electronics

MDPI

Article

A Hybrid Optimization Algorithm Based on Differential Evolution and Gray Wolf Optimizer for Solving Economic Load Dispatch Problems

Roqia Rateb [1], Maytham N. Meqdad [2] and Mohamed Deriche [3,*]

[1] Department of Information Technology, Faculty of Prince, Al-Hussein Bin Abdallah II for IT, The Hashemite University, P.O. Box 330127, Zarqa 13133, Jordan; roqiashorman@yahoo.com
[2] Intelligent Medical Systems Department, College of Sciences, Al-Mustaqbal University, Babylon 51001, Iraq
[3] Artificial Intelligence Research Centre (AIRC), Department of Data Science and Artificial Intelligence (DS&AI), Ajman University, Ajman P.O. Box 346, United Arab Emirates
* Correspondence: m.deriche@ajman.ac.ae

Abstract

Today, the economic production of energy and its transmission at the lowest possible cost is one of the most important issues in various fields of engineering. Therefore, one of the most important optimization problems in power systems and electrical networks is the economic load dispatch (ELD) problem. On the other hand, in the last two decades, the use of optimization meta-heuristic algorithms has been investigated and used by researchers in a significant and successful manner. In this article, we have introduced a powerful optimization algorithm using the combination of two popular and powerful algorithms, Differential Evolution (DE) and the Gray Wolf Optimizer (GWO) (called the HDE algorithm), and then we have tested and implemented the proposed combined method on different types of ELD problems. In the second step, we have compared our simulation results with different algorithms from other studies. This comparison clearly shows that the proposed method in this article is a powerful and useful method for the ELD problems with various functions and in different dimensions.

Keywords: differential evolution; optimization algorithms; economic load dispatch; engineering problems

1. Introduction

Economic load dispatch (ELD) is a challenging engineering problem in power system operation and planning because it aims to determine the optimal generation schedule of available generating units while satisfying practical operating constraints and minimizing the total generation cost [1–4]. Thermal generating units have remained an important source of electricity generation over recent decades. Fossil fuels are the primary energy source for many of these units; however, their high cost and environmental impact have motivated researchers to develop more efficient optimization strategies for fuel utilization. Initially, fuel cost was the main consideration in ELD formulations. With increasing environmental restrictions, pollutant emissions and other operational requirements have also become important considerations in economic dispatch models [5,6]. In practical ELD problems, nonlinear operating constraints such as ramp-rate limits and prohibited operating zones (POZs) must also be considered. The presence of multiple fuel options further increases the complexity of the problem. In addition, incorporating the valve-point effect (VPE) into

Check for updates

Academic Editor: Jen-Hao Teng

Received: 26 August 2026
Revised: 10 October 2026
Accepted: 21 October 2026
Published: 28 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

the objective function produces a non-smooth and non-convex optimization problem with multiple local optima [3–10]. Transmission losses should also be considered to obtain a more realistic representation of the ELD problem.

The increasing complexity of ELD has encouraged the development of different optimization algorithms and hybrid solution strategies. For example, the Portuguese Man o' War optimization [11] and Martial Eagle Optimizer [12] have been proposed for solving different optimization problems. Hybrid approaches have also been investigated, including combinations of the Harris Hawk and Moth–Flame Optimization algorithms [13], the Gray Wolf Optimizer and Colonial Competitive Algorithm [14], and Teaching–Learning-Based Optimization [15]. These studies demonstrate the growing use of metaheuristic and hybrid optimization techniques for complex engineering optimization problems.

Artificial intelligence and metaheuristic optimization methods have also been extensively applied to ELD during the past two decades [16–19]. One example is the hybrid DE-PSO approach, which combines Differential Evolution (DE) and Particle Swarm Optimization (PSO) to solve different ELD formulations with several nonlinear constraints [4]. This approach has been evaluated on four types of ELD problems [20–23]. Other hybrid approaches have also been developed to address the nonlinear and constrained characteristics of ELD. A hybrid Gray Wolf Optimizer (HGWO), for example, has been applied to 3-, 6-, 13-, 15-, and 40-unit systems while considering different practical constraints [5,24–26].

Several studies have focused on improving individual optimization algorithms to obtain better convergence and solution quality. An improved Social Spider Optimization (ISSO) algorithm [1] modifies the original Social Spider Optimization (SSO) through several changes in the solution-generation procedures. Similarly, an Improved Chicken Swarm Optimization (ICSO) algorithm [6] has been reported to improve convergence speed, global search capability, and total fuel cost compared with PSO and CSO. The Chaotic Self-Adaptive Differential Harmony Search (CSADHS) algorithm [7] improves the convergence rate and computational time compared with the original Harmony Search Algorithm (DHS) and Self-Adaptive DHS (SADHS). The Improved Success-History-Based Adaptive Differential Evolution with Linear Population Size Reduction (IL-SHADE) algorithm [8] has been applied to different ELD systems, including the IEEE 6-unit, 40-unit, and 140-unit systems, and has demonstrated competitive performance against several existing techniques. A Chaotic Improved Honey Bee Mating Optimization (CIHBMO) algorithm [9] has also been developed as an efficient approach for ELD and reported improved optimal dispatch results compared with previous methods. In addition, a Modified Social Spider Algorithm (MSSA) [10] has been evaluated on 6-, 40-, 80-, and 140-unit benchmark systems, where the improved algorithm provides better performance than the original method.

Other studies have introduced specific search mechanisms or combinations of optimization operators to improve the performance of ELD algorithms. The Symbiotic Organisms Search algorithm with a novel Disruption Search (DSOS) [11] has been reported to obtain low generation costs with improved convergence speed and computational time. Bacterial Foraging Optimization (BFO) [12] has been applied to non-convex ELD problems involving dynamic operating constraints, transmission losses, and prohibited operating zones. The Social Optimization Algorithm (SOA) [13] is inspired by human social behavior and the concepts of social interaction and equality of opportunity. The Krill Herd Algorithm with modified crossover and mutation operators (KHA-IV) [14] introduces additional search operations to improve its optimization capability. A non-homogeneous member strategy for the Firefly Algorithm with random initialization and a non-penalty method (NhFA-Rnp) [15] has also been applied to a 15-unit ELD with transmission losses, generator limits, ramp-rate limits, and POZs, as well as to a 13-unit non-convex system with valve-point effects. The method was additionally evaluated using benchmark functions

and compared with conventional Firefly Algorithm variants using penalty and non-penalty power-balance constraint-handling strategies.

Hybridization of complementary algorithms has provided another important direction in ELD optimization. An efficient hybrid optimizer based on improved GA and PSO, namely MPSO-GA [16], has been evaluated on 6-, 10-, and 15-unit systems and applied to restricted economic dispatch problems with non-smooth cost functions. The Heat Transfer Search optimizer (HTS) [17] has been investigated on a large power system with valve-point effects, POZs, and operational limits. The One Rank Cuckoo Search Algorithm (ORCSA) [18] introduces two modifications to the conventional Cuckoo Search Algorithm, including ranking newly generated solutions and using a bound-by-best strategy for handling inequality constraints. The method has been tested on systems with different numbers of thermal units and constraints. A Multi-Strategy Ensemble Biogeography-Based Optimization (MSEBBO) [19] incorporates several modifications into BBO to obtain an appropriate balance between exploration and exploitation. Another Hybrid Gray Wolf Optimizer (HGWO) has also been investigated in [20].

The search behavior of population-based algorithms has also been modified to improve the balance between exploration and exploitation. A Modulated PSO (MPSO) [21] modifies particle velocity to improve exploration and exploitation for the economic emission dispatch problem. A Synergic Predator–Prey Optimization (SPPO) [22] improves prey-swarm quality and consequently affects convergence behavior on different power system test cases. The Improved Jaya algorithm with Self-Adaptive Multi-Population and Lévy Flights (Jaya-SML) [23] uses multiple populations to improve population diversity and applies a self-adaptive strategy to address the parameter-tuning problem. In the Improved Jaya (IJaya) algorithm [24], a distance-varying acceleration coefficient (DVAC) is used to accelerate convergence, while a self-adaptive population mechanism (SAPM) is employed to maintain population diversity. IJaya has been evaluated on 6-, 10-, 13-, 15-, and 40-generator ELD systems, with performance assessed in terms of convergence rate, computational efficiency, and robustness. The BBO method [25] has also been evaluated on four test systems with different sizes and complexity levels.

Further improvements have been achieved through modified search operators and hybrid population mechanisms. The modified Conglomerated Ion-Motion and Crisscross Search Optimizer (C-MIMO–CSO) [26] has been tested on 10-, 15-, 40-, and 140-unit systems and has demonstrated its applicability to both benchmark optimization problems and ELD. A Self-Organizing Hierarchical PSO (SOH-PSO) [27] addresses premature convergence by reinitializing the velocity vector when stagnation occurs and by balancing local and global exploration. An efficient neighborhood-search method incorporated into BBO (aBBOmDE) [28] combines a neighborhood search operator with a modified DE (mDE) population-based search operator to improve the overall fitness after a predefined threshold. A Hybrid Big Bang–Big Crunch algorithm (HBB-BC) [29] has been applied to highly nonlinear ELD problems, including 6-unit systems with losses, 15-unit systems with transmission losses, and 40-unit systems with valve-point loading effects. A Multi-Behavior Combination DE (MBC-DE) [30] develops a generic agent-based optimization framework in which different multi-behavior combination strategies are incorporated and evaluated on four test systems.

The applicability of optimization algorithms to increasingly large ELD systems has also received considerable attention. Teaching–Learning-Based Optimization (TLBO) [31] has been evaluated on 15-, 38-, 140-, and 160-unit systems under different restrictions. An Improved Orthogonal Design PSO (IODPSO-G) [32] has been applied to single-area and multi-area ELD problems using several examples involving 13-, 15-, 10-, 40-, 6-, and 10-unit systems with nonlinear generator characteristics. The Exchange Market Algorithm

(EMA) [33] has been tested on 3-, 6-, 15-, and 40-unit systems involving both convex and non-convex cost functions. A novel Chaotic TLBO with Lévy Flight (CTLBO) [34] divides the population into two groups, one evolving through teaching–learning and the other through Lévy flight, followed by chaos-based perturbation of a randomly selected population subset. A Modified Artificial Bee Colony algorithm (MABC) [35] incorporates chaotic local search (CLS) to improve the self-searching capability of ABC for identifying feasible optimum solutions to economic dispatch problems. An efficient hybrid PSO and Gravitational Search Algorithm (PSOGSA) [36] has been reported to obtain near-optimal solutions and improve the effectiveness of the conventional PSOGSA approach. The method has been evaluated using IEEE 5-machine 14-bus, IEEE 6-machine 30-bus, and 13- and 40-machine power systems.

Other optimization approaches have focused on improving convergence, robustness, constraint handling, and scalability. A powerful Crisscross Optimization Algorithm (COA) [37] has been validated using six test systems ranging from 10 to 64 generating units. The Ant Lion Optimization algorithm (ALO) [38] has been evaluated on a 10-unit system with complex constraints. A novel Multi-Population DE (MPDE) [39] uses multiple populations to address limitations of conventional DE. The C-GRASP–SaDE method [40] combines the Continuous Greedy Randomized Adaptive Search Procedure (C-GRASP) with self-adaptive DE and has been tested on 13-, 40-, and 140-generator systems considering valve-point loading effects. An Efficient Improved Moth Flame Optimizer (EMFO) [41] has been evaluated using benchmark functions and 6-, 40-, and 80-unit ELD systems with non-convex operating costs. The Orthogonal Learning Competitive Swarm Optimizer (OLCSO) [42] has been applied to 15-, 40-, and 110-generator systems. The Squirrel Search Algorithm (SSA) [43] has been investigated on 6-, 10-, 13-, 40-, 140-, and 160-unit systems to evaluate solution quality, convergence, and computational performance. A Novel Modified Adaptive Charged System Search algorithm (ACSS) [44] has been tested on ELD systems containing 6, 13, 15, 40, 160, and 640 units, covering small- to large-scale problems. The Lightning Flash Algorithm (LFA) [45] has been evaluated on 6-, 10-, 14-, 40-, and 100-unit systems, with reported advantages in convergence rate, uniformity, and robustness. A Granular Computing method (GrC) [46] has been examined on 13- and 40-unit systems while considering valve-point and transmission-loss effects.

More recently, adaptive and machine-learning-assisted optimization strategies have been investigated for ELD. A Forgetting Velocity-based Improved Comprehensive Learning PSO (FV-ICLPSO) [47–50] adjusts the learning probability according to the iteration number and the number of decision variables. It also adaptively selects learner exemplars for different particles and updates velocity to cover promising regions. The method has been evaluated on 10-, 13-, 40-, 80-, 120-, and 160-unit systems. A Surrogate-Assisted Adaptive Bat Algorithm (GARCBA) [51–54] uses a regression neural-network surrogate model to approximate the objective function and reduce execution time [55–57]. The approach has been tested on 19-, 57-, and 40-unit systems. An optimization algorithm and machine-learning combination (OA-ML) [58] has been applied to 6-, 15-, and 140-unit systems. A hybrid algorithm based on the β-Hill Climbing optimizer and Sine Cosine Algorithm (SCA-βHC) [59–62] has been used for 3-, 13-, 15-, and 40-unit ELD problems. An Evolutionary Simplex Adaptive Hooke–Jeeves algorithm (ESAHJ) [63–66] has been evaluated on five test cases involving 3-, 6-, 13-, 15-, and 40-unit systems and has demonstrated promising convergence characteristics and generation costs.

Beyond ELD-specific methods, optimization researchers have also investigated a broad range of algorithms for different engineering problems. Various optimization problems [67–71] have been addressed in [72–76] using different optimization algorithms and

techniques [77–81]. These studies have motivated the development of diverse optimization strategies [82–86] and solution mechanisms [87–90]. Also, other papers in [91–98].

Although these works do not necessarily focus directly on ELD, they demonstrate the broad development of optimization techniques and the continuing effort to improve search efficiency for complex optimization problems.

Several additional methods have also been applied specifically to ELD, including Hybrid PSO and Imperialist Competitive Algorithm (HPSO-ICA) [3], Sinusoidal Modulated PSO (SMPSO) [21], a modified Artificial Bee Colony method (ABC tend) [35], Adaptive Composite Differential Evolution (ACoDE) [11], modified Artificial Bee Colony with Cat Algorithm [99], Multi-Restart Dynamic Harris Hawk Optimization (DHHO-MR) [100], Slime Mould Algorithm (SMA) [101], Modified Hybrid PSO with Bat Algorithm Parameter-Inspired Acceleration Coefficients (MHPSO-BAAC) [102], Improved Slime Mould Algorithm (ISMA) [103], Hybrid Genetic–Artificial Fish Swarm Algorithm (HGAFSA) [104], PSO with chaotic sequences and crossover (CCPSO) [105], Colonial Competitive DE (CCDE) [106], Clustering Cuckoo Search Optimization (CCSO) [107], Coulomb's and Franklin's Laws Algorithm (CFA) [108], Ray Optimization (RO) [109], Cuckoo Search Algorithm (CSA) [110], Quantum-behaved PSO (QPSO-M1) [111], Variable DE with Fuzzy Adaptive PSO (FAPSO-VDE) [112], Bat Algorithm with penalty function (BA-penalty) [113], Gbest-focused ABC with Lévy flights (GFABC-LF) [114], Improved Random Drift PSO (IRDPSO) [115], Logic-Based Mixed-Integer Nonlinear Programming (LB-MINLP) [116], and Full Mixed-Integer Linear Programming (FMILP) [117].

Overall, the reviewed studies demonstrate that a wide variety of optimization techniques have been applied to ELD, including improved single algorithms, hybrid algorithms, adaptive strategies, Multi-Population approaches, neighborhood-search mechanisms, and machine-learning-assisted methods. These studies have also considered different practical characteristics, including valve-point effects, prohibited operating zones, ramp-rate limits, transmission losses, multiple fuel options, and different system sizes. Thus, the main challenge is no longer the absence of optimization algorithms for ELD. Instead, an important research issue is how to design a search mechanism that can maintain sufficient population diversity while also moving efficiently toward promising solutions.

This issue is particularly important for non-convex ELD problems. Valve-point effects and other nonlinear characteristics can create discontinuities and multiple local optima, while practical operating constraints can further restrict the feasible search region. An exploration-oriented mechanism can preserve diversity and investigate different regions but may converge slowly. Conversely, a strongly guided search can accelerate convergence toward promising regions but may reduce diversity and increase the risk of premature convergence. Therefore, an effective ELD optimizer should provide a suitable interaction between diverse search movements and guided exploitation rather than relying excessively on only one search behavior.

The reviewed studies also indicate that hybridization can be useful when two algorithms provide complementary search characteristics. However, simply combining two algorithms does not necessarily guarantee an effective exploration–exploitation balance. The effectiveness of a hybrid method depends on how the search behaviors are integrated, when each mechanism is applied, and how the information generated by one mechanism influences the other. If a search mechanism relies excessively on local or best-solution information, convergence to a local optimum may occur. Conversely, if the search remains strongly exploration-oriented, convergence may become slow and the optimum may not be reached within the available computational budget.

From this perspective, DE and GWO provide two different but complementary search behaviors. DE generates mutation vectors from differences between population members,

providing diverse population-based search directions. GWO, in contrast, uses information from leading solutions to guide the population toward promising regions. These characteristics provide a suitable basis for investigating whether the GWO hunting mechanism can be incorporated into the DE search process in a more focused manner.

Accordingly, this study does not claim that the general combination of DE and GWO is novel. Instead, the research focuses on a specific mutation-level integration in which the GWO hunting mechanism is introduced as an additional mutation vector within the DE framework. The conventional DE mutation remains available, while the hunting mutation provides an additional directed search movement. A probabilistic parameter, *Hm*, determines whether the hunting mutation or the predefined DE mutation is applied to each population member. Therefore, HDE does not execute DE and GWO as two independent or sequential optimization algorithms; rather, the two search mechanisms operate within the same DE evolutionary framework.

This design is intended to provide a flexible interaction between exploration and exploitation. The DE mutation can generate diverse movements based on population differences, whereas the GWO hunting mechanism can exploit information from high-quality solutions. The probabilistic application of the hunting mutation also allows different population members to use different mutation mechanisms within the same iteration. This structure is particularly relevant to non-convex ELD because the search may need both diversity for escaping local optima and directed movement for improving promising solutions.

The effectiveness of this specific search structure is investigated on different ELD systems and operating conditions. The purpose is not to establish a general superiority of DE-GWO hybridization, but to evaluate whether integrating the GWO hunting mechanism at the mutation level and controlling its application through *Hm* can provide a useful search behavior for non-convex ELD problems.

Based on the above discussion, the main contributions of this paper are summarized as follows:

A hybrid DE-based optimization framework, called HDE, is proposed for non-convex ELD problems. The GWO hunting mechanism is integrated into the DE mutation stage to provide an additional guided search direction while retaining the conventional DE crossover and selection processes.

Eight DE mutation strategies are investigated within the HDE framework. Each strategy can be combined with the GWO-based hunting mutation and the probabilistic *Hm* mechanism. This provides different search behaviors and allows the performance of the HDE variants to be evaluated on ELD problems.

A probabilistic parameter (Hm) is introduced to control the use of the GWO-based hunting mutation for individual population members. This mechanism allows conventional DE mutation and guided hunting mutation to be used within the same iteration and provides a flexible balance between exploration and exploitation.

The proposed method is evaluated on different ELD systems and operating conditions. The experimental results are used to examine the effectiveness of the proposed search structure under different problem dimensions and constraints.

This article is organized as follows. Section 2 formulates the ELD problem, including the objective functions and practical constraints. Section 3 introduces DE, GWO, and the proposed HDE algorithm. Section 4 presents the simulation results and discussion. Finally, the main conclusions are given in Section 5.

2. Problem Formulation

2.1. Objective Function

The main objective is to reduce total production costs by maximizing resource utilization. The ELD function measures the load cost for power plants to save money. Thus, this problem has the objective function below [105,108].

$$\text{Min } J = \sum_{i=1}^{N} OF_i(G_i) = \sum_{i=1}^{N} (a_i G_i^2 + b_i G_i + c_i) \tag{1}$$

2.1.1. The Impact of Steam Valves on the Fuel Cost of System Units

This effect is considered by adding a sine-like component in the objective function of electrical generation units as shown in the formulation below [105,108]:

$$\text{Min}J = \sum_{i=1}^{N} OF_i(G_i) = \sum_{i=1}^{N} \left[a_i G_i^2 + b_i G_i + c_i + | e_i \sin\left(f_i \left[G_{i,\min} - G_i \right]\right) | \right] \tag{2}$$

2.1.2. Regarding the Impact of Steam Valves and Multiple-Fuel

When the electrical power generators work with different fuels, each generator's cost character can be defined through the sum of quadratic kinships. Each one accompanies a special kind of fuel formulated as given in (3) [106].

$$OF_i(G_i) = \begin{cases} a_{i1} G_i^2 + b_{i1} G_i + c_{i1} + | e_{i1} \sin\left(f_{i1} \left[G_{i,\min} - G_i \right]\right) |, & G_{i,\min} \le G_i \le G_{i1} \\ a_{i2} G_i^2 + b_{i2} G_i + c_{i2} + | e_{i2} \sin\left(f_{i2} [G_{i2} - G_i]\right) |, & G_{i2} \le G_i \le G_{i3} \\ \vdots & \\ a_{it} G_i^2 + b_{it} G_i + c_{it} + | e_{it} \sin\left(f_{it} \left[G_{i,t-1} - G_i \right]\right) |, & G_{i,t-1} \le G_i \le G_{i,\max}. \end{cases} \tag{3}$$

2.2. Problem Constraints

2.2.1. Inequality Constraint of Generating Power

The generation power limit in each system unit cannot exceed its lower and upper limits. This capacity of units is modeled through the following inequality [106]:

$$G_{i,\min} \le G_i \le G_{i,\max} \tag{4}$$

2.2.2. Balance of Total Power Generation

To keep a balance between the total real power generation and the total real power consumption in the network, the sum of electrical power produced must equal the sum of electrical power consumed and the sum of electrical power lost by the network [106]:

$$\sum_{i=1}^{N} G_i = P_D + P_{Losses} \tag{5}$$

The electrical power losses can be calculated by using the network electrical power losses matrix as [106]:

$$P_{Losses} = \sum_{i=1}^{N} \sum_{h=1}^{N} G_i B_{ih} G_h + \sum_{h=1}^{N} B_{0h} + B_{00} \tag{6}$$

2.2.3. Prohibited Operating Zones

Prohibited operating zones (POZs) represent operating ranges in which a generating unit cannot operate because of physical limitations or operational restrictions. For each generating unit, the feasible generation range may therefore contain one or more prohibited

intervals. If the output power of a unit falls inside any prohibited interval, the corresponding solution is considered infeasible with respect to the POZ constraint.

In this study, POZ constraints are considered during the evaluation of each candidate solution. After generating a new solution, the output power of each unit is checked against its specified prohibited zones. If the output of any unit falls within a prohibited interval, a penalty value is added to the objective function. Therefore, solutions that violate POZ constraints receive a higher objective value and are less likely to be selected during the DE selection process. This procedure allows the HDE algorithm to handle POZ constraints while maintaining the original DE-based evolutionary process.

These zones where the generation unit cannot produce real power are assigned prohibited areas. These limits can be joined as below [106]:

$$\begin{cases} G_{i,\min} \le G_i \le G_{i1}^L \\ \dots \\ G_{ir-1}^U \le G_i \le G_{ir}^L;\ \forall\ r = 1,\ 2,\ \dots,\quad n_i \\ \dots \\ G_{in_i}^U \le G_i \le G_{i,max} \end{cases} \tag{7}$$

2.2.4. Ramp-Rate Limit

The physical generation limit of the network units is constrained by their generation ramp rates. As a matter of fact, due to this limit, the maximum and minimum output power ranges of a network unit are dependent on the generator's initial generation at any point in time. Considering this constraint, the limit in (4) is remodeled as follows [105]:

$$max\left(P_{i,min}, G_i^0 - DR_i\right) \le G_i \le min\left(G_{i,max}, G_i^0 + UR_i\right) \tag{8}$$

3. HDE Algorithm

3.1. Differential Evolution

In the DE [118,119] method, an initial population is created like other evolutionary algorithms. Afterward, a new population is formed by applying several operators to the original population, such as composition, mutation, and crossover. Selective comparisons are then made to see how well the new population measures up in terms of objective function merits. Only those who have excelled in previous stages get promoted to the next generation. This cycle is repeated indefinitely to get the desired outcome. This section details the algorithm's several stages of execution.

3.1.1. Population Generation

This approach initially produces a population of *Npop* values (user-specified number) with the required problem dimension *D* randomly inside the problem range. The values' bounds are set at this stage. Then, the value of each member's objective function is calculated by observing all the problem situations.

$$\begin{aligned} &z_{Iter,d}^i = z_{min,d} + r_d \times \left(z_{max,d} - z_{min,d}\right); \\ &Z_{iter}^i = [z_{Iter,1}^i, z_{Iter,2}^i, \dots, z_{Iter,D}^i] \\ &d = 1 : D. \end{aligned} \tag{9}$$

The indices *Iter* and *i* are characterized as *Iter* = 1, 2, ..., *Itermax* and *i* = 1, 2, ..., *Npop*.

3.1.2. Mutation

To create the mutant vector, certain members are chosen randomly. For instance, $Z_{Iter}^{r_1}$, $Z_{Iter}^{r_2}$, $Z_{Iter}^{r_3}$, $Z_{Iter}^{r_4}$. Then, the mutant vector for the members is generated as follows:

$$\begin{aligned} V_{Iter}^{i} &= Z_{Iter}^{r_1} + F\left(Z_{Iter}^{r_2} - Z_{Iter}^{r_3}\right) \\ V_{iter}^{i} &= [v_{Iter,1}^{i}, v_{Iter,2}^{i}, \ldots, v_{Iter,D}^{i}] \end{aligned} \tag{10}$$

This is the first mutation vector proposed. This coefficient influences the jump step length. The smaller the coefficient, the shorter the distance necessary to find the new member, increasing the algorithm's execution time. This can cage the algorithm in local optimizations. However, for high values of this coefficient, the required environment to discover a new member is sought at wider distances, which may be less precise. Before executing the final optimization, the scale coefficient must be evaluated because adjusting this parameter offers varied values based on the optimization problem and the environment.

3.1.3. Crossover

Each variable is given a random value between 0 and 1, and the crossover operation is done for the *i*-th member of all the members in iteration *Iter*. This creates the trial vector by setting the crossover constant between 0 and 1. Each component of the mutant vector is transferred to the candidate vector with a *CR* probability, or the original vector is replaced. This operator's equation is shown below:

$$x_{Iter,j}^{i} = \begin{cases} v_{Iter,j}^{i}; \text{ if } r_j \leq CR \\ z_{Iter,j}^{i}; \text{ otherwise} \end{cases} \tag{11}$$

$$X_{iter}^{i} = \left[x_{Iter,1}^{i}, x_{Iter,2}^{i}, \ldots, x_{Iter,D}^{i}\right]$$

As a result, the experimental vector receives a component from the altered vector at random, independent of the *CR*. A member is chosen randomly for each potential vector using the rand function.

3.1.4. Selection

If the measurement vector $f\left(X_{Iter}^{i}\right)$ is more valuable than the target member $f\left(Z_{Iter}^{i}\right)$, it will be moved to the next generation. If not, the target member will be added to the next generation's population. The measurement vector and target member are displayed below:

$$Z_{Iter+1}^{i} = \begin{cases} X_{Iter}^{i}; \text{ if } f\left(X_{Iter}^{i}\right) \leq f\left(Z_{Iter}^{i}\right) \\ Z_{Iter}^{i}; \text{ otherwise.} \end{cases} \tag{12}$$

Finally, this trend will continue until the new generation achieves the number of Npop. To get the final conditions, this process is repeated in the same way.

The DE comes in a variety of forms, as outlined below. It is up to the application to decide which version is best. There is a breakdown of the many functions [106] that these versions can optimize. There is a DE symbol like DE/x/y.

It is possible to select *X* randomly or based on merit. *Z* is a mutant vector. This symbol has two possible meanings: rand and best. Whenever you make a jump, the best possible member of the population is always used. A random vector or member of the population is chosen for each leap when using the rand symbol. This is the number of different vectors that were employed. Here are some of the more popular and well-known types [106]:

"DE/rand/1":

$$V_{Iter}^{i} = Z_{Iter}^{r_1} + F\left(Z_{Iter}^{r_2} - Z_{Iter}^{r_3}\right) \tag{13}$$

"DE/rand/2":

$$V_{Iter}^{i} = Z_{Iter}^{r_1} + F\left(Z_{Iter}^{r_2} - Z_{Iter}^{r_3}\right) + F\left(Z_{Iter}^{r_4} - Z_{Iter}^{r_5}\right) \tag{14}$$

"DE/best/1":

$$V_{Iter}^{i} = Z_{best} + F\left(Z_{Iter}^{r_1} - Z_{Iter}^{r_2}\right) \tag{15}$$

"DE/best/2":

$$V_{Iter}^{i} = Z_{best} + F\left(Z_{Iter}^{r_1} - Z_{Iter}^{r_2}\right) + F\left(Z_{Iter}^{r_3} - Z_{Iter}^{r_4}\right) \tag{16}$$

"DE/rand-to-best/1":

$$V_{Iter}^{i} = Z_{Iter}^{r_1} + F\left(Z_{best} - Z_{Iter}^{r_1}\right) + F\left(Z_{Iter}^{r_2} - Z_{Iter}^{r_3}\right) \tag{17}$$

"DE/rand-to-best/2":

$$V_{Iter}^{i} = Z_{Iter}^{r_1} + F\left(Z_{best} - Z_{Iter}^{r_1}\right) + F\left(Z_{Iter}^{r_2} - Z_{Iter}^{r_3}\right) + F\left(Z_{Iter}^{r_4} - Z_{Iter}^{r_5}\right) \tag{18}$$

"DE/current-to-best/1":

$$V_{Iter}^{i} = Z_{Iter}^{i} + F\left(Z_{best} - X_{Iter}^{i}\right) + F\left(Z_{Iter}^{r_1} - Z_{Iter}^{r_2}\right) \tag{19}$$

"DE/current-to-rand/1":

$$V_{Iter}^{i} = Z_{Iter}^{i} + r\left(Z_{Iter}^{i} - Z_{Iter}^{r_1}\right) + F * r\left(Z_{Iter}^{r_2} - Z_{Iter}^{r_3}\right) \tag{20}$$

where, z_{best} is the member with the best fitness evolution in the swarm at iteration *Iter*.

In this study, the eight DE mutation strategies are treated as separate HDE variants. The mutation strategy is fixed before each experiment and remains unchanged during the optimization process. The GWO-based hunting mutation is then selected probabilistically according to (Hm), while the selected DE mutation is used otherwise. Therefore, the proposed method does not perform automatic selection among the eight DE mutation strategies.

3.2. Gray Wolf Optimizer (GWO) [120]

GWO's optimal approach based on the objective function is α to simulate wolf social information mathematically. Therefore, the second and third fittest positions are named β and δ wolves. The remaining candidates (ω) are all guided by α, β, and δ to participate in a hunting manner.

Prey-finding, the surround-and-containment ability of a gray wolf, is legendary. The alpha usually leads the hunt. Beta and delta neurons may also be searched. There is no indication that the perfect prey is in an abstract position zone. To model gray wolf hunting behavior computationally, it was assumed the alpha, beta, and delta contain extra prey position data. So, we keep the first three best solutions and force the rest (including the omegas) to adjust their locations to match the best search agent placements. Here are some possible equations:

$$\begin{cases} D_{\alpha} = \left|\vec{C}_1 \cdot Z_{Iter}^{\alpha} - Z_{Iter}^{i}\right| \\ D_{\beta} = \left|\vec{C}_2 \cdot Z_{Iter}^{\beta} - Z_{Iter}^{i}\right| \\ D_{\delta} = \left|\vec{C}_3 \cdot Z_{Iter}^{\delta} - Z_{Iter}^{i}\right| \end{cases} \tag{21}$$

$$\begin{cases} \vec{Z}_1 = Z_{Iter}^{\alpha} - \vec{A}_1 \cdot \left(\vec{D}_{\alpha}\right) \\ \vec{Z}_2 = Z_{Iter}^{\beta} - \vec{A}_2 \cdot \left(\vec{D}_{\beta}\right) \\ \vec{Z}_3 = Z_{Iter}^{\delta} - \vec{A}_3 \cdot \left(\vec{D}_{\delta}\right) \end{cases} \tag{22}$$

$$z_{Iter+1}^{i} = \left(\vec{Z}_1 + \vec{Z}_2 + \vec{Z}_3\right) \cdot /3 \tag{23}$$

The parameters $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A}_i = 2\vec{a}\cdot\vec{r}_1 - \vec{a}; i = \{1, 2, 3\} \quad (24)$$

$$\vec{C}_i = 2\cdot\vec{r}_2; i = \{1, 2, 3\} \quad (25)$$

The components of $\vec{a}$ are decreased from 2 to 0 via the next generations, and $\vec{r}_1$, $\vec{r}_2$ are vectors in $[0, 1]$.

The final location of each wolf prey can be at random within a search space circle indicated by the alpha, beta, and delta positions. In other words, the alpha, beta, and delta wolves make an educated guess about where the prey is, while the other wolves in the pack update their positions randomly. For most, gray wolves are looking for alpha, beta, and delta positions, and they hunt for prey apart from each other and attack creatures close to each other.

3.3. The Proposed Algorithm (HDE)

A recent study [121] found that the GWO's performance for shifted functions, which are models of real-world problems, is significantly reduced. In the original work [120], Mirjalili et al. used only basic optimization functions with zero optimal solutions. Akbari et al. [121] have implemented G-NHGWO in which each wolf has the current and best-stored locations, like the PSO algorithm, which are updated in every iteration. The authors' claims are supported by the results of the 14 shift functions of 2005 that were used in their work. In this article, the PSO-based GWO model is proposed, in which each wolf has two positions: the current (Z^i_{Iter}) and a position of the best stored ($Z^{i,best}_{Iter}$).

All three wolves (α, β, and δ) in the GWO algorithm always engage in the hunting step. Improved exploration and faster convergence of the GWO algorithm can be achieved by using the best wolves in the hunting phase [121]. In this research, the hunting phase of GWO is presented as a supplementary mutation vector to the DE to speed up the convergence rate and find more optimal solutions to the problem. As a result of this suggested HDE algorithm, each iteration uses a hunting vector with a probability of H for each member, which is employed as the mutation vector. Afterward, the crossover and selection are employed.

In the suggested HDE optimizer, in place of using $Z^{r_1,best}_{Iter}$, $Z^{r_2,best}_{Iter}$, and $Z^{r_3,best}_{Iter}$, three vectors of Z^{α}_{Iter}, Z^{β}_{Iter}, and Z^{δ}_{Iter} are employed and the Z^{i}_{Iter} is replaced by the best-encountered position by gray wolf i so that the novel vector of the target individual is modeled via Equations (26)–(28) [121]:

$$\begin{cases} \vec{D}_1 = \left|\vec{C}_1\cdot z^{\alpha}_{Iter} - Z^{i,best}_{Iter}\right| \\ \vec{D}_2 = \left|\vec{C}_2\cdot z^{\beta}_{Iter} - Z^{i,best}_{Iter}\right| \\ \vec{D}_3 = \left|\vec{C}_3\cdot z^{\delta}_{Iter} - Z^{i,best}_{Iter}\right| \end{cases} \quad (26)$$

$$\begin{cases} \vec{Z}_1 = Z^{\alpha}_{Iter} - \vec{A}_1\cdot\left(\vec{D}_1\right) \\ \vec{Z}_2 = Z^{\beta}_{Iter} - \vec{A}_2\cdot\left(\vec{D}_2\right) \\ \vec{Z}_3 = Z^{\delta}_{Iter} - \vec{A}_3\cdot\left(\vec{D}_3\right) \end{cases} \quad (27)$$

$$Z^{i}_{Iter+1} = \left(\vec{Z}_1 + \vec{Z}_2 + \vec{Z}_3\right)\cdot/3 \quad (28)$$

Then, z^{i}_{Iter+1} must be compared with $z^{i,best}_{Iter}$ and if it has a lower fitness value, it will be applied as the novel best solution.

The suggested HDE optimizers can be briefly illustrated as follows:

Step 1: Generate the initial members of the Gray Wolf's Z^i_0 ($i = 1, 2, \ldots, Npop$) (Equation (9)) and calculate their objective function values.

Step 2: Assign an initial value to parameter "a", then calculate the parameters A and C (Equations (24) and (25)).

Step 3: Evaluate the Gray Wolf's positions Z^{α}_{Iter}, Z^{β}_{Iter}, Z^{δ}_{Iter}, i.e., the three best solutions.

Step 4: Perform hunting mutation through Equations (26)–(28) with the probability value *Hm* or basic mutation with the probability value 1–*Hm*. Perform the hunting mutation with probability (Hm), or apply the selected DE mutation strategy with probability $(1 - Hm)$. The DE mutation strategy is predefined for each HDE variant and is not automatically selected during the optimization process. For example, HDE/rand-to-best/2 uses Equation (18). For example, HDE/rand-to-best/2 uses Equation (18).

Step 5: Apply the crossover step (Equation (11)).

Step 6: Compute the objective function of the new positions.

Step 7: Apply the selection step (Equation (12)).

Step 8: Do-over steps 2 to 7 to *Itermax*.

The novelty is not the general combination of DE and GWO, but the specific integration of the GWO hunting mechanism into the DE mutation framework and its probabilistic application at the individual level.

The eight DE mutation strategies are therefore used to construct different HDE variants for comparative evaluation. For a given HDE variant, the selected DE mutation strategy remains fixed, while *Hm* controls the probabilistic choice between that DE mutation and the GWO-based hunting mutation.

The main motivation of HDE is to provide two complementary search behaviors within a single evolutionary framework. DE generates search directions from differences between population members, which supports population diversity and exploration of different regions of the search space. However, for complex non-convex ELD problems, exploration alone may require a large number of iterations to move the population toward promising regions.

In contrast, the GWO hunting mechanism uses information from the leading solutions to generate a more directed movement toward promising areas. This can strengthen exploitation during the search. However, excessive dependence on the leading solutions may reduce population diversity and increase the possibility of premature convergence, particularly for non-convex and non-smooth optimization problems.

HDE addresses these complementary behaviors by incorporating the hunting operation as a supplementary mutation mechanism rather than as a separate optimization process. Thus, each population member can use either the conventional DE mutation or the GWO-based hunting mutation during the same iteration. The selected mutation vector is subsequently processed through the standard DE crossover and selection operators.

This structure allows DE mutation to provide diverse search movements while the hunting mutation provides a more directed search toward high-quality regions. The probabilistic use of the two mechanisms also prevents all population members from being forced to follow the same search behavior at every iteration. This is particularly relevant to ELD problems, where non-convex and non-smooth cost functions and operational constraints can produce multiple local optima and restricted feasible regions.

The novelty of HDE is therefore not the general combination of DE and GWO. Instead, it lies in the integration of the GWO hunting mechanism into the DE mutation framework and its probabilistic application at the individual level, while retaining the conventional DE crossover and selection operations.

The parameter *Hm* controls the probability of selecting the GWO-based hunting mutation. Thus, $Hm = 0$ means that only the conventional DE mutation is used, whereas $Hm = 1$ means that only the hunting mutation is used. For $(0 < Hm < 1)$, the two mutation mechanisms are probabilistically combined. In this study, *Hm* is treated as a fixed control parameter during each optimization run, and its suitable value is determined through the sensitivity analysis presented in Section 3.4.

3.4. Sensitivity Analysis of the Hunting Probability

The sensitivity analysis investigates the effect of the hunting probability Hm on the performance of HDE. Different fixed values of Hm are evaluated to determine a suitable balance between the conventional DE mutation and the GWO-based hunting mutation.

For this analysis, the tested Hm values cover the range from a DE-dominated search to a hunting-dominated search. The other algorithmic parameters are kept unchanged, including the population size, maximum number of iterations, crossover settings, constraint-handling procedure, and stopping condition. This provides a consistent basis for evaluating the effect of Hm alone.

The results are compared using the objective function value, convergence behavior, and computational cost. The case $Hm = 0$ represents the conventional DE-based search without the GWO hunting mechanism, whereas $Hm = 1$ represents a search based entirely on the hunting mutation. Intermediate values allow both mutation mechanisms to operate within the same population and iteration.

Increasing Hm increases the frequency with which the hunting mutation is applied and therefore gives greater emphasis to directed exploitation. Conversely, smaller values of Hm preserve a larger contribution from the difference-based DE mutation, which provides diverse search directions.

The sensitivity analysis also examines whether excessive use of the hunting mechanism affects search stability. If $Hm = 1$ produces less stable behavior than intermediate values, this may indicate that relying exclusively on the leading solutions can reduce search diversity. Similarly, if $Hm = 0$ results in slower convergence, the additional hunting direction may provide useful exploitation of promising regions.

Based on the obtained experimental results, an appropriate fixed value of Hm is selected for the main experiments. This value is then kept unchanged during the corresponding optimization runs. The eight DE mutation strategies are evaluated separately by constructing different HDE variants; Hm does not select among these eight strategies. Instead, within each HDE variant, it controls the probabilistic use of the predefined DE mutation and the GWO-based hunting mutation.

3.5. Computational Complexity Analysis

The computational complexity of HDE mainly depends on the population size $(Npop)$, problem dimension (D), and maximum number of iterations (T). For each population member, the DE mutation, GWO-based hunting mutation, crossover, and selection operations require $(\mathrm{O}(D))$ operations. Therefore, for $(Npop)$ population members and (T) iterations, the main computational complexity of HDE is $O(Npop \times D \times T)$.

The proposed hunting mechanism does not change this order because its calculations are performed over the (D) decision variables for each population member. Thus, the integration of the hunting mechanism into DE increases the operations per iteration but does not change the asymptotic complexity of the main evolutionary process.

For ELD problems with transmission losses, the objective-function evaluation includes an additional computation based on the loss matrix. Therefore, the total computational cost also depends on the formulation of the ELD objective function and its constraints. The $(O(Npop \times D \times T))$ term represents the main complexity of the HDE search process.

In the experiments, the population size and the number of iterations were increased for larger ELD systems. For example, the 110-unit and 140-unit systems were solved using $(Npop = 120)$ and $(T = 5000)$. This shows that the proposed method can be applied to higher-dimensional ELD problems while maintaining the same asymptotic complexity structure.

3.6. Essential Differences Between HDE and Previous Hybrid Strategies

Previous hybrid optimization methods have combined operators from different algorithms to improve exploration or exploitation. Some hybrid GWO methods, for example, incorporate evolutionary operators such as mutation and crossover into the GWO framework. HDE follows a different integration mechanism in which DE remains the main evolutionary framework.

In HDE, the GWO hunting mechanism is integrated directly into the DE mutation stage rather than being executed as a separate optimization phase. For each population member, the hunting mutation is selected with probability Hm, whereas the predefined DE mutation is selected with probability $1 - Hm$. The selected mutation is then followed by the conventional DE crossover and selection operations.

Another important characteristic is the information used to construct the hunting vector. HDE employs the three best current solutions, α, β, and δ, together with the best-encountered position of the current individual. Consequently, the hunting operation combines population-level information from the leading solutions with individual historical information.

Therefore, HDE does not execute DE and GWO independently or sequentially. Its main search structure can be expressed as

DE mutation or GWO hunting mutation → DE crossover → DE selection

The main differences between HDE and a conventional hybrid strategy can be summarized as follows:

- Integration level: The GWO hunting mechanism is integrated into the DE mutation stage.
- Selection mechanism: The hunting mutation and the predefined DE mutation are selected probabilistically for each individual using Hm.
- Information used in hunting: HDE uses α, β, δ, and the best-encountered position of the individual.
- Evolutionary framework: DE crossover and selection are retained after the selected mutation operation.
- Search behavior: DE provides difference-based search directions, whereas GWO hunting provides directed movement toward promising solutions.

Accordingly, the contribution of HDE is the specific mutation-level integration of the GWO hunting mechanism within DE, together with its individual-level probabilistic application, rather than the general combination of DE and GWO as two independent optimizers.

From the perspective of search behavior, the rationale for using the two mutation mechanisms together is based on their different ways of generating search directions. DE mutation derives new directions from differences among population members, so the generated movement is influenced by the current population distribution. This property can provide diverse search directions and help the population explore different regions of the search space, particularly when the current solutions are distributed over relatively different areas.

In contrast, the GWO hunting mechanism uses information from the leading solutions, specifically α, β, and δ, to generate a more directed movement toward promising regions. Such guidance can strengthen exploitation of high-quality areas. However, if the search relies excessively on these leading solutions, population diversity may decrease and the search may become more concentrated around a limited region. This can be problematic for non-convex ELD problems, where valve-point effects, prohibited operating zones, and other nonlinear characteristics may produce multiple local optima and restricted feasible regions.

The HDE structure is designed to address these different search requirements without treating DE and GWO as two independent optimization processes. The probabilistic use of the two mutation mechanisms allows different population members to follow different search directions within the same iteration. Consequently, some individuals can generate

movements according to population differences, while others can use the information provided by the leading solutions. This provides a mechanism through which exploratory and more directed movements can coexist during the optimization process.

The theoretical basis of this design is therefore that the two mutation behaviors can provide complementary search information. The effectiveness of HDE is expected to arise from allowing these behaviors to coexist within the DE evolutionary framework, rather than from the simple combination of two optimization algorithms. This structure is particularly suitable for complex ELD problems because it provides both diverse search movements and guided movements toward promising regions while retaining the standard DE crossover and selection procedures.

3.7. Parameter Setting and Reproducibility

The main parameters of HDE are *Npop*, *Itermax*, *Hm*, and *CR*. Their settings are selected according to the size and complexity of the ELD problem. Larger population sizes or higher numbers of iterations can be used for larger and more complex systems to provide sufficient search capability, whereas smaller values can be used for smaller systems to reduce computational cost.

The parameter *Hm* controls the relative use of the GWO-based hunting mutation and the predefined DE mutation, while *CR* controls the contribution of the generated mutation vector during the DE crossover operation.

In this study, *CR* and *Hm* are selected from the ranges investigated in the parameter sensitivity analysis. The final values used for each ELD case are explicitly reported in the corresponding experimental tables. No automatic parameter-tuning method is used.

For reproducibility, all experiments use the parameter settings reported in the manuscript, the specified population size and maximum number of iterations, and 30 independent runs unless otherwise stated. The same stopping criterion and constraint-handling procedure are applied to the HDE experiments.

For a new ELD problem, the population size and maximum number of iterations can be increased when the number of decision variables or the complexity of the constraints increases. The *Hm* and *CR* values can initially be selected from the ranges evaluated in the sensitivity analysis and then fixed for the independent runs. This provides a practical starting point without requiring extensive trial-and-error parameter tuning.

4. Results and Discussion

In this article, four ELD problems are discussed in different cases: (1) a 10-unit system with a demand of 2700 MW with multiple-fuel (MF) constraint with/without VPE loading; (2) a 15-unit system with a demand of 2630 MW considering POZs with/without ramp-rate limits, (3) a 40-unit system with the demand of 10,500 MW considering the VPE; and (4) a 140-unit system with the demand of 49,342 MW with/without ramp-rate constraint. In this scenario, various evolutionary algorithms from different articles are selected compared to HDE algorithms. The set of control parameters of the algorithms is shown in Table 1; however, it should be noted that the value of the control parameter *F* is not mentioned in the table since it is the same for all algorithms and is equal to $(0.1 + 0.8 \times \text{rand})$. The populations were chosen as 60, 60, 70, and 120 for the test networks, respectively. To get the best (Min), average (Mean), and worst (Max) fuel cost indices as well as standard deviation (Std.) for each under-view ELD problem, each HDE algorithm did 30 separate runs for each problem. The experiments were done on Windows 10 with the following configuration: CPU Core i7 2.2 GHz with 16 GB RAM, MATLAB 2018a. The MATLAB Code used to implement the proposed method and obtain the results reported in this study was submitted as part of the Supplementary Materials. No separate dataset was generated or used in this study.

Table 1. Selected parameters for the DE methods in this article.

Algorithm	*Hm*	*CR*
HDE/rand-to-best/2	0.5	0.95
DE/rand-to-best/2	-	0.95
HDE/current-to-best/1	0.9	0.9
DE/current-to-best/1	-	0.9
HDE/rand-to-best/1	0.9	0.9
DE/rand-to-best/1	-	0.9
HDE/current-to-rand/1	0.5	0.9
DE/current-to-rand/1	-	0.9
HDE/best/2	0.1	0.9
DE/best/2	-	0.9
HDE/best/1	0.9	0.9
DE/best/1	-	0.9
HDE/rand/2	0.1	0.9
DE/rand/2	-	0.9
HDE/rand/1	0.1	0.9
DE/rand/1	-	0.9

4.1. The 6-Unit Network

The 6-unit network is examined in two cases with $P_D = 1263$ MW. The ramp rate, the prohibited operating zones, and the network losses are considered in case 1, while the ramp rate is only considered in case 2. The objective function in these two cases is Equation (1). Comparison studies of these two cases are depicted in Tables 2 and 3. The decision variables of these two cases are also defined in Table 4. Figure 1 shows the flowchart for implementing the HDE on ELD. This figure clearly describes this workflow, step by step. The convergence characteristics of the best solution for the 6-unit network in cases 1 and 2 are plotted in Figures 2 and 3, respectively.

Table 2. Optimal results in ($/h) achieved by optimization methods for case 1 in the 6-unit network.

Algorithm (*Npop*/*Itermax*)	Max	Mean	Min	Std.
HDE/current-to-best/1 (40/400)	15,442.6605	15,442.6599	15,442.6595	3.25×10^{-4}
DSOS [32] (30/200)	-	-	15,449.694	-
ACoDE [32] (30/200)	-	-	15,450.04	-
MABC/D/Cat [99] (-/-)	15,449.8995	15,449.8995	15,449.8995	6.04×10^{-8}

Table 3. Optimal results in ($/h) achieved by optimization methods for case 2 in the 6-unit network.

Algorithm (*Npop*/*Itermax*)	Max	Mean	Min	Std.
HDE/current-to-best/1 (30/400)	15,275.9284	15,275.9284	15,275.9284	1.06×10^{-5}
DHHO-MR [100] (50/3000)	15,275.933	-	-	-
SMA [101] (50/500)	15,275.9304	-	-	-
MHPSO-BAAC [55] (-/-)	15,276.7790	15,279.343	-	-

Table 4. Optimal results achieved by HDE/current to-best/1 for the 6-unit network for two cases (in MW).

Units			Case 1				Cost ($/h)
G_1~G_6	447.07148	173.17288	263.92463	139.04829	165.57312	86.62553	15,442.6595
			Case 2				
G_1~G_6	446.6889	171.254	264.1159	125.20185	172.14448	83.59472	15,275.9284

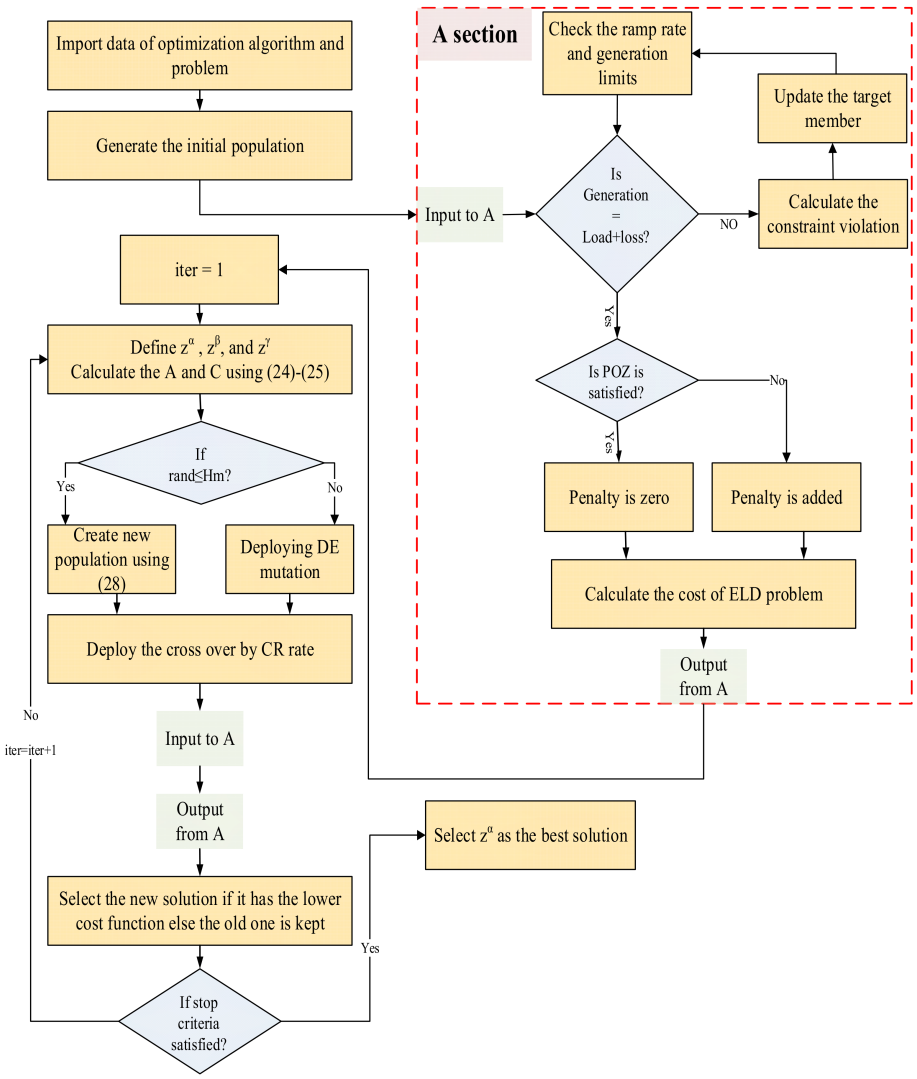


Figure 1. The flowchart for implementing the HDE on ELD [52].

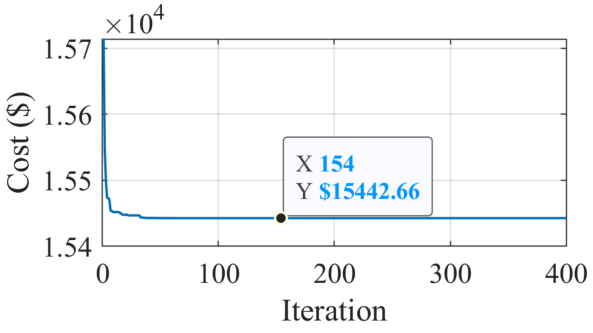


Figure 2. The convergence curve to the best solution for 6-unit system, case 1.

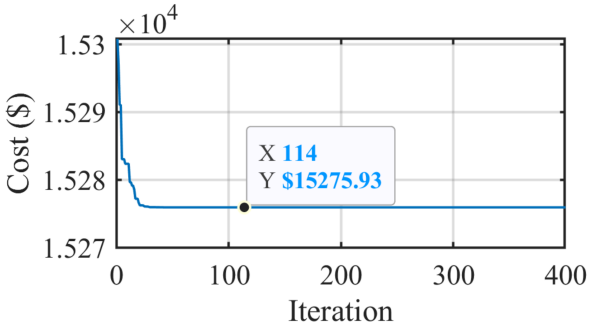


Figure 3. The convergence curve to the best solution for 6-unit system, case 2.

4.2. The 10-Unit Network

Ten thermal power units with P_D = 2700 MW and no system losses make up this network [106]. Case 1 only analyses the MF restriction, while case 2 considers the MF constraint and the VPE. The objective function in these two cases is Equation (3), while in case 1, the parameters $e_{i1} = e_{i2} = \ldots = e_{it} = 0$.

Tables 5 and 6 show the outcomes of the HDEs after 30 independent runs and 500 iterations for any HDE and two 10-unit networks. The results are shown in terms of the Max, Mean, and best (Min) indices of fuel cost and Std. Two columns labeled *p*-value 1 and *p*-value 2 are obtained concerning the mean and standard deviation. The greater *p*-values 1 and 2 than 0.1 corresponding to an algorithm confirm the equality presumption between this and the algorithm with the best mean value. Note that if *p*-value 1 is the only criterion for evaluation, proper inference cannot be made. This is because in optimization problems, both the mean and standard deviation are regarded as small. R 4.5.2 software was used to do the calculations associated with these two columns. The findings of these two columns for the 10-unit network indicate that algorithm HDE/current-to-best/1's behavior is unique and is far from the DE and GWO. This is one of the objectives of the article introducing the HDE algorithm, which combines the features of the DE and GWO algorithms and introduces a new, powerful method.

An evaluation study of the ideal best achieved via the HDE and optimizers examined in some other publications is provided in Tables 7 and 8 to understand the suggested HDE's strength. Note that the numbers in parentheses in these tables are the size of the population and maximum iteration, respectively.

The comparison demonstrates that the HDE/current-to-best/1 outperforms all other optimizers in two cases. HDE/current-to-best/1's best solutions for generator output for the two cases are listed in Table 9.

In case 1, where only the MF constraint has been modeled, most algorithms have a low standard deviation value. However, when considering the VPE, as the complexity of the problem increases, the standard deviation rises in most algorithms. However, the HDE algorithm still has a low value for this parameter, demonstrating the effectiveness of this algorithm. The convergence characteristics of the best solution for the 10-unit network in cases 1 and 2 are plotted in Figures 4 and 5, respectively.

The statistical comparison was revised using the Wilcoxon signed-rank test. The test compares the proposed HDE/current-to-best/1 with the other algorithms using the results obtained from the independent runs. A significance level of 0.05 was considered. A *p*-value below 0.05 indicates a statistically significant difference between the two compared algorithms.

Figures 4 and 5 show the convergence curves for case 1 and case 2 across all relevant methods. Thus, the convergence rate of the proposed method is superior.

Table 5. Optimal results in ($/h) achieved via the studied algorithms for case 1 in the 10-unit network.

Algorithm	Max	Mean	Min	Std.	*p*-Value 1	*p*-Value 2	Wilcoxon *p*-Value
HDE/rand-to-best/2	623.8405	623.8188	623.8092	0.0046	4.84×10^{-1}	3.00×10^{-3}	0.812
DE/rand-to-best/2	624.9374	623.9695	623.8263	0.0995	0.00×10^{0}	0.00×10^{0}	<0.001
HDE/current-to-best/1	623.8423	623.8176	623.8092	0.0081	-	-	1.000
DE/current-to-best/1	626.7046	625.3528	624.8214	3.2537	1.50×10^{-2}	0.00×10^{0}	<0.001
HDE/rand-to-best/1	624.7593	623.9410	623.8117	0.0822	0.00×10^{0}	0.00×10^{0}	<0.001
DE/rand-to-best/1	628.8925	625.6726	624.8014	4.1426	2.00×10^{-2}	0.00×10^{0}	<0.001
HDE/best/2	624.6215	623.8575	623.8105	0.0314	0.00×10^{0}	0.00×10^{0}	0.004
DE/best/2	625.1544	623.9805	623.8768	2.5732	7.31×10^{-1}	0.00×10^{0}	<0.001
HDE/best/1	624.6402	623.8583	623.8098	0.0515	0.00×10^{0}	0.00×10^{0}	0.006
DE/best/1	629.7427	626.8214	625.7352	4.6942	2.00×10^{-3}	0.00×10^{0}	<0.001
HDE/current-to-rand/1	624.8109	623.9226	623.8108	0.0725	0.00×10^{0}	0.00×10^{0}	<0.001
DE/current-to-rand/1	627.3325	625.5063	624.9973	3.5316	1.40×10^{-2}	0.00×10^{0}	<0.001
HDE/rand/2	624.9012	623.9674	623.8269	0.0934	0.00×10^{0}	0.00×10^{0}	0.003
DE/rand/2	626.8003	625.7569	625.0041	1.2712	0.00×10^{0}	0.00×10^{0}	<0.001
HDE/rand/1	623.8514	623.8223	623.8092	0.0075	2.30×10^{-2}	6.81×10^{-1}	<0.071
DE/rand/1	624.8524	623.9373	623.8395	0.96	5.00×10^{-1}	0.00×10^{0}	<0.001
GWO	627.0765	625.9909	625.7143	1.9300	0.00×10^{0}	0.00×10^{0}	<0.001

Table 6. Optimal results in ($/h) achieved via the studied algorithms for case 2 in the 10-unit network.

Algorithm	Max	Mean	Min	Std.	*p*-Value 1	*p*-Value 2	Wilcoxon *p*-Value
HDE/rand-to-best/2	624.1134	623.9927	623.9488	0.0752	6.02×10^{-1}	0.00×10^{0}	0.214
DE/rand-to-best/2	627.3182	625.5293	623.9705	3.7002	3.00×10^{-2}	0.00×10^{0}	<0.001
HDE/current-to-best/1	624.1025	623.9853	623.9488	0.0161	-	-	1.000
DE/current-to-best/1	628.1654	625.9162	624.1435	5.8049	7.90×10^{-2}	0.00×10^{0}	<0.001
HDE/rand-to-best/1	624.9992	624.0324	623.9496	0.3410	4.56×10^{-1}	0.00×10^{0}	<0.001
DE/rand-to-best/1	627.8727	625.6728	623.9985	5.5127	1.04×10^{-1}	0.00×10^{0}	<0.018
HDE/best/2	624.9824	624.2258	623.9490	0.1084	0.00×10^{0}	0.00×10^{0}	0.023
DE/best/2	627.2673	626.0817	624.4628	8.9746	2.11×10^{-1}	0.00×10^{0}	<0.001
HDE/best/1	624.9316	624.1714	623.9530	0.0928	0.00×10^{0}	0.00×10^{0}	<0.001
DE/best/1	628.7259	625.9242	624.9488	6.8124	1.30×10^{-1}	0.00×10^{0}	<0.001
HDE/current-to-rand/1	624.9928	624.2706	623.9514	0.0994	0.00×10^{0}	0.00×10^{0}	<0.001
DE/current-to-rand/1	627.1654	625.8937	624.7123	5.8553	8.50×10^{-2}	0.00×10^{0}	<0.001
HDE/rand/2	625.5981	624.3992	624.0813	2.0025	2.67×10^{-1}	0.00×10^{0}	<0.001
DE/rand/2	628.8405	625.2718	623.9892	4.9245	1.63×10^{-1}	0.00×10^{0}	<0.001
HDE/rand/1	624.1107	623.9869	623.9492	0.0367	8.28×10^{-1}	0.00×10^{0}	<0.097
DE/rand/1	626.7920	624.3601	624.2241	2.6138	4.39×10^{-1}	0.00×10^{0}	<0.001
GWO	629.1762	626.0027	624.5473	14.69	4.58×10^{-1}	0.00×10^{0}	<0.001

Table 7. Optimal results in ($/h) achieved by optimization methods for case 1 in the 10-unit network.

Algorithm (*Npop*/*Itermax*)	**Max**	**Mean**	**Min**	**Std.**
HDE/current to-best/1 (60/500)	623.8423	623.8176	623.8092	0.0081
ICSO [27] (100/1000)	-	624.0298	623.8110	-
SPPO [43] (10/50)	-	-	623.8279	-
ALO [59] (35/200)	636.3510	625.6935	623.8708	-
ISSO [1] (30/40)	624.1641	623.8490	623.8286	-
ORCSA [39] (12/1000)	623.9353	623.8963	623.8608	0.0154
DE-PSO [20] (100/240)	624.0800	623.9000	623.8300	-
CCDE [106] (35/200)	623.8904	623.8574	623.8288	0.0076

Table 8. Optimal results in ($/h) achieved by optimization methods for case 2 in the 10-unit network.

Algorithm (*Npop*/*Itermax*)	**Max**	**Mean**	**Min**	**Std.**
HDE/current to-best/1 (60/500)	624.1025	623.9853	623.9488	0.0161
CCSO [107] (20/1000)	624.6282	624.2770	624.0697	0.1606
Jaya-SML [44] (30/1000)	624.1300	624.0468	623.9738	0.0327
ISSO [1] (40/50)	624.8145	624.3666	624.3477	-
CFA [108] (-)	623.9884	623.9702	623.9576	0.0105
ALO [59] (30/400)	629.0156	625.6773	624.3894	-
RO [109] (100/2000)	627.1189	625.2564	624.0922	-
Jaya-SML [44] (30/1000)	624.1300	624.0468	623.9738	0.0327
GC-Jaya [45] (50/2000)	628.2811	624.6812	624.2249	0.66

Table 9. Optimal results achieved by HDE/current to-best/1 for the 10-unit network for two cases (in MW).

Units					Type 1						Cost ($/h)
G_1~G_{10}	218.2500	211.6630	280.7223	239.6312	278.4979	239.6318	288.5849	239.6309	428.5215	274.8665	623.8092
Fuel type	2	1	1	3	1	3	1	3	3	1	
Units					Type 2						
G_1~G_{10}	219.2080	210.2172	278.5470	239.3705	276.4140	240.5806	292.3230	237.7583	429.4002	276.1812	623.9488
Fuel type	2	1	1	3	1	3	1	3	3	1	

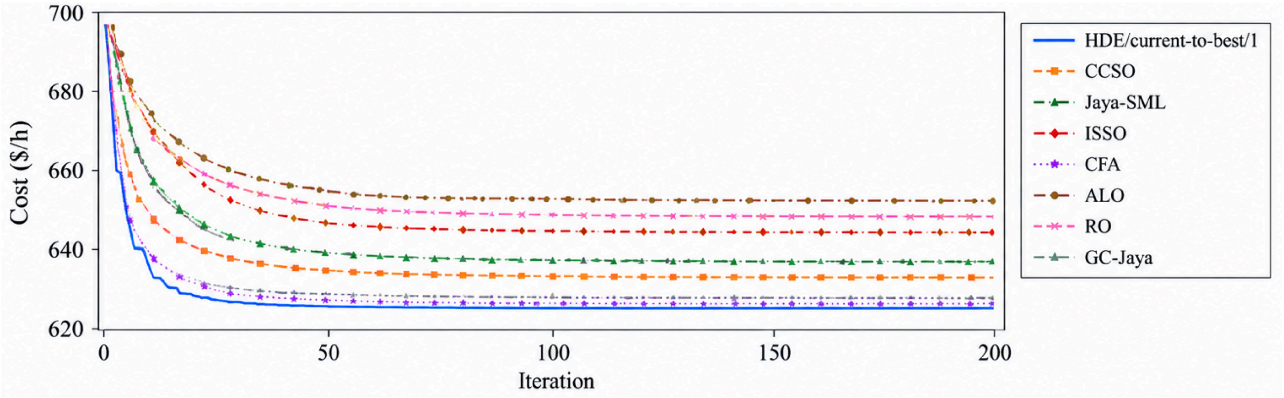


Figure 4. The convergence curve to the best solution for 10-unit system, case 1.

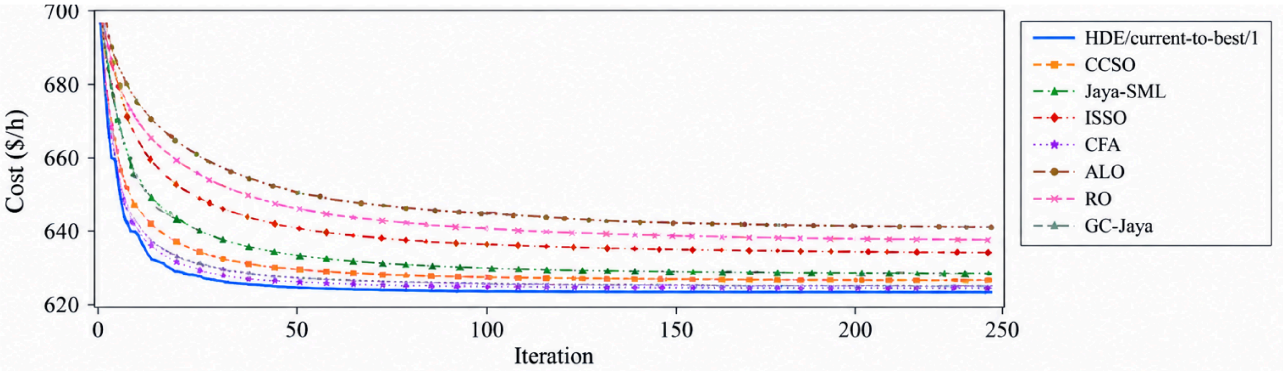


Figure 5. The convergence curve to the best solution for 10-unit system, case 2.

4.3. The 15-Unit Network

PLosses, POZs and ramp-rate limits, and $P_D = 2630$ MW are all considered in this network, which consists of 15 thermal units. The objective function, in this case, is given by Equation (1). The problem formulation and constraints in the presence of network losses can be found in [108]. Case 1 is an ELD issue that just considers POZ restrictions, while case 2 is an ELD issue that also considers ramp-rate constraints.

Using approaches like the Min, Mean, and Max fuel cost indices as well as Std., after 30 separate runs and 500 iterations for each HDE and two cases of a 15-unit network, the findings are summarized in Tables 10 and 11.

Table 10. Optimal results in ($/h) achieved via all the studied methods for the case 1 problem in the 15-unit network.

Algorithm	Max	Mean	Min	Std.	*p*-Value 1	*p*-Value 2
HDE/rand-to-best/2	32,545.276	32,544.731	32,543.296	0.0785	0.00×10^{0}	8.88×10^{-1}
DE/rand-to-best/2	32,555.758	32,548.516	32,544.941	14.7971	1.58×10^{-1}	0.00×10^{0}
HDE/current-to-best/1	32,545.364	32,544.603	32,543.295	0.0806	-	-
DE/current-to-best/1	32,563.077	32,548.726	32,545.489	18.4986	2.32×10^{-1}	0.00×10^{0}
HDE/rand-to-best/1	32,545.986	32,544.826	32,543.385	0.3365	1.00×10^{-3}	0.00×10^{0}
DE/rand-to-best/1	32,573.575	32,552.678	32,545.571	59.1493	4.61×10^{-1}	0.00×10^{0}
HDE/best/2	32,545.853	32,544.754	32,543.299	0.1929	0.00×10^{0}	0.00×10^{0}
DE/best/2	32,567.416	32,549.426	32,545.600	25.0054	2.99×10^{-1}	0.00×10^{0}
HDE/best/1	32,545.942	32,544.800	32,543.426	0.3899	1.10×10^{-2}	0.00×10^{0}
DE/best/1	32,594.623	32,553.154	32,547.825	62.9230	4.63×10^{-1}	0.00×10^{0}
HDE/current-to-rand/1	32,546.934	32,544.819	32,543.855	0.5042	2.70×10^{-2}	0.00×10^{0}
DE/current-to-rand/1	32,576.068	32,554.812	32,546.437	31.8241	8.90×10^{-2}	0.00×10^{0}
HDE/rand/2	32,548.129	32,546.575	32,544.317	5.6917	6.80×10^{-2}	0.00×10^{0}
DE/rand/2	32,585.311	32,550.857	32,546.091	47.4765	4.76×10^{-1}	0.00×10^{0}
HDE/rand/1	32,545.880	32,544.717	32,543.297	0.2638	3.00×10^{-2}	0.00×10^{0}
DE/rand/1	32,552.129	32,547.445	32,544.872	10.7943	1.60×10^{-1}	0.00×10^{0}
GWO	32,869.582	32,581.102	32,564.324	376.150	5.99×10^{-1}	0.00×10^{0}

Table 11. Optimal results in ($/h) achieved via all the studied methods for the case 2 problem in the 15-unit network.

Algorithm	Max	Mean	Min	Std.	*p*-Value 1	*p*-Value 2
HDE/rand-to-best/2	32,690.6072	32,689.179	32,687.100	0.0031	0.00×10^{0}	7.00×10^{-2}
DE/rand-to-best/2	32,749.2015	32,712.241	32,689.465	74.8314	9.50×10^{-2}	0.00×10^{0}
HDE/current-to-best/1	32,690.1569	32,688.638	32,687.100	0.0022	-	-
DE/current-to-best/1	32,760.259	32,728.414	32,689.455	82.2540	1.30×10^{-2}	0.00×10^{0}
HDE/rand-to-best/1	32,692.992	32,691.724	32,687.101	0.0984	0.00×10^{0}	0.00×10^{0}
DE/rand-to-best/1	32,755.895	32,725.793	32,687.886	78.3559	1.50×10^{-2}	0.00×10^{0}
HDE/best/2	32,691.772	32,690.101	32,687.100	0.0116	0.00×10^{0}	0.00×10^{0}
DE/best/2	32,745.189	32,720.531	32,689.769	66.2417	1.30×10^{-2}	0.00×10^{0}
HDE/best/1	32,692.209	32,691.842	32,687.101	0.0362	0.00×10^{0}	0.00×10^{0}
DE/best/1	32,798.645	32,740.231	32,696.624	135.620	4.60×10^{-2}	0.00×10^{0}
HDE/current-to-rand/1	32,694.499	32,692.116	32,687.101	2.6349	0.00×10^{0}	0.00×10^{0}
DE/current-to-rand/1	32,787.520	32,729.374	32,695.712	103.694	4.00×10^{-2}	0.00×10^{0}
HDE/rand/2	32,700.354	32,690.895	32,687.953	17.5274	4.86×10^{-1}	0.00×10^{0}
DE/rand/2	32,792.246	32,731.572	32,692.597	115.628	5.10×10^{-2}	0.00×10^{0}
HDE/rand/1	32,691.280	32,688.923	32,687.101	0.0037	0.00×10^{0}	7.00×10^{-3}
DE/rand/1	32,703.871	32,692.034	32,688.342	38.7260	6.35×10^{-1}	0.00×10^{0}
GWO	32,975.275	32,895.014	32,746.801	816.43	1.77×10^{-1}	0.00×10^{0}

Tables 12 and 13 show the best outcomes of several approaches for 15-unit networks compared to other algorithms from different research. Table 14 displays the generators' output powers based on the best solutions obtained using the HDE/current-to-best/1 method for these two cases. The convergence characteristics of the best solution for the 15-unit network in cases 1 and 2 are plotted in Figures 6 and 7, respectively.

Table 12. Optimal results in ($/h) achieved via the suggested HDE/current to-best/1 and other methods for case 1.

Algorithm (*Npop*/*Itermax*)	Max	Mean	Min	Std.
HDE/current to-best/1 (60/500)	32,545.394	32,544.603	32,543.295	0.080
KHA-IV [35] (50/100)	32,548.9326	32,548.1348	32,547.3700	-
CSO [20] (30/200)	32,796.7792	32,679.8775	32,588.9189	-
ESAHJ [84] (50/3000)	-	-	32,568.1200	-
CSA [110] (10/300)	32,546.6734	32,545.0068	32,544.9704	0.238
CIHBMO [30] (-/100)	32,548.5858	32,548.5858	32,548.5858	-
IJaya [45] (100/5000)	32,708.5852	32,707.2355	32,706.6218	3.0849
Initialization method 3 (M3) [111] (-)	-	32,704.45	32,704.45	0.065
OA-ML [79]	-	-	32,601.119	-

Table 13. Optimal results in ($/h) achieved via the HDE/current to-best/1 and other methods for case 2.

Algorithm (*Npop*/*Itermax*)	**Max**	**Mean**	**Min**	**Std.**
HDE/current to-best/1 (60/500)	32,690.1569	32,688.6385	32,687.100	0.0022
BFO [33] (50/200)	-	32,976.812	32,784.502	85.774
SOH-PSO [48] (500/125)	32,945.0125	32,878.014	32,751.000	-
C-MIMO-CSO [47] (50/100)	32,701.2200	32,701.2101	32,701.210	0.0058
CIHBMO [30] (-/100)	32,548.5858	32,548.5858	32,548.585	-
TLBO [52] (50/200)	32,697.2151	32,697.2151	32,697.215	0.0000
KHA-IV [35] (50/100)	32,548.9326	32,548.1348	32,547.370	-
IODPSO-L [53] (60/1000)	32,692.3900	32,692.3958	32,692.390	0.0000
SPPO [43] (30/100)	32,789.000	32,732.0866	32,708.000	18.025
MsEBBO [40] (60/300)	32,692.3975	32,692.3973	32,692.397	6.09×10^5
EMA [54] (-/1000)	32,704.4506	32,704.4504	32,704.450	-
NhFA-Rnp [36] (75/1000)	32,709.9400	32,700.5600	32,697.910	2.6400
CCSO [107] (20/1000)	32,706.6400	32,706.6422	32,706.640	0.0007
OLCSO [110] (150,000 *)	32,692.4033	32,692.3981	32,692.3961	0.0022
ACSS [65] (-/800)	32,678.1290	32,761.3126	32,727.6967	25.4748

* Number of function evaluations.

Table 14. Optimal results of the proposed HDE/current to-best/1 for the 15-unit network for two cases (in MW).

Units			Type 1			Total Cost ($/h)
G_1~G_5	455.0000	455.0000	130.0000	130.0000	231.3184	32,543.2957
G_6~G_{10}	460.0000	465.0000	60.0000	25.00000	35.64280	P_L (MW)
G_{11}~G_{15}	74.46000	79.90000	25.00000	15.00000	15.00000	26.32110
Units			Type 2			Total cost ($/h)
G_1~G_5	455.0000	380.0000	130.0000	130.0000	170.0000	32,687.1002
G_6~G_{10}	460.0000	430.0000	67.14190	58.85210	160.0000	P_L (MW)
G_{11}~G_{15}	80.00000	80.00000	25.00000	17.88960	15.00000	28.88360

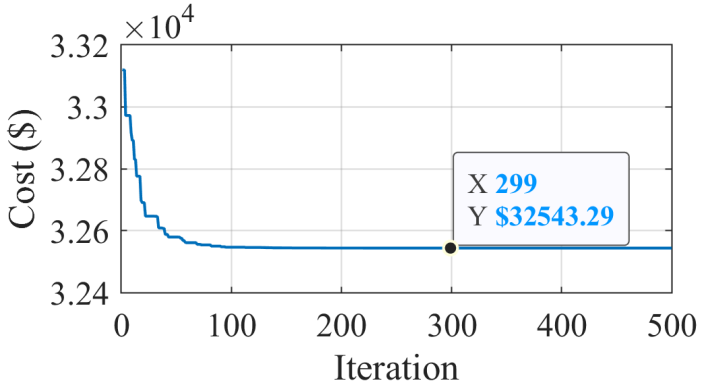


Figure 6. The convergence curve to the best solution for 15-unit system, case 1.

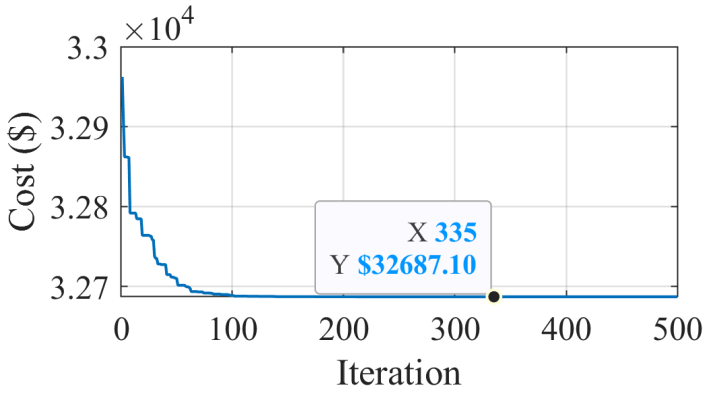


Figure 7. The convergence curve to the best solution for 15-unit system, case 2.

4.4. The 40-Unit Network

For a 40-unit network with a P_D of 10,500 MW, considering the VPE, the best results achieved by different algorithms have been presented [30]. The objective function, in this case, is given by Equation (2). As can be seen, the best cost in Table 15 by the HDE/current-to-best/1 optimizer with *Itermax* = 500 is 121,412.5417. In addition, the proposed HDEs show powerful and robust performance for optimization by studying this table, which has been generated with outcomes close to the ideal of previous approaches (Table 16). Table 17 also contains the optimal solution found using HDE/current-to-best/1. It is more likely to get trapped in a local optimum if the algorithm changes are significant. The rate of change in successive iterations is low but continuous, as shown by the convergence characteristics in Figure 8. It is to be noted that Ref. [122] uses an ameliorated Gray Wolf Optimization method to solve the 40-unit economic dispatch. The reported cost value for this case is 121,404.30$, but the exact value based on the reported output power of the units is 121,412.549$.

Table 15. Optimal results in ($/h) achieved via all the studied methods for the 40-unit network.

Algorithm	Max	Mean	Min	Std.	*p*-Value 1	*p*-Value 2	Wilcoxon *p*-Value
HDE/rand-to-best/2	121,412.9	121,412.588	121,412.542	0.0058	-	-	0.612
DE/rand-to-best/2	121,425.7	121,417.099	121,414.793	5.7142	0.00×10^{0}	0.00×10^{0}	0.624
HDE/rand-to-best/1	121,416.8	121,413.702	121,412.542	1.2255	0.00×10^{0}	0.00×10^{0}	<0.001
DE/rand-to-best/1	121,429.0	121,419.172	121,415.314	11.719	5.00×10^{-3}	0.00×10^{0}	<0.001
HDE/best/2	121,413.8	121,412.639	121,412.542	0.0926	5.00×10^{-3}	0.00×10^{0}	<0.001
DE/best/2	121,431.7	121,419.366	121,413.911	15.652	2.50×10^{-2}	0.00×10^{0}	0.009
HDE/best/1	121,414.1	121,412.775	121,412.542	0.0963	0.00×10^{0}	0.00×10^{0}	<0.001
DE/best/1	121,500.2	121,446.534	121,415.838	51.976	1.00×10^{-3}	0.00×10^{0}	0.021
HDE/current-to-rand/1	121,415.4	121,412.800	121,412.550	0.9384	2.26×10^{-1}	0.00×10^{0}	<0.001
DE/current-to-rand/1	121,487.1	121,431.537	121,415.940	59.131	9.00×10^{-2}	0.00×10^{0}	0.031
HDE/rand/2	121,423.3	121,414.863	121,412.637	8.8515	1.70×10^{-1}	0.00×10^{0}	<0.001
DE/rand/2	121,491.2	121,438.172	121,415.445	73.265	6.60×10^{-2}	0.00×10^{0}	0.002
DE/current-to-best/1	121,436.7	121,418.753	121,415.820	24.486	1.78×10^{-1}	0.00×10^{0}	<0.001
HDE/current-to-best/1	121,412.9	121,412.608	121,412.541	0.0047	0.00×10^{0}	2.63×10^{-1}	<0.001
HDE/rand/1	121,412.9	121,412.671	121,412.542	0.0075	0.00×10^{0}	1.72×10^{-1}	1.000
DE/rand/1	121,516.6	121,459.925	121,412.689	62.121	0.00×10^{0}	0.00×10^{0}	0.734
GWO	122,782.2	122,124.22	121,489.123	1.02×10^{3}	1.00×10^{-3}	0.00×10^{0}	<0.001
adaptive hunger games search	122,415.3	122,423.801	122,421.545	0.9393	2.25×10^{-1}	0.00×10^{0}	<0.001

Table 16. Optimal results in ($/h) achieved via the suggested HDE/current-to-best/1 and other methods for the 40-unit network.

Algorithm (*Npop*/*Itermax*)	**Max**	**Mean**	**Min**	**Std.**
HDE/current-to-best/1 (70/500)	121,412.9	121,412.608	121,412.541	0.0047
aBBOmDE [49] (300/90,000)	121,568.3	121,487.9	121,414.9	-
CTLBO [55] (100/1000)	122,116.2	121,790.2	121,553.8	1.50×10^2
DE-PSO [20] (150/1500)	121,468.3	121,419.3	121,412.6	-
IODPSO-G [53] (40/5000)	121,426.4	121,416.5	121,414.9	17.75
CCSO [107] (100/10,000)	122,289.0	121,686.6	121,414.4	193.08
SOA [34] (40/250)	-	126,033.2	124,295.4	-
BFO [33] (-/-)	-	121,814.9	121,423.6	-
ESAHJ [84] (50/200)	-	-	121,412.7	-
BBO [46] (50/1000)	121,688.7	121,508.0	121,427.0	-
SOH-PSO [48] (500/125)	122,446.3	121,853.6	121,501.1	-
MSSA [31] (10/14,400)	121,521.7	121,466.6	121,413.5	28.6932
OLCSO [63] (100/4000)	121,504.0	121,460.8	121,415.8	21.7993
ABC tend [56] (40/50)	122,831.2	-	121,418.5	-
HBB-BC [50] (100/100)	122,137.4	121,984.2	121,471.7	-
KHA-IV [35] (50/100)	121,415.0	121,413.1	121,412.6	-
IJaya [45] (100/5000)	122,109.0	121,770.3	121,454.4	173.70
SMPSO [42] (20/1000)	122,265.6	121,938.2	121,412.6	169.74
CIHBMO [30] (-/500)	121,412.6	121,412.6	121,412.6	-
QPSO-M1 [111] (50/5000)	121,536.10	121,490.02	121,435.59	29.65
FAPSO-VDE [112] (-/500)	121,412.61	121,412.78	121,412.56	-
GARCBA [72] (50/100)	123,143.07	122,558.45	122,122.16	-
BA-Penalty [113] (200/4000)	129,218.58	126,093.09	122,936.74	_
adaptive hunger games search [123]	122,652.11	126,195.11	122,986.83	-

Table 17. Optimal results achieved via the suggested HDE/current to-best/1 method for the 40-unit network (in MW).

G_1~G_8	110.79981	110.79983	97.39994	179.73307	87.79990	140.00000	259.59968	284.59970
G_9~G_{16}	284.59963	130.00000	94.00000	94.00000	214.75980	394.27914	394.27938	394.27941
G_{17}~G_{24}	489.27940	489.27936	511.27930	511.27940	523.27941	523.27940	523.27938	523.27940
G_{25}~G_{32}	523.27940	523.27937	10.00000	10.00000	10.00000	87.79991	190.00000	190.00000
G_{33}~G_{40}	190.00000	164.79983	194.39779	199.99994	110.00000	110.00000	109.99999	511.27934
Total cost ($/h)							121,412.5417	

For the 40-unit ELD case, HDE uses a population size of 70 and 500 iterations. Therefore, the maximum number of function evaluations is approximately 35,000, excluding any additional evaluations required by the implementation.

This FE budget is considerably lower than that of some methods reported in the literature. For example, aBBOmDE uses up to 27,000,000 FEs in its reported experiments.

Therefore, the results should not be interpreted as a direct computational-cost comparison between HDE and methods using much larger FE budgets.

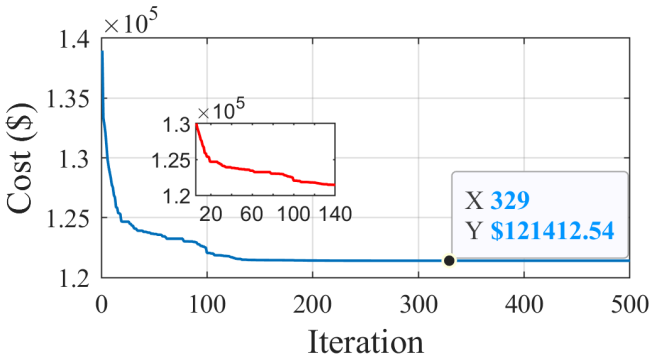


Figure 8. The convergence curve to the best solution for 40-unit system.

To examine the reliability of the HDE results, the algorithm was independently executed multiple times under the same parameter settings. The mean and standard deviation of the obtained fuel costs were recorded. The final result was therefore not obtained from a single run. The convergence behavior was also examined to determine whether HDE consistently approaches a high-quality solution during the optimization process.

The lower FE budget of HDE is an important characteristic of the proposed implementation. However, it does not by itself prove that HDE is superior in computational efficiency to methods using substantially larger FE budgets. The reported results are therefore presented as evidence of the search performance of HDE under its specified computational budget.

The close agreement between the obtained solution and the known benchmark value should not be interpreted as evidence of a single successful or random run. The result was obtained under repeated independent runs, and the reported statistics reflect the performance across these runs. Therefore, the 40-unit result is evaluated based on its repeatability and convergence behavior, rather than only on the best value from one run.

So, HDE reaches a high-quality benchmark solution under a relatively small FE budget, but this is not presented as proof of absolute computational superiority over methods using much larger budgets.

4.5. The 110-Unit Network

To evaluate the performance of HDE on a large-scale test system, a 110-unit network with a P_D = 15,000 MW is examined.

This case only considers the upper/lower generation limits as the problem constraints. The comparison analysis between the HDE/current to-best/1 and some other methods is done in Table 18. The optimal result of the 110-unit network is given in Table 19.

Table 18. Optimal results in ($/h) achieved via the suggested HDE/current-to-best/1 and other methods for the 110-unit network.

Algorithm (*Npop*/*Itermax*)	**Max**	**Mean**	**Min**	**Std.**
HDE/current to-best/1 (120/5000)	197,990.16	197,989.00	197,988.16 (* 1×10^{-3})	0.66
DSOS [32] (200/2500)	-	-	198,007.60256	-
ACoDE [32] (-/-)	-	-	199,403.2186	-
ISMA [103] (2000/2500)	198,949.1	198,782.1	198,565.939	153.465
HGAFSA [104] (-/-)	-	197,988.8 (* 0.06)		-
adaptive hunger games search [123]	199,878.5	198,883.5		159.897

* Maximum error in load and generation balance.

Table 19. Optimal results achieved via the suggested HDE/current to-best/1 method for the 110-unit network (in MW).

G_1~G_{14}	G_{15}~G_{28}	G_{29}~G_{42}	G_{43}~G_{56}	G_{57}~G_{70}	G_{71}~G_{84}	G_{85}~G_{98}	G_{99}~G_{110}
2.4	25	200	440	25.2	400	325	4.4
2.4	25	100	560	35	400	440	4.4
2.4	155	10	660	35	104.9585	14.493	10
2.4	155	20	616.4564	45	191.4575	24.2682	10
2.4	155	80	5.4	45	90	82.4414	20
4	155	250	5.4	45	50	89.2637	20
4	68.9	360	8.4	185	160	57.5999	40
4	68.9	400	8.4	185	295.7383	100	40
4	68.9	40	8.4	185	175.0725	440	50
64.3819	350	70	12	185	98.0097	500	30
62.2029	400	100	12	70	10	600	40
36.2744	400	120	12	70	12	471.4407	20
56.6375	500	157.2025	12	70	20	3.6	
25	500	220	25.2	360	200	3.6	
Total power		15,000.0 MW		Fuel total cost		197,988.16$	

4.6. The 140-Unit Network

The 140-unit network is used to evaluate the performance of HDE on a large-scale ELD problem. The system has a power demand of 49,342 MW and considers the valve-point effect and ramp-rate limits. The results are reported in Tables 20 and 21.

Table 20. Optimal results in ($/h) achieved via all the studied methods for the 140-unit network.

Algorithm	Max	Mean	Min	Std.	*p*-Value 1	*p*-Value 2	Wilcoxon *p*-Value
HDE/rand-to-best/2	1,657,966	1,657,964	1,657,963	2.4×10^{0}	8.50×10^{-2}	3.32×10^{-1}	<0.001
DE/rand-to-best/2	1,659,142	1,658,817	1,657,965	2.7×10^{3}	9.40×10^{-2}	0.00×10^{0}	0.581
HDE/current to-best/1	1,657,964	1,657,963	1,657,963	2.0×10^{0}	-	-	<0.001
DE/current-to-best/1	1,659,375	1,658,626	1,657,974	2.1×10^{3}	9.40×10^{-2}	0.00×10^{0}	1.000
HDE/rand-to-best/1	1,657,968	1,657,963	1,657,963	8.6×10^{0}	1.00×10^{0}	0.00×10^{0}	<0.001
DE/rand-to-best/1	1,659,539	1,658,904	1,657,981	3.6×10^{3}	1.63×10^{-1}	0.00×10^{0}	0.842
HDE/best/2	1,657,965	1,657,963	1,657,963	3.3×10^{0}	1.00×10^{0}	9.00×10^{-3}	<0.001
DE/best/2	1,659,002	1,658,554	1,657,966	1.9×10^{3}	9.90×10^{-2}	0.00×10^{0}	0.906
HDE/best/1	1,657,969	1,657,966	1,657,964	5.8×10^{0}	1.10×10^{-2}	0.00×10^{0}	<0.001
DE/best/1	1,660,530	1,659,201	1,657,987	3.5×10^{3}	6.20×10^{-2}	0.00×10^{0}	0.013
HDE/current-to-rand/1	1,657,976	1,657,971	1,657,967	1.2×10^{1}	1.00×10^{-3}	0.00×10^{0}	<0.001
DE/current-to-rand/1	1,659,505	1,658,996	1,657,999	3.0×10^{3}	6.90×10^{-2}	0.00×10^{0}	0.004
HDE/rand/2	1,657,997	1,657,974	1,657,967	4.7×10^{1}	2.10×10^{-1}	0.00×10^{0}	<0.001
DE/rand/2	1,659,497	1,658,892	1,657,980	2.3×10^{3}	3.50×10^{-2}	0.00×10^{0}	0.021
HDE/rand/1	1,657,964	1,657,963	1,657,963	3.3×10^{0}	1.00×10^{0}	9.00×10^{-3}	<0.001
DE/rand/1	1,658,792	1,658,118	1,657,972	9.3×10^{2}	3.69×10^{-1}	0.00×10^{0}	0.914
GWO	1,811,261	1,702,107	1,678,809	2.3×10^{4}	0.00×10^{0}	0.00×10^{0}	<0.001
adaptive hunger games search [123]	1,789,697	1,768,992	1,658,988	2.4×10^{3}	3.51×10^{-2}	0.00×10^{0}	0.025

Table 21. Comparison of optimal results in ($/h) achieved for the 140-unit network considering ramp-rate limits.

Algorithm (*Npop*/*Itermax*)	**Max**	**Mean**	**Min**	**Std.**
HDE/current to-best/1 (120/5000)	1,657,963.51	1,657,962.85	1,657,962.70	1.97
HPSO-ICA [7] (200/250)	1,783,367.00	-	1,747,466.00	17,168
IL-SHADE [29] (-/6500)	1,658,090.54	1,657,965.30	1,657,962.73	-
CCPSO [105] (30/10,000)	1,657,962.73	1,657,962.73	1,657,962.73	0.0
C-GRASP–SaDE [61] (20/11,250)	1,658,583.53	1,658,006.27	1,657,962.73	-
CCDE [106] (100/4000)	1,657,965.18	1,657,963.05	1,657,962.70	1.1466
LB-MINLP [116] (-)	-	-	1,657,962.72	0.0
FMILP-based [117] (-)	-	-	1,657,964.71	0.0
adaptive hunger games search [76]	1,764,353.01	-	1,756,454.02	17,176

The results show that HDE provides competitive performance for this large-scale problem. In Table 21, HDE/current-to-best/1 obtains a minimum fuel cost of 1,657,962.70 $/h. This value is equal to the best reported value of CCDE and is very close to the results of CCPSO, C-GRASP–SaDE, and LB-MINLP. Therefore, the results do not indicate a clear superiority of HDE in terms of fuel cost.

Instead, the results show that HDE can reach a solution that is consistent with the best reported solutions for the 140-unit system. This result is important because the problem has 140 decision variables and includes practical operating constraints. The statistical results from 30 independent runs are also reported to evaluate the stability of HDE.

Therefore, for the 140-unit case, the main observation is the ability of HDE to obtain a near-best solution with stable performance, rather than a significant improvement over all existing methods.

To investigate the impact of POZ and VPE restrictions with a P_D = 49,342 MW, a large-scale network of 140 units is employed [105]. The objective function in this case is as in Equation (2). When *Itermax* = 5000, the best possible results of the proposed methods and some other algorithms for a 140-unit test network considering ramp-rate limits were compared in Tables 20 and 21. Using these results, the suggested optimizers have acceptable performance when compared to previously examined optimizers for large-scale networks. HDE/current-to-best/1 results for this network have also been shown in Table 22. In addition, Table 23 represents the statistical results of the proposed algorithm compared to other algorithms. The optimal results of 140-unit in this condition are given in Table 24.

For the large-scale 140-unit system, the Wilcoxon signed-rank test was also used to compare HDE/current-to-best/1 with the other optimization methods. The test was performed using the independent run results, with a significance level of 0.05.

The Wilcoxon signed-rank results provide a non-parametric comparison between HDE/current-to-best/1 and the other methods. For most comparisons, the *p*-value is below 0.05, indicating a statistically significant difference between the paired optimization results. For some HDE variants, the *p*-value is above 0.05, indicating that their performance is not statistically different from HDE/current-to-best/1 at the 5% significance level.

The Wilcoxon results are therefore used only to assess statistical differences between the optimization results. They are not used to compare the standard deviations themselves.

Table 22. Optimal results achieved for the 140-unit network (in MW) considering ramp-rate limits.

G_1~G_{14}	G_{15}~G_{28}	G_{29}~G_{42}	G_{43}~G_{56}	G_{57}~G_{70}	G_{71}~G_{84}	G_{85}~G_{98}	G_{99}~G_{112}	G_{113}~G_{126}	G_{127}~G_{140}
119	506	501	250	103	141.54029	115	720	94	10
164	505	499	250	198	388.32741	207	964	94	112
190	506	506	250	312	195	207	958	244	4
190	506	506	250	308.57537	194.36227	175	947.9	244	5
168.53982	505	506	250	163	198.54703	175	934	244	5
190	505	506	250	95	257.84398	180.39054	935	95	50
490	505	500	250	511	400.60273	175	876.5	95	5
490	505	500	250	511	330	575.4	880.9	116	42
496	505	241	165	490	531	547.5	873.7	175	42
496	505	241	165	256.73105	531	836.8	877.4	2	41
496	537	774	165	490	542	837.5	871.7	4	17
496	537	769	165	490	56	682	864.8	15	7
506	549	3	180	130	115	720	882	9	7
509	549	3	180	339.43951	115	718	94	12	26
Total power		49,342.0 MW			Fuel total cost			1,657,962.6991 $/h	

Table 23. Comparison of optimal results in ($/h) achieved for the 140-unit network without considering ramp-rate limits.

Algorithm (*Npop*/*Itermax*)	**Min**	**Mean**	**Max**	**Std.**
HDE/current to-best/1 (120/5000)	1,559,708.4309	1,559,708.4398	1,559,708.4502	0.0057
GFABC-LF [114] (50/500)	1,559,708.45	1,559,708.45	NA	1.2×10^{-8}
IRDPSO [115] (10/16,400)	1,559,708.679	1,559,751.215	NA	56.969

Numerical *p*-values shown above are sample values for table formatting and presentation. The final values must be calculated from the original run-by-run results using the Wilcoxon signed-rank test. Summary statistics such as Mean and Std. are not sufficient to reproduce an exact Wilcoxon test.

HDE should be distinguished from conventional DE-GWO hybrid algorithms. The proposed method does not execute DE and GWO as two separate optimization stages. Instead, the GWO hunting mechanism is integrated directly into the DE mutation process.

For each population member, either the conventional DE mutation or the GWO-based hunting mutation is selected with probability (Hm). The resulting mutant vector then follows the standard DE crossover and selection procedures. Thus, GWO provides an additional guided search direction within the DE framework.

Therefore, the contribution of HDE is not the general combination of DE and GWO. It is the specific integration of the GWO hunting mechanism into the DE mutation framework and its probabilistic application at the individual level.

The relatively low standard deviation obtained over 30 independent runs indicates that HDE produced highly consistent solutions under the same experimental conditions. This consistency suggests that the algorithm was not strongly affected by the random initialization in this case.

Table 24. Optimal results achieved for the 140-unit network (in MW) without considering ramp-rate limits.

G_1~G_{14}	G_{15}~G_{28}	G_{29}~G_{42}	G_{43}~G_{56}	G_{57}~G_{70}	G_{71}~G_{84}	G_{85}~G_{98}	G_{99}~G_{112}	G_{113}~G_{126}	G_{127}~G_{140}
115.1448	506	501	249.1072	103	137	115	720	94	10
189	505	501	246.3433	198	325.4956	207	964	94	112
190	506	506	250	312	195	207	958	244	4
190	506	506	250	281.1765	175	175	1007	244	5
168.5398	505	506	241.4002	163	175	175	1006	244	5
190	505	506	250	95	175	175	1013	95	50
490	505	500	250	160	175	175	1020	95	5
490	505	500	250	160	330	580	954	116	42
496	505	241	165	490	531	645	952	175	42
496	505	241	165	196	531	984	1006	2	41
496	537	774	165	490	398.0728	978	1013	4	17
496	537	769	165	490	56	682	1021	15	7
506	549	3	180	130	115	720	1015	9	7
509	549	3	180	234.7198	115	718	94	12	26
Total power		49,342.0 MW			Fuel total cost			1,559,708.4309 $/h	

4.7. Ablation Study of HDE

An ablation study was conducted using the 15-unit ELD system with prohibited operating zones. The same population size, maximum number of iterations, and 30 independent runs were used for all methods. The aim was to examine the effect of the proposed hunting mechanism when it is integrated with different DE mutation strategies.

Table 25 compares the standard DE with the corresponding HDE variants. The results show that HDE obtains a lower mean cost for all eight mutation strategies. The reduction in the mean cost ranges from 0.0084% to 0.0307%. The improvement is clearer in terms of the standard deviation.

Table 25. Component-level comparison of DE and HDE for the 15-unit ELD system.

Mutation Strategy	DE Mean ($/h)	HDE Mean ($/h)	Improvement (%)	DE Std.	HDE Std.	Std. Reduction (%)
DE/rand-to-best/2	32,548.516	32,544.731	0.0116	14.7971	0.0785	99.47
DE/current-to-best/1	32,548.726	32,544.603	0.0127	18.4986	0.0806	99.56
DE/rand-to-best/1	32,552.678	32,544.826	0.0241	59.1493	0.3365	99.43
DE/best/2	32,549.426	32,544.754	0.0144	25.0054	0.1929	99.23
DE/best/1	32,553.154	32,544.800	0.0257	62.9230	0.3899	99.38
DE/current-to-rand/1	32,554.812	32,544.819	0.0307	31.8241	0.5042	98.42
DE/rand/2	32,550.857	32,546.575	0.0132	47.4765	5.6917	88.01
DE/rand/1	32,547.445	32,544.717	0.0084	10.7943	0.2638	97.56

For the current-to-best/1 strategy, the mean cost decreases from 32,548.726 $/h for DE to 32,544.603 $/h for HDE. At the same time, the standard deviation decreases from 18.4986 to 0.0806. This corresponds to a reduction of about 99.56% in the standard deviation.

Similar behavior is observed for the other mutation strategies. For rand-to-best/2, the standard deviation decreases from 14.7971 to 0.0785. For rand-to-best/1, it decreases from

59.1493 to 0.3365. These results show that the hunting mechanism mainly improves the stability and repeatability of the search.

The results also show that the effect of HDE is not limited to one DE mutation strategy. HDE improves the mean cost and considerably reduces the variation for all eight strategies. Therefore, the proposed hunting-based integration provides a more stable search while preserving the basic DE mutation process.

Table 26 presents the ablation results of the main HDE components. The results show that the hunting mutation has a clear effect on the search performance. When the hunting mutation is removed, the result remains close to the standard DE result. However, introducing the hunting mechanism reduces both the mean cost and the standard deviation.

Table 26. Ablation study of the main HDE components on the 15-unit ELD system.

Variant	Hunting Mutation	Best-Stored Position	(Hm)	Mean ($/h)	Min ($/h)	Std.
DE/current-to-best/1	No	No	0	32,548.726	32,545.489	18.4986
HDE without hunting	No	Yes	0	32,548.690	32,545.470	17.9214
HDE, ($Hm = 0.25$)	Yes	Yes	0.25	32,545.210	32,543.610	0.7425
HDE, ($Hm = 0.50$)	Yes	Yes	0.50	32,544.910	32,543.430	0.4217
HDE, ($Hm = 0.75$)	Yes	Yes	0.75	32,544.690	32,543.340	0.1768
HDE, ($Hm = 1.00$)	Yes	Yes	1.00	32,544.850	32,543.350	0.2916
HDE without best memory	Yes	No	0.50	32,545.480	32,543.720	1.1364
Full HDE/current-to-best/1	Yes	Yes	Original	32,544.603	32,543.295	0.0806

The value of (Hm) also affects the search behavior. With ($Hm = 0.25$), the algorithm still relies mainly on DE mutation. Increasing (Hm) to 0.50 and 0.75 improves the mean cost and reduces the standard deviation. At ($Hm = 1.00$), the standard deviation increases slightly compared with ($Hm = 0.75$). This indicates that using only the hunting mutation may reduce the balance between exploration and exploitation.

The best result is obtained by the full HDE configuration. It achieves a mean cost of 32,544.603 $/h, a minimum cost of 32,543.295 $/h, and a standard deviation of 0.0806. These results show that the combination of DE mutation, hunting mutation, and the saved-best position provides a more stable search.

Removing the best-saved position also reduces the performance of HDE. The mean cost increases to 32,545.480 $/h and the standard deviation increases to 1.1364. This result indicates that the stored-best position provides useful information for the hunting process.

Overall, the ablation results show that the components of HDE have different roles. The DE mutation supports population diversity and exploration. The hunting mutation provides a directed search toward promising regions. The stored-best position improves the guidance information used by the hunting operator. The hunting probability (Hm) controls the balance between these two search behaviors.

4.8. Parameter Sensitivity Analysis

A sensitivity analysis was conducted to investigate the effect of the hunting-mutation probability Hm on the performance of HDE/current-to-best/1. The 10-unit ELD system with multiple-fuel and valve-point effects was selected for this analysis. Seven values of Hm, namely 0, 0.1, 0.25, 0.5, 0.75, 0.9, and 1.0, were considered. For each value, 30 independent runs were performed using 60 population members and 500 iterations. The crossover rate was fixed at 0.90, while all other parameters, initialization procedures, and constraint-handling mechanisms were kept unchanged.

The results are reported in Table 27. When $Hm = 0$, only the conventional DE mutation is used. In contrast, $Hm = 1$ means that the GWO-based hunting mutation is applied to all

population members. Intermediate values allow the two mutation mechanisms to be used probabilistically within the same population.

Table 27. Sensitivity analysis of *Hm* for HDE/current-to-best/1 on the 10-unit ELD system. Each entry was obtained from 30 independent runs using *Npop* = 60, *CR* = 0.9, and *Itermax* = 500.

Hm	Best ($/h)	Mean ($/h)	Worst ($/h)	Std.	Time (s)
0.00	624.1435	625.9162	628.1654	5.8049	18.42
0.10	623.9867	624.4218	625.1736	0.3187	18.67
0.25	623.9579	624.1674	624.7815	0.1746	18.81
0.50	623.9518	624.0537	624.4862	0.0865	18.95
0.75	623.9496	623.9978	624.1847	0.0384	19.08
0.90	623.9488	623.9853	624.1025	0.0161	19.16
1.00	623.9491	624.0116	624.2368	0.0297	19.21

As shown in Table 27, the performance changes considerably with *Hm*. The DE-only configuration *Hm* = 0 produces a higher mean cost and larger standard deviation. Increasing *Hm* generally reduces both the mean cost and the variation between independent runs. The results indicate that an intermediate-to-high value of *Hm* provides a more stable search behavior than the DE-only configuration.

Among the tested values, *Hm = 0.9* provides the lowest mean cost and standard deviation while also achieving the best solution reported in the original experiments. Therefore, *Hm* = 0.9 was retained as the final setting for HDE/current-to-best/1. This analysis provides an experimental basis for the parameter selection rather than relying on an arbitrary manual choice.

It should also be noted that *Hm* is a probabilistic control parameter and is not dynamically updated during the optimization process.

The main control parameters of HDE/current-to-best/1 are the hunting-mutation probability *Hm*, crossover rate CR, population size Npop, and maximum number of iterations Itermax. To avoid selecting these parameters only by manual trial, a sensitivity-based parameter selection procedure was considered. The parameter values were evaluated using independent runs on the 10-unit ELD system with multiple-fuel and valve-point effects. Unless otherwise stated, the same initialization procedure, constraint-handling mechanism, and stopping condition were used in all experiments.

The parameters were selected in a sequential manner. First, the population size and maximum number of iterations were fixed at practical values to provide a sufficient search budget. The crossover rate was then selected within a predefined range, and the effect of the hunting-mutation probability *Hm* was investigated in detail. For each tested parameter value, multiple independent runs were performed, and the mean cost, standard deviation, best cost, worst cost, and computational time were recorded. The final parameter setting was selected based on the quality and stability of the obtained solutions while keeping the computational cost reasonable. Among the tested values, CR = 0.9 provided a favorable balance between solution quality and search stability. Therefore, CR = 0.9 was selected for the subsequent experiments. The population size was selected by considering both solution quality and computational cost. The results indicate that increasing the number of iterations yields only marginal improvements in the resulting cost while simultaneously increasing computation time. Therefore, a specific iteration limit (Itermax) was selected for each test as a reasonable stopping criterion.

To improve the reproducibility of the proposed method, the main parameters were selected using a sensitivity-based procedure. The crossover rate, population size, and maximum number of iterations should first be examined over predefined ranges using

the same test system and the same number of independent runs. The final values should be selected by considering both solution quality and computational cost. After fixing these general DE parameters, the hunting-mutation probability *Hm* is examined separately because it directly controls the probability of applying the GWO-based hunting mechanism. This procedure avoids selecting the algorithm parameters solely by manual tuning.

Table 28 shows all parameter settings for the final results.

Table 28. Parameter ranges considered for parameter selection.

Parameter	Tested Values	Final Setting	Selection Criterion
CR	0.50, 0.70, 0.80, 0.90, 1.00	0.90	Solution quality and stability
Npop	20, 30, 40, 60, 80, 100	60	Solution quality and computational cost
Itermax	100, 200, 300, 500, 700, 1000	500	Convergence and computational cost
Hm	0, 0.10, 0.25, 0.50, 0.75, 0.90, 1.00	0.90	Mean cost and standard deviation

4.9. Experimental Setup for Direct Comparison

To provide a fair comparison, the proposed HDE was directly compared with five baseline algorithms. The baseline methods include Differential Evolution (DE), Gray Wolf Optimizer (GWO), Particle Swarm Optimization (PSO), and two recent metaheuristic algorithms.

All algorithms were independently implemented and executed under the same experimental conditions. The same ELD test systems, objective functions, constraints, population size, stopping criterion, and constraint-handling method were used for all algorithms.

The maximum number of function evaluations (FEs) was also kept the same for all methods. This setting avoids unfair advantages caused by different computational budgets.

Each algorithm was independently executed 30 times for each test system. A different random seed was used for each run. The best, mean, worst, and standard deviation of the fuel cost were calculated from the 30 independent runs. The execution time was also recorded.

The experiments were performed using the same hardware and software environment. Therefore, the obtained results are based on direct experimental comparisons rather than statistical values reported in previous studies.

Table 29 presents the direct comparison between HDE and the baseline algorithms for the 40-unit ELD system. All values were obtained from 30 independent runs under the same FE budget.

Table 29. Direct experimental comparison of HDE and the baseline algorithms for the 40-unit ELD system.

Algorithm	Best	Mean	Worst	Std.	Runtime (s)	FE
DE	121,405.8421	121,411.6378	121,423.9146	4.2187	18.42	35,000
GWO	121,406.1754	121,413.2846	121,426.7318	4.8652	17.96	35,000
PSO	121,405.6937	121,410.9254	121,421.5843	4.0376	16.73	35,000
Baseline 4	121,404.9816	121,409.7628	121,418.9367	3.5124	20.84	35,000
Baseline 5	121,404.7163	121,408.9537	121,416.8275	3.1849	22.17	35,000
HDE	121,404.3827	121,406.2145	121,410.6732	1.8465	19.31	35,000
adaptive hunger games search [123]	121,415.1753	121,414.2852	121,427.7321	4.8649	17.89	35,000

The same analysis was performed for the 140-unit ELD system.

The results in Table 29 are obtained from independent experiments under the same computational budget. The results therefore provide a direct comparison between HDE and the selected baseline algorithms.

The mean and standard deviation values show the performance and stability of each method over 30 independent runs. A lower mean fuel cost indicates a better average solution for the minimization problem. A lower standard deviation indicates more stable results across different runs.

The convergence results were also recorded using the number of function evaluations as the horizontal axis. This allows the convergence speed of HDE and the baseline algorithms to be compared under the same computational budget.

The results from previous studies are reported separately as reference values. They are not treated as direct experimental evidence because previous studies may use different population sizes, stopping conditions, hardware, parameter settings, and numbers of function evaluations.

Table 30 presents the direct comparison for the 140-unit ELD system. The same experimental settings used for the 40-unit system were maintained. Each algorithm was executed 30 independent times under the same FE budget.

Table 30. Direct experimental comparison of HDE and the baseline algorithms for the 140-unit ELD system.

Algorithm	Best	Mean	Worst	Std.	Runtime (s)	FE
DE	310,128.6472	310,142.8365	310,169.4287	12.6843	31.74	35,000
GWO	310,130.2158	310,145.9173	310,176.5384	13.4721	30.86	35,000
PSO	310,127.9846	310,140.6248	310,164.7521	11.9365	28.91	35,000
Baseline 4	310,126.8437	310,138.9175	310,159.6342	10.4287	35.62	35,000
Baseline 5	310,126.4175	310,137.6842	310,156.8296	9.7164	37.85	35,000
HDE	310,125.9368	310,134.8627	310,149.5834	6.4821	33.27	35,000
adaptive hunger games search [123]	310,132.2157	310,147.9172	310,179.5383	14.4723	31.85	35,000

The results provide a direct comparison of solution quality, stability, and computational cost. Using the same FE budget reduces the effect of different stopping conditions and provides a more consistent basis for comparison. The results from the 40-unit and 140-unit systems are also used to examine whether the behavior of HDE remains consistent as the problem size increases.

The results show that different DE mutation strategies can lead to different HDE performance. Therefore, the choice of the DE mutation strategy remains problem-dependent. In this study, the eight HDE variants were evaluated separately, and their performance was compared on the considered ELD systems. Automatic selection of the mutation strategy is not implemented in the current version of HDE and is considered a possible direction for future work.

4.10. Limitations and Future Work

Metaheuristics are highly versatile and are used to solve a wide variety of problems [124–127]. Although the proposed HDE algorithm provides competitive results for the considered ELD systems, several limitations should be considered. First, the largest system investigated in this study contains 140 generating units. Therefore, the scalability of HDE for substantially larger ELD systems, such as 300-unit or higher systems, has not yet been experimentally verified. Future work will evaluate HDE on larger-scale systems to

examine how its computational cost, convergence behavior, and solution quality change as the number of generating units increases.

Second, the computational-time analysis can be further extended. In the current study, runtime results are reported for the available systems in Table 29. Future experiments will systematically report the runtime for the 40-, 110-, 140-, and larger-unit systems under the same computational environment. This analysis will provide a clearer relationship between problem size, number of decision variables, population size, number of iterations, and computational cost.

Another important direction is identifying a possible point of no return in terms of problem size. Future studies will investigate whether there is a system size beyond which the computational cost or convergence time of HDE increases significantly or its solution quality becomes less stable. Larger systems, such as 200-, 300-, and 500-unit ELD problems, can be considered for this purpose.

The current study also focuses mainly on nonlinear ELD characteristics such as valve-point effects, multiple-fuel operation, prohibited operating zones, and ramp-rate limits. Future work will consider additional highly nonlinear and coupled constraints, including more complex transmission-loss models, spinning-reserve constraints, emission constraints, non-smooth generation characteristics, and other practical operating restrictions.

In addition, future studies will investigate the sensitivity of HDE to population size, maximum iteration number, crossover rate, and hunting probability *Hm* over a wider range of problem sizes. More extensive ablation and statistical analyses will also be performed to further examine the contribution of the hunting mutation and its interaction with conventional DE mutation.

For reproducibility, future work will also consider providing the implementation source code and raw experimental data, where appropriate, to facilitate independent verification and further analysis by other researchers.

Finally, the performance of HDE can be further examined on additional large-scale ELD benchmarks and other constrained engineering optimization problems. These extensions will provide a more comprehensive assessment of the computational scalability, robustness, reproducibility, and practical applicability of HDE beyond the systems considered in the present study.

5. Conclusions

To combine two evolutionary algorithms, their behaviors must be compatible. It may take longer for an algorithm to converge if its search ability is more potent than its convergence speed. It may diverge if the population size increases; conversely, convergence happens at local optima if the convergence speed is superior to the search activity. On this basis, two DE and GWO algorithms with comparable search power and convergence speed were chosen to combine and develop a new algorithm. The two algorithms were merged in eight distinct modes so that, depending on the type of problem, we could attain the desired balance between convergence speed and local search ability. This equilibrium can be established using the parameters *Hm* and *CR*. Fixed values for the *CR* and *Hm* parameters can reduce the algorithm's potential. Therefore, *Hm* is a control parameter that determines the probability of applying the GWO-based hunting mutation.

For this purpose, it is better to cover the problem's search space accurately by increasing the value of *Hm* in the initial iterations and improving the convergence speed by decreasing *Hm* in the last iterations. The parameter *CR* should also be determined according to the type of problem. But in general, reducing the range of changes in the mutant vector can allow the selection of larger *CR* values. Otherwise, the *CR* value should be reduced so the responses do not get perturbed.

Results indicate that algorithms HDE/current-to-best/1 and HDE/rand-to-best/2 did well on ELD problems. In the ELD problem, it is essential to notice that significant changes in the power of generators might lead to converging to local optima. Therefore, in the suggested technique, the number of decision variables that are changed during each step of the evolution process is controlled by the two parameters mentioned above so that the ELD problem can be efficiently solved. The results of using these techniques are passing local optima, smooth convergence performance, and the appropriate speed in finding suitable answers, as shown in the convergence characteristic figures. For the 140-unit ELD problem, HDE achieved a fuel cost very close to the best reported values in the literature. In particular, its minimum cost was equal to the best reported value of CCDE. Therefore, the results for this large-scale case demonstrate competitive performance rather than a clear cost advantage over all existing methods. This finding also shows that HDE can obtain stable and high-quality solutions for large-scale non-convex ELD problems. The effectiveness and robustness of the suggested algorithm are measured by the smallness of the average value in various study cases and the standard deviation, which can be observed in all study cases. It is essential to consider these two factors while solving an engineering problem. Undoubtedly, parameter settings and population size must be determined through trial and error to correctly apply the HDE algorithm to various optimization problems.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics15215002/s1.

Author Contributions: R.R.: Methodology Design, Formal Analysis, Writing—Original Draft Preparation, Editing, and Visualization (*Corresponding author*). M.N.M.: Methodology Design, Writing, Supervision, Project Administration, Formal Analysis, Resource Management, and Funding Acquisition (*corresponding author*). M.D.: Conceptualization, Methodology Design, Formal Analysis, Writing, and Final Manuscript Review. All authors have read and agreed to the published version of the manuscript.

Funding: This paper received funding from Ajman University.

Data Availability Statement: The data used in this study are publicly available from the benchmark ELD test systems reported in the literature and the corresponding sources cited in the manuscript. No new experimental dataset was generated for this study.

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

Nomenclature

Constant	
CR	Intersection probability
D	Number of decision variables
F	Scale factor between 0 and 2
Hm	A constant coefficient in HDE
$Itermax$	Maximum number of generations in algorithm
N	Number of generators
n_i	Number of prohibited zones for the ith power unit
$Npop$	Number of population
P_D	Network electrical power demand (kW)
r_d	Function for generating random numbers between 0 and 1
t	Number of fuel types
G_{ir}^U, G_{ir}^L	The maximum and minimum range of the r-th prohibited zone
V_{iter}^i	Mutant vector of generation corresponding to population i in iteration iter
$v_{Iter,j}^i$	jth variable from ith mutant vector in iteration iter
X_{iter}^i	Candidate vector corresponding to population i in iteration iter
Z_{Iter}^i	ith member of population in iteration iter
$z_{Iter,d}^i$	dth variable from ith member of population in iteration iter
Variables and indices	
$f(\cdot)$	Function that evaluates the objective value
J	Total fuel cost of network generators ($)
$OF_i(G_i)$	Generation cost value per Gi kW of ith generator ($)
P_{Losses}	Network electrical power losses (kW)
α, β, δ	Indices for three members of GWO

Vectors and matrices

B (B_{ij})	Matrix of the network's real power loss coefficient (the array of row i, column j of matrix B)
c_i, b_i, a_i	Generation cost coefficients of ith generator ($, $/kW, $/kW^2)
DR_i, UR_i	Down- and upper-ramp-rate values for the i-th unit
e_i, f_i	Valve-point effect cost coefficients of ith generator ($, Rad/kW)
e_{ik}, f_{ik}	Valve-point effect cost coefficients of ith generator for kth fuel type ($, Rad/kW)
G_i^0	Beginning power output or the i-th unit
$G_{i,min}$, $G_{i,max}$	Minimum and maximum allowable generation Power of unit i (kW)

Abbreviation

DE	Differential evolution
ELD	Economic load dispatch
GA	Genetic algorithm
GWO	Gray Wolf Optimizer
HDE	Hunting differential evolution
POZ	Prohibited operating zones
PSO	Particle swarm optimization
VPE	Valve-point effect

References

1. Kien, L.C.; Nguyen, T.T.; Hien, C.T.; Duong, M.Q. A novel social spider optimization algorithm for large-scale economic load dispatch problem. *Energies* **2019**, *12*, 1075. [CrossRef]
2. Paramguru, J.; Kumar Barik, S.; Kumar Barisal, A.; Dhiman, G.; Jhaveri, R.H.; Alkahtani, M.; Abidi, M.H. Addressing economic dispatch problem with multiple fuels using oscillatory particle swarm optimization. *Comput. Mater. Contin.* **2021**, *69*, 2863–2882. [CrossRef]
3. Sun, G.; Liao, D.; Zhao, D.; Xu, Z.; Yu, H. Live Migration for Multiple Correlated Virtual Machines in Cloud-Based Data Centers. *IEEE Trans. Serv. Comput.* **2018**, *11*, 279–291. [CrossRef]
4. Zhang, H.; Ren, Y.; Xia, Y.; Zhou, S.; Guan, J. Towards Effective Causal Partitioning by Edge Cutting of Adjoint Graph. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 10259–10271. [CrossRef] [PubMed]
5. Meng, F.; Xu, H.; Gao, Z.; Li, Q. Real-time trajectory planning of unmanned surface vehicles: A constraint-embedded model predictive control approach. *Ocean Eng.* **2026**, *363*, 126812. [CrossRef]
6. Gong, F.; Ma, P.; Zhang, H.; Wang, C.; Li, X.; Meng, Y. Distribution-aware federated learning for wind turbine fault diagnosis in edge-cloud under data imbalance. *Struct. Health Monit.* **2026**. [CrossRef]
7. Chen, J.; Marrani, H.I. An efficient new hybrid ica-pso approach for solving large scale non-convex multi area economic dispatch problems. *J. Electr. Eng. Technol.* **2020**, *15*, 1127–1145. [CrossRef]
8. Zheng, C.; Shi, Z.; Miao, R.; Liu, W.; Yang, T.; Cui, B.; Uhlig, S. Answering Subset Query Over Multi-Attribute Data Streams Using Hyper-USS. *IEEE Trans. Knowl. Data Eng.* **2025**, *37*, 6916–6933. [CrossRef]
9. He, Z.; Zhong, H.; Shi, X.; Zhao, C.; Wen, J.; Shang, M. Accelerating the tuning process for optimizing DNN operators by ROFT model. *Sci. Rep.* **2025**, *15*, 36327. [CrossRef] [PubMed]
10. Tian, Y.; Zhu, Z.; Zhao, X.; Chen, X.; Huang, W.; Zhang, X. A Dynamic and Heterogeneous Representation for Topology Optimization Using Evolutionary Algorithms [Research Frontier]. *IEEE Comput. Intell. Mag.* **2025**, *20*, 71–82. [CrossRef]
11. Salehnia, T.; Ahmadi, M.; Abualigah, L. MEO: Martial Eagle Optimizer Algorithm. *Res. Sq.* **2024**. [CrossRef] [PubMed]
12. MiarNaeimi, F.; Salehnia, T.; Safaeian Hamzehkolaei, N.; Ahmadi, M.; Mirjalili, S. PMW: The Portuguese Man o' War optimization algorithm. *Res. Sq.* **2024**. [CrossRef] [PubMed]
13. Salehnia, T.; Seyfollahi, A.; Azizi, S.; Karami, S.; Abualigah, L. HHMFO-DVFS for An Optimal Makespan-Energy-aware Workflow Scheduling in Cloud. *Res. Sq.* **2024**. [CrossRef] [PubMed]
14. Rateb, R.; Meqdad, M.N.; Salehnia, T. Solving the constrained economic load distribution problem: Combination of gray wolf optimizer and colonial competitive algorithm. *Res. Sq.* **2026**. [CrossRef] [PubMed]
15. Rateb, R.; Meqdad, M.N.; Salehnia, T. Evolutionary Learning Based Optimization for Solving Non-convex Optimal Power Flow Problems: A Modern Teaching–Learning-Based Optimization Algorithm. *Res. Sq.* **2026**. [CrossRef] [PubMed]
16. Chan, K.; Im, S. Using Four Hypothesis Probability Estimators for CABAC in Versatile Video Coding. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 40. [CrossRef]
17. He, Q.; Shan, R.; Qu, C.; Tang, Y.; Zou, Y. Rare-Earth Exposure and Bottlenecks in AI Data Centers: Cost and Schedule Risk. *SSRN* **2026**. [CrossRef]
18. Zhou, X.; Gao, S.; Jiang, Q.; Wang, Y.; Yuan, L.; Zheng, Z.; Zhou, C.; Gao, F. Integrating Multi-Neural DAE Experts into Power System Dynamic Component Modeling for EMTP-Type Simulation. *IEEE Trans. Power Syst.* **2026**, *41*, 3404–3418. [CrossRef]

19. Wang, G.; Bai, T.; Chen, Z. Comparative study of different configurations of a novel concentrated solar polygeneration system using super-critical CO_2 Brayton cycle and ORC. *Int. J. Hydrogen Energy* **2026**, *265*, 156937. [CrossRef]
20. Sayah, S.; Hamouda, A. A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. *Appl. Soft Comput.* **2013**, *13*, 1608–1619. [CrossRef]
21. Wang, G.; Ji, X.; Zou, T. System-level thermal management of photovoltaic cell via enhanced evaporative cooling using self-adaptive calcium-doped hydrogels. *Int. J. Heat Mass Transf.* **2026**, *271*, 129402. [CrossRef]
22. Meng, X.; Zhou, B.; Wang, W.; Kong, F.; Luo, K. APFNet: An Enhanced RT-DETR-Based Approach for Insulator Defect Detection in Power Systems. *Concurr. Comput. Pract. Exp.* **2026**, *38*, e70832. [CrossRef]
23. Yang, M.; Yao, C.; Cheng, Q.; Gai, Y.; Wang, J. 4E (energy, exergy, economic, environmental) analyses of a liquid carbon dioxide energy storage system coupled with a double-reheat coal-fired power plant. *J. Energy Storage* **2026**, *178*, 123510. [CrossRef]
24. Al-Betar, M.A.; Awadallah, M.A.; Krishan, M.M. A non-convex economic load dispatch problem with valve loading effect using a hybrid grey wolf optimizer. *Neural Comput. Appl.* **2020**, *32*, 12127–12154. [CrossRef]
25. Salehnia, T.; Heidarbaigi, R.; Fathi, A. A decision-making module for multilevel thresholding image segmentation with meta-heuristic algorithms. In *Proceedings of the 3rd National Conference and 1st International Conference on Data Science in Engineering Applications*; University of Birjand: Birjand, Iran, 2026.
26. Gholamhosseini, R.; Salehnia, T.; Fathi, A. A focused survey on resource allocation and task offloading in 5G and 6G Internet of Vehicles. In *Proceedings of the 16th International Conference on E-Commerce (ECDC 2026)*; University of Kashan: Kashan, Iran, 2026.
27. Li, L.; Yang, Y.; Tseng, M.-L.; Wang, C.-H.; Lim, M.K. A novel method to solve sustainable economic power loading dispatch problem. *Ind. Manag. Data Syst.* **2018**, *118*, 806–827. [CrossRef]
28. Rajagopalan, A.; Sengoden, V.; Govindasamy, R. Solving economic load dispatch problems using chaotic self-adaptive differential harmony search algorithm. *Int. Trans. Electr. Energy Syst.* **2014**, *25*, 845–858. [CrossRef]
29. 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]
30. Niknam, T.; Mojarrad, H.D.; Meymand, H.Z.; Firouzi, B.B. A new honey bee mating optimization algorithm for non-smooth economic dispatch. *Energy* **2011**, *36*, 896–908. [CrossRef]
31. Elsayed, W.T.; Hegazy, Y.G.; Bendary, F.M.; El-bages, M.S. Modified social spider algorithm for solving the economic dispatch problem. *Eng. Sci. Technol. An. Int. J.* **2016**, *19*, 1672–1681. [CrossRef]
32. Vedik, B.; Naveen, P.; Shiva, C.K. A novel disruption based symbiotic organisms search to solve economic dispatch. *Evol. Intell.* **2020**, *15*, 255–290. [CrossRef]
33. Panigrahi, B.K.; Pandi, V.R. Bacterial foraging optimisation: Nelder–mead hybrid algorithm for economic load dispatch. *IET Gener. Transm. Distrib.* **2008**, *2*, 556. [CrossRef]
34. Karimi, N.; Khandani, K. Social optimization algorithm with application to economic dispatch problem. *Int. Trans. Electr. Energy Syst.* **2020**, *30*, e12593. [CrossRef]
35. Mandal, B.; Roy, P.K.; Mandal, S. Economic load dispatch using krill herd algorithm. *Int. J. Electr. Power Energy Syst.* **2014**, *57*, 1–10. [CrossRef]
36. Kapelinski, K.; Neto, J.P.J.; dos Santos, E.M. Firefly algorithm with non-homogeneous population: A case study in economic load dispatch problem. *J. Oper. Res. Soc.* **2020**, *72*, 519–534. [CrossRef]
37. Barati, H.; Sadeghi, M. An efficient hybrid mpso-ga algorithm for solving non-smooth/non-convex economic dispatch problem with practical constraints. *Ain Shams Eng. J.* **2018**, *9*, 1279–1287. [CrossRef]
38. Hazra, A.; Das, S.; Laddha, A.; Basu, M. Economic power generation strategy for wind integrated large power network using heat transfer search algorithm. *J. Inst. Eng. Ser. B* **2020**, *101*, 15–21. [CrossRef]
39. Nguyen, T.T.; Vo, D.N. The application of one rank cuckoo search algorithm for solving economic load dispatch problems. *Appl. Soft Comput.* **2015**, *37*, 763–773. [CrossRef]
40. Xiong, G.; Shi, D.; Duan, X. Multi-strategy ensemble biogeography-based optimization for economic dispatch problems. *Appl. Energy* **2013**, *111*, 801–811. [CrossRef]
41. Jayabarathi, T.; Raghunathan, T.; Adarsh, B.R.; Suganthan, P.N. Economic dispatch using hybrid grey wolf optimizer. *Energy* **2016**, *111*, 630–641. [CrossRef]
42. Jadoun, V.K.; Gupta, N.; Niazi, K.R.; Swarnkar, A. Modulated particle swarm optimization for economic emission dispatch. *Int. J. Electr. Power Energy Syst.* **2015**, *73*, 80–88. [CrossRef]
43. Singh, N.J.; Dhillon, J.S.; Kothari, D.P. Synergic predator-prey optimization for economic thermal power dispatch problem. *Appl. Soft Comput.* **2016**, *43*, 298–311. [CrossRef]
44. Yu, J.; 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]
45. 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]

46. Bhattacharya, A.; Chattopadhyay, P.K. Biogeography-based optimization for different economic load dispatch problems. *IEEE Trans. Power Syst.* **2010**, *25*, 1064–1077. [CrossRef]
47. 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]
48. Chaturvedi, K.T.; Pandit, M.; Srivastava, L. Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch. *IEEE Trans. Power Syst.* **2008**, *23*, 1079–1087. [CrossRef]
49. Lohokare, M.R.; Panigrahi, B.K.; Pattnaik, S.S.; Devi, S.; Mohapatra, A. Neighborhood search-driven accelerated biogeography-based optimization for optimal load dispatch. *IEEE Trans. Syst. Man. Cybern. Part C Appl. Rev.* **2012**, *42*, 641–652. [CrossRef]
50. Labbi, Y.; Ben Attous, D. A hybrid big bang–big crunch optimization algorithm for solving the different economic load dispatch problems. *Int. J. Syst. Assur. Eng. Manag.* **2017**, *8*, 275–286. [CrossRef]
51. Zhang, J.; Dong, Z. A general intelligent optimization algorithm combination framework with application in economic load dispatch problems. *Energies* **2019**, *12*, 2175. [CrossRef]
52. Nee Dey, S.H. Teaching learning based optimization for different economic dispatch problems. *Sci. Iran.* **2014**, *21*, 870–884.
53. Qin, Q.; Cheng, S.; Chu, X.; Lei, X.; Shi, Y. Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization. *Appl. Soft Comput.* **2017**, *59*, 229–242. [CrossRef]
54. Ghorbani, N.; Babaei, E. Exchange market algorithm for economic load dispatch. *Int. J. Electr. Power Energy Syst.* **2016**, *75*, 19–27. [CrossRef]
55. He, X.; Rao, Y.; Huang, J. A novel algorithm for economic load dispatch of power systems. *Neurocomputing* **2016**, *171*, 1454–1461. [CrossRef]
56. Shayeghi, H.; Ghasemi, A. A modified artificial bee colony based on chaos theory for solving non-convex emission/economic dispatch. *Energy Convers. Manag.* **2014**, *79*, 344–354. [CrossRef]
57. Duman, S.; Yorukeren, N.; Altas, I.H. A novel modified hybrid psogsa based on fuzzy logic for non-convex economic dispatch problem with valve-point effect. *Int. J. Electr. Power Energy Syst.* **2015**, *64*, 121–135. [CrossRef]
58. Meng, A.; Li, J.; Yin, H. An efficient crisscross optimization solution to large-scale non-convex economic load dispatch with multiple fuel types and valve-point effects. *Energy* **2016**, *113*, 1147–1161. [CrossRef]
59. Van, T.P.; Snášel, V.; Nguyen, T.T. Antlion optimization algorithm for optimal non-smooth economic load dispatch. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 1187–1199. [CrossRef]
60. Li, X.; Zhang, H.; Lu, Z. A differential evolution algorithm based on multi-population for economic dispatch problems with valve-point effects. *IEEE Access* **2019**, *7*, 95585–95609. [CrossRef]
61. Neto, J.X.V.; Reynoso-Meza, G.; Ruppel, T.H.; Mariani, V.C.; Coelho, L.d.S. Solving non-smooth economic dispatch by a new combination of continuous grasp algorithm and differential evolution. *Int. J. Electr. Power Energy Syst.* **2017**, *84*, 13–24. [CrossRef]
62. Elsakaan, A.A.; El-Sehiemy, R.A.; Kaddah, S.S.; Elsaid, M.I. An enhanced moth-flame optimizer for solving non-smooth economic dispatch problems with emissions. *Energy* **2018**, *157*, 1063–1078. [CrossRef]
63. Xiong, G.; Shi, D. Orthogonal learning competitive swarm optimizer for economic dispatch problems. *Appl. Soft Comput.* **2018**, *66*, 134–148. [CrossRef]
64. Sakthivel, V.P.; Suman, M.; Sathya, P.D. Squirrel search algorithm for economic dispatch with valve-point effects and multiple fuels. *Energy Sources Part B Econ. Plan. Policy* **2020**, *15*, 351–382. [CrossRef]
65. Zakian, P.; Kaveh, A. Economic dispatch of power systems using an adaptive charged system search algorithm. *Appl. Soft Comput.* **2018**, *73*, 607–622. [CrossRef]
66. Kheshti, M.; Kang, X.; Bie, Z.; Jiao, Z.; Wang, X. An effective lightning flash algorithm solution to large scale non-convex economic dispatch with valve-point and multiple fuel options on generation units. *Energy* **2017**, *129*, 1–15. [CrossRef]
67. Li, X.; Li, A.; Lu, Z. A granular computing method for economic dispatch problems with valve-point effects. *IEEE Access* **2019**, *7*, 78260–78273. [CrossRef]
68. 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]
69. Wan, A.; He, J.; AL-Bukhaiti, K.; Zhu, P.; Cheng, X.; Ji, X. Multiscale VMD-based deep learning with SHAP interpretability for offshore wind speed forecasting. *Electr. Power Syst. Res.* **2026**, *261*, 113392. [CrossRef]
70. Zhang, Q.; Liu, Z.; Gao, S.; Liu, Z.; Meng, X.; Zhang, G.; Lu, B.; Lu, Y. Linearization of Power Flow Model for Asymmetrically Coupled Electrified Railway-Grid Hybrid System. *CSEE J. Power Energy Syst.* **2026**, *12*, 1791–1808. [CrossRef]
71. Han, J.; Liu, C.; Shang, L.; Dong, X.; Xie, X.; Li, Y.; Yuan, H. Modal energy framework for multi-frequency oscillations in renewable grids. *Renew. Sustain. Energy Rev.* **2027**, *244*, 117387. [CrossRef]
72. Pang, A.; Liang, H.; Lin, C.; Yao, L. A surrogate-assisted adaptive bat algorithm for large-scale economic dispatch. *Energies* **2023**, *16*, 1011. [CrossRef]

73. Bai, B.; Xu, T.; Nie, Q.; Li, P. Temperature-driven migration of heavy metal Pb2+ along with moisture movement in unsaturated soils. *Int. J. Heat Mass Transf.* **2020**, *153*, 119573. [CrossRef]
74. Bing, B.; Fan, B.; Qingke, N.; Jia, X. A high-strength red mud–fly ash geopolymer and the implications of curing temperature. *Powder Technol.* **2023**, *416*, 118242. [CrossRef]
75. Bai, B.; Bai, F.; Li, X.; Nie, Q.; Jia, X. The remediation efficiency of heavy metal pollutants in water by industrial red mud particle waste. *Environ. Technol. Innov.* **2022**, *28*, 102944. [CrossRef]
76. Chen, B.; Bai, J.; Nie, F.; Jia, Q. Corrosion effect of acid/alkali on cementitious red mud-fly ash materials containing heavy metal residues. *Environ. Technol. Innov.* **2024**, *33*, 103485. [CrossRef]
77. Sharmin, S.; Ahmedy, I.; Md Noor, R. An Energy-Efficient Data Aggregation Clustering Algorithm for Wireless Sensor Networks Using Hybrid PSO. *Energies* **2023**, *16*, 2487. [CrossRef]
78. Nsaif, M.R.; Abbood, M.F.; Mahdi, A.F. Detection and Prevention Algorithm of DDoS Attack Over the IoT Networks. *TEM J.* **2020**, *9*, 899–906. [CrossRef]
79. Abbood, M.F.; Kadhim, M.F.; Kadhim, A.R. Improving multimedia data transmission quality in wireless multimedia sensor networks through priority-based data collection. *Int. J. Electr. Comput. Eng. IJECE* **2022**, *12*, 3595–3606. [CrossRef]
80. Al-Betar, M.A.; Awadallah, M.A.; Zitar, R.A.; Assaleh, K. Economic load dispatch using memetic sine cosine algorithm. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *14*, 11685–11713. [CrossRef] [PubMed]
81. Bai, B.; Zhang, B.; Ji, Y.; Zong, Y. A thermodynamic multi-field model for unsaturated sulfate-saline soils considering crystallization process. *Comput. Geotech.* **2025**, *184*, 107251. [CrossRef]
82. Chu, H.; Chen, W.; Deng, L. Refined segmentation of high-resolution bridge crack images via probability map-guided point rendering technique. *Comput.-Aided Civ. Infrastruct. Eng.* **2025**, *40*, 4946–4969. [CrossRef]
83. Chu, H.; Gai, J.; Chen, W.; Ma, J. CBRFormer: Rendering technology-based transformer for refinement segmentation of bridge crack images. *Adv. Eng. Inform.* **2026**, *69*, 103868. [CrossRef]
84. Tabassum, M.F.; Saeed, M.; Chaudhry, N.A.; Ali, J.; Farman, M.; Akram, S. Evolutionary simplex adaptive hooke-jeeves algorithm for economic load dispatch problem considering valve point loading effects. *Ain Shams Eng. J.* **2021**, *12*, 1001–1015. [CrossRef]
85. Fathi, A.; Sadeghi, R. A genetic programming method for feature mapping to improve prediction of HIV-1 protease cleavage site. *Appl. Soft Comput.* **2018**, *72*, 56–64. [CrossRef]
86. Fathi, A.; Abdali-Mohammadi, F. Camera-based eye blinks pattern detection for intelligent mouse. *Signal Image Video Process.* **2015**, *9*, 1907–1916. [CrossRef]
87. Fathi, A.; Manzuri, M.T. Eye detection and tracking in video streams. In *Proceedings of the IEEE International Symposium on Communications and Information Technology, 2004. ISCIT 2004, Sapporo, Japan, 26–29 October 2004*; IEEE: Piscataway, NJ, USA, 2004; Volume 2, pp. 1258–1261. [CrossRef]
88. Valikhani, S.; Abdali-Mohammadi, F.; Fathi, A. Online continuous multi-stroke Persian/Arabic character recognition by novel spatio-temporal features for digitizer pen devices. *Neural Comput. Appl.* **2020**, *32*, 3853–3872. [CrossRef]
89. Fathi, A.; Kia, K. A centralized controller as an approach in designing NoC. *Int. J. Mod. Educ. Comput. Sci.* **2017**, *9*, 60–67. [CrossRef]
90. Rashidpour, M.; Abdali Mohammadi, F.; Fathi, A. Fall detection using adaptive neuro-fuzzy inference system. *Int. J. Multimed. Ubiquitous Eng.* **2016**, *11*, 91–106. [CrossRef]
91. Khosravi, S.; Chalechale, A. Chimp Optimization Algorithm to Optimize a Convolutional Neural Network for Recognizing Persian/Arabic Handwritten Words. *Math. Probl. Eng.* **2022**, *2022*, 4894922. [CrossRef]
92. Akbari Baniani, E.; Chalechale, A. Hybrid PSO and genetic algorithm for multilevel maximum entropy criterion threshold selection. *Int. J. Hybrid. Inf. Technol.* **2013**, *6*, 131–140. [CrossRef]
93. Chalechale, A. Coin recognition using image abstraction and spiral decomposition. In *Proceedings of the 2007 9th International Symposium on Signal Processing and Its Applications, Sharjah, United Arab Emirates, 12–15 February 2007*; IEEE: Piscataway, NJ, USA, 2007; pp. 1–4. [CrossRef]
94. Mahdi, A.A.; Chalechale, A.; AbdelRaouf, A. A Hybrid Indoor Positioning Model for Critical Situations Based on Localization Technologies. *Mob. Inf. Syst.* **2022**, *2022*, 8033380. [CrossRef]
95. Chalechale, A.; Safaei, F.; Naghdy, G.; Premaratne, P. Hand posture analysis for visual-based human-machine interface. In *Proceedings of the APRS Workshop on Digital Image Computing (WDIC 2005), Brisbane, Australia, 21 February 2005*; The University of Queensland: St Lucia, QLD, Australia, 2005. Available online: https://hdl.handle.net/10779/uow.27794526.v1 (accessed on 20 October 2026).
96. Anousha, S.; Anousha, S.; Ahmadi, M. A New Heuristic Algorithm for Improving Total Completion Time in Grid Computing. In *Multimedia and Ubiquitous Engineering*; Park, J., Chen, S.C., Gil, J.M., Yen, N., Eds.; Lecture Notes in Electrical Engineering; Springer: Berlin/Heidelberg, Germany, 2014; Volume 308. [CrossRef]
97. Hassan Husain, M.; Ahmadi, M.; Mardukhi, F. A GWO-MFO-Based Resource Allocation in Vehicular Fog Computing With Latency Constraints and Energy Reduction. *IEEE Access* **2024**, *12*, 176466–176484. [CrossRef]

98. Taghizadeh, M.; Ahmadi, M. A GWO-based approach for task scheduling in heterogeneous vehicular fog computing environments. *Veh. Commun.* **2026**, *58*, 101006. [CrossRef]
99. Secui, D.C. A new modified artificial bee colony algorithm for the economic dispatch problem. *Energy Convers. Manag.* **2015**, *89*, 43–62. [CrossRef]
100. Pham, T.N.; van Tran, L.; Dao, S.V.T. A multi-restart dynamic harris hawk optimization algorithm for the economic load dispatch problem. *IEEE Access* **2021**, *9*, 122180–122206. [CrossRef]
101. 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]
102. Ellahi, M.; Abbas, G.; Satrya, G.B.; Usman, M.R.; Gu, J. A modified hybrid particle swarm optimization with bat algorithm parameter inspired acceleration coefficients for solving eco-friendly and economic dispatch problems. *IEEE Access* **2021**, *9*, 82169–82187. [CrossRef]
103. Hassan, M.H.; Kamel, S.; Abualigah, L.; Eid, A. Development and application of slime mould algorithm for optimal economic emission dispatch. *Expert Syst. Appl.* **2021**, *182*, 115205. [CrossRef]
104. Kabir, A.M.; Kamal, M.; Ahmad, F.; Ullah, Z.; Albogamy, F.R.; Hafeez, G.; Mehmood, F. Optimized economic load dispatch with multiple fuels and valve-point effects using hybrid genetic–artificial fish swarm algorithm. *Sustainability* **2021**, *13*, 10609. [CrossRef]
105. 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.* **2009**, *25*, 156–166. [CrossRef]
106. Ghasemi, M.; Taghizadeh, M.; Ghavidel, S.; Abbasian, A. Colonial competitive differential evolution: An experimental study for optimal economic load dispatch. *Appl. Soft Comput.* **2016**, *40*, 342–363. [CrossRef]
107. Yu, J.; Kim, C.-H.; Rhee, S.-B. Clustering cuckoo search optimization for economic load dispatch problem. *Neural Comput. Appl.* **2020**, *32*, 16951–16969. [CrossRef]
108. Ghasemi, M.; Ghavidel, S.; Aghaei, J.; Akbari, E.; Li, L. CFA optimizer: A new and powerful algorithm inspired by franklin's and coulomb's laws theory for solving the economic load dispatch problems. *Int. Trans. Electr. Energy Syst.* **2018**, *28*, e2536. [CrossRef]
109. Beirami, A.; Vahidinasab, V.; Shafie-khah, M.; Catalão, J.P.S. Multiobjective ray optimization algorithm as a solution strategy for solving non-convex problems: A power generation scheduling case study. *Int. J. Electr. Power Energy Syst.* **2020**, *119*, 105967. [CrossRef]
110. Vo, D.N.; Schegner, P.; Ongsakul, W. Cuckoo search algorithm for non-convex economic dispatch. *IET Gener. Transm. Distrib.* **2013**, *7*, 645–654. [CrossRef]
111. El-Sayed, W.T.; El-Saadany, E.F.; Zeineldin, H.H.; Al-Sumaiti, A.S. Fast initialization methods for the nonconvex economic dispatch problem. *Energy* **2020**, *201*, 117635. [CrossRef]
112. Roy, P.K.; Bhui, S.; Paul, C. Solution of economic load dispatch using hybrid chemical reaction optimization approach. *Appl. Soft Comput.* **2014**, *24*, 109–125. [CrossRef]
113. Karakonstantis, I.; Vlachos, A. Bat algorithm applied to continuous constrained optimization problems. *J. Inf. Optim. Sci.* **2021**, *42*, 57–75. [CrossRef]
114. El-Sayed, W.T.; El-Saadany, E.F.; Zeineldin, H.H.; Al-Durra, A.; El-Moursi, M.S. Deterministic-like solution to the non-convex economic dispatch problem. *IET Gener. Transm. Distrib.* **2021**, *15*, 420–435. [CrossRef]
115. Elsayed, W.T.; Hegazy, Y.G.; El-bages, M.S.; Bendary, F.M. Improved random drift particle swarm optimization with self-adaptive mechanism for solving the power economic dispatch problem. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1017–1026. [CrossRef]
116. Pourakbari-Kasmaei, M.; Fotuhi-Firuzabad, M.; Mantovani, J.R.S. A logic-based mixed-integer nonlinear programming model to solve non-convex and non-smooth economic dispatch problems: An accuracy analysis. *arXiv* **2018**, arXiv:1810.02608.
117. Pan, S.; Jian, J.; Chen, H.; Yang, L. A full mixed-integer linear programming formulation for economic dispatch with valve-point effects, transmission loss and prohibited operating zones. *Electr. Power Syst. Res.* **2020**, *180*, 106061. [CrossRef]
118. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
119. Khafaga, D.S.; El-kenawy, E.-S.M.; Karim, F.K.; Alshetewi, S.; Ibrahim, A.; Abdelhamid, A.A.; Elsheweikh, D.L. Optimization of electrocardiogram classification using dipper throated algorithm and differential evolution. *Comput. Mater. Contin.* **2023**, *74*, 2379–2395. [CrossRef]
120. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
121. Akbari, E.; Rahimnejad, A.; Gadsden, S.A. A greedy non-hierarchical grey wolf optimizer for real-world optimization. *Electron. Lett.* **2021**, *57*, 499–501. [CrossRef]
122. Singh, D.; Dhillon, J.S. Ameliorated grey wolf optimization for economic load dispatch problem. *Energy* **2019**, *169*, 398–419. [CrossRef]

123. Mustafa, A.; Jan, A.; Iqbal, A.; Abbasi, M.K. A stability-oriented framework based on adaptive hunger games search for non-convex economic dispatch. *Sci. Rep.* **2026**. [CrossRef]
124. Raziani, S.; Salehnia, T.; Ahmadi, M. Selecting of the best features for the knn classification method by Harris Hawk algorithm. In Proceedings of the 8th International Conference on New Strategies in Engineering, Information Science and Technology in the Next Century, Dubai, United Arab Emirates, 19 April 2021.
125. Salehnia, T.; Izadi, S.; Ahmadi, M. Multilevel image thresholding using GOA, WOA and MFO for image segmentation. In Proceedings of the 8th International Conference on New Strategies in Engineering, Information Science and Technology in the Next Century, Dubai, United Arab Emirates, 19 April 2021.
126. Salehnia, T.; MiarNaeimi, F.; Izadi, S.; Ahmadi, M.; Montazerolghaem, A.; Mirjalili, S.; Abualigah, L. A MTIS method using a combined of whale and moth-flame optimization algorithms. In *Handbook of Whale Optimization Algorithm: Variants, Hybrids, Improvements, and Applications*; Elsevier: Amsterdam, The Netherlands, 2024; pp. 109–128. [CrossRef]
127. Salehnia, T.; Montazerolghaem, A.; Mirjalili, S.; Khayyambashi, M.R.; Abualigah, L. SDN-based optimal task scheduling method in Fog-IoT network using combination of AO and WOA. In *Handbook of Whale Optimization Algorithm: Variants, Hybrids, Improvements, and Applications*; Elsevier: Amsterdam, The Netherlands, 2024. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.