Chapter 2

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

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.76666

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:

$$\underset{P}{\text{minimize}} \quad [F_1(P), F_2(P)] \tag{1}$$

where *F1(P)* and *F2(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 Pi [8, 22]:

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

where a_i, b_i, and c_i are the fuel cost coefficients of the ith 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]:

$$F_2(P_i) = \sum_{i=1}^{n} \left(d_i + e_i P_i + f_i P_i^2 \right) \quad \mathrm{kg/h} \tag{3}$$

where di, ei, and fi 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]:

$$\sum_{i=1}^{n} P_i - P^{\mathrm{D}} - P^{\mathrm{L}} = 0 \tag{4}$$

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.

$$\sum_{i=1}^{n} P_i = P^{\mathrm{D}} + P^{\mathrm{L}} \tag{5}$$

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]:

$$P^L = \sum_{i=1}^{n} B_i P_i^2 \tag{6}$$

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]:

$$P_L = \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00}, \tag{7}$$

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]

$$P_{\min.i} \leq P_i \leq P_{\max.i} \tag{8}$$

where P_i is the output power of the *i* generator, $P_{min.i}$, is the minimal power of the *i* generator, and $P_{max.i}$, 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 $F_{min,i}$ and its upper limit $F_{max,i}$ so that [21].

$$F_{\min.i} \leq F_{im} \leq F_{\max.i}, \; i \in N, m \in M, \tag{9}$$

where $F_{i,m}$ is the fuel supplied to the engine *i* at the interval *m*, $F_{i,min}$ is the minimum amount of fuel supplied to *i* generator, and $F_{max,i}$ is the maximum amount of fuel supplied to *i* generator.

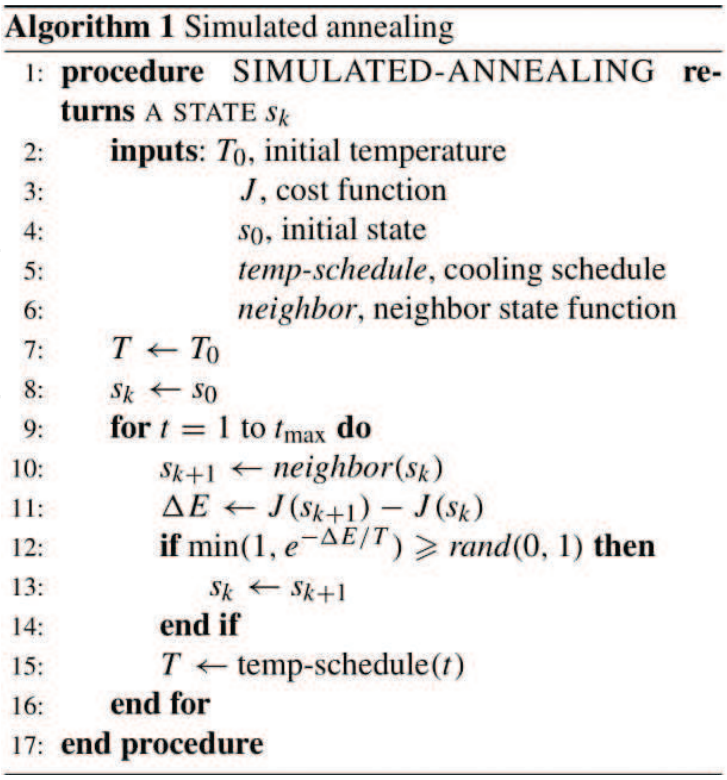
3. Multi-objective optimization techniques to solve EELD

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].



Algorithm 1 Simulated annealing

```
procedure SIMULATED-ANNEALING returns A STATE $s_k$
    inputs: $T_0$, initial temperature
            $J$, cost function
            $s_0$, initial state
            temp-schedule, cooling schedule
            neighbor, neighbor state function
    $T \leftarrow T_0$
    $s_k \leftarrow s_0$
    for $t = 1$ to $t_{\max}$ do
        $s_{k+1} \leftarrow neighbor(s_k)$
        $\Delta E \leftarrow J(s_{k+1}) - J(s_k)$
        if $\min(1, e^{-\Delta E/T}) \geqslant rand(0, 1)$ then
            $s_k \leftarrow s_{k+1}$
        end if
        $T \leftarrow$ temp-schedule($t$)
    end for
end procedure
```

Figure 1. Simulated annealing algorithm. Source: [28].

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]:

$$Pr(\Delta) = \exp\left(-\Delta/T\right) \tag{10}$$

where Δ 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:

$$T_v = r^{(v-1)} T_0 \tag{11}$$

where v is the iteration number and r is temperature reduction factor. T_0 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**.

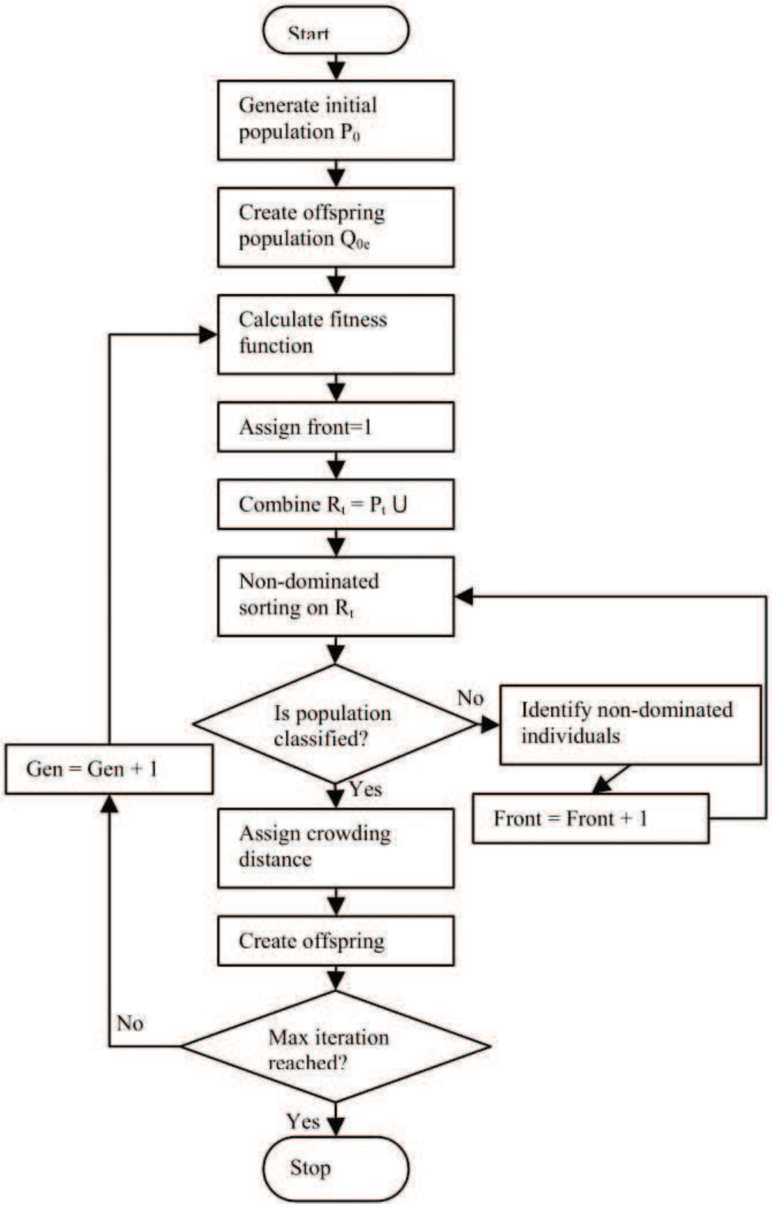

Start
Generate initial population P_0
Create offspring population Q_{0e}
Calculate fitness function
Assign front=1
Combine $R_t = P_t$ U
Non-dominated sorting on R_t
Is population classified?
No
Identify non-dominated individuals
Front = Front + 1
Yes
Gen = Gen + 1
Assign crowding distance
Create offspring
Max iteration reached?
No
Yes
Stop

Figure 2. Operation of NSGA II. Source: [35].

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 metamorphism to become adult [41].

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

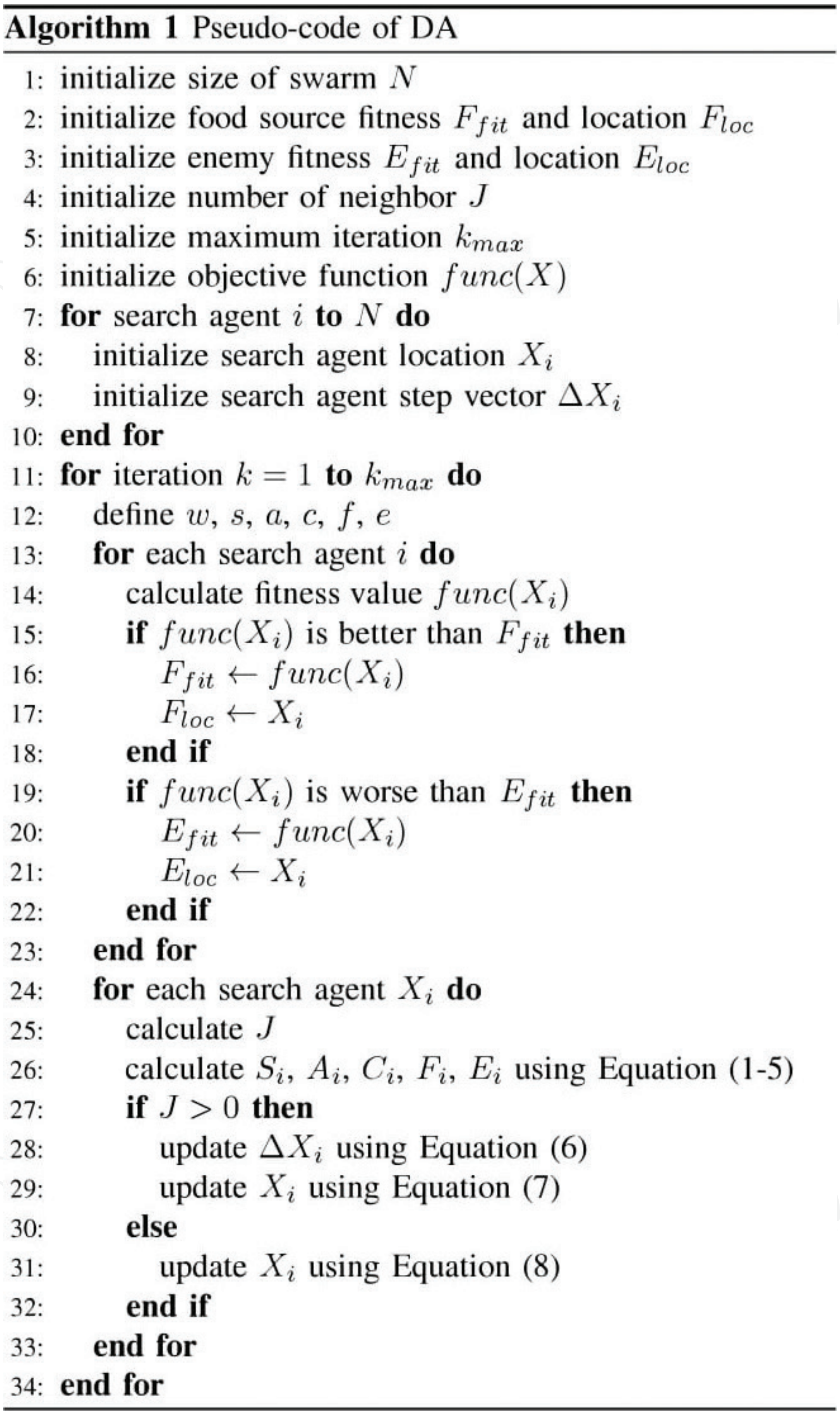
Dragonflies are considered as small predators that hunt almost all other small insects in nature. Nymph dragonflies also predate on other marine insects and even small fishes. The interesting fact about dragonflies is their unique and rare swarming behavior. Dragonflies swarm for only two purposes: hunting 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.

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

Source: [42].

Table 1. Characteristics of the DA.



Algorithm 1 Pseudo-code of DA

```
initialize size of swarm $N$
initialize food source fitness $F_{fit}$ and location $F_{loc}$
initialize enemy fitness $E_{fit}$ and location $E_{loc}$
initialize number of neighbor $J$
initialize maximum iteration $k_{max}$
initialize objective function $func(X)$
for search agent $i$ to $N$ do
    initialize search agent location $X_i$
    initialize search agent step vector $\Delta X_i$
end for
for iteration $k = 1$ to $k_{max}$ do
    define $w$, $s$, $a$, $c$, $f$, $e$
    for each search agent $i$ do
        calculate fitness value $func(X_i)$
        if $func(X_i)$ is better than $F_{fit}$ then
            $F_{fit} \leftarrow func(X_i)$
            $F_{loc} \leftarrow X_i$
        end if
        if $func(X_i)$ is worse than $E_{fit}$ then
            $E_{fit} \leftarrow func(X_i)$
            $E_{loc} \leftarrow X_i$
        end if
    end for
    for each search agent $X_i$ do
        calculate $J$
        calculate $S_i$, $A_i$, $C_i$, $F_i$, $E_i$ using Equation (1-5)
        if $J > 0$ then
            update $\Delta X_i$ using Equation (6)
            update $X_i$ using Equation (7)
        else
            update $X_i$ using Equation (8)
        end if
    end for
end for
```

Figure 3. Dragonfly pseudocode. Source: [43].

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].

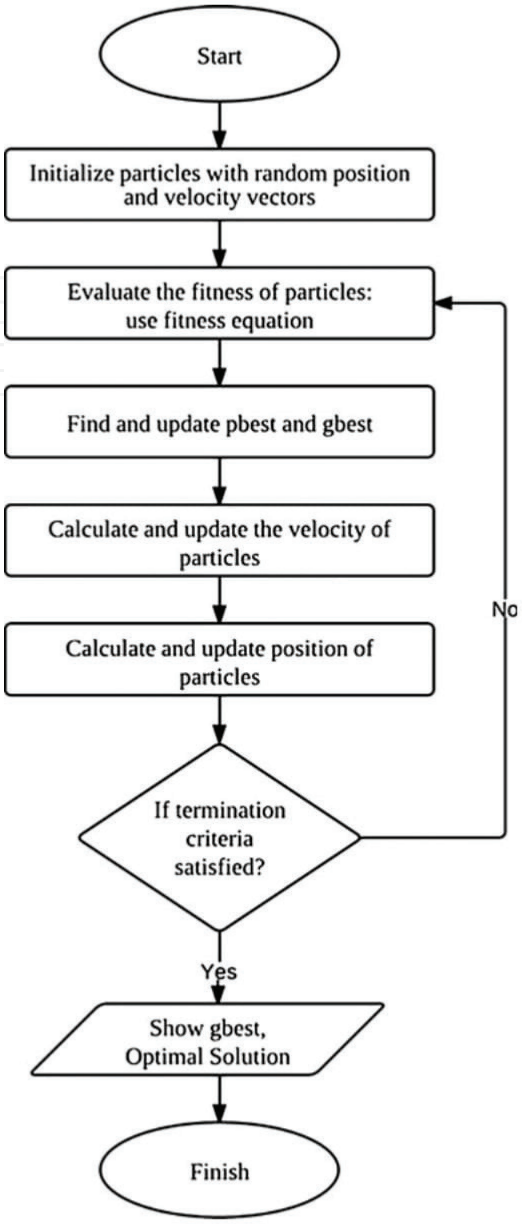

Start
Initialize particles with random position and velocity vectors
Evaluate the fitness of particles: use fitness equation
Find and update pbest and gbest
Calculate and update the velocity of particles
Calculate and update position of particles
If termination criteria satisfied?
No
Yes
Show gbest, Optimal Solution
Finish

Figure 4. PSO flow chart. Source: [46].

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 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 predefined 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 CO_2, NO_X, and SO_2 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,

Motor	c_i ($/Mw^2)	b_i ($/Mw)	a_i($)	P_{min}(Mw)	P_{max} (Mw)
1	0.007	7	240	0.66	3.35
2	0.0095	10	200	0.9	3.7
3	0.009	8.5	220	0.8	3.6
4	0.009	11	200	0.66	3.35
5	0.008	10.5	220	0.72	3.45
6	0.0075	12	120	0.66	2.97
7	0.0075	14	130	0.88	3.5
8	0.0075	14	130	0.754	3.33
9	0.0075	14	130	0.9	3.9
10	0.0075	14	130	0.56	2.35

Source: [1, 21].

Table 2. Cost coefficients from the thermal plant case study.

Motor	f	e	d
1	0.00419	1.32767	73.85932
2	0.00419	0.32767	13.85932
3	0.00683	−0.54551	40.2669
4	0.00683	−0.54551	40.2669
5	0.00461	−0.51116	42.89553
6	0.00461	−0.51116	42.8955
7	0.00461	−0.51116	42.8955
8	0.00461	−0.51116	42.8955
9	0.00061	−0.51116	10.8955
10	0.00461	−0.51116	42.8955

Source: [1].

Table 3. Emission coefficients from the thermal power plant used as case study.

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**.

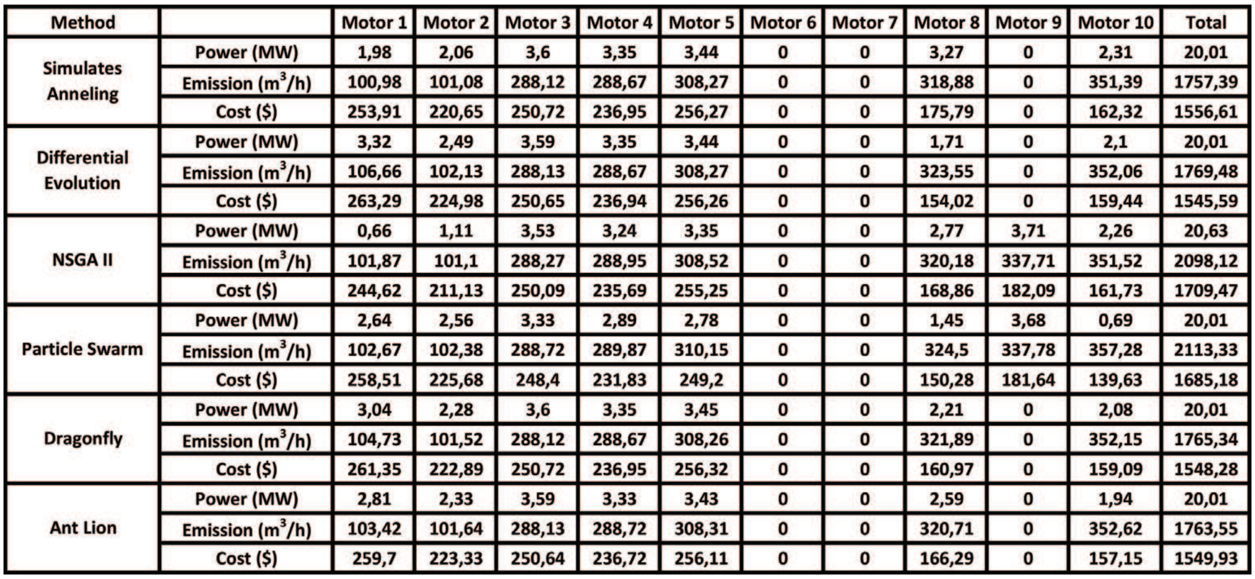
M	1	2	3	4	5	6	7	8	9	10
1	0.14	0.17	0.15	0.19	0.26	0.22	0.34	0.38	0.43	0.45
2	0.17	0.6	0.13	0.16	0.15	0.2	0.23	0.56	0.23	0.51
3	0.15	0.13	0.65	0.17	0.24	0.19	0.25	0.38	0.43	0.45
4	0.19	0.16	0.17	0.71	0.3	0.25	0.43	0.56	0.23	0.51
5	0.26	0.15	0.24	0.3	0.69	0.32	0.18	0.37	0.42	0.48
6	0.22	0.2	0.19	0.25	0.32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

Fonte: [1, 21].

Table 4. Loss coefficients (all values have to be multiplied by 1e-4).

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

Table 5. Comparative table with all costs and emissions.



Method		Motor 1	Motor 2	Motor 3	Motor 4	Motor 5	Motor 6	Motor 7	Motor 8	Motor 9	Motor 10	Total
Simulates Anneling	Power (MW)	1,98	2,06	3,6	3,35	3,44	0	0	3,27	0	2,31	20,01
	Emission (m^3/h)	100,98	101,08	288,12	288,67	308,27	0	0	318,88	0	351,39	1757,39
	Cost ($)	253,91	220,65	250,72	236,95	256,27	0	0	175,79	0	162,32	1556,61
Differential Evolution	Power (MW)	3,32	2,49	3,59	3,35	3,44	0	0	1,71	0	2,1	20,01
	Emission (m^3/h)	106,66	102,13	288,13	288,67	308,27	0	0	323,55	0	352,06	1769,48
	Cost ($)	263,29	224,98	250,65	236,94	256,26	0	0	154,02	0	159,44	1545,59
NSGA II	Power (MW)	0,66	1,11	3,53	3,24	3,35	0	0	2,77	3,71	2,26	20,63
	Emission (m^3/h)	101,87	101,1	288,27	288,95	308,52	0	0	320,18	337,71	351,52	2098,12
	Cost ($)	244,62	211,13	250,09	235,69	255,25	0	0	168,86	182,09	161,73	1709,47
Particle Swarm	Power (MW)	2,64	2,56	3,33	2,89	2,78	0	0	1,45	3,68	0,69	20,01
	Emission (m^3/h)	102,67	102,38	288,72	289,87	310,15	0	0	324,5	337,78	357,28	2113,33
	Cost ($)	258,51	225,68	248,4	231,83	249,2	0	0	150,28	181,64	139,63	1685,18
Dragonfly	Power (MW)	3,04	2,28	3,6	3,35	3,45	0	0	2,21	0	2,08	20,01
	Emission (m^3/h)	104,73	101,52	288,12	288,67	308,26	0	0	321,89	0	352,15	1765,34
	Cost ($)	261,35	222,89	250,72	236,95	256,32	0	0	160,97	0	159,09	1548,28
Ant Lion	Power (MW)	2,81	2,33	3,59	3,33	3,43	0	0	2,59	0	1,94	20,01
	Emission (m^3/h)	103,42	101,64	288,13	288,72	308,31	0	0	320,71	0	352,62	1763,55
	Cost ($)	259,7	223,33	250,64	236,72	256,11	0	0	166,29	0	157,15	1549,93

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[1]*, Marcus Vinicius Alves Nunes[2],
Manoel Henrique Reis Nascimento[1], Jandecy Cabral Leite[1],
Jorge Laureano Moya Rodriguez[3], Carlos Alberto Oliveira de Freitas[1],
Milton Fonseca Júnior[4], Edson Farias de Oliveira[1], David Barbosa de Alencar[1],
Nadime Mustafa Moraes[5], Tirso Lorenzo Reyes Carvajal[1] and Haroldo Melo de Oliveira[1]

*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

[1] Júnior JAB, Nunes MVA, Nascimento MHR, et al. Solution to economic emission load dispatch by simulated annealing: Case study. Electrical Engineering. 2017. https://doi.org/10.1007/s00202-017-0544-0

[2] Zhou J, Wang C, Li Y, Wang P, Li C, Lu P, et al. A multi-objective multi-population ant colony optimization for economic emission dispatch considering power system security. Applied Mathematical Modelling. 2017;**45**:684-704

[3] Roy PK, Bhui S. A multi-objective hybrid evolutionary algorithm for dynamic economic emission load dispatch. International Transactions on Electrical Energy Systems. 2016;**26**(1):49-78

[4] Modiri-Delshad M, Kaboli SHA, Taslimi-Renani E, Rahim NA. Backtracking search algorithm for solving economic dispatch problems with valve-point effects and multiple fuel options. Energy. 2016;**116**:637-649

[5] Granelli G, Montagna M, Pasini G, Marannino P. Emission constrained dynamic dispatch. Electric Power Systems Research. 1992;**24**(1):55-64

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

[7] Krishnamurthy S, Tzoneva R, editors. Comparative analyses of Min-Max and Max-Max price penalty factor approaches for multi criteria power system dispatch problem with valve point effect loading using Lagrange's method. 2011 International Conference on Power and Energy Systems; 2011:22-24 Dec. 2011

[8] Nascimento MHR, Nunes MVA, Rodríguez JLM, Leite JC, Junior JAB. New solution for resolution of the economic load dispatch by different mathematical optimization methods, turning off the less efficient generators. Journal of Engineering and Technology for Industrial Applications. 2017;**03**:37-46

[9] De M, Das G, Mandal S, Mandal K. Investigating economic emission dispatch problem using improved particle swarm optimization technique. Industry Interactive Innovations in Science, Engineering and Technology: Springer. 2018:37-45

[10] Mason K, Duggan J, Howley E. Multi-objective dynamic economic emission dispatch using particle swarm optimisation variants. Neurocomputing. 2017;**270**(Suppl C):188-197

[11] Mohan M, Kuppusamy K, Khan MA. Optimal short-term hydrothermal scheduling using decomposition approach and linear programming method. International Journal of Electrical Power & Energy Systems. 1992;**14**(1):39-44

[12] Farag A, Al-Baiyat S, Cheng T. Economic load dispatch multiobjective optimization procedures using linear programming techniques. IEEE Transactions on Power Systems. 1995;**10**(2):731-738

[13] Huang S-J. Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches. IEEE Transactions on Energy Conversion. 2001;**16**(3):296-301

[14] Ma H, Yang Z, You P, Fei M. Multi-objective biogeography-based optimization for dynamic economic emission load dispatch considering plug-in electric vehicles charging. Energy. 2017;**135**(Suppl C):101-111

[15] Liu Hd, Ma Zl, Liu S, Lan H, editors. A new solution to economic emission load dispatch using immune genetic algorithm. 2006 IEEE Conference on Cybernetics and Intelligent Systems; 2006 7-9 June 2006

[16] Damousis IG, Bakirtzis AG, Dokopoulos PS. Network-constrained economic dispatch using real-coded genetic algorithm. IEEE Transactions on Power Systems. 2003;**18**(1):198-205

[17] Kumarappan N, Mohan MR, editors. Hybrid genetic algorithm based combined economic and emission dispatch for utility system. International Conference on Intelligent Sensing and Information Processing, 2004 Proceedings of; 2004

[18] Momoh JA, Reddy SS, editors. Combined Economic and Emission Dispatch using Radial Basis Function. 2014 IEEE PES General Meeting | Conference & Exposition; 2014 27-31 July 2014

[19] Jebaraj L, Venkatesan C, Soubache I. Rajan CCA. Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: A review. Renewable and Sustainable Energy Reviews. 2017;**77**(Suppl C):1206-1220

[20] Chatterjee A, Ghoshal S, Mukherjee V. Solution of combined economic and emission dispatch problems of power systems by an opposition-based harmony search algorithm. International Journal of Electrical Power & Energy Systems. 2012;**39**(1):9-20

[21] Nascimento MHR, Nunes MVA, Rodríguez JLM, et al. A new solution to the economical load dispatch of power plants and optimization using differential evolution. Electrical Engineering. 2017;**99**:561. https://doi.org/10.1007/s00202-016-0385-2

[22] Basu M. Fuel constrained economic emission dispatch using nondominated sorting genetic algorithm-II. Energy. 2014;**78**:649-664

[23] Miranda V, Hang PS. Economic dispatch model with fuzzy wind constraints and attitudes of dispatchers. IEEE Transactions on Power Systems. 2005;**20**(4):2143-2145

[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

[25] Ashish D, Arunesh D, Surya P, Bhardwaj AK. A traditional approach to solve economic load dispatch problem of thermal generating unit using MATLAB programming. International Journal of Engineering Research & Technology (IJERT). 2013;**2**(9):2013

[26] Wang L, Singh C. Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm. Electric Power Systems Research. 2007;**77**(12):1654-1664

[27] Kamboj VK, Bath S, Dhillon J. Solution of non-convex economic load dispatch problem using Grey wolf optimizer. Neural Computing and Applications. 2015:1-16

[28] Fraga-Gonzalez LF, Fuentes-Aguilar RQ, Garcia-Gonzalez A, Sanchez-Ante G. Adaptive simulated annealing for tuning PID controllers. AI Communications. 2017;**30**(5):347-362

[29] Wong K, Fung C, editors. Simulated annealing based economic dispatch algorithm. IEE proceedings C (generation, transmission and distribution). IET; 1993

[30] Ziane I, Benhamida F, Amel G. Simulated annealing optimization for multi-objective economic dispatch solution. Leonardo Journal of Sciences. 2014;**13**(25):43-56

[31] Abul'Wafa AR. Optimization of economic/emission load dispatch for hybrid generating systems using controlled elitist NSGA-II. Electric Power Systems Research. 2013;**105**:142-151

[32] Basu M. Combined heat and power economic emission dispatch using nondominated sorting genetic algorithm-II. International Journal of Electrical Power & Energy Systems. 2013;**53**:135-141

[33] Cococcioni M, Lazzerini B, Marcelloni F, Pistolesi F, editors. Solving the environmental economic dispatch problem with prohibited operating zones in microgrids using NSGA-II and TOPSIS. Proceedings of the 31st Annual ACM Symposium on Applied Computing. ACM; 2016

[34] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 2002;**6**(2):182-197

[35] Golchha A, Qureshi SG. Non-dominated sorting genetic algorithm-II – A succinct survey. International Journal of Computer Science and Information Technologies - (IJCSIT). 2015;**6**(1):252-255

[36] Kalaivani L, Subburaj P, Willjuice Iruthayarajan M. Speed control of switched reluctance motor with torque ripple reduction using non-dominated sorting genetic algorithm (NSGA-II). International Journal of Electrical Power & Energy Systems. 2013;**53**:69-77

[37] Golchha A, Qureshi SG. Non-dominated sorting genetic algorithm-II–A succinct survey. International Journal of Computer Science and Information Technologies. 2015;**6**(1):252-255

[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

[39] Liu T, Gao X, Wang L. Multi-objective optimization method using an improved NSGA-II algorithm for oil–gas production process. Journal of the Taiwan Institute of Chemical Engineers 2015(0)

[40] Mirjalili S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Computing and Applications. 2016;**27**(4):1053-1073

[41] Zolghadr-Asli B, Bozorg-Haddad O, Chu X. Dragonfly Algorithm (DA). Advanced Optimization by Nature-Inspired Algorithms. Springer; 2018. pp. 151-159

[42] Bozorg-Haddad O. Advanced Optimization by Nature-Inspired Algorithms. Springer; 2018

[43] Daely PT, Shin SY, editors. Range based wireless node localization using dragonfly algorithm. Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on. IEEE; 2016

[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

[45] Park JB, Jeong YW, Shin JR, Lee KY. An improved particle swarm optimization for nonconvex economic dispatch problems. IEEE Transactions on Power Systems. 2010;**25**(1):156-166

[46] Armaghani DJ, Hajihassani M, Mohamad ET, Marto A, Noorani S. Blasting-induced flyrock and ground vibration prediction through an expert artificial neural network based on particle swarm optimization. Arabian Journal of Geosciences. 2014;**7**(12):5383-5396

[47] Idoumghar L, Chérin N, Siarry P, Roche R, Miraoui A. Hybrid ICA–PSO algorithm for continuous optimization. Applied Mathematics and Computation. 2013;**219**(24):11149-11170

[48] Behera S, Sahoo S, Pati BB. A review on optimization algorithms and application to wind energy integration to grid. Renewable and Sustainable Energy Reviews. 2015;**48**:214-227

[49] Coelho LS, Bora TC, Mariani VC. Differential evolution based on truncated Lévy-type flights and population diversity measure to solve economic load dispatch problems. International Journal of Electrical Power & Energy Systems. 2014;**57**:178-188

[50] Roque CMC, Martins PALS. Differential evolution for optimization of functionally graded beams. Composite Structures. 2015;**133**:1191-1197

[51] Mirjalili S. The ant lion optimizer. Advances in Engineering Software. 2015;**83**:80-98

[52] Raju M, Saikia LC, Sinha N. Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller. International Journal of Electrical Power & Energy Systems. 2016;**80**:52-63

[53] Kamboj VK, Bhadoria A, Bath S. Solution of non-convex economic load dispatch problem for small-scale power systems using ant lion optimizer. Neural Computing and Applications. 2017;**28**(8):2181-2192

[54] Mirjalili S, Jangir P, Saremi S. Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems. Applied Intelligence. 2017;**46**(1): 79-95

[55] Qu B, Zhu Y, Jiao Y, Wu M, Suganthan P, Liang J. A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems. Swarm and Evolutionary Computation;**2017**