

Modified Differential Evolution Algorithm Applied to Economic Load Dispatch Problems

Gabriella Lopes Andrade[1], Claudio Schepke[2(✉)], Natiele Lucca[2], and João Plinio Juchem Neto[3]

[1] School of Technology, Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, RS, Brazil
gabriella.andrade@edu.pucrs.br

[2] Laboratory of Advances Studies in Computation, Federal University of Pampa (UNIPAMPA), Alegrete, RS, Brazil
{claudioschepke,natielelucca.aluno}@unipampa.edu.br

[3] Department of Economics and International Relations, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brazil
plinio.juchem@ufrgs.br

Abstract. This paper proposes a modification of the Differential Evolution (DE) algorithm to solve the problem of Economic Load Dispatch (ELD). DE is an algorithm based on the theory of natural selection of species, where the fittest are more likely to survive. In the original DE, each possible solution to the target problem composes an initial population. This population evolved through genetic operators of mutation, selection, and crossover of individuals. In each iteration of the DE, the newly generated population replaces and discards the old population input. At the end of execution, the DE should return the best solution found. The Modification of the Differential Evolution (MDE) present in this paper considers that, in the selection stage, the ablest individual will replace the old one in the current population instead of being inserted in the new sample. To verify the performance of the MDE about the original DE, we solve a set of test functions to obtain the global minimum and different instances of the ELD. Both algorithms were effective in minimizing the three least-dimensional functions. Our results showed that DE proved more effective than MDE in minimizing the set of higher dimensional test functions, presenting a solution up to 99.99% better. However, none of the algorithms managed to obtain the optimal solution. In the ELD resolution, where it is to find the production level of each thermoelectric generating unit, satisfying the total system demand at the lowest cost, MDE was more effective than DE in all cases, finding a solution up to 1.10% better, solving the constraints of the problem. In addition, the computation time reduction of MDE concerning DE was up to 95.98%. Therefore, we confirm the efficiency of the proposed modification over the original DE version.

Keywords: Economic Load Dispatch · Modified Differential Evolution · Differential Evolution · Optimization

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13956, pp. 19–37, 2023.
https://doi.org/10.1007/978-3-031-36805-9_2

1 Introduction

Many engineering problems are challenging or impossible to solve using classical mathematical optimization methods. These occur due to the difficulty in formulating the modelling or the mathematical effort required to solve it because they are non-linear, non-differentiable, or discontinuous [10]. Thus, bio-inspired algorithms based on populations are widely applied to solve this problem type.

Bio-inspired algorithms use a constructive method to obtain an initial population. From the initial population, applying local and global search techniques optimize population solutions. Among the most prominent classes of bio-inspired algorithms is Evolutionary Computing [10]. Algorithms belonging to the Evolutionary Computing class use rules of the natural selection theory, where individuals that best adapt to the environment are more likely to survive.

This paper proposes changes in the Differential Evolution (DE) algorithm [13], which belongs to the class of Evolutionary Computing algorithms. We present a Modification to the DE algorithm (MDE) to make it more efficient. In the original DE, the initial population advances to the evolutionary process, which consists of applying the genetic operators of mutation and crossover and the selection of individuals for insertion in the new input. At each iteration, a newly generated population replaces the old discarded one. In the end, the DE returns the best solution found. In MDE, instead of inserting the fittest individual in the new population, it replaces the old one in the current population. In this way, the population is updated in the greedy selection step and not replaced at the end of the iteration.

The target problem of this work is the Economic Load Dispatch (ELD) [7]. The purpose of the ELD is to scale the outputs of the power generation units of a thermoelectric plant to meet the consumer load demand at a minimum operation cost, which satisfies all the operating restrictions of the generating units and the restrictions of equality and inequality imposed [10]. Due to the increased demand for electricity in recent years and the scarcity of water resources [2], thermoelectric plants are of paramount importance for the contribution of the supply of electricity demand [14]. However, the fuel used for energy generation by thermoelectric plants presents a high cost. Soon, the importance of using ELD in optimizing the operation of energy systems is justified. In addition, with the resolution of the ELD, we will be able to verify the performance of the MDE compared to the original DE.

The **objective** of this paper is to propose a modification of the classical Differential Evolution algorithm to solve Economic Load Dispatch problems. The **contributions** of the paper are:

- Implementation of the Differential Evolution method and the proposed modification in Matlab;
- Application and comparison of methods in problems of the Economic Dispatch of a load of different sizes;
- Verification of the performance of methods about other methods already used in the literature to solve the Economic Dispatch.

The organization of this paper is into seven sections. Section 2 presents a general description of the ELD problem with the main restrictions applied to it. Section 3 discusses the DE heuristic. Section 4 points to the modifications of the DE, resulting in the proposed MDE heuristic. Section 5 shows aspects of methodology, like implementation and parameters for the executions. Section 6 details the results obtained through the execution of the DE and MDE heuristics to solve ELD. Section 7 presents the conclusion of this paper and future works.

2 Economic Load Dispatch Problem Description

Economic Load Dispatch (ELD) is a non-linear mathematical problem considered a sub-problem of the Unit Commitment (UC) [4,7]. The problem consists of allocating the total demand among generating units to minimize the total cost of production [8]. In addition, it must satisfy the system load demand, spinning reserve capacity, and practical operation constraints of generators such as valve-point effects, ramp rate limit, and prohibited operating zone [7,9].

This paper considers thermal sources (biomass, natural gas, coal, nuclear) to generate electricity. The generating units offer different production costs from the energy source used to produce the electricity [8]. In this way, ELD is an optimization problem in which the objective function to determine the total cost of electricity generation satisfies an equality problem and inequality of the problem [10].

2.1 Objective Function

In our ELD problem, we want to find the generation level of each thermoelectric power-generating unit that satisfies the total demand of the system at a minimum cost [10]. Considering a set of thermal groups with n generating units, the total fuel cost F to minimize the sum of the contributions from the generating units is given by the following equation:

$$\min C(P) = \sum_{i=1}^{n} C_i(P_i) \tag{1}$$

In this equation, C_i is the operating cost for the generating unit i (in \$/h), and P_i (in MW) is the power supplied by this unit. The operating cost C_i of each generating unit can be expressed as a function of the output power P_i through the following equation [7,10]:

$$\sum_{i=1}^{n} C_i(P_i) = \sum_{i=1}^{n} \alpha_i + \beta_i P_i + \gamma_i P_i^2 \tag{2}$$

Where the variables α_i, β_i and γ_i are restrictions on the characteristics of the generating unit i [10].

In the basic ELD, the supplied power must satisfy the power balance. That is, the sum of the contributions from the generating units must meet the total demand of the system P_D [8]. The following equation defines the power balance:

$$\sum_{i=1}^{n} P_i = P_D \tag{3}$$

Where P_D is the constant load power demanded. Furthermore, in the basic DE, transmission losses are not considered ($P_L = 0$).

According to [10], the objective function (2) must be modified to contemplate not only the load distribution with low cost in the generating units but also to satisfy the equality constraint of the system. Thus, we have the following objective function:

$$\sum_{i=1}^{n} F_i(P_i) + \phi \left(abs \left(\sum_{i=1}^{n} P_i - P_D \right) \right) \tag{4}$$

In this equation, ϕ is the penalty factor for solutions that do not meet the equilibrium in the power balance, which is a positive constant [10].

2.2 Practical Operation Constraints of Generator

The ELD problem generally has practical operation constraints of generating units. These restrictions are the balance of power and transmission losses, minimum and maximum limits of power operation by generating units, valve points, ramp limits, and prohibited operating zones [7].

Power Balance Constraint. The totality of generated power by the thermal units should be equal to the sum of the total system demand (P_D) and the total transmission network losses (P_L), according to the following equation:

$$\sum_{i=1}^{n} P_i = P_D + P_L \tag{5}$$

Where P_D is a constant value and P_L is a function of unit power outputs that uses the B coefficients according to the Kron equation [7]:

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} B_{ij} P_i P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} \tag{6}$$

Where B_{ij} is the i-th element of the loss coefficient matrix, B_{0i} is the i-th element of the loss coefficient vector, and B_{00} is the loss coefficient constant [10].

The objective function (2) must be modified to contemplate not only the low-cost load distribution in the generating units, but also to satisfy the equality constraint of the system. Thus, we have the following objective function:

$$\sum_{i=1}^{n} F_i(P_i) + \phi \left(abs \left(\sum_{i=1}^{n} P_i - P_D - P_L \right) \right) \tag{7}$$

In this equation, ϕ is the penalty factor for solutions that do not meet the equilibrium in the load balance, which is a positive constant [10].

Power Output Limits. The power P_i (in MW) of each generating unit i must be within its minimum and maximum limits. The restriction on the power P_i is given by the following inequality:

$$P_i^{min} \leq P_i \leq P_i^{max} \tag{8}$$

Where P_i^{min} and P_i^{max} are, respectively, the minimum and maximum operating outputs of the generating unit i (in MW) [10]. According to [8], when power limit constraints are considered, along with power balance constraints and transmission losses, the classification of the DE problem is classic.

Valve Point. The modelling of the objective function of the ELD is as a quadratic polynomial. The effect of valve points in the form of a rectified sinusoidal function is by adding to the quadratic cost function. This ripple effect occurs when each valve in a turbine starts to open [15]. The description of sinusoidal components is in the following way:

$$F_i(P_i) = \alpha_i P_i^2 + \beta_i P_i + \gamma_i + \sigma |e_i \times \sin[f_i \times (P_i^{min} - P_i)]| \tag{9}$$

Where σ is equal to 1 if valve points exist, otherwise σ is setting to zero, e_i the coefficient due to the effect of valve points, P_i^{min} is the minimum output limit on the power P_i (in MW) generated [4,9].

Prohibited Zones. The operating prohibited zones divide the operating limits into several regions. In addition to considering the minimum and maximum operating limits, we will have intermediate limits representing the restriction of operating prohibited zones. In practice, it is difficult to determine the shape of the input-output curve close to the prohibited zones, either by carrying out machine performance tests or through operational records [4,7]. Thus, the best way to deal with the problem is to adjust the output of the P_i generation to avoid the unit operation in the prohibited zones [7]. The possible mathematical description of operating zones for an energy-generating unit i is as follows:

$$\begin{array}{l} P_i^{min} \leq P_i \leq P_{i,1}^l \\ P_{i,j-1}^u \leq P_i \leq P_{i,j}^l, \ j = 2, 3, \ldots, n_i \\ P_{i,n_i}^u \leq P_i \leq P_i^{max} \end{array} \tag{10}$$

Where $P_{i,j}^l$ and $P_{i,j}^u$ are the lower and upper limits of the prohibited operating zone j of the generating unit i [7,9].

Ramp Limits. The adjustment of the generator output could not be instantly in practice. Therefore, the restricting the operating range of all generating units

is by their ramp limits to force it to operate continuously between two adjacent specific operating periods [7]. The inequality restrictions due to ramp-rate limits for unit generation changes are given by:

– As the generation grows:

$$P_i - P_i^0 \leq UR_i \tag{11}$$

– As the generation decreases:

$$P_i - P_i^0 \leq DR_i \tag{12}$$

Where P_i is the current output power of the generating unit i at time t; P_i^0 is the previous output power of the generating unit at time $t-1$; UR_i and DR_i (in MW) are respectively the ascent and descent restrictions of generator i [7,9]. The inclusion of ramp rate limits in Eq. 11 and Eq. 12 modifies the output power limits, as can be seen in the following Equation [9]:

$$\max(P_i^{min}, P_i^0 - DR_i) \leq P_i \leq \min(P_i^{max}, P_i^0 + UR_i) \tag{13}$$

3 Differential Evolution

The Differential Evolution algorithm is a direct parallel search method based on the theory of species evolution [12]. In DE, each possible solution of the target problem composes the initial population, which is evaluated and applied to the evolution process, which consists of mutating individuals with a certain probability, crossing the mutant individual with the current individual generating a new individual, and selecting the fittest individuals.

3.1 Algorithm

Algorithm 1 illustrates DE's behaviour. The algorithm returns the best solution upon satisfying the stopping criterion (by the number of iterations or solutions quality). Initially, three parameters are defined: the population size (NP), the disturbance factor (F), and the probability of crossing over (CR). These parameters will remain the same throughout the [13] optimization process. According to [12], the most used values for F and CR must respect the following restrictions: $F \in [0;\ 1{,}2]$ and $CR \in [0;\ 1]$. The generation of the initial population is after defining the parameters, which is composed of NP possible solutions for the target problem. Considering a target problem with D dimensions, each x_i solution is a D-dimensional vector, where the generation of each position is randomly performed from the following equation:

$$x_{i,G} = x_{i,G}^{min} + rand(0,1)(x_{i,G}^{max} - x_{i,G}^{min}); \tag{14}$$

Where min and max are the lower and upper bounds of the parameter G (1, ..., D) of the solution x_i and $rand(0,1)$ generates a number between $[0,1]$ [12].

Algorithm 1: Differential Evolution Algorithm

```
Input: NP,F, CR
Output: Best Solution Found
Generates initial population;
Evaluates all individuals from the initial population;
while Stop method is not satisfied do
    for each individual in the population do
        Selects an individual from the population;
        Performs mutation in the selected individual;
        Crosses the current individual with the mutant in a certain probability;
        Evaluates the generated descendant;
        Performs the greedy selection;
    end
    Replaces the old population with the new one;
end
return Best Solution Found
```

Behind the generation of the initial population, it is necessary to calculate the value of the objective function for each x_i of the population NP, where the value of each position i of the vector $x_{i,G}$ is given as a parameter of the objective function. If the target problem is a maximization problem, the best individual will be the one that produces the large solution. If it is a minimization problem, the best individual will be the one that produces the smallest solution.

After the evaluation of the population, the application of the evolutionary process occurs. The fundamental idea behind DE is a scheme for generating vectors of experimental parameters. In this way, the evolutionary process generates new parameter vectors by adding a weighted difference vector between two individuals to a third individual of the population [13]. The performance of this process is by the mutation and crossover genetic operators.

3.2 Mutation and Crossover Operators

The mutation is responsible for modifying a vector through the weighted difference between two individuals in the population. For each $x_{i,G}$ belonging to the population NP, the generation of an experimental vector $v_{i,G+1}$ is done.

After finishing the mutation process, the crossover occurs, which is the exchange of attributes among the individuals of the population and the mutant vectors to create a new modified vector [13]. The performed crossover increases the diversity of the parameters of the perturbed vectors. It occurs with a certain probability [12]. There are different ways to mutate the DE vector and two crossover modes. The following DE notation: $/x/y/z$ defines mutation and crossover, where x specifies whether to choose a random vector to mutate (*rand*) or the vector that produces the best solution (*best*), y defines the number of weighted differences used for the perturbation, and z specifies whether the perform type of crossover will be binomial (*bin*) or exponential (*exp*).

4 Modified Differential Evolution

This paper proposes a modification to the original DE algorithm. We aim to optimize it and improve the quality of the solutions found. In the original DE, a new population is generated at each iteration, replacing the old discarded one. This process performs a copy of the current population at the beginning of the evolutionary process. The update of this copy is according to the greedy selection step. That is, the insertion of the fittest individuals occurs into the new population. The new population replaces the old one after all individuals have gone through the evolution process [12].

The population update is after the greedy selection step in the proposed modification. If the solution of the new individual $v_{i,G+1}$ is better than the solution of the current $x_{i,G}$, the replacement of the individual $x_{i,G}$ will be immediately in the population. Otherwise, it kept the current individual. Therefore, there will be no need to create a new population at each iteration. In this way, we will increase the possibility of improving the solution of a given individual.

Algorithm 2 shows the Modified Differential Evolution (MDE). Initially, the definition of parameters NP, F, and CR occurs according to the restrictions imposed by [12]. The generation and evaluation of the initial population are then performed, where the individual who obtains the smallest solution will be the fittest.

The application of the evolutionary process to the population is after the evaluation, which involves the selection of a random individual to the mutation and crossover genetic operators. Crossover occurs with a certain probability. Soon after, an individual is generated and evaluated, and then performs the greedy selection and the updating of the population. In the end, the MDE algorithm should return the best solution found.

Algorithm 2: Modified Differential Evolution Algorithm - Pseudo-code

```
   Input: NP, F, CR
   Output: Better Solution Found
 1 Generates initial population;
 2 Evaluates all individuals in the initial population;
 3 while The stopping criterion is not met do
 4 |   for each individual in the population do
 5 |   |   Selects an individual from the population;
 6 |   |   Performs mutation on the selected individual;
 7 |   |   Performs the Crossing of the current individual with the mutant;
 8 |   |   Evaluates the generated descendant;
 9 |   |   Performs the greedy selection;
10 |   |   Updates the population;
11 |   end
12 end
13 return Best Solution Found
```

5 Methodology

We will solve the ELD by considering systems of different sizes and constraints to verify how the DE and MDE[1] heuristics behave. One of the issues in solving the ELD is how the algorithm handles the equality and inequality constraints of the problem [9]. The address of five constraints must be power balance, operating limits, forbidden zones, ramp limits, and valve points. We used the penalty factor ϕ, according to Eq. 7, to satisfy the power balance restriction [10]. In this way, the objective function contemplates not only the load distribution, with low cost in the generating units but also satisfies the power balance constraint.

To satisfy the restriction of the operating limits on the power P_i generated by each generating unit i (in MW), the value of P_i is adjusted when it exceeds the limits. The power P_i is scared to the nearest to the limit. The performance adjustment is by adopting the following strategy:

$$P_i = \begin{cases} P_i^{min}, \text{ se} & P_i < P_i^{min}, \\ P_i^{max}, \text{ se} & P_i > P_i^{max} \end{cases} \tag{15}$$

Where P_i^{min} and P_i^{max} are, respectively, the minimum and maximum operating outputs of the generating unit i (in MW) [10].

The power value P_i that is in the j-th zone of prohibited operation must be readjusted to the closest limit of this zone to satisfy the restriction of prohibited zones, as follows:

$$P_i = \begin{cases} P_{i,j}^l, \text{ se} & P_{i,j}^l \leq P_i \leq (P_{i,j}^l + P_{i,j}^u)/2, \\ P_{i,j}^u, \text{ se} & (P_{i,j}^l + P_{i,j}^u)/2 \leq P_i \leq P_{i,j}^u \end{cases} \tag{16}$$

Where $P_{i,j}^l$ and $P_{i,j}^u$ are respectively the lower and upper limits of the prohibited operating zone j of the generating unit i [7,9].

The restriction inclusion of ramp rate limits modifies the output power limits as seen in Eq. 13 [9]. Therefore, we deal with the constraints of operating limits and ramp limits simultaneously, adopting the following strategy:

$$P_i = \begin{cases} \max(P_i^{min}, P_i^0 - DR_i), \text{ se} & P_i \leq \max(P_i^{min}, P_i^0 - DR_i) \\ \min(P_i^{max}, P_i^0 + UR_i), \text{ se} & P_i \geq \min(P_i^{max}, P_i^0 + UR_i) \\ P_i & \text{other cases.} \end{cases} \tag{17}$$

Rectified sinusoidal components must be added to the objective function (Eq. 2), as can be seen in Eq. 9 to satisfy the valve point restriction.

We implemented the DE and MDE algorithms in the Matlab programming language. The executions running on a microcomputer with an Intel i5-7200U processor with two physical cores (four logical cores) of 2.5 GHz frequency, 4 GB DDR4 RAM, and Windows operating system version 10 of 64-bit.

To verify the performance of the MDE to the original DE, we solved two cases of the Economic Load Dispatch problem: Classic DE with a system of 15 generating units (Subsect. 5.1) and Non-Convex DE with 15 generating units (Subsect. 5.2).

[1] Available at: https://github.com/gabriella-andrade/DifferentialEvolution.

5.1 Classic Economic Load Dispatch with 15 Generating Units

The ELD with 15 thermoelectric energy generating units considers power balance restrictions, transmission losses, and each generating unit operating between minimum and maximum operating levels. It is a classic problem if it considers only these restrictions, being just nonlinear [8]. The classic ELD with 15 generating units can be mathematically described as the following nonlinear constrained optimization problem [7].

$$\begin{aligned} &\min\ F = \sum_{i=1}^{15} F_i(P_i) \\ &\text{s.a}\ \sum_{i=1}^{15} P_i = P_D + P_L, \\ &\quad P_i^{min} \leq P_i \leq P_i^{max}, i = 1, \ldots, 15 \end{aligned} \tag{18}$$

For this system, the total power demand is $P_D = 2{,}630\,\text{MW}$ [5]. The penalty factor was $\phi = 50$ to satisfy the power balance constraint [5]. The values of the minimum (P_i^{min}) and maximum (P_i^{max}) limits of each of the i ($i \in \{1, 2, \ldots, 15\}$) generating units, in addition to the parameters α_i, β_i and γ_i, which can be consulted in the Table 1 [7]. The values of the loss coefficients B_{ij}, B_{0i} and B_{00} are in the Eq. 19, Eq. 20, and Eq. 21.

$$B_{ij} = 10^{-3} \times \begin{bmatrix} 1.4 & 1.2 & 0.7 & -0.1 & -0.3 & -0.1 & -0.1 & -0.1 & -0.3 & 0.5 & -0.3 & -0.2 & 0.4 & 0.3 & -0.1 \\ 1.2 & 1.5 & 1.3 & 0.0 & -0.5 & -0.2 & 0.0 & 0.1 & -0.2 & -0.4 & -0.4 & 0.0 & 0.4 & 1.0 & -0.2 \\ 0.7 & 1.3 & 7.6 & -0.1 & -1.3 & -0.9 & -0.1 & , & 0.0 & -0.8 & -1.2 & -1.7 & -0.0 & -2.6 & 11.1 & -2.8 \\ -0.1 & 0.0 & -0.1 & -3.4 & , & -0.7 & -0.4 & -1.1 & 5.0 & -2.9 & -3.2 & -1.1 & -0.0 & 0,1 & 0.1 & -2.6 \\ -0.3 & -0.5 & -1.3 & -0.7 & 9.0 & 1.4 & -0.3 & -1.2 & -1.0 & -1.3 & 0.7 & -0.2 & -0.2 & -2.4 & -0.3 \\ -0.1 & -0.2 & -0.9 & -0.4 & 1.4 & 1.6 & -0.0 & -0.6 & -0.5 & -0.8 & 1,1 & -0,1 & -0.2 & -1.7 & 0.3 \\ -0.1 & 0.0 & -0.1 & 1.1 & -0.3 & -0.0 & 1.5 & , & 1,7 & 1.5 & 0.9 & -0,5 & 0.7 & -0.0 & -0.2 & -0,8 \\ -0,1 & 0,1 & 0,0 & 5.0 & -1.2 & -0.6 & 1.7 & 16.8 & 8,2 & 7.9 & -2,3 & -3.6 & 0.1 & 0.5 & -7.8 \\ -0.3 & -0.2 & -0.8 & 2.9 & , & -1.0 & -0.5 & 1.5 & 8,2 & 12.9 & 11.6 & -2.1 & -2,5 & 0.7 & -1.2 & -7.2 \\ -0.5 & -0.4 & -1.2 & 3.2 & , & -1.3 & -0.8 & 0.9 & 7.9 & 11.6 & 20.0 & -2.7 & -3,4 & 0,9 & -1,1 & -8,8 \\ -0.3 & -0.4 & -1.7 & -1.1 & 0.7 & 1.1 & -0.5 & -2.3 & -2.1 & -2.7 & 14.0 & 0.1 & 0.4 & -3.8 & 16,8 \\ -0.2 & -0.0 & -0.0 & -0.0 & -0.2 & -0.1 & 0.7 & -3.6 & , & -2.5 & -3.4 & 0.1 & 5.4 & , & -0.1 & -0.4 & 2.8 \\ 0.4 & 0.4 & -2.6 & 0.1 & -0.2 & -0.2 & -0.0 & , & 0.1 & 0.7 & 0.9 & 0.4 & -0.1 & 10.3 & -10.1 & 2.8 \\ 0.3 & 1.0 & 11,1 & , & 0.1 & -2.4 & -1.7 & -0.2 & 0.5 & -1.2 & -1.1 & -3.8 & -0,4 & -10.1 & 57.8 & -9.4 \\ -0.1 & -0.2 & -2.8 & -2.6 & -0.3 & 0.3 & -0.8 & -7.8 & -7.2 & -8.8 & 16.8 & 2.8 & 2.8 & -9.4 & 128.3 \end{bmatrix} \tag{19}$$

$$B_{0i} = 10^{-3} \times [-0.1\ -0.2\ 2.8\ -0.1\ 0.1\ -0.3\ -0.2\ -0.2\ 0,6\ 3.9\ -1.7\ -0.0\ -3.2\ 6.7\ -6.4] \tag{20}$$

$$B_{00} = 0.0055 \tag{21}$$

As parameters of the DE and MDE algorithms, we used a population NP of 75 individuals, and [13] indicate that NP must be between $5 \times D$ and $10 \times D$, where D is the dimension of the problem, with $D = 15$ in this case. The disturbance factor F and the crossing probability CR must respect the following conditions: $F \in [0; 1{,}2]$ and $CR \in [0; 1]$. According to [13], for good results, F should be from 0.5 up to 1.0, and CR can start at 0.1 to the crossover probability.

Table 1. Input data from the 15 Thermoelectric Energy Generating Units of the Economic Dispatch Problem

Unit	P_i^{min}	P_i^{max}	α_i ($)	β_i ($/MW)	γ_i ($/MW^2)
1	150	455	671	10.1	0.000299
2	150	455	574	10.2	0.000183
3	20	130	374	8.8	0.001126
4	20	130	374	8.8	0.001126
5	150	470	461	10.4	0.000205
6	135	460	630	10.1	0.000301
7	135	465	548	9.8	0.000364
8	60	300	227	11.2	0.000338
9	25	162	173	11.2	0.000807
10	25	160	175	10.7	0.001203
11	20	80	186	10.2	0.003586
12	20	80	230	9.9	0.005513
13	25	85	225	13.1	0.000371
14	15	55	309	12.1	0.001929
15	15	55	323	12.4	0.004487

In this way, we vary the value of F from 0.5 to 1 and CR from 0.1 to 1. We use the mutation $DE/rand/1/bin$, where *rand* determines a randomly chosen vector to mutate, 1 determines the value of weighted differences used for the perturbation, and *bin* specifies a binomial crossover.

Based on the results of [5], the stopping criterion used for both algorithms was the percentage relative error in the last 500 iterations of the algorithm. We check if the error for tolerance is less than 10^{-6} in this stopping criterion. In addition, we perform 20 runs for each configuration of both algorithms with a random initial population. We compare the results obtained by the DE and MDE methods to verify the performance of the MDE to the original DE. Furthermore, we compared the results obtained with those of other methods in the literature.

5.2 Non-Convex Economic Load Dispatch with 15 Generating Units

The non-convex ELD with 15 generating units can be mathematically described as a constrained non-linear, discontinuous, and non-convex optimization problem [7]. There are constraints regarding prohibited operating zones and ramp limits in the non-convex ELD with 15 generating units. This problem is challenging due to a non-convex decision space of 192 convex subspaces [9]. Therefore, the outputs of the generators have a high degree of nonlinearities and discontinuities as characteristics [7]:

Table 2. Data from the 15 Thermoelectric Energy Generating Units of the Non-Convex Economic Load Dispatch

Unit	UR_i	DR_i	P_i^0	Forbidden Zones (MW)
1	80	120	400	-
2	80	120	300	[185 225] [305 335] [420 450]
3	130	130	105	-
4	130	130	100	-
5	80	120	90	[180 200] [305 335] [390 420]
6	80	120	400	[230 255] [365 395] [430 455]
7	80	120	350	-
8	65	100	95	-
9	60	100	105	-
10	60	100	110	-
11	80	80	60	-
12	80	80	40	[30 40] [55 65]
13	80	80	30	-
14	55	55	20	-
15	55	55	20	-

$$
\begin{aligned}
&\min\ F = \sum_{i=1}^{15} F_i(P_i) \\
&\text{s.a}\ \sum_{i=1}^{15} P_i = P_D + P_L, \\
&\quad P_i^{min} \leq P_i \leq P_i^{max}, i = 1, \ldots, 15 \\
&\quad P_{i,j-1}^{u} \leq P_i \leq P_{i,j}^{l}, j = 2, \ldots, n_i \\
&\quad P_{i,n_i}^{u} \leq P_i \leq P_i^{max} \\
&\quad max(P_i^{min}, P_i^0 - DR_i \leq P_i \leq min(P_i^{max}, P_i^0 + UR_i)
\end{aligned}
\tag{22}
$$

For this system, the total power demand is $P_D = 2{,}630$ MW [5]. To satisfy the power balance constraint, the penalty factor was $\phi = 50$ [5]. The values of the minimum (P_i^{min}) and maximum (P_i^{max}) limits of each of the i ($i \in \{1, 2, \ldots, 15\}$) generating units, in addition to the parameters α_i, β_i and γ_i, which can be consulted in the Table 2 [7]. The values of the loss coefficients B_{ij}, B_{0i} and B_{00} are in the Eq. 19, Eq. 20, and Eq. 21.

Based on the results obtained by [9], the stopping criterion used for both algorithms was the maximum number of the objective function (FEs) evaluation. To control the stopping criterion FEs, we use a variable called $fitc$, incremented in each time of the evaluation of the objective function. The algorithm ends its execution when the value of $fitc$ is the value of FEs. Conform [9], the number of FEs was defined as 6×10^4. We ran 20 executions for each DE and MDE configuration, and in each execution the solution of the problem was with a random initial solution.

6 Results

In this section, we present the evaluation of MDE in relation to DE heuristic, considering the classic (Subsect. 6.1) and non-convex economic dispatch (Subsect. 6.2). In both cases, we use 15 thermoelectric power-generating units that satisfy the total demand of the system. We also compare both implementations with find results in the literature to other optimization algorithms. At the end, we show the execution time of DE and MDE.

6.1 Classic Economic Load Dispatch with 15 Generating Units

We seek the best solutions for each DE and MDE configuration, considering that the penalty factor F was varied from 0.5 to 1.0 and the crossing probability CR ranged from 0.1 to 1.0. For the DE, the best solution was 32,763.74$/h with F equal to 1 and CR equal to 0.2. For MDE, the best solution was 32,548.78 $/h with F equal to 1 and CR equal to 0.8. The best solution found by MDE is 0.66% smaller than the one found by DE. Therefore, the modification performed in the DE achieved the best result.

We compare the performance of the best solution found by the DE and MDE algorithms with the results of [5] to verify the performance of the implemented heuristics. [5] proposed three methods to solve the classic ELD with 15 units: Particle Swarm Optimization (PSO), Firefly Algorithm (FA), and the Lagrange Multiplier Method (LMM). PSO is an algorithm based on a population composed of individuals capable of interacting with each other and the environment, changing direction and speed to find the location in the search space that leads to the best [6] solution. The FA associates the light of fireflies with the objective function. Therefore, the more intense the light of the firefly, or its brightness, the better the [16] solution. LMM is a classic mathematical optimization method [11]. In comparison with the results obtained by PSO and FA and with the solution obtained via LMM, we verify the performance of the implemented heuristics. The parameters used in the execution of PSO and FA and the results used in the comparison are at [5].

Table 5 show the statistics of the total costs ($/h) for the ELD problem found by the PSO, FA, DE, and MDE algorithms. It presents the lowest cost ($/h) found, the highest cost ($/h), the average cost ($/h) of the 20 executions performed, the standard deviation of this average, and the coefficient of variation. Observing Table 5, we see that the algorithm that led to the best solution was the MDE, obtaining a minimum cost of 32,548.78 $/h. The solution obtained by MDE is approximately 1.10% smaller than the solution obtained by PSO, 0.01% smaller than the FA solution, and 0.66% smaller than the DE.

We compared the results obtained through the DE and MDE methods with the results obtained by 25 algorithms from the [9] literature to verify the performance of the implemented heuristics. Table 3 shows the lowest cost ($/h), the highest cost ($/h), the average cost ($/h), and the standard deviation of this average. Solutions not evaluated have the value marked as NA. We highlighted

Table 3. Total Costs ($/h) found by PSO [5], FA [5], ABC [1], DE, and MDE algorithms in solving the classic ELD problem with 15 generating units.

Method	Min. Cost ($/h)	Max. Cost ($/h)	Average Cost ($/h)	Std. Deviation	Var. Coefficient
PSO	32912.00	33313.11	33108.61	109.95	0.33%
FA	32551.70	32686.94	32584.91	40.08	0.12%
ABC	32894.05	33375.11	33112.15	112.30	0.33%
DE	32763.74	33134.21	33009.36	76.57	0.23%
MDE	**32548.78**	32608.87	32564.79	17.33	0.05%

Table 4. Min, max, and average costs of each method

Method	Min. Cost ($/h)	Max. Cost ($/h)	Average Cost ($/h)	Standard Deviation
PSO	32858.00	33331.00	33039.00	NA
GA	33113.00	33337.00	33228.00	NA
AIS	32854.00	32892.00	32873.25	10.81
MTS	32716.87	32796.15	32767.21	3.65
TSA	32762.12	32942.71	32822.84	60.59
SA	32786.40	33028.95	32869.51	112.32
APSO	32742.78	NA	32976.68	133.93
SA-PSO	32708.00	32789.00	32732.00	18.03
PC PSO	32775.36	NA	NA	NA
SOH-PSO	32751.00	32945.00	32878.00	NA
DSPSO-TSA	32715.06	32730.39	32724.63	8.40
MDE	32704.90	32711.50	32708.10	NA
CTPSO	32704.45	32704.45	32704.45	0.00
CSPSO	32704.45	32704.45	32704.45	0.00
COPSO	32704.45	32704.45	32704.45	0.00
CCPSO	32704.45	32704.45	32704.45	0.00
MPSO	32738.42	NA	NA	NA
ABC	32707.85	32708.27	32707.95	NA
FA	32704.50	33175.00	32856.10	141.17
SWT-PSO	32704.45	NA	NA	NA
MsEBBO	32692.40	32692.40	32692.40	0.00
SQPSO	32704.57	32711.62	32707.08	NA
IA EDP	32698.20	32823.78	32750.22	9.30
IODPSO-G	32692.39	32692.39	32692.39	0.00
IODPSO-L	32692.39	32692.39	32692.39	0.00
DE	32804.11	33076.68	32996.08	75.84
MDE	32706.66	32719.72	32709.76	4.22

DE and MDE solutions in bold. Observing Table 3, we see that the DE and MDE algorithms failed to obtain the best result compared to the literature.

Observing the standard deviation values in Table 4, we verify that the PSO algorithm presents the large variability of solutions between one execution and another. However, as the coefficient of variation for the PSO is 0.33%, the

variability can be considered small. Therefore, all algorithms present a low variability of solutions found, where the data set present low dispersion since all coefficients of variation are below 0.34% about the mean.

The computational cost of each algorithm is shown in Table 4 and measured by the total number of iterations executed. We measure the computational cost from the total number of iterations executed. Table 4 presents the minimum number of iterations, the maximum and average of the iterations performed, the standard deviation of this average, and the coefficient of variation. Observing Table 4, we see that the MDE algorithm, which presents the best solution for the DE, performs an average of iterations approximately 2.90 times greater than the original DE and 1.15 times greater than the PSO. Although the average computational cost of the MDE is higher than that of the DE and PSO algorithms, the MDE manages to obtain the lowest cost ($/h) for the DE problem. Therefore, for the stopping criterion used, the DE and PSO algorithms cannot converge to the best solution. An alternative to these algorithms would be to use a different stopping criterion, such as the maximum number of the objective function evaluations [9].

In relation to the FA algorithm, the MDE performs an average of iterations approximately 3.5 times small. Therefore, the MDE method can find the best solution with less computational effort in relation to the FA, which is the most expensive algorithm.

Observing the standard deviation values presented in Table 4, we verify that the variability of the number of iterations between one execution and another is significantly greater for the FA algorithm than other algorithms. However, both algorithms have large variability since the value of the coefficient of variation is more than 25%. It means that the number of iterations required until convergence varies from one to the next run. Therefore, the dispersion of the data to the mean is large.

6.2 Non-Convex Economic Load Dispatch with 15 Generating Units

In order to make a comparison with the results found in the literature, we used a population NP of 40 individuals [9] as parameters for the DE and MDE. As in the previous test case, the disturbance factor F was varied from 0.5 to 1 and the crossing probability CR was varied from 0.1 to 1. Therefore, it can select parameters that lead to the best result. As in the previous test case, we used the mutation $DE/rand/1/bin$. It is the most general and used mutation form [3]. The conduction of results evaluations are to verify the performance of the MDE to the original DE. Furthermore, we compared the results obtained by the DE and MDE methods with those obtained by other methods in the literature.

Considering that the penalty factor F ranges from 0.5 to 1.0 and the crossing probability CR ranged from 0.1 to 1.0 the best solution obtained by each algorithm is in bold highlight. For DE, the best solution of 32,804.11 $/h considers F equal to 1 and CR equal to 0.9. For MDE, the best solution was 32,706.66 $/h with F equal to 1 and CR equal to 0.8. The best solution found by MDE is

Table 5. Comparison of Different Methods in Solving Non-Convex DE with 15 Generating Units

Method	Min Cost ($/h)	Max Cost ($/h)	Average Cost ($/h)	Std. Deviation
PSO	32,858.00	33,331.00	33,039.00	NA
GA	33,113.00	33,337.00	33,228.00	NA
AIS	32,854.00	32,892.00	32,873.25	10.81
MTS	32,716.87	32,796.15	32,767.21	3.65
TSA	32,762.12	32,942.71	32,822.84	60.59
SA	32,786.40	33,028.95	32,869.51	112.32
APSO	32,742.78	NA	32,976.68	133.93
SA-PSO	32,708.00	32,789.00	32,732.00	18.03
PC PSO	32,775.36	NA	NA	NA
SOH-PSO	32,751.00	32,945.00	32,878.00	NA
DSPSO-TSA	32,715.06	32,730.39	32,724.63	8.40
MDE	32,704.90	32,711.50	32,708.10	NA
CTPSO	32,704.45	32,704.45	32,704.45	0.00
CSPSO	32,704.45	32,704.45	32,704.45	0.00
COPSO	32,704.45	32,704.45	32,704.45	0.00
CCPSO	32,704.45	32,704.45	32,704.45	0.00
MPSO	32,738.42	NA	NA	NA
ABC	32,707.85	32,708.27	32,707.95	NA
FA	32,704.50	33,175.00	32,856.10	141.17
SWT-PSO	32,704.45	NA	NA	NA
MsEBBO	32,692.40	32,692.40	32,692.40	0.00
SQPSO	32,704.57	32,711.62	32,707.08	NA
AI EDP	32,698.20	32,823.78	32,750.22	9.30
IODPSO-G	32,692.39	32,692.39	32,692.39	0.00
IODPSO-L	32,692.39	32,692.39	32,692.39	0.00
DE	**32,804.11**	**33,076.68**	**32,996.08**	**75,84**
MDE	**32,706.66**	**32,719.72**	**32,709.76**	**4,22**

0.30% smaller than the one found by DE. Therefore, the modification performed in the DE algorithm enables to find the best result, proving to be more effective.

We compared the results obtained through the DE and MDE methods with the results obtained by 25 algorithms from the [9] literature to verify the performance of the implemented heuristics. Table 5 shows the lowest cost ($/h), the highest cost ($/h), the average cost ($/h), and the standard deviation of this average. We marked as NA the values of solutions not evaluated. Solutions from DE and MDE are in bold highlights. Observing Table 5, we see that the DE and MDE algorithms failed to obtain the best result compared to the literature.

The DE obtained the best solution compared only to the PSO, GA, and AIS methods. It is up to 0.93% better. The MDE obtained the best solution

against the PSO, GA, AIS, MTS, TSA, SA, APSO, SA-PSO, PC PSO, SOH-PSO, DSPSO-TA, MPSO, and ABC methods, being up to 1.23% better as can be seen in Table 5. The best solution of 32,692.39 $/h for this case was obtained using the IODPSO-G and IODPSO-L methods implemented by [9], which are PSO variants. To the best solution obtained by the literature, the solutions obtained by the DE and MDE methods are 0.34% and 0.04% worse, respectively.

We analyze the computation time of the DE and MDE algorithms to verify the efficiency of the modification performed to the original version. This comparison did not carry out with the other methods in the literature because they ran in different architectures. The computational cost was evaluated by the average execution time (in seconds) of the 20 executions. We see that the average cost of DE is 23.31 s, and the MDE had 0.93 s. Therefore, although the MDE does not obtain the best solution concerning the literature, this method is quite efficient to the original DE, presenting a reduction in the average execution time of 95.98%.

7 Conclusion and Future Works

This article presented a Modification of the Differential Evolution algorithm (MDE), which belongs to the class of Evolutionary Computation algorithms. In the original Differential Evolution (DE) algorithm, each possible solution to the target problem represents an individual in the population. The population suffers from genetic operators of mutation, crossover, and greedy selection. At each iteration, a newly generated population replaces the discarded old one. The step of greedy selection is responsible for choosing the fittest elements and inserting them into the new population. In the proposed MDE, we modify the selection step. Instead of inserting the fittest individual into the new population, the old one gives place to the fittest elements in the current population. In this way, the population is updated in the greedy selection stage and not replaced at the end of the iteration.

We apply the algorithms to an Economic Load Dispatch problem to find the generation level of each thermoelectric energy generating unit that satisfies the total demand of the system at a minimum cost and to satisfy the problem constraints. Furthermore, we saw that restrictions such as forbidden zones, ramp limits, and valve points make the problem non-convex and hard to solve by classical optimization methods.

We solve the ELD problem by considering different sizes and restrictions to verify the performance of the algorithms applied to different scenarios. Initially, we solved the ELD problem with 15 thermoelectric energy generating units, considering the classic constraints of power balance, transmission line losses, and minimum and maximum operating limits for each one. Our results showed that the MDE algorithm obtained the lowest cost ($/h) for the ELD compared to the other optimization algorithms. The best solution found by MDE is up to 1.10% better than DE and other heuristics in the literature. Therefore, we concluded that the MDE is the most effective method concerning others algorithms. In

addition, we conclude that solving the DE via LMM becomes costly due to the great system of nonlinear equations and the initial effort in formulating the problem and manually inserting the system in Matlab. Therefore, the DE using an optimization heuristic is more efficient to be solved.

We solve an ELD problem with 15 units, which in addition to considering the classic constraints of the issue, also considers forbidden zones and ramp limits. The restriction of forbidden zones results in a non-convex decision space and is difficult to be solved by classical methods. Our results showed that the MDE algorithm obtained the lowest cost ($/h) if compared to the DE, being 0.30% better. Compared to other methods in the literature, MDE failed to obtain the best solution, presenting a result up to 0.043% worse.

In future works, we intend to solve the Economic Dispatch Problem by considering another instance of the problem. We want to use the specifications of a thermoelectric power plant. Soon, we will be able to verify if the performance of the MDE concerning the DE is maintained. Furthermore, we intend to implement a version of MDE combined with another optimization algorithm like PSO because hybrid versions have shown very effective results in the literature.

References

1. Andrade, G.L.: Algoritmo evolução diferencial modificado aplicado ao problema do despacho econômico de carga. Master thesis, Federal University of Pampa, Alegrete (2020)
2. BEN: Balanço energético nacional 2018: Ano base 2017/empresa de pesquisa energética. Technical report, EPE, Rio de Janeiro (2018)
3. De Carvalho, L., Morais, M.d.F., Ceolho, L.d.S., Da Rocha, R.P., Beline, E.L.: Evolução diferencial: Características dos métodos de solução para programação em ambientes flow shop permutacional. In: Anais do XXXVI Encontro Nacional de Engenharia de Produção. ABEPRO, João Pessoa, PB (2016)
4. De Oliveira, K.W.R.C.B., Nascimento Jr., N.T., Saavedra, O.R.: Uma abordagem via estratégias evolutivas para o despacho econômico considerando restrições de geração. IEEE Latin America Trans. **6**(1), 42–50 (2008)
5. Dos Santos, J.O., Kapelinski, K., dos Santos, E.M., Juchem Neto, J.P.: Resolução de um problema de despacho econômico de carga utilizando enxames de partículas e vaga-lumes. Proc. Ser. Braz. Soc. Comput. Appl. Math. **6**(1) (2018)
6. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: 1995 Proceedings of the 6th International Symposium on Micro Machine and Human Science, MHS 1995, pp. 39–43. IEEE (1995)
7. Gaing, Z.L.: Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans. Power Syst. **18**(3), 1187–1195 (2003)
8. Gomez-Exposito, A., Conejo, A.J., Canizares, C.: Electric Energy Systems: Analysis and Operation. CRC Press (2009)
9. Qin, Q., Cheng, S., Chu, X., Lei, X., Shi, Y.: Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization. Appl. Soft Comput. **59**, 229–242 (2017)
10. Serapião, A.B.d.S.: Fundamentos de otimização por inteligência de enxames: uma visão geral. Sba: Controle & Automação Sociedade Brasileira de Automatica **20**(3), 271–304 (2009)

11. Simon, C.P., Blume, L.: Matemática para Economistas. Bookman, Porto Alegre - RS (2004)
12. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)
13. Storn, R., Price, K.: Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. ICSI, Berkeley (1995)
14. Tolmasquim, M.T.: Energia termelétrica: gás natural, biomassa, carvão e nuclear. Rio de Janeiro: EPE (2016)
15. Walters, D.C., Sheble, G.B.: Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans. Power Syst. **8**(3), 1325–1332 (1993)
16. Yang, X.S., Hosseini, S.S.S., Gandomi, A.H.: Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. Appl. Soft Comput. **12**(3), 1180–1186 (2012)