sustainability

MDPI

Article

Modified Social Group Optimization to Solve the Problem of Economic Emission Dispatch with the Incorporation of Wind Power

Dinu Calin Secui [1], Cristina Hora [1,*], Codruta Bendea [1], Monica Liana Secui [2], Gabriel Bendea [1] and Florin Ciprian Dan [1]

[1] Department of Energy Engineering, Faculty of Energy Engineering and Industrial Management, University of Oradea, 410058 Oradea, Romania; gbendea@uoradea.ro (G.B.); florin.dan@uoradea.ro (F.C.D.)

[2] Department of Psychology, Faculty of Social-Humanistic Sciences, University of Oradea, 410058 Oradea, Romania; msecui@uoradea.ro

* Correspondence: chora@uoradea.ro

Abstract: Economic dispatch, emission dispatch, or their combination (EcD, EmD, EED) are essential issues in power systems optimization that focus on optimizing the efficient and sustainable use of energy resources to meet power demand. A new algorithm is proposed in this article to solve the dispatch problems with/without considering wind units. It is based on the Social Group Optimization (SGO) algorithm, but some features related to the selection and update of heuristics used to generate new solutions are changed. By applying the highly disruptive polynomial operator (HDP) and by generating sequences of random and chaotic numbers, the perturbation of the vectors composing the heuristics is achieved in our Modified Social Group Optimization (MSGO). Its effectiveness was investigated in 10-unit and 40-unit power systems, considering valve-point effects, transmission line losses, and inclusion of wind-based sources, implemented in four case studies. The results obtained for the 10-unit system indicate a very good MSGO performance, in terms of cost and emissions. The average cost reduction of MSGO compared to SGO is 368.1 $/h, 416.7 $/h, and 525.0 $/h for the 40-unit systems. The inclusion of wind units leads to 10% reduction in cost and 45% in emissions. Our modifications to MSGO lead to better convergence and higher-quality solutions than SGO or other competing algorithms.

Keywords: economic/emission dispatch; wind power; logistic map; highly disruptive polynomial operator; optimization

check for updates

Citation: Secui, D.C.; Hora, C.; Bendea, C.; Secui, M.L.; Bendea, G.; Dan, F.C. Modified Social Group Optimization to Solve the Problem of Economic Emission Dispatch with the Incorporation of Wind Power. *Sustainability* **2024**, *16*, 397. https://doi.org/10.3390/su16010397

Academic Editor: Yagang Zhang

Received: 21 September 2023
Revised: 3 November 2023
Accepted: 6 November 2023
Published: 2 January 2024

1. Introduction

The power generation sector has been undergoing continuous development in recent years, with a focus on diversification of energy sources and production technologies, but also on efficiency and constant innovation. Decision makers aim through regulations and policies to create a competitive and attractive framework for investors while ensuring the sustainable development of the energy system [1]. In order for electricity producers and companies to remain in a competitive market, it is necessary for them to operate with the lowest possible costs and to use sources/technologies with the lowest possible environmental impact. One way to optimize operating costs while taking emissions into account is to solve the economic emission dispatch (EED) problem [2]. The goal of the EED problem is to determine the optimal operating mode of energy sources to minimize the two objectives—cost and emissions—considering a given power demand, as well as the operating limits of the generating units [3]. If the EED problem aims only at cost optimization (without considering emissions), then this is called the economic dispatch problem (EcD) [4]. If EED only aims to optimize emissions (without considering costs), then it is called the emission dispatch (EmD) problem [5]. For a more complete approach, both the economic aspects and the level of emissions released into the atmosphere must be taken into account, which implies solving the EED problem.

Initially, the solution to EcD, EmD, and EED problems focused mainly on conventional unit power plants (firing coal, gas, etc.), as they had a very large share of the power system structure. However, last year's warnings about the environmental impact of these types of power plants led to an increased use of both high-efficiency co-generation units (CU) (that simultaneously produce both electricity and heat), and renewable energy power generation systems (RESPSs) to steer the electricity generation sector towards sustainable development [6,7]. The EED problem that includes CU aims at the optimal scheduling (in terms of cost and emissions) of power-only units, heat-only units, and their combination: co-generation units [8]. It is a similar situation for EcD [9] or EmD [8] problems, with both aiming to manage all categories of units operating in the power system: power-only, heat-only, and CU units. For systems that include CU units, in addition to costs and constraints specific to power-only units, the EED optimization model must also consider the fuel costs related to heat-only and CU units, power-heat dependencies, and the produced-demand heat balance [10]. In the case of including RESPSs in the power system, an important option is the use of wind energy, which can be converted into electricity without producing greenhouse gases. Wind energy is considered a renewable and clean energy source, having only a low secondary impact on the environment due to the manufacturing process of the equipment and its transport. Increasing the share of wind energy in the energy mix has a positive impact on the quality of the environment and reduces dependence on conventional sources. Since wind is an uncontrollable and variable source of energy, the energy production from wind units is influenced by weather conditions and wind speed. Thus, the operation of power systems that include wind units must also take into account the unpredictable variations in the power of these units [4].

The integration of wind units into EcD, EmD, or EED problems brings challenges related to wind speed distribution, mathematical optimization models, and solution algorithms. One of the most common parametric distributions used in wind speed modeling is the Weibull distribution [11–13], but other distributions can be considered, such as [14] Gamma, inverse Gamma, Gaussian, Burr, Halphen, etc.

In the following, some articles are presented which propose mathematical models and solution algorithms for solving EcD, EmD, or EED problems considering the uncertainties caused by unpredictable wind speed fluctuations. Thus, in [4], an optimization model is presented that includes the costs related to the overestimation and underestimation of the wind power. The case study considers a Weibull distribution of the wind speed, and it shows that the optimal solutions to the EcD problem can be influenced by factors associated with the overestimation and underestimation of wind power. Based on a linear relationship between wind speed and wind power, and considering a Weibull distribution for wind speed, in [11], an optimization model is developed for the minimization of emissions, in which the uncertainty of the wind power is included in the constraints of the model. In [12], both the cost of emissions and the cost of overestimation and underestimation of wind power are included in the bi-objective EED problem. Starting from the classical formulation of EcD and EmD problems, in [2], the objective functions related to costs and emissions are extended with terms that include the uncertainty of the wind power. The resulting models are tested on a system consisting of two conventional units and two wind units, considering a mixed Gamma-Weibull distribution for the wind speed. To solve the EED problem with the inclusion of wind power, in [15], the Honey Bee Mating Optimizer (HBMO) is applied, where a so-called 2m-point model is used to estimate the uncertainty of the wind power. In [16], a new evolutionary technique called Lightning Flash Algorithm (LFA) is proposed to solve the bi-objective EED problem by considering different levels of wind power penetration and multiple fuel sources. The LFA technique is efficient for the EED problem, having a good convergence and achieving lower costs and emissions than other algorithms. A robust and efficient optimization model for dispatching wind-thermal power under uncertainties is developed in [1], taking into account robustness, economics, and environmental aspects. In [17], another robust model for the optimization of the EcD problem is presented. It is based on the identification of a set of discrete bad scenarios, where the objective function aims at minimizing all the penalties associated with the bad scenarios.

The EcD, EmD, or EED problems that include or do not include wind power uncertainty are non-convex with multiple local optima and nonlinear constraints [12,15,16] requiring advanced solution algorithms.

For the study of both systems (the ones comprising conventional-only units and the ones comprising thermal-wind units), several algorithms are presented in the following lines. Considering the first category of systems, in [18], an Improved Class Topper Optimization (ICTO) has been developed by including in the classical CTO three new concepts: adaptive improvement factor, adaptive acceleration coefficient, and chaos local search, which help to enhance the exploration and exploitation capability of the ICTO. The ICTO is tested on five systems with conventional units, and it performs better, in terms of cost and emissions, than the original CTO and several other competing algorithms. A quasi-oppositional-based political optimizer is used, in [19], to solve the bi-objective EED problem considering valve-point loading effect, transmission line losses, and other constraints related to conventional units. Recently, two new population-based algorithms (Criminal Search Optimization Algorithm (CSOA) [20] and Kho-Kho optimization algorithm (KKO) [21]) have been developed and applied to solve the EED problem. Both algorithms (CSOA and KKO) show good exploitation and exploration capabilities, and they can be used to solve complex problems in various domains. Another option for solving the EED problem is presented in [22], where two metaheuristic algorithms, Exchange Market Algorithm (EMA) and Adaptive Inertia Weight Particle Swarm Optimization (AIWPSO), are combined to obtain a new algorithm with improved global and local search abilities. The constraints of the EED problem are maintained using the multiple constraints ranking technique. The best compromise solution (BCS) obtained using EMA-AIWPSO dominates the BCSs obtained using other algorithms (such as KKO, ISA, or GSA). Also, several multi-objective algorithms have been proposed for solving the EED problem, such as multi-objective SSA [23], multi-objective cultural algorithm [24], NSGA-III algorithm [25], or multi-objective quasi-oppositional TLBO [26], each of which uses a basic metaheuristic algorithm (SSA, CA, GA, or TLBO) that is endowed with the Pareto-dominance principle to generate successive Pareto fronts. The multi-objective algorithms mentioned [23–26] have been tested on various power systems with conventional units, their performance being superior to other recognized multi-objective techniques (MODE, NSGA II, or SPEA-2).

Over the time, when trying to solve the EED problem, various techniques have been used to optimally program the units of the thermal-wind systems. Thus, in [6], the techniques of weighted goal programming and the progressive bounded constraint method are combined to generate a set of Pareto solutions that are efficient in the cost–emission space, and then extract the best compromise solution. For instance, four cases are analyzed to quantify and demonstrate the benefits of including wind units in the structure of a power system comprising only thermal units. In [12], the Artificial Bee Colony (ABC) algorithm is strengthened by including the best solution in the update equations, and in [27], chaos is inserted into the sine–cosine algorithm to obtain better-quality solutions to the EED problem. Additionally, in [13]—where wind units are considered—a number of eight metaheuristic algorithms (Flower Pollination Algorithm (FPA), Mine Blast Algorithm (MBA), Backtracking Search Algorithm (BSA), Symbiotic Organisms Search (SOS), Ant Lion Optimizer (ALO), Moth-Flame Optimization (MFO), Stochastic Fractal Search (SFS), and Lightning Search Algorithm (LSA)) are applied to identify the best solutions to the EcD, EmD, or EED problem. The FPA and BSA algorithms provide the best scheduling of thermal/wind units for cost and/or emission minimization. To study the behavior of systems in which thermal, wind, and solar units operate, in [28], the Dragonfly Algorithm (DA) is proposed for the EcD problem. Uncertainties related to the power generated by using wind and solar energy are modeled using the 2-m point estimation technique. The

DA algorithm outperforms other algorithms, such as Crow Search Algorithm (CSA), Ant Lion Optimizer (ALO), oppositional RCCRO, Biogeography-Based Optimization (BBO), PSO, and GA, in terms of cost and execution time.

SGO is a metaheuristic algorithm proposed relatively recently by Satapathy SC (2016) [29], which is based on the fact that a social group of individuals has a greater ability to solve a real-life problem than a single individual. For some mathematical functions, SGO is more efficient than other well-known metaheuristic algorithms (such as [29]: DE, PSO, ABC, GA, TLBO, etc.), being an easy-to-implement algorithm, having two main phases (improving phase and acquiring phase) and a single specific parameter. However, for some mathematical functions or applications, it can provide a low performance due to an imbalance between exploration and exploitation, which ultimately leads to getting stuck in a local optimum [30]. Also, SGO may suffer due to the reduced diversity of the population [31]. To overcome the mentioned shortcomings and to obtain better quality solutions, there have been attempts to improve the performance of SGO by different strategies, such as modifying the relations for updating the solutions [30], inertia weight strategies [32], and hybridizations with other algorithms [31].

In this paper, some changes have been made to the original SGO algorithm, aiming to increase efficiency, resulting in the MSGO algorithm. In order to improve the exploration–exploitation balance, the commutation condition between the equations for updating the solutions in the acquiring phase was changed. Also, to increase the diversity of the population, a logistic map and a highly disruptive polynomial operator were inserted.

The main contributions of this paper are:

- We propose a modified version of SGO (called MSGO) in which the way of updating and adapting the individuals in the social group is changed by inserting chaos and an HDP operator (in the original SGO only uniformly randomly generated number sequences are used). The operators associated with chaos and HDP aim at increasing the efficiency of the MSGO algorithm by reducing the number of close solutions and overcoming some drawbacks related to slow convergence. To the best of the authors' knowledge the HDP operator has never been used to improve the performance of the SGO algorithm.
- Implementation of MSGO to solve EcD, EmD, and EED problems with or without consideration of wind units.
- Conducting experiments to evaluate and statistically compare the effectiveness of MSGO with SGO and other well-known algorithms (or their varieties) for thermal or wind-thermal power systems of different sizes and characteristics.

2. Statement of the EcD, EmD, and EED Problems

2.1. Statement of the EcD Problem

The EcD problem for a power system that includes thermal and wind units aims to determine the power outputs of these generating units so that the operating cost of the entire system is minimized, and a number of technical constraints are met. For the formulation of the mathematical optimization model, we consider a power system comprising Nt thermal units and Nw wind units, and the total power demand (P_D) of the consumers in the system is considered known and constant for the period of analysis. The variables to be optimized are continuos, being represented by the output power vectors of the thermal units $PT = [PT_1, PT_2, \ldots, PT_i, \ldots, PT_{Nt}]$ and of wind units $PW = [PW_1, PW_2, \ldots, PW_j, \ldots, PW_{Nw}]$.

The objective function $C(PT, PW)$ is represented by two components; one related to the fuel cost of the thermal units, $C^T(PT)$, and the other related to the operating cost of the wind units, $C^W(PW)$ [4]:

$$C(PT, PW) = C^T(PT) + C^W(PW) \tag{1}$$

$$C^T(PT) = \sum_{i=1}^{Nt} C_i^T(PT_i) \tag{2}$$

$$C^W(PW) = \sum_{j=1}^{Nw} C_j^W(PW_j) = \sum_{j=1}^{Nw} \left\{ c_j^d \cdot PW_j + c_j^o \cdot E_j^o(PW_j) + c_j^u \cdot E_j^u(PW_j) \right\} \quad (3)$$

where, ${C^T}_i(PT_i)$ is the fuel cost corresponding to thermal unit i, and ${C^W}_j(PW_j)$ is the operating cost corresponding to the wind unit j.

In this paper, the relationship between the fuel cost ${C^T}_i(PT_i)$ and output power PT_i of a thermal unit i is modeled via a non-convex function consisting of a quadratic and a sinusoidal term [33]:

$$C_i^T(PT_i) = a_i PT_i^2 + b_i PT_i + c_i + |e_i sin(f_i(PT_{min,i} - PT_i))|, \; i = 1, 2, \ldots, Nt \quad (4)$$

The operating cost of wind unit j, ${C^W}_j(PW_j)$, consists of three terms [4]: the direct operating cost of unit j ($c_j^d \cdot PW_j$), the cost corresponding to the overestimation of the wind power ($c_j^o \cdot E_j^o(PW_j)$), and the cost corresponding the underestimation the wind power ($c_j^u \cdot E_j^u(PW_j)$).

The wind power overestimation for unit j occurs if the estimated wind power for this unit PW_j is higher than the available power represented by a random variable W_j. In this situation, the difference between the two powers is covered by a reserve source and implies the reserve cost $c_j^o \cdot E_j^o(PW_j)$. In the opposite situation of the underestimation (if PW_j is less than W_j), a part of the available power of unit j remains unused, which implies a penalty cost $c_j^u \cdot E_j^u(PW_j)$. Since W_j is a random variable, the powers $E_j^o(PW_j)$ and $E_j^u(PW_j)$ will include the uncertainty of the wind power, and their calculation method is presented below.

For the calculation of the average powers $E_j^o(PW_j)$ and $E_j^u(PW_j)$, it is considered that the wind speed is modeled by a Weibull distribution whose probability density function (pdf) is given by relation (5) [11]:

$$f_V(v) = \frac{k}{c}\left(\frac{v}{c}\right)^{k-1} exp\left[-\left(\frac{v}{c}\right)^k\right] \quad (5)$$

where, V is the random variable wind speed, v denotes the wind speed (a value of V), and $f_V(v)$ is the pdf of the variable V, k is the shape parameter, and c is the scale parameter.

Also, between the random variable wind power (W) and the random variable wind speed (V), we consider a linear relationship expressed as follows [11]:

$$W = \begin{cases} 0, & if \; V < v_{in} or \; V > v_{out} \\ \frac{(V - v_{in})PWr}{v_r - v_{in}}, & if \; v_{in} \le V \le v_r \\ PWr, & if \; v_r \le V \le v_{out} \end{cases} \quad (6)$$

The use of the linear model (6) requires knowledge of three limit speeds: cut-in wind speed (v_{in}), rated wind speed (v_r), and cut-out wind speed (v_{out}). PWr is the rated output power of the wind unit, which corresponds to the speed v_r.

The pdf for wind power (W) over the continuous interval (0, PWr) has the expression:

$$f_W(w) = \frac{k \cdot v_{in} \cdot R}{c \cdot PWr}\left(\frac{v_{in}}{c} \cdot \left(1 + \frac{w \cdot R}{PWr}\right)\right)^{k-1} exp\left[-\left(\frac{v_{in}}{c} \cdot \left(1 + \frac{w \cdot R}{PWr}\right)\right)^k\right] \quad (7)$$

where $R = (v_r - v_{in})/v_{in}$. Given relation (6) and the Weibull distribution of the wind speed, the event probabilities $W = 0$ and $W = PWr$ are calculated using relations [4]:

$$Prob(W = 0) = Prob(V < v_{in}) + Prob(V > v_{out}) = 1 - exp[-(v_{in}/c)^k] + exp[-(v_{out}/c)^k] \quad (8)$$

$$Prob(W = PWr) = Prob(v_r < V < v_{out}) = exp[-(v_r/c)^k] - exp[-(v_{out}/c)^k] \quad (9)$$

We must say that, in general, the distributions of the variables V and W, as well as the pdfs derived from them, differ depending on the location of the wind turbine units.

Thus, for wind unit j, the average powers associated with the overestimation and the underestimation of the wind power are mathematically expressed as follows [4]:

$$E_j^o\left(PW_j\right) = \int_0^{PW_j} \left(PW_j - w\right) f_{W_j}(w) dw + \left(PW_j - 0\right) \cdot \text{Prob}\left(W_j = 0\right) \tag{10}$$

$$E_j^u\left(PW_j\right) = \int_{PW_j}^{PWr_j} \left(w - PW_j\right) f_{W_j}(w) dw + \left(PWr_j - PW_j\right) \cdot \text{Prob}\left(W_j = PWr_j\right) \tag{11}$$

where $f_{W_j}(w)$ is probability density function for the random variable W_j having the form given by relation (7), while w are the values of the continuous random variable W_j. The discrete probabilities $Prob\left(W_j = 0\right)$ and $Prob(W_j = PW_{rj})$, for unit j, are calculated using relations (8) and (9), where PWr_j represents rated wind power for unit j.

In Table A1 are presented the steps to evaluate the cost related to wind power, as well as a calculation example for a wind unit.

The feasible space of EcD problem solutions is limited by the following constraints [4,12]:

1. The thermal units i must be operated between the minimum capacity ($PT_{min,i}$) and the maximum capacity ($PT_{max,i}$):

$$PT_{min,i} \leq PT_i \leq PT_{max,i},\ i = 1, 2, \ldots, Nt \tag{12a}$$

If the power of the thermal unit i in the previous hour is known or specified (PT_i^0), then the operating limits are given by the constraint:

$$Max\left(PT_{min,i},\ PT_i^0 - DR_i\right) \leq PT_i \leq Min\left(PT_{max,i},\ PT_i^0 + UR_i\right),\ i = 1,\ 2, \ldots,\ Nt \tag{12b}$$

where DR_i and UR_i are the down-ramp and up-ramp limits of the unit i.

2. The wind units j must be operated between the minimum ($PW_{min,j}$) and maximum ($PW_{max,j}$) capacity:

$$PW_{min,j} \leq PW_j \leq PW_{max,j},\ j = 1, 2, \ldots, Nw \tag{13}$$

3. The actual powers generated by the thermal and wind units must cover the power consumed in the system:

$$\sum_{i=1}^{Nt} PT_i + \sum_{j=1}^{Nw} PW_j - P_L - P_D = 0 \tag{14}$$

where P_D is the load demand of the system, and P_L represents the transmission line losses, which can be determined considering the constant B coefficients formula:

$$P_L = \sum_{i=1}^{Nt+Nw} \sum_{j=1}^{Nt+Nw} PTW_i \cdot B_{ij} \cdot PTW_j + \sum_{i=1}^{Nt+Nw} B_{0i} \cdot PTW_i + B_{00} \tag{15}$$

where B_{ij} is an element of the loss coefficients matrix, B_{0i} is i element of the loss coefficients vector, and B_{00} is the loss coefficient constant; $PTW = [PTW_1, PTW_2, \ldots, PTW_i, \ldots, PTW_{Nt+Nw}]$ is a vector that combines the powers of thermal PT and wind PW units, while PTW_i represents the i^{th} component of the PTW vector.

2.2. Statement of the EmD Problem

The EmD problem has a high level of similarity with the EcD problem, aiming to determine the PT and PW vectors, so that the emissions released into the atmosphere is minimal while maintaining some technical constrains at the units and system level. Thus, in the EmD problem, the variables to be optimized (PT and PW vectors), as well as the constraint relations (12)–(15), are identical to those in the EcD problem. The objective function is represented by the total amount of emissions released into the atmosphere, mathematically defined by the following relation [2]:

$$E(PT, PW) = E^T(PT) + E^W(PW) \tag{16}$$

$$E^T(PT) = \sum_{i=1}^{Nt} E_i^T(PT_i) \tag{17}$$

$$E^W(PW) = \sum_{j=1}^{Nw} E_j^W(PW_j) = \sum_{j=1}^{Nw} \left\{ e_j^o \cdot E_j^o(PW_j) + e_j^u \cdot E_j^u(PW_j) \right\} \tag{18}$$

where $E_i^T(PT_i)$ and $E^T(PT)$ represent the pollutant emissions released into the atmosphere due to the operation of thermal unit i, respective of all thermal units in the analyzed power system. $E_j^W(PW_j)$ and $E^W(PW)$ are the pollutant emissions produced due to the need to use other thermal units to cover the uncertainty of the availability of wind unit j, respective of all wind units considered. Average powers, $E_j^o(PW_j)$ and $E_j^u(PW_j)$, are calculated with relations (10) and (11), which include the uncertainty of wind power in the estimation of pollutant emissions.

The term $e_j^o \cdot E_j^o(PW_j)$ represents the emissions released into the environment due to the need to use some thermal units in the system to cover the difference between the scheduled power of the wind unit PW_j (power that cannot be realized due to the unavailability of the wind resource) and its available power W_j (the case of overestimation of wind power). The term $e_j^u \cdot E_j^u(PW_j)$ represents the emissions released into the environment by other thermal units due to the non-use of the full available power of wind unit j (the case of underestimation of wind power). The quantity of emissions $E_i^T(PT_i)$ released into the atmosphere by a thermal unit i can be defined by relation [2]:

$$E_i^T(PT_i) = \sum_{i=1}^{Nt} (\gamma_i + \beta_i PT_i + \alpha_i PT_i^2 + \delta_i exp(\lambda_i PT_i)) \tag{19}$$

2.3. Statement of the EED Problem

The EED problem is similar to the EcD and EmD problems, in that the variables to be optimized and the constraints of the problem are the same, but the objective function is different. In the EED problem, the objective function $\Phi(PT, PW)$ can be formed by the weighted and normalized summation of the cost $C(PT, PW)$ and emissions $E(PT, PW)$ objectives [34]:

$$\Phi(PT, PW) = \omega \frac{C(PT, PW) - C_{min}}{C_{max} - C_{min}} + (1 - \omega) \frac{E(PT, PW) - E_{min}}{E_{max} - E_{min}} \tag{20}$$

where C_{min}, C_{max} are the minimum and maximum costs corresponding to the function $C(PT, PW)$; E_{min}, E_{max} are the minimum and maximum emissions corresponding to the function $E(PT, PW)$; ω and $(1 - \omega)$ are weighting factors associated with the normalized cost and emissions objectives, $0 \le \omega \le 1$.

To use relation (20) it is necessary to calculate the minimum (C_{min} and E_{min}) and maximum (C_{max} and E_{max}) values for the two objectives. The determination of the minimum values (C_{min}, E_{min}) is done by first solving the EcD and EmD problems formulated in Sections 2.1 and 2.2. The maximum values (C_{max}, E_{max}) are determined while considering that the EcD and EmD problems have opposite tendencies [35]. Thus, the solution for which a minimum cost is obtained in the EcD problem will be considered to result in maximum emissions, and vice versa.

In current practice, system operators or decision makers look for a single solution that takes into account both objectives, which is generally the best compromise solution. By solving the EED problem for different values of the ω factor, one can estimate the set of non-dominant solutions that form the discrete Pareto front [5] (from which the BCS between the two objectives is extracted). To extract the BCS from the set of solutions belonging to the Pareto front, a fuzzy approach is used [5]. Thus, the solutions in the Pareto front are ordered according to one of the objectives, then for each objective i and non-dominant solution s, a fuzzy membership function $\mu_{i,s}$ is assigned. This is defined by relation [5]: $\mu_{i,s} = (f_{i,max} - f_{i,s})/(f_{i,max} - f_{i,min})$, $i = \{1, 2\}$ and $s = \{1, 2,\ldots, P\}$, where $f_{i,min}$ and $f_{i,max}$ are the minimum and maximum value of the i^{th} objective function; $f_{i,s}$ is the value of the objective function corresponding to solution s and objective i; P is the number of non-dominated solutions from the Pareto front. In order to demonstrate the merit of all the objectives corresponding to solution s, the normalized membership function μ_S^* is calculated [5]:

$$\mu_s^* = \frac{\sum_{i=1}^{2} \mu_{i,s}}{\sum_{s=1}^{P} \sum_{i=1}^{2} \mu_{i,s}},\ s = 1, 2, \ldots, P \tag{21}$$

The BCS corresponds to the maximum value of the index μ_S^*: $\mu_{max} = Max(\mu_S^*, s = 1, 2,\ldots, P)$.

3. The Modified SGO Algorithm

3.1. Classic SGO

The SGO is a metaheuristic, population-based algorithm that is inspired by human social group behavior and is used for solving complex problems [29]. In the SGO, the population consists of a social group of N individuals $X_i = [x_{1,i}, x_{2,i},\ldots, x_{j,i},\ldots, x_{n,i}]_{\,|\, i = 1,2,\ldots, N}$ interacting and exchanging knowledge with each other, each individual representing a solution X_i to the problem. The characteristics of the individuals represent the components $x_{j,i}$, $j = 1, 2,\ldots, n$ (n is the number of characteristics of an individual, which equals the size of the problem to be solved) of the solutions X_i, and the ability of an individual to find a solution to the problem is measured by the fitness function $f_i = f(X_i)$. The SGO algorithm has three phases: initialization phase, improving phase, and acquiring phase. In the initialization phase, each component $x_{j,i}$, $j = 1, 2,\ldots, n$ of the solution X_i, is randomly generated between the minimum ($x_{min,j}$) and maximum ($x_{max,j}$) limits:

$$x_{j,i} = x_{min,j} + r_1(x_{max,j} - x_{min,j}) \tag{22}$$

where r_1 is a random number uniformly distributed in the range (0, 1).

In the improving phase, each individual X_i, $i = 1, 2,\ldots, N$ of the group seeks to improve their traits $x_{j,I}$ by interacting with the best individual of the group at that moment, $X^{best} = [x_1{}^{best}, x_2{}^{best},\ldots, x_j{}^{best},\ldots x_n{}^{best}]$. Thus, the new traits ($x_{j,i}{}^{new}$) of the individual $X_i{}^{new}$ are determined with relation [29]:

$$x^{new}{}_{j,i} = c\, x_{j,i} + r_2\, (x_j{}^{best} - x_{j,i}) \tag{23}$$

where c is the self-introspection parameter with values between 0 and 1, and r_2 is a uniformly generated random number in the range (0, 1). If the new vector $X_i{}^{new}$ is better, then it is retained; otherwise, the old solution is kept.

In the acquiring phase, each individual X_i aims to improve its traits (level of knowledge) through mutual, cumulative interactions with both the best individual X^{best} and an individual randomly r ($r \neq i$) chosen from the population, $X_r = [x_{1,r}, x_{2,r},\ldots, x_{j,r},\ldots, x_{n,r}]$. Depending on the quality of the individuals X_i and X_r, the new traits ($x_{j,i}{}^{new}$) of the individual $X_i{}^{new}$ are determined as follows [29]:

If $f(X_i) < f(X_r)$ Then

$$x_{j,i}{}^{new} = x_{j,i} + r_3\, (x_{j,i} - x_{j,r}) + r_4\, (x_j{}^{best} - x_{j,i}) \tag{24a}$$

Else

$$x_{j,i}{}^{new} = x_{j,i} + r_3\, (x_{j,r} - x_{j,i}) + r_4\, (x_j{}^{best} - x_{j,i}) \tag{24b}$$

where, r_3 and r_4 denote independent arbitrary value in the (0, 1).

The solutions obtained using the SGO algorithm are iteratively improved using the updating relations from the improving and acquiring phases, until the maximum number of iterations t_{max} is reached. A pseudo-code of the SGO algorithm is presented in detail in [29].

3.2. Modified SGO (MSGO)

The proposed MSGO algorithm includes the same phases as the original SGO algorithm, but some features of SGO have been modified to increase the efficiency of the MSGO algorithm. Two of these changes, presented below, aim at inserting two operators (one chaotic type and the other being highly disruptive polynomial (HDP)) in the structure of the solutions update relations to improve MSGO's ability to escape from local minima and obtain better quality solutions.

Chaotic operator: Chaos, due to its properties (unpredictability, non-periodic, ergodicity, non-converging, pseudo-randomness etc. [18]), has been successfully incorporated into the structure of various optimization algorithms (such as adaptive sparrow search algorithm [36] and moth–flame optimizer [37]) to cover a number of shortcomings related to low population diversity, slow convergence of the optimization process, stagnation in a local area etc. In this paper, chaos is included in the MSGO algorithm using a chaotic sequence (cx_p) generated by Logistic map [37] which can be effective in solving some EcD problems [18]. The Logistic map is defined by relation [18]:

$$cx_{p+1} = 4\, cx_p\, (1 - cx_p),\ cx_p \in (0, 1) \quad (25)$$

where $\{cx_p\}_p = 1 \ldots \infty$ represents the chaotic sequence generated by the Logistic map, at the p^{th} iteration. The initial conditions are $cx_0 \in (0, 1)$ and $cx_0 \neq \{0.0, 0.25, 0.5, 0.75, 1\}$.

HDP operator: In case of more complex mathematical functions, SGO cannot provide an efficient search in the whole solutions space by relying only on randomly generated sequences through numbers ($r1$–$r4$) or on randomly extracting a solution from the population (X_r). As a result, in order to increase the exploration and exploitation capacity of the MSGO algorithm, as well as the chances of exceeding local minima, a perturbation based on the HDP operator [38] is introduced into MSGO. Mathematically, the HDP operator is defined by relation [38]:

$$\delta = \begin{cases} \delta_H = \left[2{\cdot}ru + (1 - 2{\cdot}ru)(1 - \delta_1)^{\eta+1}\right]^{\frac{1}{\eta+1}-1} & If\, ru \leq \alpha \\ \delta_L = 1 - \left[2(1 - ru) + 2(ru - 0.5)(1 - \delta_2)^{\eta+1}\right]^{\frac{1}{\eta+1}} & otherwise \end{cases} \quad (26)$$

where, ru is a random number uniformly generated in the range [0, 1]; LB_i and UB_i indicate the lower and upper boundary of the variables $x_{j,i}$; η is the index for the polynomial operator; α is a control parameter that is considered to have a value of 0.5 [39] (in this paper, the values of the parameter α are determined by experimental trials in the range (0, 1)); δ_H is the equation that alternates moderate values with high values for δ, while δ_L is the equation that generates low values for δ; δ_1 and δ_2 are calculated, based on the LB_i and UB_i imposed boundaries for the variable $x_{j,i}$, with the relations $\delta_1 = (x_{j,i} - LB_i)/(UB_i - LB_i)$ and $\delta_2 = (UB_i - x_{j,i})/(UB_i - LB_i)$.

In general, the HDP operator is used as a mutation operator that intervenes in altering some $x_{j,i}$ values of a X_i solution with a certain predetermined probability. In MSGO, the HDP operator is not used as a mutation operator, but as an operator applied to modify each variable $x_{j,i}$ of a solution X_i (practically, the mutation probability is considered equal to 1). Thus, depending on the HDP parameters (η, ru, δ_1, α), the δ values generated by (26) may be high, which could cause some of the $x_{j,i}$ components of some X_i solutions to reach the limits of the search space relatively quickly (the δ values could generate excessively large search steps). To avoid this problem, the δ values generated by (26) will be attenuated/smoothed according to the current iteration t, if δ exceeds a certain limit value (δ_{max}):

$$\delta_a = \begin{cases} \delta \cdot \left(1 - \frac{t}{t_{max}}\right) & \text{If } \delta \geq \delta_{max} \\ \delta & \text{otherwise} \end{cases} \tag{27}$$

Figure 1 shows the first 10,000 values of δ generated by relation (26), considering the following settings for the HDP operator parameters: δ_1 = $rnd(1)$, $\delta_2 = 1 - \delta_1$; η = 4, α = 0.5, δ_{max} = 1.5. The δ_H values generated by (26) under the condition $ru \leq \alpha = 0.5$ are marked in blue, and the δ_L values in red. Since the δ_H values are much higher than δ_L, in order to have a clearer picture, the δ_L values have been detailed in a separate graph on the interval (1, 10,000). It is seen that the δ_H equation alternates moderate values and high ones, while the δ_L equation generates only low values. Figure 2 shows the attenuated δ_a values obtained using (27). Initially, the δ_a values are higher (being able to ensure an efficient search in the whole solutions space), then with the increase in the number of simulations they attenuate, the search step is reduced, making the transition from exploration to exploitation, and at the end of the process the values are equal to or lower than the δ_{max} limit (which facilitates the exploitation phase).

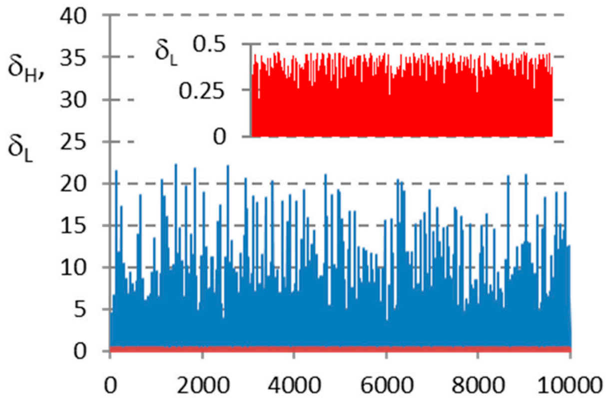

Figure 1. HDP values generated for δ_H and δ_L.

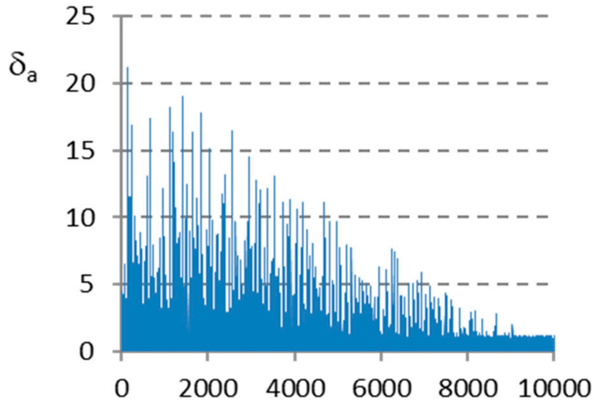

Figure 2. The attenuated values obtained for δ_a.

The changes implemented in MSGO compared to SGO are presented below:

a. The improving phase of MSGO is similar to that of SGO but the sequences of numbers r_2 randomly generated in SGO by (23) are replaced by sequences of numbers generated by the attenuated DHP operator δ_a by relation (27) in the form:

$$x_{j,i}^{new} = x_{j,i} + \delta_a \left(x_j^{best} - x_{j,i} \right) \tag{28}$$

b. The acquiring phase of MSGO is similar to that of SGO, except for the following three modifications:

   b1. The sequences of numbers r_3 randomly generated in SGO by (24a) are replaced by sequences of numbers generated by the attenuated DHP operator δ_a by relation (27).

   b2. The sequences of numbers r_3 randomly generated in SGO by (24b) are replaced by chaotic sequences (cx) generated by the Logistic map with relation (25).

   b3. In SGO, switching between relations (24a) and (24b) is performed by considering the fitness of the competitive solutions X_i and X_r, based on the condition $f(X_i) < f(X_r)$. In MSGO, this condition is replaced by a random one, having the form $rnd(1) < \beta$, where $rnd(1)$ is a uniformly generated random number in the range (0, 1); β is a value determined by experimental trials.

The three changes (b1)–(b3) made in MSGO are embodied in relations (29a) and (29b):
If $rnd(1) < \beta$ Then

$$x_{j,i}{}^{new} = x_{j,i} + \delta_a\,(x_{j,i} - x_{j,r}) + r_4\,(x_j{}^{best} - x_{j,i}) \tag{29a}$$

Else

$$x_{j,i}{}^{new} = x_{j,i} + cx\,(x_{j,r} - x_{j,i}) + r_4\,(x_j{}^{best} - x_{j,i}) \tag{29b}$$

In Algorithm 1, the calculation steps for applying the MSGO algorithm are presented in detail, with the modifications (b1)–(b3) mentioned above.

Algorithm 1: MSGO algorithm

{Initialization phase}
Initialize the iterations (t = 0), t is counter of iterations;
Initialize the solutions X_i, i = 1, 2,. . ., N using relation (22);
Evaluate the initial solutions and identify the best X^{best} solution;
repeat
    $t = t + 1$
    For i = 1 To N Do {*improving phase*}
    For j = 1 To n Do
    Generate a value δ of the HDP operator using (26);
    Determine δ_a using relation (27);
    Update the components $x_{j,i}$ using relation (28), obtaining the new solution $X_i{}^{new}$; End For j
If the new solution $X_i{}^{new}$ is better, then it is retained; otherwise, the old solution is maintained;
Find the best X^{best} solution from the population; End For i
For i = 1 To N Do {acquiring phase}
Randomly select a solution X_r, $r \in \{1, 2, \ldots, N\}$, $r \neq i$;
If $rnd(1) < \beta$ Then
For j = 1 To n Do
    Generate a value δ of the HDP operator using (26);
    Determine δ_a using relation (27);
    Update $x_{j,i}$ using (29a), obtaining $X_i{}^{new}$; End For j; End If
Else
For j = 1 To n Do
    Generate a chaotic value cx using (25);
    Update $x_{j,i}$ using (29b), obtaining $X_i{}^{new}$; End For j; End Else
If the new solution $X_i{}^{new}$ is better, then it is retained; otherwise, the old solution is maintained;
Find the best X^{best} solution from the population; End For i
Until $t \geq t_{max}$ {*stopping criterion*}
The best solution X^{best} and the fitness $f(X^{best})$ are retained.

The main steps of the MSGO algorithm are summarized in a flowchart shown in Figure 3.

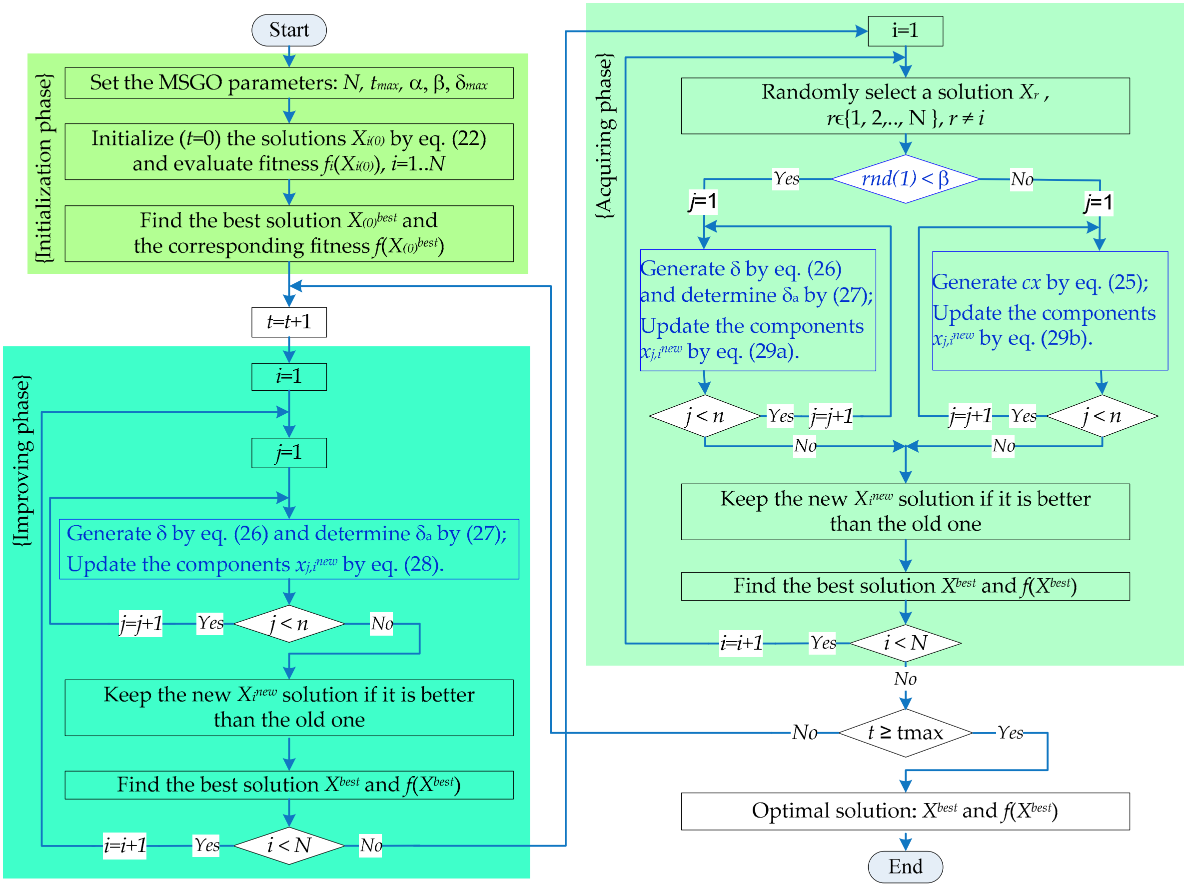

Figure 3. Flowchart of the MSGO algorithm.

4. Implementing the MSGO for the EcD, EmD, or EED Problems

This section shows how to implement the MSGO algorithm for solving EcD, EmD, or EED problems. The application of MSGO for solving EcD, EmD, or EED problems is similar, the difference being given by the type of objective function to be minimized: cost $C(PT, PW)$ defined by (1) for EcD problems, emission $E(PT, PW)$ defined by (16) for EmD problems or $\phi(PT, PW)$ defined by (20) for EED problems. For the implementation of the MSGO algorithm, a solution i is considered to be represented by the vector $PTW_i = [PTW_{1,i}, PTW_{2,i}, \ldots, PTW_{j,i}, \ldots, PTW_{Nt+Nw,i}]\,|_{\,i\,=\,1,\,2,\ldots\,N}$ which includes the powers of thermal units ($PTW_{j,i}\,|_{\,j\,=\,1,2,\ldots\,Nt}$) and wind units ($PTW_{j,i}\,|_{\,j\,=\,Nt+1,\,Nt+2,\ldots,\,Nt+Nw}$). If the system contains only thermal units, then the vector PTW_i consists only of the powers generated by these units. We also denote $PTW^{best(t)}$ as the best solution obtained using MSGO up to iteration t.

4.1. Stages of MSGO Implementation for the EcD, EmD, or EED Problem

The MSGO algorithm applied to solve the EcD, Em, or EED problems formulated in Section 2 includes the following steps:

Step 1: Specify test system input data for the EcD, EmD, or EED problem: the number of thermal (Nt) and wind (Nw) units; cost coefficients for the thermal units (a, b, c, e, f); parameters (c, k) to the Weibull distribution; the characteristic values of the wind speed (v_{in}, v_r, v_{out}); the rated output power of the wind units (PWr); active power limits for thermal and wind units (PT_{min}, PT_{max}; PW_{min}, PW_{max}); the cost coefficients (c^d, c^o, c^u) and emissions coefficients (e^o, e^u) to the wind units; the B-loss coefficients (B_{ij}, B_{0i}, B_{00}); load demand (P_D) and accuracy ε.
Step 2: Set the parameters of the MSGO algorithm: N and t_{max};

Step 3: Random initialization of a population of N solutions, represented by the vectors $PTW_{i\,|\,i=1,2,\ldots N}$

3.1: $t = 0$;
3.2: Chaotic sequence cx are initialized using the logistic map;
3.3: Randomly generate an initial population with N solutions ($PTW_i^{(0)}\,|_{\,i=1,2,\ldots,N}$) using relation (22). Each solution respects the constraints defined by relations (12)–(14); The constraint (14) is handled by a heuristic procedure CHM presented in Section 4.2;
3.4: Evaluate the initial solutions $PTW_i^{(0)}\,|_{\,i=1,2,\ldots,N}$ using relation (1), (16) or (20) associated with the EcD, EmD, or EED problems;
3.5: Find the best initial solution $PTW^{best(0)}$ and the objective function associated with the addressed problem (EcD, EmD, or EED).

For $t = 1$ Do t_{max}

Step 4: Update solutions $PTW_i^{(t)}\,|_{\,i=1,2,\ldots,N}$ in the improving phase.

4.1: For $i = 1$ To N Do
4.2: For $j = 1$ to n Do
4.3: Generate a value δ of the HDP operator using (26);
4.4: Determine δ_a using relation (27);
4.5: Updating the components $PTW_{j,i}^{(t)}$ using relation (28), obtaining the new components $PTW_{j,i}^{new(t)}$, at iteration t;
4.6: Checking the inequality constraints (12) and (13): if the $PTW_{j,i}^{new(t)}$ power is outside the limits, then CHM from Section 4.2 is applied; End For j;
4.7: Checking the equality constraint (14): the new solution $PTW_i^{new(t)}$ is adjusted using the equality constraint handling mechanism from Section 4.2;
4.8: Evaluate the new solution $PTW_i^{new(t)}$ using relation (1), (16) or (20) depending on the addressed problem (EcD, EmD or EED): If the new solution $PTW_i^{new(t)}$ is better, then it is retained; otherwise, the old solution is maintained;
4.9: Update the best solution $PTW^{best(t)}$; End For i;

Step 5: Update solutions $PTW_i^{(t)}\,|_{\,i=1,2,\ldots,N}$ in the acquiring phase.

5.1: For $i = 1$ To N Do
5.2: Randomly select a solution $PTW_r^{(t)}$, $r \epsilon \{1, 2, \ldots, N\}$, $r \neq i$;
5.3: If $rnd(1) < \beta$ Then
5.4: For $j = 1$ to n Do
5.5: Generate a δ value of the HDP operator using (26);
5.6: Determine δ_a using relation (27);
5.7: Updating $PTW_{j,i}^{(t)}$ using (29a), getting the new components $PTW_{j,i}^{new(t)}$ of the new solutions $PTW_i^{new(t)}$;
5.8: Apply the procedure for handling inequality constraints (12) and (13) from Section 4.2; End For j; End If
5.9: Else
5.10: For $j = 1$ to n Do
5.11: Generate a chaotic value cx by (25);
5.12: Updating $PTW_{j,i}^{(t)}$ by (29b), obtaining $PTW_{j,i}^{new(t)}$;
5.13: Apply CHM from Section 4.2 for constraints (12) and (13); End For j; End Else
5.14: Checking the equality constraint (14): the new solution $PTW_i^{new(t)}$ is adjusted using the equality constraint handling mechanism from Section 4.2;
5.15: Evaluate the new solution $PTW_i^{new(t)}$ using relation (1), (16) or (20) depending on the addressed problem (EcD, EmD, or EED): if the new solution $PTW_i^{new(t)}$ is better, then it is retained; otherwise the old solution is maintained;
5.16: Update the best solution $PTW^{new(t)}$; End For i;

Step 6: Stop the process: the calculation process is stopped when the maximum number of iterations (t_{max}) is reached. {End For t}

Step 7: Memorize the best solution: The best solution PTW^{best} and the objective function associated with the problem addressed (EcD, EmD, or EED).

4.2. Constraints Handling Mechanism (CHM)

The CHM refers to the adjusting method of the solutions that do not satisfy the inequality and equality constraints of the EcD, EmD, or EED problems defined by relations (12)–(14). In the case of inequality constraints, defined by relations (12) and (13), if the power of a thermal or wind unit is outside the imposed operating limits (PT_{min} or PT_{max}, and PW_{min} or PW_{max}, respectively), then the power of this unit is set with the exceeded limit [40]. To handle the equality constraint, defined by (14), we apply a CHM procedure presented in [41]. The CHM(PTW) procedure has a single parameter, represented by the power vector PTW, and finally it returns a feasible solution that respects relation (14) with the imposed error ε. The CHM(PTW) is called whenever a PTW vector is updated in the initializing, the improving, or the acquiring phases of the MSGO or SGO algorithms.

5. Case Studies

The efficiency of the MSGO algorithm was tested on medium-sized (10-unit) and large (40-unit) systems, having different characteristics, such as operating limits of the units, transmission line losses, valve-point effect, wind power. According to the characteristics of the systems, four cases were studied, described below.

Case 1 (C_1): This case analyses a 10-unit system taking into account the transmission line losses. The power demand is 2000 MW. The cost coefficients (a_i, b_i, c_i, e_i, f_i, i = 1, 2,. . ., 10), B-loss coefficients (B_{ij}), and emission coefficients (α_i, β_i, γ_i, η_i, δ_i) are taken from [42].

Case 2 (C_2): The second case is a system having 40 units that considers the valve-point effects. The system is analyzed without considering transmission line losses. The power demand is 10,500 MW. The cost coefficients are considered from [33] and emission coefficients are provided by [43].

Case 3 (C_3): The case C_3 is a system with 40 units similar to case C_2 (the cost and emission coefficients are identical to those in case C_2 [33,43]), but transmission losses are considered, as well. The power demand is 10,500 MW. The B-loss coefficients are taken from [44].

Case 4 (C_4): A 40-unit system derived from case C_3 by replacing the first two thermal units (PT_1 and PT_2) with the two wind units (PW_1 and PW_2). Each wind unit has a nominal power of 550 MW, and the minimum and maximum capacities are $PW_{min,1}$ = $PW_{min,2}$ = 0, $PW_{max,1}$ = $PW_{max,2}$ = 550 MW. The power demand is P_D = 10,500 MW. We consider that the wind speed has a Weibull distribution. The shape and scale parameters (k, c) corresponding to the sites of the two wind units have the values [13]: k_1 = 1.5; c_1 = 15; k_2 = 1.5; c_2 = 15. Other wind-related characteristics have the following values [13]: cut-in wind speed (v_{in} = 5 m/s), rated wind speed (v_r = 15 m/s), cut-out wind speed (v_{out} = 45 m/s), the cost coefficients associated with underestimation ($c_1^u = c_2^u = 5$) and overestimation ($c_1^o = c_2^o = 5$). The cost and emission coefficients for the thermal units (P_3–P_{40}) and the B-loss coefficients are identical to those mentioned in C_3 case, being taken from [44]. The characteristics of the thermal units and the B-loss coefficients for the 10-unit and 40-unit systems are presented in Appendix A, Tables A2–A5.

Each of the cases mentioned in Table 1 (C_1–C_4) were studied based on the mathematical optimization models (presented in Section 2) of the problems EcD (cases C_{1a}–C_{4a}), EmD (cases C_{1b}–C_{4b}), and EED (cases C_{1c}–C_{4c}). The SGO and MSGO algorithms perform 50 independent runs for each case study, retaining four statistical indicators: Best cost/emission (B), Average cost/emission (A), Worst cost/emission (W), and the standard deviation (SD). Algorithms SGO and MSGO have been implemented in MathCAD and run on a PC with an Intel i5 processor, 2.2 GHz CPU and 4 GB of RAM.

5.1. Setting the Parameters

Adjusting the parameters of metaheuristic algorithms is an essential task with multiple positive effects, such as improved performance, adaptation of the algorithms to a specific problem, better execution times, and stability, etc. The proposed MSGO algorithm has three specific parameters (α, β, and δ_{max}). A procedure based on the design of an experimental plan is used to set the specific parameters. They are successively set (for example, in the following order α, β, and δ_{max}), starting from a set of initial values [45]. Thus, one parameter is varied, and the others are fixed either with the initial values or with the already set values. This procedure allows the evaluation of the MSGO algorithm performance through testing different combinations of the interest parameter values, followed by the selection of the best combination based on a predetermined criterion. The test values considered for setting the specific parameters are α = {0, 0.25, 0.5, 0.75, 1}, β = {0, 0.25, 0.5, 0.75, 1}, and δ_{max} = {1.2, 1.5, 2}. The initial values are α = 0, β = 0, and δ_{max} = 1.2. The parameter selection criterion is based on Average cost/emission obtained from 25 runs for each case study. Note that the experimental analysis does not guarantee the best values for the parameters of MSGO algorithm. However, the results obtained using MSGO following this selection process show that the algorithm parameters were reasonably set Table 1 shows the parameters of the MSGO algorithm (N, t_{max}, α, β, δ_{max}) for all the analyzed cases.

Table 1. Characteristics of the analyzed systems and values set for the SGO and MSGO parameters.

Cases	Type of Problem	SGO and MSGO				MSGO Parameters			P_L	VPE	Wind
		n	N	t_{max}	NE	α	β	δ_{max}			
C_{1a}	EcD	10	15	30	900	0.75	0.25	1.2	√	√	-
C_{1b}	EmD	10	15	30	900	0.75	0.25	1.2	√	√	-
C_{1c}	EED	10	15	30	900	0.75	0.25	1.2	-	√	-
C_{2a}	EcD	40	50	350	35,000	0.05	0.5	1.2	-	√	-
C_{2b}	EmD	40	50	350	35,000	0.05	0.5	1.2	-	√	-
C_{2c}	EED	40	50	350	35,000	0.05	0.5	1.2	√	√	-
C_{3a}	EcD	40	50	350	35,000	0.05	0.5	1.5	√	√	-
C_{3b}	EmD	40	50	200	20,000	0.05	0.5	1.5	√	√	-
C_{3c}	EED	40	50	350	35,000	0.05	0.5	1.5	√	√	-
C_{4a}	EcD-Wind	40	50	350	35,000	0.05	0.5	1.5	√	√	√
C_{4b}	EmD-Wind	40	50	200	20,000	0.05	0.25	1.5	√	√	√
C_{4c}	EED-Wind	40	50	350	35,000	0.05	0.5	1.5	√	√	√

In the case of the SGO algorithm, the values of the common parameters (N and t_{max}) are the same as the values set for the MSGO algorithm in each case analyzed (these are shown in Table 1). The SGO algorithm has only one specific parameter c, the value of which being set to the one recommended in [29] (c = 0.2). For the fair comparison of the SGO and MSGO algorithms, the number of evaluations of the objective functions (NE) is considered equal in all analyzed cases (C_1–C_4), which is mentioned in Table 1.

5.2. Results for EcD (Cases C_{1a}–C_{4a}) and EmD (Cases C_{1b}–C_{4b}) Problems

The best solutions obtained using SGO and MSGO: The optimal scheduling of the thermal units for the cases C_{1a}, C_{1b} (10 units), and C_{2a}, C_{2b} (40 units) obtained using the MSGO and SGO algorithms are presented in Tables 2 and 3, respectively.

In the cases C_{1a} and C_{1b}, the system being of relatively small sizes, the MSGO and SGO algorithms find approximately the same optimal operating solution, and as a result, the derived quantities (Cost, Emission, P_L) will be approximately the same. However, mathematically, MSGO performs slightly better than SGO. Thus, in the case of C_{1a} the best costs obtained using MSGO and SGO are $\text{Cost}_{\text{C1a(MSGO)}}$ = 111,497.6301 $/h and $Cost_{C1a(SGO)}$ = 111,497.6302 $/h, respectively. For the C_{1b} case, the best emissions found by MSGO and SGO are $\text{Emission}_{\text{C1b(MSGO)}}$ = 3932.2432 lb/h and $\text{Emission}_{\text{C1b(SGO)}}$ = 3932.2433 lb/h, respectively. Also, in the case of C_{2b} (Table 3), the best solutions (P_1–P_{40}, Cost, Emission) found by MSGO and SGO are very close (both algorithms have the ability to identify quality solutions): $\text{Emission}_{\text{C2b(MSGO)}}$ = 176,682.26363 t/h and $\text{Emission}_{\text{C2b(SGO)}}$ = 176,682.26364 t/h, respectively.

Table 2. Generation schedule obtained using SGO and MSGO for cases C_{1a}, $C_{1b,}$ and C_{1c} (P_D = 2000 MW).

Outputs Algorithms	Best Cost (C_{1a})		Best Emission (C_{1b})		BCS (C_{1c})	
	SGO	MSGO	SGO	MSGO	SGO	MSGO
PT_1 (MW)	55	55	55	55	55	55
PT_2 (MW)	80	80	80	80	79.9919138	80
PT_3 (MW)	106.9541594	106.9339816	81.1395049	81.13389758	84.9307435	86.1178873
PT_4 (MW)	100.5724243	100.5785797	81.3521442	81.365596459	84.4692580	84.4754324
PT_5 (MW)	81.4829314	81.5030072	160	160	132.5861458	132.0947090
PT_6 (MW)	83.0293259	83.0232077	240	240	148.8844943	150.1405606
PT_7 (MW)	299.9999999	300	294.5013582	294.50918872	298.3856220	300
PT_8 (MW)	340	340	297.2767111	297.26088151	317.3109360	318.6699829
PT_9 (MW)	470	470	396.7327820	396.76630122	436.6060140	437.5797908
PT_{10} (MW)	470	470	395.5926758	395.55896593	446.4576771	440.4116407
Cost C^T (PT) ($/h)	111,497.6302221	111,497.6301430	116,412.4961603	116,412.5597722	112,884.4848042	112,913.8454551
Emission E^T (PT) (lb/h)	4572.2503628	4572.1759096	3932.2432876	3932.2432406	4177.6632977	4173.0883561
P_L (MW)	87.0388508	87.03878611	81.59518555	81.594840685	84.62280525	84.49001346
P_D (MW)	2000	2000	2000	2000	2000	2000
ΔP * (MW)	-9.9×10^{-6}	-9.8×10^{-6}	-9.3×10^{-6}	-9.2×10^{-6}	-8.3×10^{-7}	-9.8×10^{-6}
Time (s)	0.17	0.18	0.17	0.18	0.18	0.19

* ΔP represents the accuracy with which the power balance is satisfied: $\Delta P = \Sigma PT_i - P_L - P_D$.

Table 3. Generation schedule obtained using SGO and MSGO for cases C_{2a}, $C_{2b,}$ and C_{2c} (P_D = 10,500 MW).

Outputs Algorithms	Best Cost (C_{2a})		Best Emission (C_{2b})		BCS (C_{2c})	
	SGO	MSGO	SGO	MSGO	SGO	MSGO
PT_1 (MW)	110.922179	111.716634	114.000000	114.000000	113.999966	110.800341
PT_2 (MW)	111.243322	110.860214	114.000000	114.000000	113.999796	110.800220
PT_3 (MW)	119.999923	97.403591	120.000000	120.000000	120.000000	119.999963
PT_4 (MW)	179.733198	179.743937	169.368013	169.367866	179.733101	179.733075
PT_5 (MW)	87.968669	87.780202	97.000000	97.000000	97.000000	87.801870
PT_6 (MW)	139.999997	139.999967	124.257317	124.257125	140.000000	139.999941
PT_7 (MW)	299.999977	259.611600	299.711092	299.711096	299.999999	299.999869
PT_8 (MW)	284.599970	284.603778	297.914825	297.914414	284.599650	284.599715
PT_9 (MW)	284.600310	284.603707	297.259861	297.260410	284.599652	284.599803
PT_{10} (MW)	130.000055	130.000000	130.000000	130.000000	204.799826	130.000011
PT_{11} (MW)	168.799938	94.001033	298.409765	298.409555	243.599650	318.397097
PT_{12} (MW)	94.000100	168.802785	298.026091	298.026393	318.399211	318.396537
PT_{13} (MW)	125.000068	214.762696	433.557450	433.557827	394.279369	394.279333
PT_{14} (MW)	304.519644	394.279917	421.727984	421.729503	394.279373	394.279355
PT_{15} (MW)	394.279363	304.520136	422.779051	422.780633	394.279370	394.279352
PT_{16} (MW)	394.279396	394.281435	422.779145	422.779765	394.279371	394.279339
PT_{17} (MW)	489.279362	489.279715	439.413302	439.412095	489.279356	489.279199
PT_{18} (MW)	489.279374	489.277936	439.402981	439.402147	489.279365	489.278987
PT_{19} (MW)	511.279452	511.286046	439.413322	439.413830	472.436489	421.519583
PT_{20} (MW)	511.279469	511.288520	439.413375	439.412524	421.519581	421.519572
PT_{21} (MW)	523.279375	523.282365	439.446230	439.446547	433.519581	433.519577
PT_{22} (MW)	523.279470	523.288232	439.447404	439.446608	433.519581	433.519606
PT_{23} (MW)	523.279716	523.286753	439.771529	439.772178	433.519581	433.519604
PT_{24} (MW)	523.279630	523.298896	439.771899	439.771421	433.519585	433.519606
PT_{25} (MW)	523.279479	523.282145	440.111752	440.112274	433.519584	433.519649
PT_{26} (MW)	523.279413	523.285836	440.111656	440.111780	433.519583	433.519592
PT_{27} (MW)	10.000142	10.001290	28.994136	28.993709	10.000009	10.000005
PT_{28} (MW)	10.000000	10.000158	28.994289	28.993427	10.000000	10.000098
PT_{29} (MW)	10.000004	10.000029	28.993716	28.994114	10.000000	10.000015
PT_{30} (MW)	87.979665	96.047579	97.000000	97.000000	97.000000	96.999848
PT_{31} (MW)	189.999986	189.999914	172.332024	172.331705	190.000000	189.999825
PT_{32} (MW)	190.000000	190.000000	172.332025	172.331635	189.999867	189.999975
PT_{33} (MW)	190.000000	189.999740	172.331486	172.331960	189.999915	189.999989
PT_{34} (MW)	199.999981	164.840604	200.000000	200.000000	200.000000	199.999986
PT_{35} (MW)	199.999969	199.999540	200.000000	200.000000	200.000000	199.999994
PT_{36} (MW)	199.999996	199.999900	200.000000	200.000000	200.000000	199.999976
PT_{37} (MW)	110.000000	109.999989	100.839198	100.838331	109.999999	109.999997
PT_{38} (MW)	109.999997	109.999981	100.838254	100.838389	110.000000	109.999850
PT_{39} (MW)	110.000000	109.999727	100.838245	100.838200	110.000000	109.999659
PT_{40} (MW)	511.279414	511.283472	439.412572	439.412526	421.519580	488.039989

Table 3. *Cont.*

Outputs Algorithms	Best Cost (C_{2a})		Best Emission (C_{2b})		BCS (C_{2c})	
	SGO	MSGO	SGO	MSGO	SGO	MSGO
Cost C^T (PT) (\$/h)	121,509.82092	121,426.70390	129,995.28508	129,995.30108	125,526.34268	125,434.46554
Emission E^T (PT) (t/h)	359,251.81848	356,231.07314	176,682.26364	176,682.26363	201,944.58419	200,613.67971
ΔP * (MW)	-1.7×10^{-11}	1.5×10^{-11}	-9.9×10^{-6}	-9.9×10^{-6}	-8.7×10^{-6}	5.1×10^{-7}
Time (s)	7.64	8.03	7.34	7.65	9.28	9.97

* ΔP represents the accuracy with which the power balance is satisfied: $\Delta P = \Sigma PT_i - P_L - P_D$.

On the contrary, in the case of C_{2a} (where the functions that model the cost have a higher complexity and determine a larger number of local minima), the MSGO algorithm can identify a much better solution than SGO ($Cost_{C2a(MSGO)}$ = 121,426.7039 \$/h < $Cost_{C2a(SGO)}$ = 121,509.8209 \$/h).

Table 4 shows the best solutions in terms of cost and emissions obtained using the MSGO algorithm for the cases C_{3a}, C_{3b} (without wind), C_{4a}, and C_{4b} (with wind).

Table 4. Generation schedule obtained using MSGO for cases C_{3a}, C_{3b}, and C_{3c} (with loss of power and without wind) and C_{4a}, C_{4b}, C_{4c} (with loss of power and wind).

Outputs Cases	Best Cost		Best Emission		BCS	
	Case C_{3a} (Without Wind)	Case C_{4a} (With Wind)	Case C_{3b} (Without Wind)	Case C_{4b} (With Wind)	Case C_{3c} (Without Wind)	Case C_{4c} (With Wind)
PT_1/PW_1 (MW)	114.000000	549.9999977	114.000000	550.000000	114.0000000	549.9997828
PT_2/PW_2 (MW)	113.999999	549.9999998	114.000000	550.000000	114.0000000	549.9997818
PT_3 (MW)	120.000000	97.39994714	120.000000	120.000000	120.0000000	119.9988116
PT_4 (MW)	189.999995	179.7331053	190.000000	175.313561	182.8929206	179.7374211
PT_5 (MW)	96.999999	87.79995033	97.000000	97.000000	96.99999982	96.99883835
PT_6 (MW)	140.000000	68.00000364	132.951212	119.769488	139.9999995	105.4022758
PT_7 (MW)	300.000000	259.5996859	300.000000	300.000000	300.0000000	299.9999742
PT_8 (MW)	300.000000	284.5996597	300.000000	299.999999	299.9999998	285.7135868
PT_9 (MW)	299.999998	284.5997061	300.000000	300.000000	299.9999995	287.9519754
PT_{10} (MW)	279.599683	204.79983	270.805053	161.263885	279.5996268	204.8086989
PT_{11} (MW)	168.799860	94.00001916	322.733043	297.366373	318.3994557	243.6012566
PT_{12} (MW)	94.000003	94.00000226	315.129176	289.219449	318.3993251	243.6003501
PT_{13} (MW)	484.039161	304.519587	480.860016	441.638969	484.0391469	394.2851366
PT_{14} (MW)	484.039166	304.5195816	475.905463	431.552565	484.0391695	394.2799802
PT_{15} (MW)	484.039164	394.2793758	476.271511	433.133288	484.0391541	394.2841752
PT_{16} (MW)	484.039178	484.039163	480.455232	439.072444	484.0391647	484.0292251
PT_{17} (MW)	489.279372	489.2793773	471.830727	438.041844	489.2793744	489.2692227
PT_{18} (MW)	489.279372	489.2793735	461.973780	427.916684	399.5196438	399.5209877
PT_{19} (MW)	511.279600	511.2793699	483.440794	446.749851	510.840761	506.0067276
PT_{20} (MW)	511.279490	511.2793799	483.258780	446.772552	510.666052	421.5364187
PT_{21} (MW)	526.732209	523.2793949	483.252001	447.336714	511.7221827	433.5530376
PT_{22} (MW)	550.000000	523.2793745	486.947376	452.005128	516.1579618	514.2353651
PT_{23} (MW)	523.279384	523.2793957	472.040407	438.402949	433.5204637	433.5329212
PT_{24} (MW)	523.279383	523.2793722	462.227523	428.365213	433.5195802	433.5268561
PT_{25} (MW)	524.239856	523.2793935	483.754300	447.367210	511.6249809	433.5342317
PT_{26} (MW)	523.815577	523.2793731	483.575006	447.380310	511.4580636	433.7489257
PT_{27} (MW)	10.000020	10.00000931	67.543932	33.650576	15.28948883	10.05540017
PT_{28} (MW)	10.000113	10.00000064	72.653376	36.778266	17.79565013	10.24418652
PT_{29} (MW)	10.000007	10.00000189	54.008031	28.253062	10.36768925	10.0022597
PT_{30} (MW)	87.800155	87.79991386	97.000000	97.000000	96.99999662	89.80426078
PT_{31} (MW)	190.000000	190.0000000	190.000000	175.455547	190.0000000	189.998884
PT_{32} (MW)	190.000000	189.9999989	190.000000	175.467999	190.0000000	189.9955403
PT_{33} (MW)	190.000000	189.9999997	190.000000	175.691210	190.0000000	189.9978565
PT_{34} (MW)	200.000000	195.3472481	200.000000	200.000000	200.0000000	199.9994215
PT_{35} (MW)	199.999999	164.7998866	200.000000	200.000000	200.0000000	199.9994151
PT_{36} (MW)	164.799870	164.7998371	200.000000	200.000000	199.9999989	199.998299
PT_{37} (MW)	110.000000	109.9999997	110.000000	103.209409	110.0000000	110.0000000
PT_{38} (MW)	109.999999	109.9999992	110.000000	103.216803	110.0000000	109.9997758
PT_{39} (MW)	110.000000	109.9999985	110.000000	103.385396	110.0000000	109.9962344
PT_{40} (MW)	550.000000	511.2794082	486.926373	451.971343	511.2792996	509.5675741

Table 4. *Cont.*

Outputs Cases	Best Cost		Best Emission		BCS	
	Case C_{3a} (Without Wind)	**Case C_{4a} (With Wind)**	**Case C_{3b} (Without Wind)**	**Case C_{4b} (With Wind)**	**Case C_{3c} (Without Wind)**	**Case C_{4c} (With Wind)**
Cost $C(PT,PW)$ ($/h)	136,454.33693	123,161.88666	147,526.93169	133,213.62916	139,049.0921	126,645.13420
Emission $E(PT,PW)$ (t/h)	501,366.97888	375,873.82903	347,578.49057	193,311.54070	388,020.2454	239,155.73184
P_L (MW)	958.6206217	936.7097306	1040.543121	1009.748098	1000.489155	962.815076
ΔP (MW)	-9.7×10^{-6}	-9.8×10^{-6}	-9.9×10^{-6}	-9.7×10^{-6}	-6.1×10^{-6}	-4.1×10^{-6}
Time (s)	28.61	30.09	16.34	17.37	30.07	32.96

ΔP represents the accuracy with which the power balance is satisfied: $\Delta P = (\Sigma PT_i + \Sigma PT_i) - P_L - P_D$.

The solutions for C_{4a} and C_{4b} cases indicate that the two wind units (PW_1, PW_2) are scheduled to operate at full capacity (550 MW) in both the EcD and EmD problems. Thus, according to the statement in case C_4, the wind units will replace two thermal units PT_1, PT_2 (that operate at maximum capacity $PT_1 \approx PT_2 \approx 114$ MW in C_{3a} and C_{3b} cases), and will also reduce the operating power level of other thermal units (PT_3–PT_{40}). Based on the EcD and EmD models, it can be seen that if the wind units operate at full capacity it results in an E^u power very close to zero ($E_1^u(PW_1 = 550) \approx E_2^u(PW_2) \approx 0\ MW$), so their corresponding costs and emissions will be close to zero. In contrast, the average powers associated with overestimation will tend towards maximum values $\left(E_1^o(PW_1 = 550) \approx E_2^o(PW_2) \approx 230.7635\ MW\right)$, while their corresponding costs and emissions are 2 × 1153.817 $/h, and 2 × 1153.817 t/h. The risks taken by the system operator when planning the wind units to operate at values close to the maximum capacity are covered by the reduction of costs and emissions from the thermal units (due to the reduction of the PT_3–PT_{40} powers in $C_{4a}(C_{4b})$ cases compared to $C_{3a}(C_{3b})$).

Comparing the solutions for the cases with wind and those without wind, it can be seen that the inclusion of the wind units (PW_1 and PW_2) results in a substantial reduction in cost (from $\text{Cost}_{C3a(MSGO)}$ = 136,454.33693 $/h to $\text{Cost}_{C4a(MSGO)}$ = 123,161.88666 $/h, representing 9.74%), and emissions (from $\text{Emission}_{C3b(MSGO)}$ = 347,578.49052 t/h to $\text{Emission}_{C4b(MSGO)}$ = 193,311.54070 t/h representing 44.38%). Also, power losses are reduced from 958.62 MW in the case of C_{3a} to 936.70 MW in the case of C_{4a} (2.28%) and from 1040.54 MW in the case of C_{3b} to 1009.74 MW in the case of C_{4b} (2.96%), respectively.

Comparison of MSGO algorithm with other algorithms: In Tables 5–12 the values of the statistical items (*B*, *A*, *W*, and *SD*) are presented. They are obtained using different algorithms for EcD problem (cases C_{1a}–C_{4a}) and EmD problem (C_{1b}–C_{4b}), respectively.

Analyzing the values from the mentioned tables, it can be seen that in all studied cases the MSGO algorithm obtains statistical indicators as good or better than the competing algorithms mentioned in these tables. The exception is the Worst cost item for the algorithms Jaya-SML [46] (case C_{2a}—Table 6), ORCCRO [47] (case C_{3a}—Table 7), DE (case C_{4a}—Table 8), and the SD item for the algorithms CSS [48] (cases C_{3a}—Table 7), DE, and SCA (cases C_{4a}—Table 8).

The MSGO algorithm shows a better performance than well-known algorithms, such as DE [49], RCCRO [35] (in cases C_{1a}—Table 5 and C_{1b}—Table 9), GA [50], DE [42], ABC [51], PSO [52], TLBO [53], SCA [54], CSO [55] (cases C_{2a}—Table 6), DE [26], TLBO [26], BSA [56] (cases C_{2b}—Table 10), ACS [57], BBO [47], ORCCRO [47], CSS [48], IMO [58] (cases C_{3a}—Table 7), PSO, DE, SCA (cases C_{3b}—Table 11, C_{4a}—Table 8, C_{4b}—Table 12). Also, the performance of MSGO is superior to other metaheuristic algorithms obtained using various procedures (such as modification of solution update relations, inclusion of chaos and/or opposite solutions, hybridizations, etc.): GQPSO [59], CSCA [27], QOPO [19] (in cases C_{1a}—Table 5), MIMO [58], HSCA [60], CTLBO [53], TLABC [61] (cases C_{2a}—Table 6), GAAPI [44], SDE [62], HPSO-DE [63], MIMO [58] (cases C_{3a}—Table 7).

Table 5. Values for the statistical items obtained using different algorithms for the cases C_{1a}—*Best Cost* (10 units with losses, 2000 MW).

Algorithm	Best Cost ($/h)	Average Cost ($/h)	Worst Cost ($/h)	SD ($/h)	Cost Saving * ($/h)
DE [49]	111,500	-	-	-	-
TLBO [26]	111,500	-	-	-	-
QOTLBO [26]	111,498	-	-	-	-
QPSO [59]	119,005.3030	121,621.7556	122,144.8454	372	10,124.12
GQPSO [59]	112,429.7444	113,102.4627	113,327.0680	256	1604.83
RCCRO [35]	111,497.6319	-	-	-	-
BSA [56]	111,497.6308	-	-	-	-
CSCA [27]	111,497.6307	-	-	-	-
QOPO [19]	111,892.4096	-	-	-	-
SGO	111,497.6302	111,497.7362	111,502.7703	7.27×10^{-1}	0.10
MSGO	111,497.6301	111,497.6302	111,497.6304	6.25×10^{-5}	0

* Indicates the cost saving through MSGO compared to other algorithms using the *Average Cost* item.

Table 6. Values of the statistical items obtained using different algorithms for the cases C_{2a}—*Best Cost* (40 units without losses, 10,500 MW).

Algorithm	Best Cost ($/h)	Average Cost ($/h)	Worst Cost ($/h)	SD ($/h)	Cost Saving ($/h)
MIMO [58]	122,758.7	124,621.8	126,059.2	866.20	2964.84
FSS-IPSO2 [64]	122,535.56	125,025.86	127,401.23	1134.43	3368.90
GA [50]	121,996.4	122,919.77	123,807.97	320.31	1262.81
HSCA [60]	121,983.5	-	-	-	-
NGWO [65]	121,881.81	122,787.77	-	-	1130.81
DE [42]	121,840	-	-	-	-
PSO [52]	122,588.5093	123,544.88	124,733.67	-	1887.92
TLBO [53]	124,517.27	126,581.56	128,207.06	1060	4924.60
CTLBO [53]	121,553.83	121,790.23	122,116.18	150	133.27
SMA [66]	121,658.6656	-	-	-	-
L-SHADE [67]	121,543.43	122,105.39	122,983.68	-	448.43
S-Jaya [68]	121,517.6513	121,948.42	122,283.83	193.57	291.46
SCA [54]	121,506.58	121,857.90	122,056.15	347.26	200.94
ABC [51]	121,479.6467	121,984.24	122,137.42	—	327.28
Jaya-SML [46]	121,476.3977	121,689.07	122,039.87	147.89	32.11
TLABC [61]	121,468.3847	121,739.4406	122,192.3263	160.88	82.48
CSO [55]	121,465.99	121,988.48	122,781.75	275.92	331.52
IJaya [68]	121,454.3785	121,770.32	122,109.01	173.70	113.36
ESCSDO10 [69]	121,626.97	122,351.7	123,128.9	412.2976	694.74
SDO [69]	121,750.2	122,460.1	123,222.7	405.019	803.14
SGO	121,509.8209	122,025.1179	123,527.6187	380	368.16
MSGO	121,426.7039	121,656.9571	122,048.2807	143	0

Table 7. Values of the statistical items obtained using different algorithms for the cases C_{3a}—*Best Cost* (40 units with losses and without wind, 10,500 MW).

Algorithm	Best Cost ($/h)	Average Cost ($/h)	Worst Cost ($/h)	SD ($/h)	Cost Saving ($/h)
GAAPI [44]	139,864.96	-	-	-	-
SDE [62]	138,157.46	-	-	-	-
ACS [57]	137,413.73	-	-	-	-
BBO [47]	137,026.82	137,116.58	137,587.8200	-	382.64
ORCCRO [47]	136,855.1900	136,855.1900	136,855.1900	-	121.25
HPSO-DE [63]	136,835.0021	-	-	-	-
CSS [48]	136,679.0228	136,993.6115	137,447.4131	171.26	259.67
MIMO [58]	137,034.2000	138,472.9000	140,124.3000	752.74	1738.96
IMO [58]	138,789.6000	140,486.3000	142,106.7000	765.71	3752.36
SGO	136,510.7626	137,150.7361	138,082.7271	415	416.80
MSGO	136,454.6072	136,733.9409	137,185.4625	174	0

Table 8. Values of the statistical items obtained using different algorithms for the cases C_{4a}—*Best Cost* (40 units with losses and with wind, 10,500 MW).

Algorithm	Best Cost ($/h)	Average Cost ($/h)	Worst Cost ($/h)	SD ($/h)	Cost Saving ($/h)
PSO	123,607.9479	124,438.1644	125,509.7725	463.89	877.02
DE	123,804.0394	123,962.8486	124,160.1466	78.42	401.70
SCA	125,895.2706	126,196.7393	126,479.6091	132.44	2635.60
SGO	123,289.7874	124,086.1871	125,789.8060	532.90	525.04
MSGO	123,161.8867	123,561.1438	124,239.9876	212.44	0

Table 9. Values of the statistical items obtained using different algorithms for the cases C_{1b}—*Best Emission* (10 units with losses, 2000 MW).

Algorithm	Best Emission (lb/h)	Average Emission (lb/h)	Worst Emission (lb/h)	SD (lb/h)	Emission Reduction * (lb/h)
DE [49]	3923.40 **	-	-	-	-
QPSO [59]	4032.3875	4041.9171	4058.3615	8.06	109.6
GQPSO [59]	4011.9244	4032.9320	4042.1878	7.55	100.6
RCCRO [35]	3932.243269	-	-	-	-
BSA [56]	3932.243269	-	-	-	-
NSGA-III [25]	3932.5	-	-	-	-
SGO	3932.243252	3932.2484	3932.2821	9.05×10^{-3}	≈0
MSGO	3932.243240	3932.2432	3932.2433	2.16×10^{-5}	0

* Indicates the emisssion reduction through MSGO compared to other algorithms using the *Average Cost* item; ** in the case of the DE algorithm, the correct value is 3932.41728 lb/h.

Table 10. Values of the statistical items obtained using different algorithms for the cases C_{2b}—*Best Emission* (40 units without losses, 10,500 MW).

Algorithm	Best Emission (t/h)	Average Emission (t/h)	Worst Emission (t/h)	SD (t/h)	Emission Reduction (t/h)
MBFA [43]	176,682.269	-	-	-	-
DE [26]	176,683.3	-	-	-	-
TLBO [26]	176,683.5	-	-	-	-
QOTLBO [26]	176,682.5	-	-	-	-
BSA [56]	176,682.2646	-	-	-	-
SGO	176,682.26364	176,682.26380	176,682.2646	2.04×10^{-4}	≈0
MSGO	176,682.26363	176,682.26376	176,682.2641	9.80×10^{-5}	0

Table 11. Values of the statistical items obtained using different algorithms for the cases C_{3b}—*Best Issue* (40 units with losses and without wind, 10,500 MW).

Algorithm	Best Emission (t/h)	Average Emission (t/h)	Worst Emission (t/h)	SD (t/h)	Emission Reduction (t/h)
PSO	347,578.65776	347,581.8140	347,594.6001	3.39	3.3
DE	347,877.89873	348,120.8380	348,591.0764	135	542.3
SCA	364,849.12062	372,931.5027	380,287.1626	3880	25,353.0
SGO	347,578.49061	347,578.4912	347,578.4922	3.57×10^{-4}	≈0
MSGO	347,578.49057	347,578.4910	347,578.4919	2.88×10^{-4}	0

Table 12. Values of the statistical items obtained using different algorithms for the cases C_{4b}—*Best Issue* (40 units with losses and with wind, 10,500 MW).

Algorithm	Best Emission (t/h)	Average Emission (t/h)	Worst Emission (t/h)	SD (t/h)	Emission Reduction (t/h)
PSO	193,313.7047	193,331.5342	193,373.1017	13.59	19.9
DE	193,953.9668	194,532.0465	195,040.0923	246.50	1220.5
SCA	210,484.9674	218,736.9094	227,559.3490	3971.80	25,425.3
SGO	193,311.54075	193,311.5414	193,311.5437	4.88×10^{-4}	≈0
MSGO	193,311.54071	193,311.5410	193,311.5415	2.02×10^{-4}	0

The stability of the MSGO algorithm is very good (SD item value being below 2.1×10^{-4}) for all cases on the EmD problem (C_{1b}–C_{4b}, Tables 9–12) and relatively good for the EcD problem (C_{1a}–C_{4a}, Tables 5–8).

We specify that the specific parameters of the PSO, DE, and SCA algorithms were set by performing several experiments in which the number of evaluations is considered to be the same as for the SGO and MSGO algorithms (shown in Table 1). Thus, the settings performed for these algorithms applied in the cases C_{3b}, C_{4a}, and C_{4b} are the following:

- For the PSO algorithm (N = 50, t_{max} = 400, $c_1 = c_2 = 2$, w_{min} = 0.3, w_{max} = 0.9 in cases C_{3b} and C_{4b}, respectively, N = 50, t_{max} = 700, $c_1 = c_2 = 2$, w_{min} = 0.3, w_{max} = 0.9 in the case of C_{4a}; where c_1 and c_2 are acceleration coefficients, w_{min} and w_{max} are the initial and final inertial weights).
- For the DE algorithm (N = 50, t_{max} = 400, CR = 0.2, F = 0.4 in cases C_{3b} and C_{4b}, respectively, N = 50, t_{max} = 700, CR = 0.1, F = 0.8 in the case of C_{4a}; where, CR is the crossover rate, and F is the scaling factor).
- For the SCA algorithm (N = 50, t_{max} = 400, a = 1 in cases C_{3b} and C_{4b}, respectively, N = 50, t_{max} = 700, a = 1 in the case of C_{4a}).

The comparison of the MSGO and SGO algorithms: Tables 5–12 present the values of the statistical items (B, A, W, SD) obtained using the SGO and MSGO algorithms for all the study cases (C_{1a}–C_{4a} and C_{1b}–C_{4b}, respectively). In all situations, the statistical items obtained using MSGO are higher than those obtained using SGO (except for the cases mentioned in the previous paragraph), but there are also cases where the differences between the SGO and MSGO algorithms are small (in particular, the cases C_{1b}–C_{4b} associated with EmD problem, to which case C_{1a} can be added). For this reason, the SGO and MSGO algorithms are compared using the non-parametric Wilcoxon statistical test, considering a significance-level of 1%. Also, 50 values/algorithm were simulated to compare SGO and MSGO, resulting in 50 pairs of values that are compared using the Wilcoxon test.

Table 13 shows the statistical item p-value after applying the Wilcoxon test for the comparison of SGO and MSGO in all cases studied. For the EcD (C_{1a}–C_{4a}) and EmD (C_{1b}–C_{4b}) problems, the p-value is less than 0.01 (except for cases C_{2b} and C_{3b}), which indicates that MSGO is statistically significantly better than the SGO algorithm (getting six wins out of a possible eight).

Table 13. Comparison of SGO and MSGO by the Wilcoxon test.

Cases	R^-	R^+	p-Values	Winner
C_{1a}	3.00	26.44	0.000	MSGO
C_{1b}	1.00	26.00	0.000	MSGO
C_{2a}	12.50	27.27	0.000	MSGO
C_{2b}	22.11	29.82	0.858	-
C_{3a}	12.67	28.32	0.000	MSGO
C_{3b}	21.95	27.87	0.055	-
C_{4a}	11.67	28.54	0.000	MSGO
C_{4b}	13.54	29.70	0.000	MSGO

R^- and R^+ represent the mean of the negative and positive ranks, respectively.

In order to investigate the ability of MSGO to identify a better solution compared to SGO, we will evaluate the cost savings, as well as the emission reduction for cases C_{1a}–C_{4a} and C_{1b}–C_{4b}. Thus, for C_{1b}–C_{4b} (Tables 9–12) the MSGO and SGO algorithms perform very well at solving the EmD problem. As a result, the emission reduction achieved by MSGO compared to SGO is insignificant for this type of problem. The situation is similar in the EcD problem, case C_{1a} (Table 5), when both algorithms perform very well on the small 10-unit system. However, the cost savings and emission reduction may be important when comparing MSGO with other competing algorithms. Thus, for C_{1a} case (Table 5), analyzing the Average cost item, the average cost reduction is 10,124.12 $/h compared to QPSO [59]. Also, in the EmD problem (Tables 9–12), based on the Average emission item, the average hourly emission reduction is 109.6 lb/h (compared to QPSO [59], in case C_{1b}), 542.3 t/h (relative to DE, in case C_{3b}), and 1220.5 t/h (relative to DE, in C_{4b}). Following the results in Tables 6–8, the cost savings achieved by MSGO compared to SGO is significant: 368.16 \$/h (for case C_{2a}), 416.80 \$/h (case C_{3a}), and 525.04 \$/h (case C_{4a}). These cost reductions are maintained or increased when comparing MSGO with other algorithms presented in Tables 6–8.

Convergence process*:* Figures 4 and 5 show the convergence characteristics in terms of cost and emissions obtained using the MSGO algorithm for all studied cases C_{1a}–C_{4a} and C_{1b}–C_{4b}. In all analyzed situations, convergence is fast in the first iterations (approximately 15% of the maximum number of iterations), and in the last iterations the convergence is slow, the iterative process being stabilized.

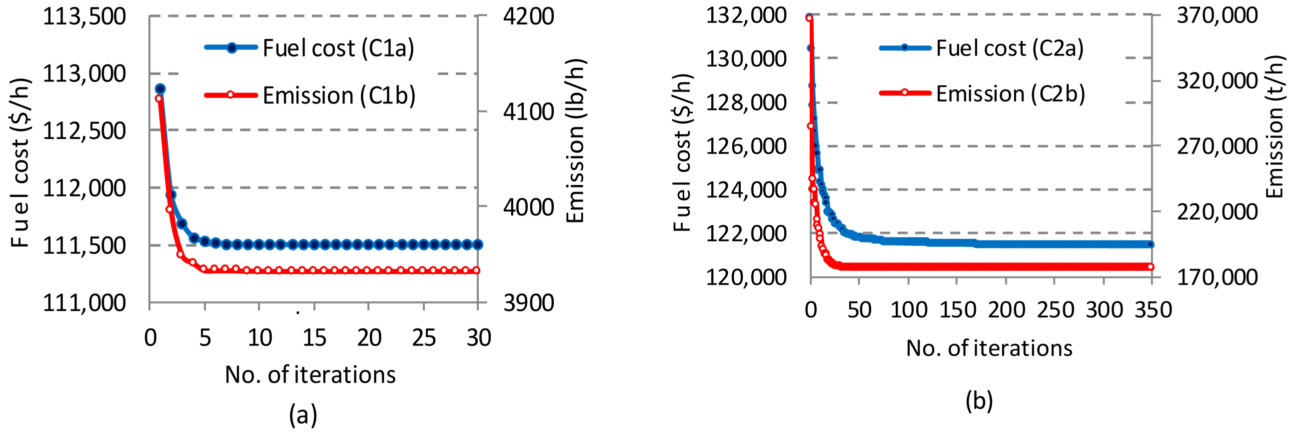

Figure 4. Cost and emission convergence characteristics obtained using MSGO: (**a**) for cases C_{1a}–C_{1b}; (**b**) for cases C_{2a}–C_{2b}.

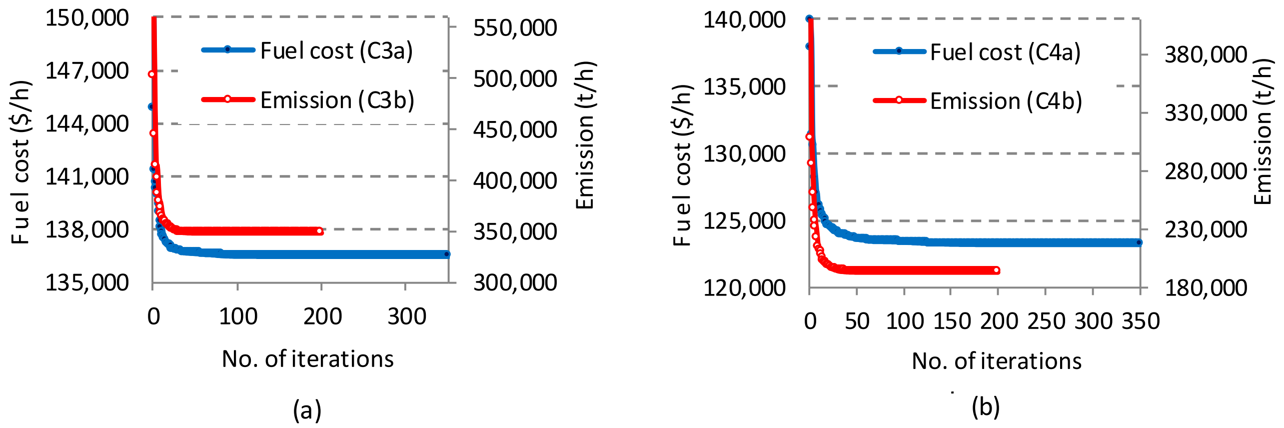

Figure 5. Cost and emission convergence characteristics obtained using MSGO: (**a**) for cases C_{3a}–C_{3b}; (**b**) for cases C_{4a}–C_{4b}.

In the case of emission convergence characteristics (C_{1b}–C_{4b}), the iterative process stabilizes much faster than in the case of cost characteristics (C_{1a}–C_{4a}). Thus, in the case of the EmD problem (C_{1b}–C_{4b}), the quasi-stable process starts after about 20–30% of the maximum number of iterations, and in the EcD problem after approximately 40–50% of the maximum number of iterations. We mention that the stable values towards which the convergence characteristics of the MSGO algorithm tend are mentioned for each case in Tables 5–12.

The accuracy and computation time*:* The accuracy (ΔP) of the computations refers to the inequality and equality constraints of the optimization model presented in Section 2 and can be seen for the best solutions presented in Tables 2–4. It is observed that the constraints are respected, the size of ΔP being less than 10^{-5} MW for all cases studied. Also, for each case (C_{1a}–C_{4a} and C_{1b}–C_{4b}), the average execution time (Time) obtained using the MSGO algorithm is indicated in Tables 2–4. In cases C_{1a} and C_{1b}, the execution time is very good (under 0.2 s); for cases C_{2a} and C_{2b} it is higher (under 10 s) due to the larger size of the analyzed system (40-unit). In the cases C_{3a}–C_{4a} and C_{3b}–C_{4b}, due to the large size of the analyzed systems (40-unit) and the need to calculate power losses, the average execution time goes up to 33 s.

5.3. Results for EED Problem (Cases C_{1c}–C_{4c})

The EED problem considers the cases C_{1c}–C_{4c}, in which the single-objective function ϕ defined by relation (20) is minimized. In these cases, it is of interest to determine the Pareto front solutions, as well as the best compromise solution (BCS) obtained using the SGO and MSGO algorithms.

To estimate the Pareto front, the weighting factor ω in relation (20) is varied between 0 and 1 with a step of 0.1 (in the end 11 points are obtained in the Cost–Emission objective plan). However, in order to obtain a Pareto front as uniform as possible and to show graphically that the solutions obtained using MSGO dominate the solutions obtained using other algorithms, another 12 points have been added to the 11 points for the case C_{1c} (the added points correspond to the values ω = {0.25, 0.35, 0.41, 0.42, 0.43, 0.435, 0.436, 0.44, 0.441, 0.445, 0.47, 0.55}) and 3 another points in the case of C_{2c} (ω = {0.51, 0.52, 0.55}), respectively. In the case of C_{3c} and C_{4c}, the number of points considered is 11.

Figures 6–8 show the Pareto fronts obtained using the SGO and MSGO algorithms for the cases C_{1c}–C_{4c}. Also, the best compromise solutions obtained using SGO, MSGO, and other competing algorithms applied in the literature are indicated.

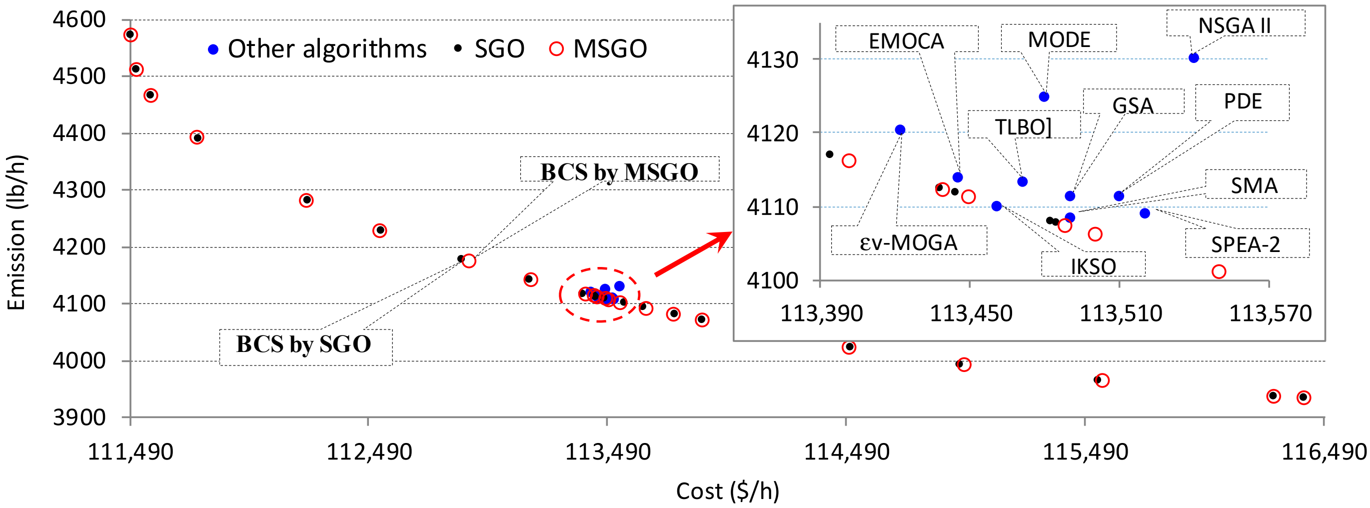

Figure 6. Pareto fronts and BCSs obtained using SGO, MSGO, and other algorithms for case C_{1c}.

Best compromise solution by SGO and MSGO: The BCSs obtained using the SGO and MSGO algorithms are identified using a fuzzy-based mechanism presented in Section 2.3. The results obtained through this mechanism indicate that BCS corresponds to a weighting factor equal to ω = 0.5, both for SGO and MSGO.

As a result, in Table 14, we present for each case C_{1c}–C_{4c}, the values for Cost, Emission, and item μ_{max} (determined according to the methodology in Section 2.3) corresponding to the best compromise solution determined by SGO and MSGO.

Comparison of solutions from the Pareto front of MSGO with BCSs obtained using other algorithms: In the case of C_{1c}, in order to test the ability of MSGO to identify solutions from the Pareto front, it was compared with BCSs obtained using a group of algorithms (denoted G1) presented in the literature G1 = {EMOCA [24], MODE [42], NSGA II [42], TLBO [26], GSA [70], PDE [42], SMA [71], εv-MOGA [49], SPES-2 [42]}. To highlight the Pareto front solutions obtained using MSGO and BCSs obtained using competing algorithms in the G1 group, in Figure 6, a detail of the Cost–Emission plan is shown. In this detail of Figure 6, it can be visually observed that the Pareto front solutions obtained using MSGO (marked with red circles) dominate, in terms of Cost and Emission, all BCSs obtained using competing algorithms G1 (marked with blue circles). Thus, the points marked with red circles (obtained using MSGO) will remove from the Pareto front of MSGO all the points marked by blue circles (obtained using competing G1 algorithms). The exception is the IKSO algorithm [72], which obtains a non-dominated solution using the solutions in the Pareto front of the MSGO (a discussion on IKSO [72] will be made below).

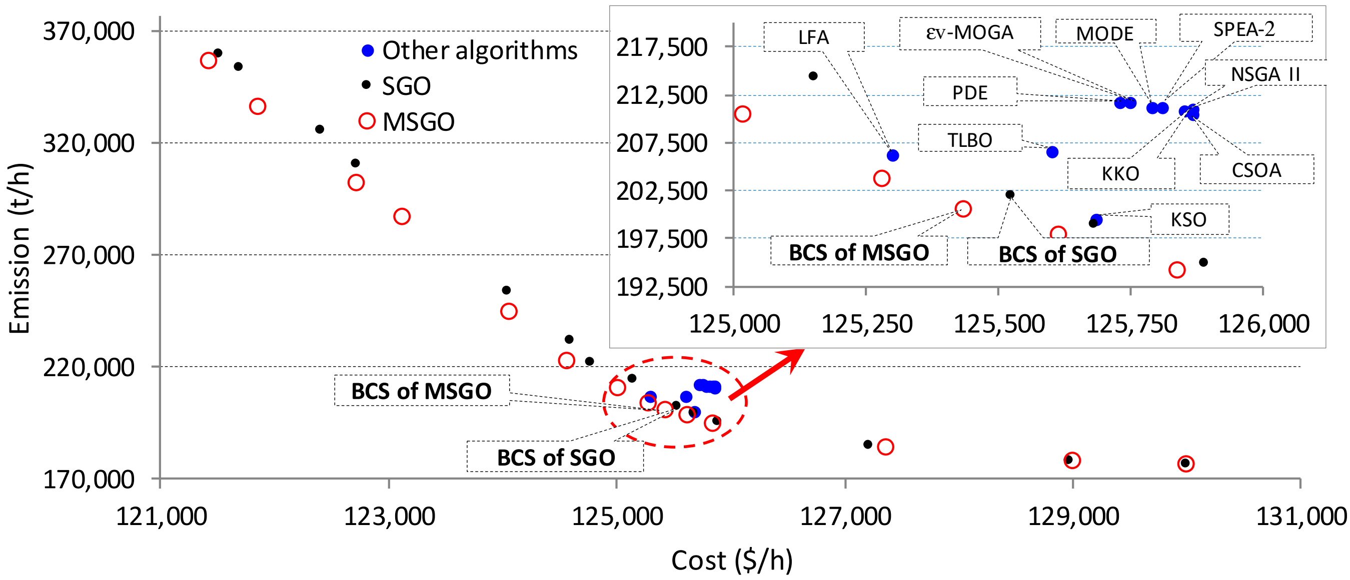

Figure 7. Pareto fronts and BCSs obtained using SGO and MSGO for case C_{2c}.

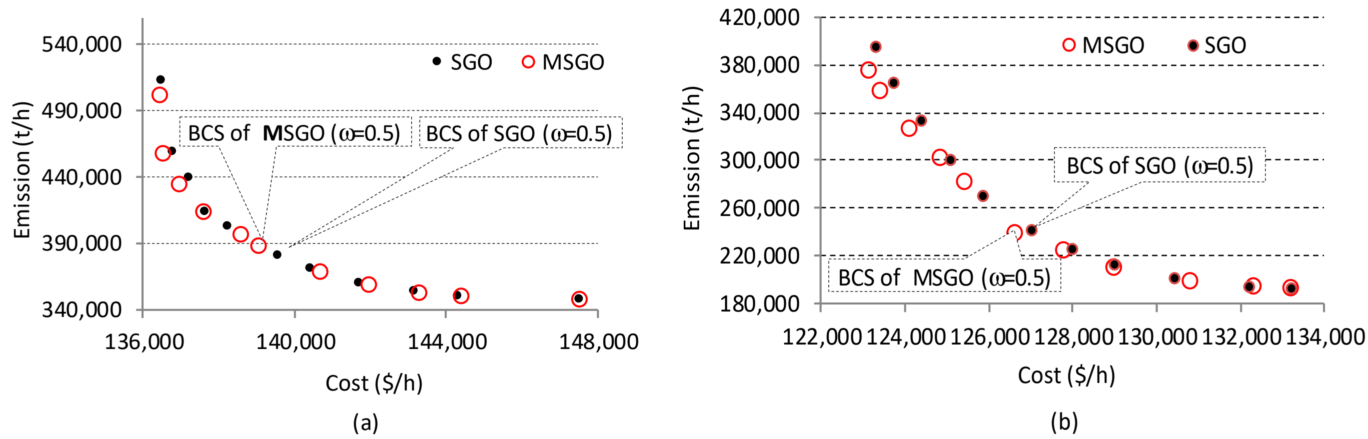

Figure 8. Pareto fronts and BCSs obtained using SGO and MSGO: (**a**) for case C_{3c}; (**b**) for case C_{4c}.

Table 14. The values corresponding to cost and emission obtained using SGO and MSGO for the cases C_{1c}–C_{4c}.

Items Cases	SGO Algorithm (ω = 0.5)			MSGO Algorithm (ω = 0.5)			Cost/Emission Reduced by MSGO **	
	Cost ($/h)	Emission (t/h)	μ_{max}	Cost ($/h)	Emission (t/h)	μ_{max}	Cost Saving ($/h)	Emission Reduction (t/h)
C_{1c}	112,884.4848	4177.6633	0.04903	112,913.8455	4173.0883	0.04908	−29.36	4.57
C_{2c}*	125,526.3426	201,944.5841	0.08106	125,434.4655	200,613.6797	0.08017	91.88	1330.90
C_{3c}	139,588.2247	380,601.7819	0.10265	139,049.0921	388,020.2454	0.10244	539.13	−7418.46
C_{4c}*	127,033.6502	242,252.893	0.10342	126,645.1342	239,155.7318	0.10661	388.52	3097.16

* MSGO's BCS dominates SGO's BCS in terms of *Cost* and *Emission*; ** MSGO Cost savings and emission reduction comparing to SGO, for each case study BSCs.

In case of C_{2c}, in Figure 7, a detail of the Cost–Emission plan is shown highlighting the Pareto front solutions obtained using MSGO, and BCSs obtained through several algorithms presented in the literature, denoted G2, G2 = {LFA [16], MODE [42], SPEA-2 [42], εv-MOGA [49], PDE [42], NSGA II [22], TLBO [26], CSOA [20], KKO [21], KSO [22]}. From the image it can be visually observed that the Pareto front solutions obtained using MSGO (marked with red circles) dominate, in terms of Cost and Emission, the BCSs obtained using the G2 group algorithms (marked with blue circles).

Comparison of BCSs obtained using MSGO and SGO: To compare BCSs obtained using the SGO and MSGO algorithms, we will include in the Pareto front of MSGO the best compromise solution identified for the SGO algorithm. Similarly, in the case of C_{1c}, the BCS identified using the IKSO algorithm [72] will be included in the Pareto front of MSGO. Note that BCSs obtained using the SGO algorithm (cases C_{1c}–C_{4c}) or IKSO [72] (in the case of C_{1c}) must be non-dominated solutions in the MSGO Pareto front. Thus, MSGO will have a Pareto front increased by one value (in cases C_{3c}, corresponding to the BCS of SGO), respectively by two values (in cases C_{1c}, corresponding to the BCSs of SGO and IKSO [72]). In cases C_{2c} and C_{4c}, the BCS of MSGO dominates the BCS of SGO in terms of Cost and Emission (see Table 14). The values from these extended Pareto fronts (with one or two other values) are subject to the methodology for establishing the BCS presented in Section 2.3, and the results of interest are given in Table 15. Table 15 shows the values of the item μ_{max} obtained using the SGO, MSGO, and IKSO algorithms (we specify that the IKSO algorithm [72] is used only in the case of C_{1c}). From Table 15, it can be seen that, for the cases C_{1c} and C_{3c}, the maximum value of the item μ corresponds to the MSGO algorithm (the weighting factor being for each case ω = 0.5). As a result, in all cases (C_{1c}–C_{4c}), the MSGO algorithm can identify a better compromise solution than SGO. Figures 5–8 show the Pareto fronts obtained using SGO and MSGO and indicate the BCSs identified using these algorithms.

Table 15. Values of item μ_{max} obtained using SGO, MSGO and IKSO for cases C_{1c}–C_{4c}.

Case	Case C_{1c}	Case C_{2c}	Case C_{3c}	Case C_{4c}
IKSO [72]	0.044284	–	–	–
SGO (ω = 0.5)	0.044676	MSGO's BCS dominates SGO's BCS, from Table 14 (MSGO winner)	0.092898	MSGO's BCS dominates SGO's BCS, from Table 14 (MSGO winner)
MSGO (ω = 0.5)	**μ_{max} = 0.044715 (MSGO winner)**		**μ_{max} = 0.092926 (MSGO winner)**	

– the IKSO algorithm [72] is not used in the C_{2c}–C_{4c} cases. The bold values correspond to the winning algorithm.

Comparison of SGO and MSGO Pareto fronts: Two metrics are used to evaluate and compare the quality of the Pareto fronts obtained using the SGO and MSGO algorithms: C-metric [73] and hyper-volume [74]. The C-metric, denoted C(S_1, S_2) and applied to the non-dominated solution sets S_1 and S_2, indicates the percentage of solutions from the S_2 set dominated by the solutions from S_1. For example, if C(S_1, S_2) = 100%, all solutions in S_2 are dominated by those in S_1. Currently, C(S_1, S_2) $\neq$ C(S_2, S_1) is required to calculate both metrics C(S_1, S_2) and C(S_2, S_1). The hyper-volume metric measures the volume (area in the case of two objectives) that is dominated by the Pareto front and is located below a predetermined reference point. Typically, higher values of the hyper-volume metric are associated with a better-quality Pareto front.

Table 16 shows, for each case, the values of the C-metric and hyper-volume metrics corresponding to the solution sets obtained using the SGO and MSGO algorithms. From Table 16 it can be seen that the C-metric (SGO, MSGO) = 0, which indicates that the solutions in the Pareto front obtained using SGO do not dominate any solution in the Pareto front obtained using the MSGO algorithm. Instead, the inverse *C*-metric (MSGO, SGO) indicates the percentages of SGO solutions dominated by MSGO solutions. Also, the hyper-volume metric shows higher values in the case of MSGO compared to SGO. Considering the values obtained using the two metrics, we estimate that MSGO can obtain a Pareto front of better quality than SGO.

Table 16. Values of the metrics for assessing the Pareto fronts of SGO and MSGO.

Item Cases	Hyper Volume		C-Metric (%)	
	SGO	MSGO	C(SGO, MSGO)	C(MSGO, SGO)
Case C_{1c}	0.9710	0.9712	0	4.55
Case C_{2c}	0.9775	0.9836	0	42.86
Case C_{3c}	1.0887	1.0942	0	27.27
Case C_{4c}	1.0007	1.0172	0	54.54

When analyzing Table 4 from the EED problem point of view—case C_{4c}—it shows that the BCS is obtained when the wind units operate at maximum capacity ($PW_1 \approx PW_2 \approx 550$ MW). When comparing the BCS-including wind (case C_{4c}) with the BCS-without wind (case C_{3c}), it results in lower costs (by 8.9%) and emissions (by 38.3%) for case C_{4c}. Table 14 shows Cost savings and Emission reduction by MSGO compared to SGO for BCSs obtained in cases C_{1c}–C_{4c}. Positive values indicate that MSGO achieves better cost or lower emissions than SGO, and negative values indicate the opposite situation.

Average execution time: Tables 2–4 show the values of the average execution time (Time) obtained using the MSGO algorithm for each case (C_{1c}–C_{4c}) of the EED problem. It can be seen that the values obtained for the EED problem are slightly higher than for the EcD or EmD problems.

6. Conclusions

In this article, a modified SGO is proposed in which several terms in the solution update relations are disturbed by including chaotic sequences generated by the Logistic map and sequences generated by the HDP operator. Also, switching between solution update relations (in the acquiring phase) is achieved by introducing a random condition instead of a condition based on the value of the fitness function of the competing solutions. MSGO has been successfully tested for solving EcD, EmD, and EED problems for medium and large power systems.

The effectiveness of MSGO was tested on four cases for each type of problem (EcD, EmD, and EED). For all case studies targeting EcD, and EmD problems, MSGO obtained solutions of better or equal quality than well-known algorithms (such as: DE, ABC, PSO, SCA, etc.) or improved varieties (except for the cases mentioned in Section 5.2). Also, the convergence process was fast and the stability of the MSGO algorithm evaluated by the SD item was very good in cases C_{1b}–C_{4b} and relatively good in the cases C_{1a}–C_{4a}. The statistical items (B, A, W and SD) and the results of the Wilcoxon test show that MSGO is more efficient than SGO in six cases (out of a possible eight) and equally good in two cases (C_{2b} and C_{3b} in Table 13), indicating that the changes made in MSGO had a positive effect. In the case of the EED problem, MSGO was able to extract a better compromise solution than SGO or other algorithms for all studied cases C_{1c}–C_{4c}. Moreover, the values of the hyper-volume and C-metric indicate that MSGO can obtain a Pareto front of superior quality compared to SGO. It should be mentioned that the inclusion of wind sources (in the case of C_{4a}/C_{4b} compared to C_{3a}/C_{3b}) brings benefits both in terms of cost reduction (about 10%) and polluting emissions (about 45%). The average computation time of MSGO and SGO algorithms is similar, and the efficiency of both algorithms is good in all case studies. By applying economic and/or emissions dispatching using MSGO, electricity producers can make informed decisions to operate a sustainable energy system.

Author Contributions: Conceptualization, D.C.S., C.H. and M.L.S.; software data curation, D.C.S.; investigation, D.C.S., C.H., M.L.S., G.B., C.B. and F.C.D.; writing—original draft preparation, D.C.S., M.L.S. and C.B.; writing—review and editing, C.H., G.B., C.B. and F.C.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the University of Oradea within the framework of the grant competition "Scientific Research of Excellence Related to Priority Areas with Capitalization through Technology Transfer: INO-TRANSFER-UO-2nd Edition" projects no. 239/2022 and no. 236/2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The research undertaken was made possible by the equal scientific involvement of all the authors concerned.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

a_i, b_i, c_i, e_i, and f_i	Fuel cost coefficients of thermal unit i;
B_{ij}, B_{0i}, B_{00}	Loss coefficients;
c	Self-introspection parameter from SGO;
c_j^d, c_j^o, c_j^u	Direct, reserve, and penalty cost coefficient for unit j;
$C(PT,PW)$	Total fuel cost;
e_j^o, e_j^u	Reserve, and penalty emission coefficient for unit j;
$E_j^o\left(PW_j\right)$, $E_j^u\left(PW_j\right)$	Mean powers associated with over and underestimation W_j for unit j;
$E(PT,PW)$	Total emission;
$f(X^{best})$	Objective function associated to X^{best};
$f_i = f(X_i), f(X_r)$	Objective function associated to X_i and X_r solutions;
k, c	Shape, and scale parameters of Weibull distribution;
n	Problem dimension;
N	Population size;
Nt, Nw	The number of thermal and wind units;
o, u	Superscript symbols attached to some quantities reflecting overestimation, and underestimation of the available wind power;
P_D	The total power demand;
P_L	Transmission line losses;
PT, PW	The output power vectors of the thermal and wind units;
PT_i	Power of the thermal unit i;
$PT_{min,i}$, $PT_{max,i}$	Real minimum and maximum power of the thermal unit i;
PW_j	Scheduled wind power of the wind unit j;
PWr	Rated power of the wind unit;
ru, δ	Parameters specific to the HDP operator;
t	Current iteration;
t_{max}	Maximum number of iterations
v_{in}, v_r, v_{out}	Cut-in, rated, and cut-out wind speeds of the wind unit;
W_j	Random variable that represents available wind power for unit j;
X_i, X_r	n-dimensional vectors associated with solutions i and r;
X_i^{new}	New solution vector X_i;
X^{best}	The vector of the best solution;
$x_{j,i}$	j^{th} component of solution X_i;
$x_{j,i}^{new}$	j^{th} component of solution X_i^{new}
x_j^{best}	j^{th} component of solution X^{best}
α, β, δ_{max}	Specific parameters of the MSGO algorithm;
α_j, β_j, γ_j, δ_j, and λ_j	Emission coefficients of the thermal unit i;
ω	Weighting factor;
$\{cx_p\}$	Chaotic sequence;
$\exp(\bullet)$	exponential function;
$f_W(\bullet)$	Probability density function (pdf) of random variable W;
$Prob(\bullet)$	The probability of the event;
min, max	Indicate the minimum and maximum limits of some variables/functions;
Min, Max	Mathematical functions that determine the minimum and maximum value in a set.

Abbreviations

ABC: Artificial bee colony algorithm; ACS: Artificial cooperative search; BBO: Biogeography-based optimization; BCS: Best compromise solution; BSA: Backtracking search optimization; CA: Cultural algorithm; CSCA: Chaotic SCA; CSO: Cuckoo search optimization; CSOA: Criminal search optimization algorithm; CSS: Charged system search; CTLBO: Chaotic TLBO; CTO: Class topper optimization; DE: Differential evolution; EcD: Economic dispatch; EED: Economic emission dispatch; EmD: Emission dispatch; EMA: Exchange market algorithm; EMOCA: Enhanced multi-objective cultural algorithm; εv-MOGA: Epsilon-multi-objective GA; ESCSDO10: Eagle-strategy supply–demand based optimization algorithm with chaotic map ten; FSS-IPSO2: Floating search space with improved PSO; GA: Genetic algorithm; GAAPI: GA with ant colony optimization; GQPSO: Gaussian quantum-behaved PSO; GSA: Gravitational search algorithm; GWO: Grey wolf optimization; HDP: Highly disruptive polynomial; HPSO-DE: Hybrid PSO with DE; HSCA: Hybrid SCA; IJaya: Improved Jaya algorithm; IKSO: Improve kernel search optimization; IMO: Ion motion optimization; ISA: Interior search algorithm; Jaya-SML: Jaya algorithm with SAMP and Lévy flights; KKO: Kho-Kho optimization algorithm; KSO: Kernel search optimization; LFA: Lightning flash algorithm; LPSR: Linear population size reduction; L-SHADE: LPSR with success-history based adaptive DE; MBFA: Modified bacterial foraging algorithm; MIMO: Modified IMO; MODE: Multi-objective DE; MSGO: Modified SGO; NGWO: Novel grey wolf optimization; NSGA II/III: Non-dominated sorting genetic algorithm-version II/III; ORCCRO: Oppositional real coded chemical reaction optimization; PDE: Pareto differential evolution; pdf: Probability density function; PSO: Particle swarm optimization; QOPO: Quasi-oppositional-based political optimizer; QOTLBO: Quasi-oppositional TLBO; QPSO: Quantum PSO; RCCRO: Real coded chemical reaction optimization; SAMP: Self-adaptive multi-population; SCA: Sine–cosine algorithm; SDE: Shuffled differential evolution; SDO: Supply–demand optimization; SGO: Social group optimization; S-Jaya: Jaya with self-adaptive po-pulation mechanism; SMA: Slime mould algorithm; SPES-2: Strength pareto evolutionary algorithm 2; SSA: Squirrel search algorithm; TLABC: Teaching-learning-based ABC; TLBO: Teaching learning-based-optimization; VPE: Valve-point effects; Wind: Indicates the inclusion of wind power in EcD, EmD, or EED problems.

Appendix A

Table A1. Assessment of the cost related to wind power: steps and numerical example.

Step	Description and Exemplification
	To Evaluate the Cost Related to the Wind Power, the Main Steps That Have to Be Done Are Presented, as Well as How Each of Them Is Applied for a Specific Case of a Wind Unit:
1	Set the input data for a wind unit: the shape (k = 1.5) and scale (c = 15) parameters; rated output power of the wind unit (PWr = 550 MW); cut-in wind speed (v_{in} = 5 m/s), rated wind speed (v_r = 15 m/s), cut-out wind speed (v_{out} = 45 m/s); direct cost coefficient (c^d = 0), reserve cost coefficient (c^o = 5), penalty cost coefficient (c^u = 5). Because during the MSGO optimization process, the variable "scheduled wind power" (PW) vary between the limits (PW_{min} = 0, PW_{max} = 550) MW, the calculation is performed for an arbitrarily chosen value, for example PW = 400 MW
2	Set the expression of the pdf $f_W(w)$ related to the wind power (W) using relation (7): $f_W(w) = \frac{1.5\cdot 5\cdot 2}{15\cdot 550}\left(\frac{5}{15}\cdot\left(1+\frac{w\cdot 2}{550}\right)\right)^{1.5-1} exp\left[-\left(\frac{5}{15}\cdot\left(1+\frac{w\cdot 2}{550}\right)\right)^{1.5}\right] = \frac{1}{550}\left(\frac{1}{3}\cdot\left(1+\frac{w}{275}\right)\right)^{0.5} exp\left[-\left(\frac{1}{3}\cdot\left(1+\frac{w}{275}\right)\right)^{1.5}\right]$, where, $R = (15-5)/5 = 2$
3	Calculate the discrete probabilities $Prob(W = 0)$, and $Prob(W = PWr)$, using relations (8) and (9): $Prob(W = 0) = 1 - exp[-(v_{in}/c)^k] + exp[-(v_{out}/c)^k] = 1 - \exp[(-(5/15)^{1.5}] + \exp[-(45/15)^{1.5}] = 0.180602$ $Prob(W = PWr) = Prob(W = 550) = exp[-(v_r/c)^k] - exp[-(v_{out}/c)^k] = \exp[-(15/15)^{1.5}] - \exp[-(45/15)^{1.5}] = 0.362342$
4	Calculate the average power E^o(PW = 400) and E^u(PW = 400) according to (10) and (11): $E^o(PW = 400) = \int_0^{400} (400 - w) f_W(w)dw + (400-0)\cdot 0.180602 = 143.087\text{MW}$ $E^u(PW = 400) = \int_{400}^{550} (w - 550) f_W(w)dw + (550-400)\cdot 0.362342 = 45.944\text{MW}$ where $f_W(w)$ is presented in Step 2 and the integrals may by directly calculated in mathcad.
5	Calculate the cost $C^W(PW)$ related to the wind power PW, using (3): $C^W(PW) = \sum_{j=1}^{Nw=1} C_j^W(PW_j) = c^d\cdot PW + c^o\cdot E^o(PW = 400) + c^u\cdot E^u(PW = 400) = 0\cdot 400 + 5\cdot 143.087 + 5\cdot 45.944 = 945.153\ \$/\text{h}$

Table A2. Thermal units' characteristics for the 10-unit test system.

Unit	Power Limits		Fuel Cost Coefficients					Emission Coefficients				
	$P_{min,i}$	$P_{max,i}$	a_i	b_i	c_i	e_i	f_i	α_i	β_i	γ_i	δ_i	λ_i
	MW	MW	\$/MW^{2}h	\$/MWh	\$/h	\$/h	rad/MW	lb/MW^{2}h	lb/MWh	lb/h	lb/h	1/MW
1	10	55	0.12951	40.5407	1000.403	33	0.0174	0.04702	−3.9864	360.0012	0.25475	0.01234
2	20	80	0.10908	39.5804	950.606	25	0.0178	0.04652	−3.9524	350.0056	0.25475	0.01234
3	47	120	0.12511	36.5104	900.705	32	0.0162	0.04652	−3.9023	330.0056	0.25163	0.01215
4	20	130	0.12111	39.5104	800.705	30	0.0168	0.04652	−3.9023	330.0056	0.25163	0.01215
5	50	160	0.15247	38.5390	756.799	30	0.0148	0.00420	0.3277	13.8593	0.24970	0.01200
6	70	240	0.10587	46.1592	451.325	20	0.0163	0.00420	0.3277	13.8593	0.24970	0.01200
7	60	300	0.03546	38.3055	1243.531	20	0.0152	0.00680	−0.5455	40.2669	0.24800	0.01290
8	70	340	0.02803	40.3965	1049.998	30	0.0128	0.00680	−0.5455	40.2669	0.24990	0.01203
9	135	470	0.02111	36.3278	1658.569	60	0.0136	0.00460	−0.5112	42.8955	0.25470	0.01234
10	150	470	0.01799	38.2704	1356.659	40	0.0141	0.00460	−0.5112	42.8955	0.25470	0.01234

Table A3. The B-loss coefficients ($[B_{ij}]$, $[B_{0i}]$, and B_{00}) for 10-unit system.

		1	2	3	4	5	6	7	8	9	10
$[B_{ij}]_{10\times10}$=	1	4.9×10^{-5}	1.4×10^{-5}	1.5×10^{-5}	1.5×10^{-5}	1.6×10^{-5}	1.7×10^{-5}	1.7×10^{-5}	1.8×10^{-5}	1.9×10^{-5}	2.0×10^{-5}
	2	1.4×10^{-5}	4.5×10^{-5}	1.6×10^{-5}	1.6×10^{-5}	1.7×10^{-5}	1.5×10^{-5}	1.5×10^{-5}	1.6×10^{-5}	1.8×10^{-5}	1.8×10^{-5}
	3	1.5×10^{-5}	1.6×10^{-5}	3.9×10^{-5}	1.0×10^{-5}	1.2×10^{-5}	1.2×10^{-5}	1.4×10^{-5}	1.4×10^{-5}	1.6×10^{-5}	1.6×10^{-5}
	4	1.5×10^{-5}	1.6×10^{-5}	1.0×10^{-5}	4.0×10^{-5}	1.4×10^{-5}	1.0×10^{-5}	1.1×10^{-5}	1.2×10^{-5}	1.4×10^{-5}	1.5×10^{-5}
	5	1.6×10^{-5}	1.7×10^{-5}	1.2×10^{-5}	1.4×10^{-5}	3.5×10^{-5}	1.1×10^{-5}	1.3×10^{-5}	1.3×10^{-5}	1.5×10^{-5}	1.6×10^{-5}
	6	1.7×10^{-5}	1.5×10^{-5}	1.2×10^{-5}	1.0×10^{-5}	1.1×10^{-5}	3.6×10^{-5}	1.2×10^{-5}	1.2×10^{-5}	1.4×10^{-5}	1.5×10^{-5}
	7	1.7×10^{-5}	1.5×10^{-5}	1.4×10^{-5}	1.1×10^{-5}	1.3×10^{-5}	1.2×10^{-5}	3.8×10^{-5}	1.6×10^{-5}	1.6×10^{-5}	1.8×10^{-5}
	8	1.8×10^{-5}	1.6×10^{-5}	1.4×10^{-5}	1.2×10^{-5}	1.3×10^{-5}	1.2×10^{-5}	1.6×10^{-5}	4.0×10^{-5}	1.5×10^{-5}	1.6×10^{-5}
	9	1.9×10^{-5}	1.8×10^{-5}	1.6×10^{-5}	1.4×10^{-5}	1.5×10^{-5}	1.4×10^{-5}	1.6×10^{-5}	1.5×10^{-5}	4.2×10^{-5}	1.9×10^{-5}
	10	2.0×10^{-5}	1.8×10^{-5}	1.6×10^{-5}	1.5×10^{-5}	1.6×10^{-5}	1.5×10^{-5}	1.8×10^{-5}	1.6×10^{-5}	1.9×10^{-5}	4.4×10^{-5}
$[B_{0i}]_{1\times10}$=		0	0	0	0	0	0	0	0	0	0
B_{00}=		0									

Table A4. Thermal units' characteristics for the 40-unit test system.

Unit	Power Limits		Fuel Cost Coefficients					Emission Coefficients				
	$P_{min,i}$	$P_{max,i}$	a_i	b_i	c_i	e_i	f_i	α_i	β_i	γ_i	δ_i	λ_i
	MW	MW	\$/MW^{2}h	\$/MWh	\$/h	\$/h	rad/MW	t/MW^{2}h	t/MWh	t/h	t/h	1/MW
1	36	114	0.0069	6.73	94.705	100	0.084	0.048	−2.22	60	1.31	0.0569
2	36	114	0.0069	6.73	94.705	100	0.084	0.048	−2.22	60	1.31	0.0569
3	60	120	0.02028	7.07	309.54	100	0.084	0.0762	−2.36	100	1.31	0.0569
4	80	190	0.00942	8.18	369.03	150	0.063	0.054	−3.14	120	0.9142	0.0454
5	47	97	0.0114	5.35	148.89	120	0.077	0.085	−1.89	50	0.9936	0.0406
6	68	140	0.01142	8.05	222.33	100	0.084	0.0854	−3.08	80	1.31	0.0569
7	110	300	0.00357	8.03	287.71	200	0.042	0.0242	−3.06	100	0.6550	0.02846
8	135	300	0.00492	6.99	391.98	200	0.042	0.031	−2.32	130	0.6550	0.02846
9	135	300	0.00573	6.6	455.76	200	0.042	0.0335	−2.11	150	0.6550	0.02846
10	130	300	0.00605	12.9	722.82	200	0.042	0.425	−4.34	280	0.6550	0.02846
11	94	375	0.00515	12.9	635.2	200	0.042	0.0322	−4.34	220	0.6550	0.02846
12	94	375	0.00569	12.8	654.69	200	0.042	0.0338	−4.28	225	0.6550	0.02846
13	125	500	0.00421	12.5	913.4	300	0.035	0.0296	−4.18	300	0.5035	0.02075
14	125	500	0.00752	8.84	1760.4	300	0.035	0.0512	−3.34	520	0.5035	0.02075
15	125	500	0.00708	9.15	1728.3	300	0.035	0.0496	−3.55	510	0.5035	0.02075
16	125	500	0.00708	9.15	1728.3	300	0.035	0.0496	−3.55	510	0.5035	0.02075
17	220	500	0.00313	7.97	647.85	300	0.035	0.0151	−2.68	220	0.5035	0.02075
18	220	500	0.00313	7.95	649.69	300	0.035	0.0151	−2.66	222	0.5035	0.02075
19	242	550	0.00313	7.97	647.83	300	0.035	0.0151	−2.68	220	0.5035	0.02075
20	242	550	0.00313	7.97	647.81	300	0.035	0.0151	−2.68	220	0.5035	0.02075
21	254	550	0.00298	6.63	785.96	300	0.035	0.0145	−2.22	290	0.5035	0.02075
22	254	550	0.00298	6.63	785.96	300	0.035	0.0145	−2.22	285	0.5035	0.02075
23	254	550	0.00284	6.66	794.53	300	0.035	0.0138	−2.26	295	0.5035	0.02075
24	254	550	0.00284	6.66	794.53	300	0.035	0.0138	−2.26	295	0.5035	0.02075

Table A4. *Cont.*

Unit	Power Limits		Fuel Cost Coefficients					Emission Coefficients				
	$P_{min,i}$	$P_{max,i}$	a_i	b_i	c_i	e_i	f_i	α_i	β_i	γ_i	δ_i	λ_i
	MW	MW	$/MW^{2}h	$/MWh	$/h	$/h	rad/MW	t/MW^{2}h	t/MWh	t/h	t/h	1/MW
25	254	550	0.00277	7.1	801.32	300	0.035	0.0132	−2.42	310	0.5035	0.02075
26	254	550	0.00277	7.1	801.32	300	0.035	0.0132	−2.42	310	0.5035	0.02075
27	10	150	0.52124	3.33	1055.1	120	0.077	1.842	−1.11	360	0.9936	0.0406
28	10	150	0.52124	3.33	1055.1	120	0.077	1.842	−1.11	360	0.9936	0.0406
29	10	150	0.52124	3.33	1055.1	120	0.077	1.842	−1.11	360	0.9936	0.0406
30	47	97	0.0114	5.35	148.89	120	0.077	0.085	−1.89	50	0.9936	0.0406
31	60	190	0.0016	6.43	222.92	150	0.063	0.0121	−2.08	80	0.9142	0.0454
32	60	190	0.0016	6.43	222.92	150	0.063	0.0121	−2.08	80	0.9142	0.0454
33	60	190	0.0016	6.43	222.92	150	0.063	0.0121	−2.08	80	0.9142	0.0454
34	90	200	0.0001	8.95	107.87	200	0.042	0.0012	−3.48	65	0.6550	0.02846
35	90	200	0.0001	8.62	116.58	200	0.042	0.0012	−3.24	70	0.6550	0.02846
36	90	200	0.0001	8.62	116.58	200	0.042	0.0012	−3.24	70	0.6550	0.02846
37	25	110	0.0161	5.88	307.45	80	0.098	0.095	−1.98	100	1.42	0.0677
38	25	110	0.0161	5.88	307.45	80	0.098	0.095	−1.98	100	1.42	0.0677
39	25	110	0.0161	5.88	307.45	80	0.098	0.095	−1.98	100	1.42	0.0677
40	242	550	0.00313	7.97	647.83	300	0.035	0.0151	−2.68	220	0.5035	0.02075

Table A5. The B-loss coefficients ($[B_{ij}]$, $[B_{0i}]$, and B_{00}) for 40-unit system.

$([B_{ij}]_{40\times40}) \times 10^{-6}$ =	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
1	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1
2	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1
3	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0
4	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24
5	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6
6	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8
7	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1
8	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1
9	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0
10	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24
11	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6
12	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8
13	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1
14	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1
15	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0
16	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24
17	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6
18	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8
19	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1
20	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1
21	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0
22	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24
23	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6
24	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8
25	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1
26	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1
27	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0
28	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24
29	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6
30	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8
31	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1
32	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1
33	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0
34	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24
35	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6	129	−2	−5	−6	−10	−6
36	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8	−2	150	−2	−1	−6	−8
37	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1	−5	−2	17	12	7	−1
38	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1	−6	−1	12	14	9	1
39	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0	−10	−6	7	9	31	0
40	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24	−6	−8	−1	1	0	24

$(B_{0i}) \times 10^{-4}$ =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	−3.908	−1.297	7.047	0.591	2.161	−6.635	−3.908	−1.297	7.047	0.591	2.161	−6.635	−3.908	−1.297	7.047	0.591	2.161	−6.635	−3.908	−1.297
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
	7.047	0.591	2.161	−6.635	−3.908	−1.297	7.047	0.591	2.161	−6.635	−3.908	−1.297	7.047	0.591	2.161	−6.635	−3.908	−1.297	7.047	0.591
B_{00} =	0.56																			

References

1. Jin, J.; Wen, Q.; Qiu, Y.; Cheng, S.; Guo, X. Distributed robust optimization for low-carbon dispatch of wind-thermal power under uncertainties. *Environ. Sci. Pollut. Res.* **2023**, *30*, 20980–20994. [CrossRef] [PubMed]
2. Jin, J.; Zhou, D.; Zhou, P.; Miao, Z. Environmental/economic power dispatch with wind power. *Renew. Energy* **2014**, *71*, 234–242. [CrossRef]
3. Nanda, J.; Kothari, D.P.; Linga Murthy, K.S. Economic emission load dispatch through goal programming techniques. *IEEE Trans. Energy Convers.* **1988**, *3*, 26–32. [CrossRef] [PubMed]
4. Hetzer, J.; Yu, D.C.; Bhattarai, K. An economic dispatch model incorporating wind power. *IEEE Trans. Energy Convers.* **2008**, *23*, 603–611. [CrossRef]
5. Dhillon, J.S.; Parti, S.C. Stochastic economic emission load dispatch. *Electr. Power Syst. Res.* **1993**, *26*, 179–186. [CrossRef]
6. Martins, A.C.S.; Balbo, A.R.; Jones, D.; Nepomuceno, L.; Soler, E.M.; Baptista, E.C. A Hybrid Multi-Criteria Methodology for Solving the Sustainable Dispatch Problem. *Sustainability* **2020**, *12*, 6780. [CrossRef]
7. Mahdy, A.; El-Sehiemy, R.; Shaheen, A.; Ginidi, A.; Elbarbary, Z.M.S. An improved artificial ecosystem algorithm for economic dispatch with combined heat and power units. *Appl. Sci.* **2022**, *12*, 11773. [CrossRef]
8. El-Sehiemy, R.; Shaheen, A.; Ginidi, A.; Elhosseini, M. A honey badger optimization for minimizing the pollutant environmental emissions-based economic dispatch model integrating combined heat and power units. *Energies* **2022**, *15*, 7603. [CrossRef]
9. Ginidi, A.; Elsayed, A.; Shaheen, A.; Elattar, E.; El-Sehiemy, R. An innovative hybrid heap-based and jellyfish search algorithm for combined heat and power economic dispatch in electrical grids. *Mathematics* **2021**, *9*, 2053. [CrossRef]
10. Shaheen, A.M.; Ginidi, A.R.; El-Sehiemy, R.A.; Ghoneim, S.S.M. Economic power and heat dispatch in cogeneration energy systems using manta ray foraging optimizer. *IEEE Access* **2020**, *8*, 208281–208295. [CrossRef]
11. Liu, X.; Xu, W. Minimum emission dispatch constrained by stochastic wind power availability and cost. *IEEE Trans. Power Syst.* **2010**, *25*, 1705–1713.
12. Jadhav, H.T.; Roy, R. Gbest guided artificial bee colony algorithm for environmental/economic dispatch considering wind power. *Expert Syst. Appl.* **2013**, *40*, 6385–6399. [CrossRef]
13. Dubey, H.M.; Pandit, M.; Panigrahi, B.K. An overview and comparative analysis of recent bio-inspired optimization techniques for wind integrated multi-objective power dispatch. *Swarm Evol. Comput.* **2018**, *38*, 12–34. [CrossRef]
14. Tsvetkova, O.; Ouarda, T.B.M.J. Use of the Halphen distribution family for mean wind speed estimation with application to Eastern Canada. *Energy Convers. Manag.* **2023**, *276*, 116502. [CrossRef]
15. Ghasemi, A.; Gheydi, M.; Golkar, M.J.; Eslami, M. Modeling of Wind/Environment/Economic Dispatch in power systemand solving via an online learning meta-heuristic method. *Appl. Soft Comput.* **2016**, *43*, 454–468. [CrossRef]
16. Kheshti, M.; Kang, X.; Li, J.; Regulski, P.; Terzija, V. Lightning Flash Algorithm for Solving Nonconvex Combined Emission Economic Dispatch with Generator Constraints. *IET Gener. Transm. Distrib.* **2018**, *12*, 104–116. [CrossRef]
17. Wang, B.; Zhang, P.; He, Y.; Wang, X.; Zhang, X. Scenario-oriented hybrid particle swarm optimization algorithm for robust economic dispatch of power system with wind power. *J. Syst. Eng. Electron.* **2022**, *33*, 1143–1150. [CrossRef]
18. Srivastava, A.; Das, D.K. An adaptive chaotic class topper optimization technique to solve economic load dispatch and emission economic dispatch problem in power system. *Soft Comput.* **2022**, *26*, 2913–2934. [CrossRef]
19. Basetti, V.; Rangarajan, S.S.; Shiva, C.K.; Pulluri, H.; Kumar, R.; Collins, R.E.; Senjyu, T. Economic Emission Load Dispatch Problem with Valve-Point Loading Using a Novel Quasi-Oppositional-Based Political Optimizer. *Electronics* **2021**, *10*, 2596. [CrossRef]
20. Srivastava, A.; Das, D.K. Criminal Search Optimization Algorithm: A Population-Based Meta-Heuristic Optimization Technique to Solve Real-World Optimization Problems. *Arab. J. Sci. Eng.* **2022**, *47*, 3551–3571. [CrossRef]
21. Srivastava, A.; Das, D.K. A new Kho-Kho optimization algorithm: An application to solve combined emission economic dispatch and combined heat and power economic dispatch problem. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103763. [CrossRef]
22. Nourianfar, H.; Abdi, H. Environmental/Economic dispatch using a new hybridizing algorithm integrated with an effective constraint handling technique. *Sustainability* **2022**, *14*, 3173. [CrossRef]
23. Sakthivel, V.P.; Suman, M.; Sathya, P.D. Combined economic and emission power dispatch problems through multi-objective squirrel search algorithm. *Appl. Soft Comput. J.* **2021**, *100*, 106950. [CrossRef]
24. Zhang, R.; Zhou, J.; Moa, L.; Ouyang, S.; Liao, X. Economic environmental dispatch using an enhanced multi-objective cultural algorithm. *Electr. Power Syst. Res.* **2013**, *99*, 18–29. [CrossRef]
25. Bhesdadiya, R.H.; Trivedi, I.N.; Jangir, P.; Jangir, N.; Kumar, A. An NSGA-III algorithm for solving multi-objective economic/environmental dispatch problem. *Cogent Eng.* **2016**, *3*, 1269383. [CrossRef]
26. Roy, P.K.; Bhui, S. Multi-objective quasi-oppositional teaching learning based optimization for economic emission load dispatch problem. *Electr. Power Energy Syst.* **2013**, *53*, 937–948. [CrossRef]
27. Guesmi, T.; Farah, A.; Marouani, I.; Alshammari, B.; Abdallah, H.H. Chaotic sine–cosine algorithm for chance-constrained economic emission dispatch problem including wind energy. *IET Renew. Power Gener.* **2020**, *14*, 1808–1821. [CrossRef]
28. Das, D.; Bhattacharya, A.; Ray, R.N. Dragonfly Algorithm for solving probabilistic Economic Load Dispatch problems. *Neural Comput. Appl.* **2020**, *32*, 3029–3045. [CrossRef]
29. Satapathy, S.C.; Naik, A. Social group optimization (SGO): A new population evolutionary optimization technique. *Complex Intell. Syst.* **2016**, *2*, 173–203. [CrossRef]

30. Naik, A.; Satapathy, S.C.; Abraham, A. Modified social group optimization—A meta-heuristic algorithm to solve short-term hydrothermal scheduling. *Appl. Soft Comput.* **2020**, *95*, 106524. [CrossRef]
31. Reddy, A.K.V.K.; Narayana, K.V.L. A combinatorial social group whale optimization algorithm for numerical and engineering optimization problems. *Appl. Soft Comput.* **2021**, *99*, 106903.
32. Jena, J.J.; Satapathy, S.C. A new adaptive tuned social group optimization (SGO) algorithm with sigmoid-adaptive inertia weight for solving engineering design problems. In *Multimedia Tools and Applications*; Springer: Berlin/Heidelberg, Germany, 2021. [CrossRef]
33. Sinha, N.; Chakrabarti, R.; Chattopadhyay, P.K. Evolutionary programming techniques for economic load dispatch. *IEEE Trans Evol. Comput.* **2003**, *7*, 83–94. [CrossRef]
34. Rajasomashekar, S.; Aravindhababu, P. Biogeography based optimization technique for best compromise solution of economic emission dispatch. *Swarm Evol. Comput.* **2012**, *7*, 47–57. [CrossRef]
35. Bhattacharjee, K.; Bhattacharya, A.; Halder, S. Solution of economic emission load dispatch problems of power systems by real coded chemical reaction algorithm. *Int. J. Electr. Power Energy Syst.* **2014**, *59*, 176–187. [CrossRef]
36. Geng, J.Q.; Sun, X.M.; Wang, H.H.; Bu, X.H.; Liu, D.H.; Li, F.; Zhao, Z.W. A modified adaptive sparrow search algorithm based on chaotic reverse learning and spiral search for global optimization. In *Neural Computing and Applications*; Springer: Berlin/Heidelberg, Germany, 2023. [CrossRef]
37. Gupta, A.; Tiwari, D.; Kumar, V.; Rana, K.P.S.; Mirjalili, S. A chaos–infused moth–flame optimizer. *Arab. J. Sci. Eng.* **2022**, *47*, 10769–10809. [CrossRef]
38. Abed-alguni, B.H.; Paul, D.; Hammad, R. Improved salp swarm algorithm for solving single-objective continuous optimization problems. *Appl. Intell.* **2022**, *52*, 17217–17236. [CrossRef]
39. Alawad, N.A.; Abed-alguni, B.H. Discrete island-based cuckoo search with highly disruptive polynomial mutation and opposition-based learning strategy for scheduling of workflow applications in cloud environments. *Arab. J. Sci. Eng.* **2021**, *46*, 3213–3233. [CrossRef]
40. Gaing, Z.-L. Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Trans. Power Syst.* **2003**, *18*, 1187–1195. [CrossRef]
41. Park, J.-B.; Jeong, Y.-W.; Shin, J.-R.; Lee, K.Y. An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Trans. Power Syst.* **2010**, *25*, 155–166.
42. Basu, M. Economic environmental dispatch using multi-objective differential evolution. *Appl. Soft Comput.* **2011**, *11*, 2845–2853. [CrossRef]
43. Hota, P.K.; Barisal, A.K.; Chakrabarti, R. Economic emission load dispatch through fuzzy based bacterial foraging algorithm. *Electr. Power Energy Syst.* **2010**, *32*, 794–803. [CrossRef]
44. Ciornei, I.; Kyriakides, E. A GA–API solution for the economic dispatch of generation in power system operation. *IEEE Trans. Power Syst.* **2012**, *27*, 233–242. [CrossRef]
45. Lam, A.Y.S.; Li, V.O.K.; Yu, J.J.Q. Real-coded chemical reaction optimization. *IEEE Trans. Evol. Comput.* **2012**, *16*, 339–353. [CrossRef]
46. Yu, J.T.; Kim, C.H.; Wadood, A.; Khurshaid, T.; Rhee, S.B. Jaya algorithm with self-adaptive multi-population and Lévy flights for solving economic load dispatch problems. *IEEE Access* **2019**, *7*, 21372–21384. [CrossRef]
47. Bhattacharjee, K.; Bhattacharya, A.; Dey, S.H.N. Oppositional real coded chemical reaction optimization for different economic dispatch problems. *Int. J. Electr. Power Energy Syst.* **2014**, *55*, 378–391. [CrossRef]
48. Zakian, P.; Kaveh, A. Economic dispatch of power systems using an adaptive charged system search algorithm. *Appl. Soft Comput. J.* **2018**, *73*, 607–622. [CrossRef]
49. Afzalan, E.; Joorabian, M. Emission, reserve and economic load dispatch problem with non-smooth and non-convex cost functions using epsilon-multi-objective genetic algorithm variable. *Electr. Power Energy Syst.* **2013**, *52*, 55–67. [CrossRef]
50. Pothiya, S.; Ngamroo, I.; Kongprawechnon, W. Ant colony optimisation for economic dispatch problem with non-smooth cost functions. *Int. J. Electr. Power Energy Syst.* **2010**, *32*, 478–487. [CrossRef]
51. Labbi, Y.; Attous, D.; Mahdad, B. Artificial bee colony optimization for economic dispatch with valve point effect. *Front. Energy* **2014**, *8*, 449–458. [CrossRef]
52. Selvakumar, A.I.; Thanushkodi, K. Optimization using civilized swarm: Solution to economic dispatch with multiple minima. *Electr. Power Syst. Res.* **2009**, *79*, 8–16. [CrossRef]
53. He, X.; Rao, Y.; Huang, J. A novel algorithm for economic load dispatch of power Systems. *Neurocomputing* **2016**, *171*, 1454–1461. [CrossRef]
54. Alghamdi, A.S. Greedy sine-cosine non-hierarchical grey wolf optimizer for solving non-convex economic load dispatch problems. *Energies* **2022**, *15*, 3904. [CrossRef]
55. Yu, J.-T.; Kim, C.-H.; Rhee, S.-B. Clustering cuckoo search optimization for economic load dispatch problem. *Neural Comput. Appl.* **2020**, *32*, 16951–16969. [CrossRef]
56. Bhattacharjee, K.; Bhattacharya, A.; Dey, S.H. Backtracking search optimization based economic environmental power dispatch problems. *Int. J. Electr. Power Energy Syst.* **2015**, *73*, 830–842. [CrossRef]
57. Kaboli, S.H.A.; Alqallaf, A.K. Solving non-convex economic load dispatch problem via artificial cooperative search algorithm. *Expert Syst. Appl.* **2019**, *128*, 14–27. [CrossRef]

58. Kumar, M.; Dhillon, J.S. A conglomerated ion-motion and crisscross search optimizer for electric power load dispatch. *Appl. Soft Comput.* **2019**, *83*, 105641. [CrossRef]
59. Coelho, L.S. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Syst. Appl.* **2010**, *37*, 1676–1683. [CrossRef]
60. Babar, M.I.; Ahmad, A.; Fayyaz, S. A hybrid sine cosine algorithm with SQP for solving convex and nonconvex economic dispatch problem. *Mehran Univ. Res. J. Eng. Technol.* **2020**, *39*, 31–46. [CrossRef]
61. Xu, S.; Xiong, G.; Mohamed, A.W.; Bouchekara, H.R.E.H. Forgetting velocity based improved comprehensive learning particle swarm optimization for non-convex economic dispatch problems with valve-point effects and multi-fuel options. *Energy* **2022**, *256*, 124511. [CrossRef]
62. Reddy, A.S.; Vaisakh, K. Shuffled differential evolution for large scale economic dispatch. *Electr. Power Syst. Res.* **2013**, *96*, 237–245. [CrossRef]
63. Naderi, E.; Azizivahed, A.; Narimani, H.; Fathi, M.; Narimani, M.R. A comprehensive study of practical economic dispatch problems by a new hybrid evolutionary algorithm. *Appl. Soft Comput.* **2017**, *61*, 1186–1206. [CrossRef]
64. Amiri, M.; Khanmohammadi, S.; Badamchizadeh, M.A. Floating search space: A new idea for efficient solving the economic and emission dispatch problem. *Energy* **2018**, *158*, 564–579. [CrossRef]
65. Xu, J.; Yan, F.; Yun, K.; Su, L.; Li, F.; Guan, J. Noninferior solution grey wolf optimizer with an independent local search mechanism for solving economic load dispatch problems. *Energies* **2019**, *12*, 2274. [CrossRef]
66. Kamboj, V.K.; Kumari, C.L.; Bath, S.K.; Prashar, D.; Rashid, M.; Alshamrani, S.S.; AlGhamdi, A.S. A cost-effective solution for non-convex economic load dispatch problems in power systems using slime mould algorithm. *Sustainability* **2022**, *14*, 2586. [CrossRef]
67. Hamdi, M.; Idomghar, L.; Chaoui, M.; Kachouri, A. An improved adaptive differential evolution optimizer for non-convex economic dispatch problems. *Appl. Soft Comput.* **2019**, *85*, 105868. [CrossRef]
68. Chen, C.; Zou, D.; Li, C. Improved Jaya algorithm for economic dispatch considering valve-point effect and multi-fuel options. *IEEE Access* **2020**, *8*, 84981–84995. [CrossRef]
69. Hassan, M.H.; Kamel, S.; Eid, A.; Nasrat, L.; Jurado, F.; Elnaggar, M.F. A developed eagle-strategy supply-demand optimizer for solving economic load dispatch problems. *Ain Shams Eng. J.* **2023**, *14*, 102083. [CrossRef]
70. Güvenç, U.; Sonmez, Y.; Duman, S. Combined economic and emission dispatch solution using gravitational search algorithm. *Sci. Iran. D Comput. Sci. Eng. Electr. Eng.* **2012**, *19*, 1754–1762. [CrossRef]
71. Ahmed, T.A.; Ebeed, M.; Refai, A.; Kamel, S. Solving combined economic and emission dispatch problem using the slime mould algorithm. *Sohag Eng. J.* **2021**, *1*, 62–70. [CrossRef]
72. Dong, R.; Chen, H.; Heidari, A.A.; Turabieh, H.; Mafarja, M.; Wang, S. Boosted kernel search: Framework, analysis and case studies on the economic emission dispatch problem. *Knowl. Based Syst.* **2021**, *233*, 107529. [CrossRef]
73. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [CrossRef]
74. Cao, Y.; Smucker, B.J.; Robinson, T.J. On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design. *J. Stat. Plan. Inference* **2015**, *160*, 60–74. [CrossRef]