist preservation approach, and a less segmented operating parameter [35]. The operation of NSGA II is shown in Figure 2.

![Figure 2. Operation of NSGA II. Source: [35].](image)
3.2.1. Initial population

Initially, a parent random population is created. The population is ordered based on non-domination. For each solution, an aptitude equal to its level of non-domination is assigned (1 is the best level). Thus, a minimization of aptitude or fitness is assumed. Tournament selection, recombination, and mutation operators are used to create N size descendant populations [36]. The population is initialized based on the problem range and constraints, if any. This population is initialized and ordered based on non-domination.

3.2.2. Selection

In the original NSGA II, the selection by binary tournament (BTS) is used, where tournament is disputed between two solutions and the best one is selected and placed in the crossing tank. Two other solutions are taken again, and another gap in the cross junction is filled. It is performed in such a way that each solution can participate in two tournaments exactly [35].

3.2.3. Crossover

The NSGA II uses the simulated binary crossover (SBX), which works with two parent solutions and generates two descendants. The step-by-step procedure is described in [35].

3.2.4. Mutation

NSGA II uses polynomial mutation, which transforms each solution separately, that is, a stock solution gives offspring after being mutated. The mathematical formulation can be described as in [37].

3.2.5. Crowded tournament selection

An estimate obtained of the density of solutions close to a particular solution \( i \) in the population, the average between the two solutions on both sides of solution \( i \) together with each one of the targets is taken. This amount is the Crowding distance [35].

The NSGA II presents an improved technique for the maintenance of the diversity of solutions, proposing a method of crowding classification distance. However, in its case the classification technique remained unchanged from its previous version.

In NSGA II, all individuals are classified into combined populations (parents and descendants) based on the Pareto dominance ratio and then classified into several layers based on the front to which an individual is located. At each front, the individuals are arranged in descending order of magnitude of the crowding distance value. In the binary tournament selection process, the algorithm first selects an individual positioned on a better non-dominated front. In cases where the individuals with an identical front are compared, the tournament selection operator chooses the winner based on the crowding distance value [38].

The elitist strategy of NSGA II considers all combinations of non-dominated population solutions as the candidates for solutions to the next generation. If the number of non-dominated
solutions is less than the size of the population, they are all maintained as the next-generation solutions. Otherwise, candidates for the next generation are selected by the criterion of crowding distance. This criterion has the advantage of maintaining the diversity of solutions in the population in order to avoid premature convergence [38, 39].

3.3. Dragonfly

Recently new stochastic optimization technique was developed by Mirjaili in 2016 [40] named dragonfly optimizer. Dragonflies (Odonata) are fancy insects. There are nearly 3000 different species of this insect around the world. A dragonfly’s lifecycle includes two main milestones: nymph and adult. They spend the major portion of their lifespan in nymph, and they undergo metamorphosis to become adult [41].

On Table 1 and Figure 3, the characteristics of dragonfly algorithm (DA) are shown.

Dragonflies are considered as small predators that hunt almost all other small insects in nature. Nymph dragonflies also predate on other marine insects and even small fishes. The interesting fact about dragonflies is their unique and rare swarming behavior. Dragonflies swarm for only two purposes: hunting and migration. The former is called static (feeding) swarm, and the latter is called dynamic (migratory) swarm [40].

The main inspiration of the DA algorithm originates from static and dynamic swarming behaviors. These two swarming behaviors are very similar to the two main phases of optimization using meta-heuristics: exploration and exploitation. Dragonflies create sub-swarms and fly over different areas in a static swarm, which is the main objective of the exploration phase. In the static swarm, however, dragonflies fly in bigger swarms and along one direction, which is favorable in the exploitation phase [40]. The dragonfly algorithm has been applied successfully to the environmental economic dispatch.

<table>
<thead>
<tr>
<th>General algorithm</th>
<th>Dragonfly algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision variable</td>
<td>Dragonfly’s position in each dimension</td>
</tr>
<tr>
<td>Solution</td>
<td>Dragonfly’s position</td>
</tr>
<tr>
<td>Old solution</td>
<td>Old position of a dragonfly</td>
</tr>
<tr>
<td>New solution</td>
<td>New position of a dragonfly</td>
</tr>
<tr>
<td>Best solution</td>
<td>Any dragonfly with the best fitness</td>
</tr>
<tr>
<td>Fitness function</td>
<td>Distance from the food source, the predator, center of the swarm, velocity matching, and collision avoidance</td>
</tr>
<tr>
<td>Initial solution</td>
<td>Randomly selected position of a dragonfly</td>
</tr>
<tr>
<td>Selection</td>
<td>—</td>
</tr>
<tr>
<td>Process of generating new solution</td>
<td>Flying with a specific velocity and direction</td>
</tr>
</tbody>
</table>

Source: [42].

Table 1. Characteristics of the DA.
3.4. Particle swarm optimization (PSO)

Particle swarm optimization (PSO) is a population-based stochastic optimization technique, inspired by the social behavior of birds or shoals of fishes. Moore and Chapman extended this idea for multi-objective optimization in 1999 [44].
The simple PSO cannot be applied directly to multi-objective optimization since there are two issues to consider when extending PSO optimization to multi-objective problems [44]. The first one is how to select the best global and local particles (leaders) to guide the search, and the second one is how to keep good points found so far. For the latter, a secondary population is usually used to maintain non-dominated solutions. PSO is one of the modern heuristic algorithms suitable for solving large-scale non-converting optimization problems. It is a population-based search algorithm and parallel search using a group of particles [45].

On Figure 4, the PSO flow chart is shown. The main idea of the PSO is that “particles” (solutions) move through the search space with velocities that are adjusted dynamically according to their historical behavior. Therefore, the particles have a tendency to move to a better search area throughout the search process [47].

3.5. Differential evolution (DE)

The DE algorithm was developed to be a promising heuristic optimization algorithm for numerical optimization problems. The DE was designed to meet the requirement of practical minimization techniques with consistent convergence to the global minimum in consecutive independent trials. It can solve non-differentiable, nonlinear functions and multimodal cost functions [48].

The DE algorithm is a simple stochastic optimization strategy. It uses a voracious and less stochastic approach with floating-point coding in problem solving, unlike other evolutionary algorithms [49]. DE uses the arithmetic operators to evolve from a randomly generated initial population to a final solution. Basically, the weighted difference between two individuals is added to a third individual in the population. Thus, no separate probability distribution has to be used, which makes the scheme completely self-organized [49]. There are several variant strategies of DE. In general expressions are divided into two parts: the first part represents a vector to be disturbed. The first part is “rand” (vector chosen at random) or “best” (best vector of the current population). The second part is the number of difference vectors (one or two) chosen for perturbation vectors of the first part, and the last part indicates the type of crossover to be used. The type of crossing can be “bin” (binomial) or “exp” (exponential) [49].

3.5.1. Initialization

The initialization of DE can be done by randomly generating candidate solutions with \( NP \) \( D \)-dimensional vectors of real parameters evaluated [50]. There are other approaches to determining the initial population, although random equality is the most common. In current optimization problems, a possible solution parameter is added to the initial population in order to improve convergence.

3.5.2. Mutation

Differential mutation adds a scale, random sampling, of a difference vector to a third vector. Mutant vectors, also called donors, are obtained through the differential mutation operation [50].
3.5.3. Crossover

The crossover increases the potential diversity of a population. In the case of binomial crossover, test vectors are produced. Crossover can be understood as a mutation rate or a probability of inheritance between successive generations. There are alternatives to binomial crossover. The most common is the exponential crossover. Both approaches are valid for all problems, although the success/improvement of one over the other varies according to the problem considered [50].

3.5.4. Selection

Selection can be understood as a form of competition, in line with many examples directly observable in nature. Many evolutionary optimization schemes, such as DE or genetic algorithms (GAs), use some form of selection.
For a selection operation, the selection of pairs, also called avid selection or elite selection, is constantly used in the algorithm. As a stopping criterion, a maximum number of generations is defined [50].

### 3.6. Ant lion

Ant lion optimizer (ALO) [51] is a new algorithm inspired by nature proposed by Seyedali Mirjalili in 2015. The ALO algorithm mimics the mechanism of ant hunting in nature. It uses five main stages of prey hunting, such as random walking of ants, building traps, trapping ants in traps, catching prey, and rebuilding traps. All these steps are implemented.

The ant lions belong to the class of insects with wings and nerves (Neuroptera). The life cycle of the ants includes two main phases: larvae and adults. A natural shelf life can take up to 3 years, which occurs mainly in larvae (3–5 weeks into adulthood). The ant lion undergoes a metamorphosis into a cocoon to become an adult.

They hunt primarily on larvae, and the adult period is for breeding. A larva of ant lion digs a well of cones into sand, moving along a circular path and throwing the sands with the massive jaw. After digging the trap, the larvae hide under the bottom of the cone and wait for the insects (preferably ants) to be trapped in the well. The edge of the cone is sharp enough so that the insects fall easily into the bottom of the trap. Once that ALO realizes that a prey is in the trap, it tries to catch it. This is one of the algorithms that are also used for EELD, and it is one of the most recent discoveries [52–54].

The ALO is governed by the following rules [51]:

1. Ants move around the search space using different random walks.
2. Random walks are affected by the traps of ant lions.
3. Ant lions can build pits proportional to their fitness (the higher the fitness, the larger the pit).
4. Size of the pits is proportional to the probability of catching prey. Hence, ant lions with larger pits have higher probability to catch ants.
5. Each ant can be caught by an ant lion as well as the elite (fittest ant lion) in each iteration.
6. When ants try to escape from the pit, the ant lions throw sand toward the top of the trap to slide the ants inside the bottom of the trap. In order to simulate this behavior of sliding ants toward ant lions, the range of random walk is decreased adaptively.
7. If an ant becomes fitter than an ant lion, this means that it is caught and pulled under the sand by the ant lion.
8. After each hunt, an ant lion repositions itself to the latest caught prey and builds a pit to improve its chance of catching another prey.

### 4. Results of the comparison among all approaches applied to a case study

The power plant selected for the case study consists of 10 gas engines Jenbacher J620. In order to take the data of the power plant, its 10 motors were placed to work at different powers
(20, 30, 40, 50, 60, 70, 80, and 100% of the total power). For each one of these regimes, the data of fuel consumption of each engine were taken, and then the curve of power vs. cost of fuel for each motor was plotted. For clarity it is convenient to say that each consume data was measured 10 times and was chosen the main. In a similar way, it was done for the emission case. The engines were set to work with different power values, the data of the emissions of NO2, CO, CO2, SO2, etc., were collected, besides the total volume of the emissions of each engine, and the curves of power vs. emissions were also plotted. In this section, comparisons of results obtained with different approaches for the same case study are presented. There are many algorithms used for the economic environmental load dispatch [55]. Results are compared using the following algorithms:

- NSGA II
- Dragonfly optimizer
- Particle swarm optimization
- Simulated annealing
- Ant lion optimizer

The comparison was made with the following data:

- Power demand, 20 MW
- Number of power plant generators, 10
- Minimum power of generators, 0.56 MW
- Maximum power of generators, 3.9 MW

All algorithms were programmed in MATLAB to simulate and compare them with the pre-defined case studies. Besides, the shutdown of less efficient motors in all these metaheuristic algorithms was implemented.

### 4.1. Motor data

**Table 3** shows the emission coefficients of 10 gas engines of the power plant used as a case study. The coefficients “a,” “b,” “c,” “d,” “e,” and “f” were determined by operating the engines of the power plant at different powers, measuring the consumption and emissions to obtain the power versus cost and power vs. emission curves of each engine. The quadratic equation of each curve of each motor was obtained by the regression methods using MATLAB’s tool box curve fitting. In the same way, the coefficients “d,” “e,” and “f” were obtained but in this case by measuring the CO2, NOx, and SO2 emissions of each engine at different powers.

On **Table 4**, the coefficients of losses of the ten motors that make up the case study are presented. In this chapter a reduction to the transmission loss model is applied as a function of the output of the generators through the Kron loss coefficients [26].

On **Table 5** cost and emission results with all metaheuristic algorithms used for comparison in this chapter are shown. Among these algorithms, applied in a plant with ten generators,
it is possible to notice that the SA has the lowest emission of pollutants with 1757.39 (m$^3$/h), however with a cost of 1556.61 ($/h), while the DE algorithm has the lowest cost with 1545.59 ($/h), however with emission of pollutants with 1769.48 (m$^3$/h).

The comparison of all the results of the different metaheuristic algorithms is provided in Table 6.
Table 4. Loss coefficients (all values have to be multiplied by 1e-4).

<table>
<thead>
<tr>
<th>M</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.14</td>
<td>0.17</td>
<td>0.15</td>
<td>0.19</td>
<td>0.26</td>
<td>0.22</td>
<td>0.34</td>
<td>0.38</td>
<td>0.43</td>
<td>0.45</td>
</tr>
<tr>
<td>2</td>
<td>0.17</td>
<td>0.6</td>
<td>0.13</td>
<td>0.16</td>
<td>0.15</td>
<td>0.2</td>
<td>0.23</td>
<td>0.56</td>
<td>0.23</td>
<td>0.51</td>
</tr>
<tr>
<td>3</td>
<td>0.15</td>
<td>0.13</td>
<td>0.65</td>
<td>0.17</td>
<td>0.24</td>
<td>0.19</td>
<td>0.25</td>
<td>0.38</td>
<td>0.43</td>
<td>0.45</td>
</tr>
<tr>
<td>4</td>
<td>0.19</td>
<td>0.16</td>
<td>0.17</td>
<td>0.71</td>
<td>0.3</td>
<td>0.25</td>
<td>0.43</td>
<td>0.56</td>
<td>0.23</td>
<td>0.51</td>
</tr>
<tr>
<td>5</td>
<td>0.26</td>
<td>0.15</td>
<td>0.24</td>
<td>0.69</td>
<td>0.32</td>
<td>0.21</td>
<td>0.37</td>
<td>0.42</td>
<td>0.48</td>
<td>0.48</td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.2</td>
<td>0.19</td>
<td>0.25</td>
<td>0.32</td>
<td>0.85</td>
<td>0.97</td>
<td>0.55</td>
<td>0.27</td>
<td>0.58</td>
</tr>
<tr>
<td>7</td>
<td>0.22</td>
<td>0.2</td>
<td>0.19</td>
<td>0.25</td>
<td>0.32</td>
<td>0.85</td>
<td>0.67</td>
<td>0.38</td>
<td>0.43</td>
<td>0.45</td>
</tr>
<tr>
<td>8</td>
<td>0.19</td>
<td>0.7</td>
<td>0.13</td>
<td>0.18</td>
<td>0.16</td>
<td>0.21</td>
<td>0.28</td>
<td>0.56</td>
<td>0.23</td>
<td>0.51</td>
</tr>
<tr>
<td>9</td>
<td>0.26</td>
<td>0.15</td>
<td>0.24</td>
<td>0.69</td>
<td>0.32</td>
<td>0.18</td>
<td>0.37</td>
<td>0.42</td>
<td>0.48</td>
<td>0.48</td>
</tr>
<tr>
<td>10</td>
<td>0.15</td>
<td>0.13</td>
<td>0.65</td>
<td>0.17</td>
<td>0.24</td>
<td>0.19</td>
<td>0.25</td>
<td>0.38</td>
<td>0.43</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Fonte: [1, 21].

Table 5. Comparative table with all costs and emissions.

<table>
<thead>
<tr>
<th>Results</th>
<th>SA</th>
<th>DE</th>
<th>DA</th>
<th>NSGA II</th>
<th>PSO</th>
<th>ALO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emission (m³/h)</td>
<td>1757,39</td>
<td>1769,48</td>
<td>1765,34</td>
<td>2098,12</td>
<td>2113,33</td>
<td>1763,55</td>
</tr>
<tr>
<td>Cost ($/h)</td>
<td>1556,61</td>
<td>1545,59</td>
<td>1548,28</td>
<td>1709,47</td>
<td>1685,18</td>
<td>1549,93</td>
</tr>
</tbody>
</table>

Table 6. Comparison between all results of the different programmed algorithms.
5. Conclusion

The model was implemented in MATLAB computing. Usually all algorithms have presented good results, because in all cases it is possible to switch off at least two motors, but some differences among results of the different algorithms were evidenced. It was possible to notice vantages in relation to cost and emission of two methods among all of them. The SA achieves the lowest emission of pollutants, while DE obtained the lowest cost for the power plant of ten generating units.

Acknowledgements

The Institute of Technology and Education “Galileo” from Amazonia (ITEGAM), the Federal University of Para (UFPA), the Research Support Foundation State of Amazonas (FAPEAM), and the National Council of Research (CNPq) Productivity of Research Funds Process 301105/2016-2 for the financial support to this research.

Author details

Jorge de Almeida Brito Júnior*, Marcus Vinicius Alves Nunes¹, Manoel Henrique Reis Nascimento¹, Jandecy Cabral Leite¹, Jorge Laureano Moya Rodriguez³, Carlos Alberto Oliveira de Freitas¹, Milton Fonseca Júnior¹, Edson Farias de Oliveira¹, David Barbosa de Alencar¹, Nadime Mustafa Moraes⁵, Tirso Lorenzo Reyes Carvajal¹ and Haroldo Melo de Oliveira¹

*Address all correspondence to: jorge.brito@itegam.org.br

1 Research Department Institute of Technology and Education Galileo da Amazônia (ITEGAM), Manaus, Brazil
2 Faculty of Electrical Engineering Institute of Technology, Federal University of Para (UFPA), Belém, Pará, Brazil
3 Federal University of Bahia (UFBA), Bahia, Brazil
4 Department of Generation of Mauá, Eletrobrás Amazonas GT, Manaus, Amazonas, Brazil
5 University of the State of Amazonas (UEA), Manaus, Amazonas, Brazil

References


[6] Gonçalves E. Métodos híbridos de pontos interiores/exteriores e de aproximantes de funções em problemas multiobjetivo de despacho econômico e ambiental; 2015


[24] Nwulu NI, Xia X. Multi-objective dynamic economic emission dispatch of electric power generation integrated with game theory based demand response programs. Energy Conversion and Management 2015;89(0):963-974


[38] Suksonghong K, Boonlong K, Goh K-L. Multi-objective genetic algorithms for solving portfolio optimization problems in the electricity Market. Electrical Power and Energy Systems, Elsevier Ltd All rights reserved; 2014


[44] Zhoua A, Bo-Yang Q, Shi-Zheng HLZ, Nagaratnam SP, Qingfu Z. Multiobjective evolutionary algorithms: A survey of the state of the art. Elsevier BV All rights reserved, Swarm and Evolutionary Computation; 2011


