Economic dispatch using metaheuristics: algorithms, problems, and solutions

Thammarsat Visutarrom, Tsung-Che Chiang
Department of Computer Science and Information Engineering, National Taiwan Normal University, Taipei, Taiwan
thammarsat@gmail.com, tcchiang@ieee.org

Abstract

Economic dispatch (ED) has received considerable interest in the field of energy management and optimization. The problem aims to determine the most cost-effective power allocation strategy that satisfies the power demand and all physical constraints of the power system. To solve this problem, we propose an algorithm based on differential evolution and adopt a hybrid mutation strategy, a linear population size reduction mechanism, and an improved single-unit repair mechanism. Experimental results confirmed that these mechanisms are useful for performance improvement. The proposed algorithm (L-HMDE) showed good performance when compared with more than 90 algorithms in solving 22 test cases. It could provide high-quality solutions stably and efficiently. In addition to designing a good algorithm, we present a review of over 100 papers and highlight their algorithm features. We also provide a comprehensive collection of test cases in the literature. Through careful examination and verification, data coefficients of these test cases and solutions to them are included in this paper as a useful reference for researchers who are interested in this problem.

Keywords: Economic dispatch, Differential evolution, Hybrid mutation strategy, Linear population size reduction, Constraint handling

---

1. The Economic Dispatch Problem

Energy management has garnered significant attention in contemporary times, reflecting an increasing interest in energy sustainability [1]. Multiple research domains are now acknowledging the energy management as a pivotal factor in their analysis and resolution [2]. In the domains of industrial and power plant operation, effective power allocation strategies are crucial to enhance a power system to reach its full potential with the minimal operating cost. One of the fundamental challenges in the field of power management is the economic dispatch (ED) problem. It is a constrained continuous optimization problem that aims to allocate power output of generators to meet the power demand and minimize the generation cost.

In the ED problem, a power system with NG generators needs to generate power output while satisfying operational constraints. The objective function F_c is mathematically formulated as a convex (1) or a nonconvex (2) quadratic function, which presents the operating cost incurred by the consumption of fossil fuel in the power system. In the convex objective function, the variable P_j denotes the power output, and a_j, b_j and, c_j are the cost coefficients of the j^{th} generator. The landscape of the solution space is a smooth curve when the objective function is convex [3].

$$min\sum_{j}^{NG} F_c(P_j) = \sum_{j=1}^{NG}\left(a_j + b_jP_j + c_jP_j^2\right) \tag{1}$$

Nevertheless, the convex function might not represent the nature of all power systems. The objective function of several ED test cases introduces a sine function, which represents the valve-point effect of the power system, as shown in (2) [4]. The variables d_j and e_j are the cost coefficients of the valve-point effect, and $P_j^{\min}$ is the minimal power output that the j^{th} generator must generate. The non-convexity changes the landscape from a smooth curve to a rugged curve with multiple local minima. Fig. 1 illustrates the landscape, where the horizontal axes are the power output of two different power generators, and the vertical axis represents the total operating cost.

$$min\sum_{j}^{NG} F_c(P_j) = \sum_{j=1}^{NG}\left(a_j + b_jP_j + c_jP_j^2 + \left|d_j\left(sin\left(e_j(P_j^{\min} - P_j)\right)\right)\right|\right) \tag{2}$$

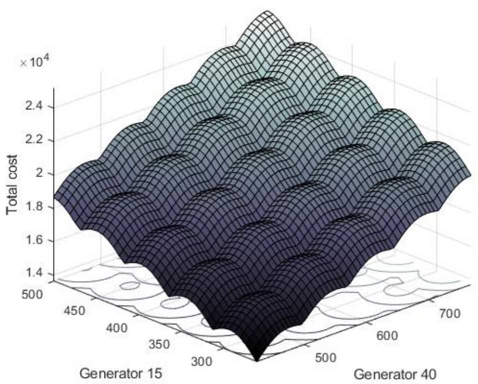

Fig. 1. Illustration of the landscape of the ED problem with the objective function in (2)

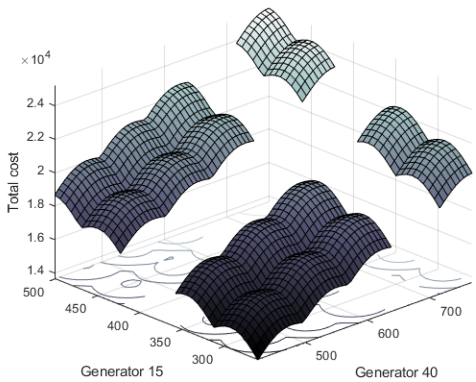

Fig. 2. Illustration of the landscape of the ED problem with the objective function in (2) and prohibited zones in (8)

Four operational constraints are typically considered to reflect the problem's nature, including power balance, power limitation, ramping rate, and prohibited zone constraints. The power balance is the only equality constraint in the problem, and the other three are inequality constraints.

The power balance constraint (3) requires total power output to be equal to the sum of power demand P_{D} and the transmission loss P_{L} of the power system. The transmission loss P_{L} is calculated by Kron's loss formula (4) [5], which can be ignored if there is no power loss in the system. The variables B_{gh}, B_{0g}, and B_{00} are the loss coefficients. Note that if the loss coefficient is presented in the MVA base format [6], it must be transformed into the actual values by (5) before loss calculation [7]–[8]. The variable $B_{gh\,(\mathrm{p.u})}$, $B_{0g\,(\mathrm{p.u})}$, and $B_{00\,(\mathrm{p.u})}$ are the loss coefficient in the MVA format, and *MVAbase* is the base MVA value. For example, if the loss coefficient is presented with the 100-MVA base capacity [6], B_{gh} must be divided by 100, and $B_{00\,(\mathrm{p.u})}$ must be multiplied by 100. The power limitation constraint (6) requires the power output P_j to lie between the minimal output $P_j^{\min}$ and the maximal output $P_j^{\max}$ of the j^{th} generator. Constraints (3) and (6) are included in all test cases.

$$\sum_{j=1}^{NG} P_j = P_{\mathrm{D}} + P_{\mathrm{L}} \tag{3}$$

$$P_{\mathrm{L}} = \sum_{g=1}^{NG}\sum_{h=1}^{NG} P_g B_{gh} P_h + \sum_{g=1}^{NG} B_{0g} P_g + B_{00} \tag{4}$$

$$B_{gh} = B_{gh\,(\mathrm{p.u})}/MVAbase\,, B_{0g} = B_{0g\,(\mathrm{p.u})}, B_{00} = B_{00\,(\mathrm{p.u})} \cdot MVAbase \tag{5}$$

$$P_j^{\min} \le P_j \le P_j^{\max} \tag{6}$$

The ramping rate and prohibited zone constraints are included in some problem models. The ramping rate constraint (7) is involved in the ED problem when the power system does not allow power generators to change the output too much between two consecutive periods [6]. The operating boundary of the current period is controlled by the power output P_j^0 in the previous period and the specified maximum decrement DR_j and increment UR_j of the output of the j^{th} generator.

$$\max\left(P_j^{\min}, P_j^0 - DR_j\right) \le P_j \le \min\left(P_j^{\max}, P_j^0 + UR_j\right) \tag{7}$$

The prohibited zone constraint (8) is applied to the problem to avoid unavailable power output ranges due to instability or physical issues [6]. The variables P^{l} and P^{u} denote the lower and upper boundaries of the prohibited zones, and NZ denotes the number of prohibited zones. Fig. 2 illustrates the discontinuity of the solution space caused by the prohibited zones.

$$P_j \in \begin{cases} P_j^{\min} \le P_j \le P_{j,1}^{\mathrm{l}} \\ P_{j,1}^{\mathrm{u}} \le P_j \le P_{j,2}^{\mathrm{l}} \\ \vdots \\ P_{j,NZ}^{\mathrm{u}} \le P_j \le P_j^{\max} \end{cases} \tag{8}$$

The objective functions can be explained as a piecewise quadratic function (9) if any power generator requires multiple fuel types to generate different levels of power [9]. The cost coefficients $a_{j,k}$, $b_{j,k}$, $c_{j,k}$, $d_{j,k}$, and $e_{j,k}$ vary with different fuel types, where the variable K is the number of fuel types (and power levels). The variables $P_{j,k}^{\min}$ and $P_{j,k}^{\max}$ are the minimal and maximal power output of each fuel type. The sine function is excluded from the problem when the valve-point effect does not happen in the power system [10]. This kind of problem model concerns not only power allocation but also the most economic fuel type.

$$F_c(P_j) = \begin{cases} a_{j,1} + b_{j,1}P_j + c_{j,1}P_j^2 + \left|d_{j,1}\left\{sin\left(e_{j,1}\left(P_{j,1}^{\min} - P_j\right)\right)\right\}\right|, & P_{j,1}^{\min} \le P_j \le P_{j,1}^{\max} \\ a_{j,2} + b_{j,2}P_j + c_{j,2}P_j^2 + \left|d_{j,2}\left\{sin\left(e_{j,2}\left(P_{j,2}^{\min} - P_j\right)\right)\right\}\right|, & P_{j,2}^{\min} \le P_j \le P_{j,2}^{\max} \\ \vdots & \\ a_{j,K} + b_{j,K}P_j + c_{j,K}P_j^2 + \left|d_{j,K}\left\{sin\left(e_{j,K}\left(P_{j,K}^{\min} - P_j\right)\right)\right\}\right|, & P_{j,K}^{\min} \le P_j \le P_{j,K}^{\max} \end{cases} \tag{9}$$

The ED model can be adapted further to many additional challenging problems depending on the power system components and inquisitive objective functions. We briefly review four primary problems extended from the ED problem model as a roadmap for subsequent further research.

1. The problem of multi-area economic dispatch (MAED) [11]–[13] lies in the operating cost minimization of the power system in multiple interconnected areas. The generating power can be transferred from one to other areas through tie-lines (connecting power wires across different areas). Apart from the conventional constraints of the ED problem, the MAED problem also considers a tie-line flow limits constraint to restrict the power flow capacity across different areas, maintaining the security and reliability of the power system.
2. The combined heat and power economic dispatch problem (CHPED) [14]–[17] aims to increase the power generation capacity of the power system. In practice, the conventional thermal power system wastes a certain amount of energy in the form of heat during the power generation process. Combined heat and power (CHP) systems are integrated into the thermal power system, serving as cogeneration units to convert the wasted heat to electrical power. The CHP system's heat balance and capacity limitation are included in the problem model, where they are the physical constraints of CHP for power generation. The objective function of the CHPED problem is to minimize the operating cost of the whole power system by satisfying the constraints of the thermal and CHP systems.
3. The dynamic economic dispatch (DED) [18]–[20] problem is an extensive practical ED problem that determines the most cost-effective allocation of the power output of generators to meet varying power demands across time intervals. The DED problem's complexity is upon time interval, as it directly controls the problem's dimensionality. The ramping rate is a security constraint typically included in the DED problem. The constraint regulates the rate at which the generator changes its power output between consecutive periods to maintain the reliability of the power system.
4. The economic emission dispatch (EED) [21]–[23] problem has received much more attention due to the increasing awareness of contemporary global warming. Apart from the operating cost objective function of the ED problem, pollution emission level is integrated as a second objective function of the problems to verify an environmental impact from the power system. In the EED problem, both objective functions are minimized simultaneously, while they may be conflicting in nature. Therefore, the EED problem can be classified as multi-objective optimization, seeking non-dominated solutions. Furthermore, the EED problem can combine with renewable resources [24]–[25], such as wind and solar energy systems, to reduce pollution emissions from the thermal power system.

The ED problem is itself important and challenging. Investigations into the ED problem provide high practical value since the ED problem serves as the basis of many extended problems as mentioned above. In the past decades, many research studies have addressed the ED problem. Abbas et al. [26] –[27] reviewed PSO-based approaches to the ED problem, and Jebaraj et al. [28] reviewed DE-based approaches. These surveys only included papers published before 2017. A recent survey by Lolla et al. [29] covered newer studies but still included only 20 papers published during 2018–2020 and no paper after 2020. In addition, we also lack of a work that collects data sets and solutions as a valid reference for researchers in this domain. The lack of a benchmark set also affects the completeness of experiments in the past literature. In this paper, we aim to fill these research gaps. The contributions of this paper are listed as follows:

1. A review based on algorithmic analysis: We review about 150 papers (about 90 papers published within recent ten years) that addressed the ED problem. We summarize the focused algorithmic components in these studies. This helps researchers to know what has been done and what may be done in the future.
2. A comprehensive collection of test cases: In the literature on the ED problem, there are more than ten test cases and more than 20 sub-cases in total. There is no collection of these test cases, and thus sometimes experiments were carried out with different/wrong test cases, which may result in misleading performance comparison results. In this paper, we make a comprehensive collection of test cases and their model coefficients. We will make these test cases public and downloadable for the convenient use of other researchers.
3. A simple but effective solver: We propose an algorithm called L-HMDE based on differential evolution (DE). It incorporates a hybrid mutation strategy, a linear population size reduction mechanism, and an improved repair mechanism. Although these components are not totally new, our integration makes the whole algorithm a simple but effective solver to the ED problem.
4. A complete and trustful performance comparison between algorithms: As mentioned, due to the lack of a collection of test cases, it is difficult for researchers in this domain to do a complete performance verification of their proposed algorithms. In most studies, the proposed algorithms were evaluated by one to three test cases. In this paper, we collect and verify solutions in past studies. Then, the performance of our algorithm is verified by comparing it with algorithms from more than 50 papers using more than 20 test cases. Together with the collected test cases, these solutions can be trustful and useful benchmarks.

The remaining of this paper is organized as follows. Section 2 presents a review of papers on the ED problem. Section 3 thoroughly describes the proposed L-HMDE. Section 4 presents the ED test cases. Section 5 presents experiments, results, and discussions. Section 6 concludes this paper and gives future research directions.

2. Literature Review

The ED problem with the convex objective function and operating boundary constraint might be solvable by deterministic approaches [30]–[32]. However, they might not be applicable for dealing with other ED characteristics like non-convexity or discontinuity. The Lagrangian approach, such as Lambda iteration, might provide an infeasible solution or get stuck in local minima because of improper initial values [33]–[35]. Dynamic programming might suffer from the curse of dimensionality in solving large-scale ED problems [36]. Linear programming has difficulty in solving the problem model with the transmission loss and prohibited zones [37]. In view of these difficulties, these approaches require problem model transformation or modification to improve the searching ability in solving the ED problem [37]–[40]. Metaheuristics are a promising approach to overcome these challenges. Over years, metaheuristics have been introduced for solving the ED problem in many studies. This section aims to give a literature review of metaheuristic algorithms for the ED problem, categorized based on the algorithm design and the connection between their proposed strategies.

Particle swarm optimization (PSO) [41]–[42] and DE [43] have gained considerable attention in solving the ED problem due to the ease of implementation and good performance. We review research studies related to DE and PSO separately in sub-sections 2.1 and 2.2, respectively. Sub-section 2.3 offers a brief review of other algorithms for solving the ED problem. Furthermore, the widely used constraint handling mechanisms for the ED problem are discussed in detail in sub-section 0. A summary of the essential features in solving the ED problem can be reviewed in Table 1.

2.1 Differential evolution

2.1.1 Solution reproduction mechanism

● ***Operator modification***: the mutation and crossover operators serve DE in reproducing new solutions. Some studies introduced novel mutation strategies to enrich DE's capability of solving the ED problem. Amjady and Sharifzadeh [44] modified the mutation operator and created mutant vectors with the guidance of a group of elite solutions. Modiri-Delshad et al. [45]–[46] presented a backtracking search algorithm (BSA) analogous to the standard DE. BSA employed similar crossover and selection operators of a standard DE. A mutant vector was generated through current and preceding solutions stored in the historical table. BSA provided high-quality solutions of small- and medium-scale test cases.

● ***Hybrid mutation strategy***: several studies showed an advantage of the hybrid mutation strategy in enhancing the performance of DE. Coelho et al. [47] applied the belief space concept of the cultural algorithm as a selection criterion to select between the rand/1 operator or the best/1 operator. Zou et al. [48] hybridized the rand/1 and rand/2 mutation operators based on probability selection. The chance to select the rand/2 operator was reduced throughout the search process. The worst half of the population was reinitialized to escape from local optima when it had no progress for a specified duration. Their proposed algorithm achieved better performance than other modified DEs in small- and medium-scale test cases. In [49] the mutation operators were selected based on quality and the number of improvement failures of each solution. Neto et al. [50] adopted self-adaptive DE (SaDE) as a local optimizer in the continuous-greedy randomized adaptive search procedure (C-GRASP) to enhance search performance. In SaDE, the rand/1 or rand/2 mutation operators were adaptively selected to create a new solution based on the probability calculated from the survival rate of new solutions. The proposed algorithm reached better solution quality over standard C-GRASP in small- to large-scale test cases.

● ***Hybrid DE with other algorithms***: several studies hybridized DE with other algorithms. In [51]─[52], they proposed hybrid frameworks that combined DE with PSO. PSO's mechanisms were employed to prevent premature convergence of DE. The hybrid algorithm provided promising results in solving a wide range of ED test cases. Xiong et al. [53] embedded DE operators and the Lévy flight function into biogeography-based optimization (BBO) to balance the exploitation and exploration. In their study, BBO parameters were controlled by a cosine function. Their algorithm outperformed the standard BBO [54] and other existing algorithms in solving small- and medium-scale test cases. Wang and Li [55] incorporated DE operators into the harmony search (HS) algorithm (DHS) to increase the global and local search capability. DHS performed effectively in solving small- and medium-scale test cases. Yang et al. [56] adapted the DE operators into Firefly Algorithm (FA) to enhance the searching ability. Their experimental results showed that the algorithm obtained better solutions quality than the standard FA in several ED test cases. Balamurugan and Subramanian [57] introduced a hybrid integer-coded DE with dynamic programming (ICDEDP) in solving the multiple-fuel ED problem. They adopted an integer encoding scheme to represent the fuel types of generators. The operating cost of each solution was minimized by dynamic programming. Liu et al. [58] incorporated the DE algorithm with the gain-sharing knowledge-based algorithm (GSK) to balance local and global searchability. In each iteration, the population was randomly divided into two sub-populations and assigned to the DE and GSK operators. At the end of each iteration, all sub-populations were combined together to share searching experiences for each other.

● ***Multiple group search***: an advantage of DE with multiple group search was discussed in [59]─[62]; the whole population was divided into multiple groups to improve the searching ability. Reddy and Vaisakh [59]─[60] proposed a shuffled DE (SDE) for tackling ED problems. A new solution was generated through the best and random solutions in the same group to maintain global and local search capability. SDE showed superior performance over existing algorithms in small- and medium-scale test cases. The concept of colonic competition was taken into DE (CCDE) by Ghasemi et al. [61]. The weakest group gradually reduces its size to increase the convergence rate. Li et al. [62] applied different mutation operators to different groups and proposed MPDE. The group without improvement was allowed to use solutions from other groups to create new solutions. MPDE obtained the optimal solution in small- to large-scale test cases.

2.1.2 Parameter control mechanism

The scaling factor and crossover rate are key parameters that influence the performance of DE. The scaling factor affects the moving distance of the mutant vector, and the crossover rate controls the number of exchanged variable values. Noman and Iba [63] investigated the parameter sensitivity of DE by fixing the parameter values during the search process. They showed that the standard DE performed effectively with small scaling factor and crossover rate in solving small- and medium-scale test cases.

● ***Dynamic parameter adjustment mechanism***: Many efforts indicated an improvement in DE by using dynamic parameter control mechanisms, which included linear functions [44], [48], uniform randomization [48], [61], or chaotic map functions [64]. Li et al. [62] applied a normal distribution to control DE's parameters; the mean value linearly decreased every iteration, and the standard deviation was fixed as a constant value. In Basu's study [65], a normal distribution was also utilized to adjust the scaling factor. The mean value was zero, and the standard deviation was calculated by the ratio of the operating cost of the current to that of the best-found solutions. His experiments demonstrated that the normal-distribution-based parameter control mechanism accelerated the convergence of DE in solving small- to large-scale test cases.

● ***Adaptive parameter adjustment***: several studies applied adaptive mechanisms to select DE parameters. Wang et al. [66] applied the one-fifth success rule to regulate the increment and decrement of the scaling factor parameter. They incorporated migrating and accelerated operators into DE to enhance solution quality. Coelho et al. [47] utilized the ratio of the diversity of the current population to the diversity of the initial population to control the crossover rate adaptively. In [67], they also applied the Lévy flight function and population diversity to control the crossover rate. Zhang et al. [49] applied the number of improvement failures of each solution as a criterion for selecting the scaling factor and crossover rate. In [68]–[69], the reinforcement learning was utilized to select DE's parameters; it selected the parameter value based on the improvement condition of new solutions generated in each iteration. The mechanism demonstrated the enhancement of DE's searching ability in solving ED and related problems.

2.2 Particle swarm optimization

2.2.1 Solution reproduction mechanism

The velocity updating mechanism is a crucial step of PSO. It updates the velocity of a particle through the cognitive, social, and inertial components. The cognitive component relies on the particle's personal best solution, and the social component relies on the best solution across the entire population. The inertial component is the velocity of a particle at the previous moment.

● ***Search trajectory improvement mechanism***: Several studies aimed to balance the exploitation and exploration of PSO by introducing new components into the standard velocity updating mechanism. In [70]–[72], the personal and global worst solutions were utilized to assist the population in escaping from poor areas. Abdullah et al. [73] introduced the neighbor's personal best solution to the velocity updating mechanism to prevent PSO from being stuck at local minima. Jadoun et al. [74] maintained the population diversity by introducing two new components to the velocity updating mechanism. The first component was a particle's preceding solution, and the second was the root-mean-square solution calculated from the current population. In [75], a new solution was updated through only one of the cognitive or social components to improve the search ability of their proposed PSO algorithm in solving the ED problem. The new solution was generated by the guidance of the personal best solution (the cognitive component) or one of the neighbors' best solutions (the social component). The orthogonal strategy was utilized to lead a population to a new promising area. Xu et al. [76] introduced a concept of comprehensive learning to the velocity updating equation to improve population diversity and maintain the convergence rate of their proposed PSO. Singh et al. [77] improved the search trajectory of the PSO by using an attraction factor vector; each particle was attracted to move forward to the global best solution to speed up the convergence rate.

● ***Hybrid PSO with other algorithms***: Some studies combined hybrid PSO with other algorithms to improve searchability of their proposed algorithms. Duman et al. [78] hybridized PSO with a gravitational search algorithm (GSA) for dealing with ED problems. The cognitive component was replaced by the updating mechanism of GSA. The proposed algorithm obtained superior solutions compared to existing algorithms in small and medium test cases. Ellahi et al. [79] hybridized particle swarm optimization with bat algorithm (BA) in solving the ED problem. The BA frequency parameter was adopted to control the behavior of the social and cognitive components, which allowed the proposed algorithm to have more flexibility in parameter turning and also enhanced the algorithm's exploration. Gacem and Benattous [80] hybridized genetic algorithm (GA) with PSO for tackling the ED problem. The new population was generated by incorporating GA and PSO operators, which provided multiple search characteristics to the proposed algorithm. This entity could allow the algorithm more opportunities to reach the optimization solution. Saber [81] integrated the updating equation of PSO with the bacterial foraging (BF) algorithm. The concept of biased random walk from the BF algorithm was introduced to the PSO updating equation, which enhanced the search performance of the proposed hybrid algorithm.

● ***Updating mechanism redefinition***: many efforts demonstrated the improvement of PSO by redefining its updating mechanism. The Quantum-behaved PSO (QPSO) and Random Drift PSO (RDPSO) were respectively utilized in [82] and [83] in tackling ED problems. QPSO and RDPSO shared a similar concept of the updating mechanism using two components. The first component was an absolute difference between the current solution and the average of personal best solutions, and the second was the weighted arithmetic mean of personal and global best solutions. The concept of escaping prey was taken into PSO to prevent premature convergence by Chen et al. [84]. The population was divided into three groups based on solution quality. The prey group (elite solutions) was updated by the Lévy flights to maintain the population diversity; the standard velocity updating mechanism was applied to the strong group; lastly, the random normal distribution was utilized to perturb the weak group. Their algorithm performed effectively in solving small and medium test cases. Kumar et al. [85] suggested a multi-agent PSO to tackle ED problems. The search space was divided into multiple regions occupied by particles. The Nelder-mean method was applied to update a particle. The final solution was created based on the obtained information from each region. The proposed algorithm obtained better solution quality than standard and modified PSOs.

● ***Other mechanism***: besides the velocity updating mechanism, some studies also discussed other aspects of enhancing PSO performance. Abdullah et al. [86] applied a tournament selection to select a survival solution for PSO. A group of solutions were randomly selected from the current and new populations to compete in tournaments, and the winner survived. The study obtained promising solutions in small- and medium-scale test cases. Hosseinnezhad and Babaei [87] introduced a new encoding scheme by mapping solutions to vectors of phase angles. This scheme might reshape the search space and allow the PSO to search potential solutions effortlessly. The proposed algorithm showed better performance than existing algorithms in solving small- and medium-scale test cases.

2.2.2 Parameter control mechanism

● ***Dynamic parameter adjustment mechanism***: Several studies incorporated parameter control mechanisms into PSO to enhance performance. The first type of control mechanism is dynamic control, which adjusts parameter values based on search iterations without feedback information. Many studies utilized exponential functions [73]─[74] or linear functions [6], [70], [73], [88] to control their PSO parameters in a time-dependent manner. The studies [75], [89] applied a chaotic map function to control their PSO parameters, where the parameters were adjusted based on a chaotic map rule and previous parameter values. In [90], Gholamghasemi et al. controlled cognitive and social components' behavior by using the cosine and sine functions; the inertia component was excluded from their velocity updating mechanism. Other studies introduced a cosine function [72], a chaotic map [91], or random functions [92]─[93]. These studies enhanced the search capability of PSO by trying a broader value range of control parameters.

● ***Adaptive parameter adjustment mechanism***: In [78], [94], they utilized adaptive parameter control mechanisms, which selected appropriate parameter values based on feedback information. In [78], the parameters of the hybrid PSO were adaptively selected by the fuzzy logic. The parameter selection criterion was ruled by the quality and progress of the best solution found in each iteration. Li et al. [94] applied population diversities of the current and personal best solutions to control parameters. In [83], Elsayed et al. applied a self-adaptive parameter control mechanism to RDPSO; each particle took the parameters as a part of the solution and sought their appropriate values through the PSO search process.

2.2.3 Local search

The local search mechanism is usually adopted in evolutionary algorithms to improve the solution quality. The advantage of sequential quadratic programming (SQP) was discussed in [95]─[96]. Coelho and Mariani [89] improved PSO by using an implicit filtering (IF) local search. In [97]─[98], PSO's searching ability was enhanced using a space reduction mechanism. When it had no progress for a period longer than the specified limit, it reduces the search space according to the position of the global best solution. Their PSO with the space reduction mechanism reached the optimal solution in a small- and medium-scale test cases.

2.3 Other existing algorithms

2.3.1 Solution reproduction mechanism

● ***Search direction improvement***: The topic of determining the search direction has been addressed in various studies to improve the efficiency of algorithms in solving the ED problem. Amjady and Nasiri-Rad [99]–[100] embedded the arithmetic-average-bound crossover operator into the real-coded genetic algorithm (GA), which had multiple operators with different search characteristics to improve global search efficiency. Many studies focused on reproducing new solutions with the guidance of the best solution. Babu et al. [101] embedded two operators into the evolutionary algorithm (EA) to balance exploitation and exploration. The first operator performed a random search, and the second one searched for a new solution with the guidance of the best solution. Their proposed algorithm found the best-known solutions when solving small- and large-scale test cases. The guidance of the best solution was also adopted in the modified pitch adjustment of HS by Secui et al. [102]. The proposed algorithm reached promising results in small- and medium-scale test cases. Many studies [103]–[109] allowed the population of the artificial bee colony (ABC) algorithm to move toward the best solution, which accelerated the search performance of the algorithm.

● ***Oppositional learning mechanism***: Some studies utilized the oppositional learning concept to produce new solutions and allowed the population to change the search direction. Pradhan et al. presented a standard [110] and a modified [111] grey wolf optimization (GWO) algorithms in tackling ED problems. In [111], an oppositional learning concept was introduced into GWO to improve the search ability. This concept changed the moving trajectory of the population to the opposite direction to escape from local optima. The oppositional learning-based GWO achieved a better convergence rate than the standard version. The same advantage of the oppositional learning concept was also discussed in [112]–[113], which integrated the concept into invasive weed optimization (IWO) and beluga whale optimization algorithm (BWO), respectively.

● ***Solution perturbation mechanism***: Various studies mentioned the improvement of their algorithm by perturbation of solutions based on random distributions. In [4], [114], they reported the performance enhancement of evolutionary programming (EP) by combining Gaussian and Cauchy mutation operators to generate new solutions. Chen et al. [115] combined Gaussian and Cauchy mutation operators into the Jaya algorithm to avoid premature convergence. In their algorithm, the population size was dynamically changed during the search process. The proposed algorithm performed more effectively than other Jaya algorithms in solving small- and medium-scale test cases. Zheng et al. [116] applied a crossover operator and a Gaussian mutation operator of GA in IWO to enhance solution quality and maintain population diversity. The proposed algorithm performed effectively in several ED test cases.

● ***Lévy flight mechanism***: the Lévy flight was another random distribution utilized as a standard or additional component to improve algorithm efficiency in solving the ED problem. El-Sayed et al. [109] applied the Lévy flight in the ABC algorithm as a new phase to assist the population to escape from local optima. The proposed algorithm showed a higher opportunity to achieve the optimal solution than other algorithms. Yu et al. [117] introduced the Lévy flight into the multiple-group search Jaya algorithm. The proposed algorithm obtained better solution quality than other Jaya algorithms in solving various ED test cases. The Lévy flight function is one of the standard components of the cuckoo search algorithm (CSA), and it provides the exploration ability to CSA. Sahoo et al. [118] compared performance of the standard CSA and other evolutionary approaches in solving ED problems. Their experiment showed that CSA obtained better results than the standard GA and PSO in several test cases. Nguyen and Vo [119] modified the solution reproduction process of CSA. This algorithm combined the Lévy flight and a crossover operator to generate new solutions in a probabilistic way, and it improved the convergence rate. The searchability of the chameleon swarm algorithm was improved in Braik's work [120] using Lévy flight and roulette wheel mechanisms. The Lévy flight mechanism was applied to the updating equation to enhance exploration, and the roulette wheel mechanism was utilized for mating selection to maintain exploitation.

● ***Hybrid algorithms***: many efforts investigated the performance improvement of hybrid algorithms in solving the ED problem. Some studies [121]–[122] discussed the advantages of problem space reduction. In [121], tabu search (TS) was utilized to regulate the feasible search of the ABC algorithm. The hybrid ABC/TS delivered better solution quality than several canonical algorithms. In [122], the lambda iteration algorithm was adopted to narrow the search space and speed up the searchability of the simulated annealing (SA) algorithm. The algorithm demonstrated a better convergence than some canonical and modified algorithms. In studies [123]–[124] discussed the advantages of using B-hill climbing to enhance the sine-cosine algorithm (SCA) exploitation to improve local searchability Basak et al. [125] conducted a study on the hybrid crow search algorithm and JAYA algorithms. The updating equation of both algorithms was merged to accelerate convergence rate.

● ***Mating selection mechanism***: Some studies discussed the selection mechanism. Al-Betar et al. [126] introduced a tournament selection into the pitch adjustment condition of HS. The tournament-based HS obtained promising results in various test cases. Al-Betar et al. [127] also investigated the performance improvement of HS by using three new selection operators to select survival solutions: tournament selection, roulette wheel, and ranking-based selection mechanisms. Their experimental results showed that new selection operators enhance the search efficiency of HS over the classic selection operator. Awadallah et al. [108] introduced four new selection schemes to the onlooker bee phase of ABC. The modified ABC achieved high-quality solutions in solving the CEC benchmark functions and several ED problems. In [128]–[129], the perturbed solution of the crossover operator was selected based on competition instead of randomization. Their modified CSA reached impressive results in small- to large-scale test cases.

2.3.2 Parameter control mechanism

● ***Dynamic parameter adjustment mechanism***: Many studies indicated a performance enhancement of algorithms by using dynamic parameter control mechanisms. Amjady and Nasiri-Rad [100] reported that adding exponential population size reduction to their proposed algorithm could speed up the convergence rate in solving small- and medium-scale test cases. Coelho and Mariani [130] utilized the population size and problem dimension to control the adjustment rate (PAR) parameter of the HS algorithm. They applied the exponential function to generate random step sizes of the bandwidth (BW) component. Jeddi and Vahidinasab [131] modified HS to obtain high-quality solutions. The parameters PAR and BW were dynamically adjusted using a linear function and an exponential function, respectively. The wavelet function was integrated into the proposed algorithm to reinitialize new solutions, which assisted the population in avoiding being trapped in local optima. Aydın and Ozyon [103]–[104] applied incremental social learning in the ABC algorithm. The population size increased during the search process until it reached the maximum size. In Secui's study [107], the step size of the updating mechanism of HS was controlled by chaotic map functions instead of pure randomization. Adarsh et al. [132] incorporated the sine function into the bat algorithm (BA) to control the loudness parameter. Liang et al. [133] utilized chaotic map functions to adjust the control parameters of BA, allowing the algorithm to escape from local minima. The random black hole model was incorporated into BA to accelerate the convergence. Lee et al. [134] introduced the adaptive Hopfield neural network (AHNN) for coping with multiple-fuel ED problems. Slope adjustment and bias adjustment mechanisms were utilized to control the HNN parameter. Their experimental results showed that AHNN reached similar solution quality with only one-half of the number of iterations used by the standard HNN [135].

2.3.3 More metaheuristic algorithms

Besides the mentioned algorithms, several nature-inspired metaheuristic algorithms were used to solve the ED problem. Examples include continuous quick group search optimizer (CQGSO) [136], social spider algorithm (SSA) [137], crisscross search optimizer (CSO) [138]–[139], water cycle algorithm (WCA) [140], grasshopper algorithm (GSO) [141], artificial algae algorithm (AAA) [142], symbiotic organisms search (SOS) [143], salp swarm algorithm (SSA) [144], turbulent flow of water-based optimization (TFWO) [145], slime mould algorithm (SMA) [146], ant colony optimization (ACO) [147], and Hooke-Jeeves algorithm (HJ) [150]. Details of these algorithms are referred to the original papers.

Table 1 A summary of the essential features in solving the ED problem

Algorithms	Algorithm features		References
DE	Solution reproduction mechanism	Operator modification	[44]–[46]
		Hybrid mutation strategy	[47]–[50]
		Hybrid DE with other algorithms	[51]–[58]
		Multiple group search	[59]–[62]
	Parameter control mechanism	Dynamic parameter adjustment mechanism	[44], [48], [61]–[62], [64]–[65]
		Adaptive parameter adjustment	[47], [49], [66]–[69]
PSO	Solution reproduction mechanism	Search trajectory improvement mechanism	[70]–[77]
		Hybrid PSO with other algorithms	[78]–[81]
		Updating mechanism redefinition	[82]–[85]
		Other mechanisms	[86]–[87]
	Parameter control mechanism	Dynamic parameter adjustment mechanism	[6], [70], [72]–[75], [88]–[93]
		Adaptive parameter adjustment mechanism	[78], [83], [94],
	Local search		[89], [95]–[98]
Other algorithms	Solution reproduction mechanism	Search direction improvement	[99]–[109]
		Oppositional learning mechanism	[110]–[113]
		Solution perturbation mechanism	[4], [114]–[116]
		Lévy flight mechanism	[109] [117]–[120]
		Hybrid algorithms	[121]–[125]
		Mating selection mechanism	[108], [126]–[129]
	Dynamic parameter adjustment mechanism		[100], [103]–[104], [107], [130]–[135]
	More metaheuristic algorithms		[136]–[147]

2.4 Constraint handling mechanisms

Constraint handling is essential to maintain the feasibility of solutions with respect to the constraints of the ED problem. This sub-section reviews popular constraint handling mechanisms in the studies on the ED problem. They are categorized into repair and penalty mechanisms. The repair mechanism fixes an infeasible solution directly; the penalty mechanism imposes penalty on solutions and expects the selection pressure pushes the population toward feasible regions.

2.4.1 Repair mechanism

The truncating mechanism [44], [53] was widely used to handle violation of the boundary constraints. It fixes the infeasible power output to the closest boundary. Nguyen and Vo [119] combined two repair mechanisms to deal with the power limit constraint. The first mechanism replaced the infeasible power output with the feasible power output from other solutions, and the second mechanism randomly regenerated the power output within the feasible range. In [139], [142], the authors integrated random shifting and truncating mechanisms to handle the prohibited zone constraints. The infeasible power output was adjusted to the closest boundary.

The power balance constraint is more complicated to handle, especially in the ED problem with the transmission loss. Several studies transformed the constraint into an inequality constraint where a small violation (tolerance error) was acceptable. The tolerance error was commonly set as a value less than or equal to 10^{-3} [74], [56], [142]. The constraint handling mechanism for the power balance constraint can be categorized into single- and multiple-unit repair mechanisms.

The single-unit repair mechanism modifies an infeasible solution by adding a compensating value to a single generator in each repair trial. In [48], [130], the authors allowed their single-unit repair mechanism to modify only generators with feasible power output. Studies [119], [128] applied a quadratic formula to handle the power balance constraint when the transmission loss was considered. The power balance constraint was rewritten as a quadratic function with a randomly selected generator. Then, the quadratic formula was solved, and a positive root was set as the new power output of the selected generator.

The multiple-unit repair mechanism modifies more than one generator of an infeasible solution in each repair trial. Jadoun et al. [74] distributed the deviation to the power demand to all generators equally. Li et al. [62] introduced a multiple-repair mechanism with proportional adjustment. The error due to the power balance was distributed to only generators that satisfied the boundary constraints in proportion to their current power output. Reddy and Vaisakh [60] combined single- and multiple-unit repair mechanisms to handle the power balance constraint. A generator was arbitrarily selected from the infeasible solution to modify using the single-unit repair mechanism. The residual error was then distributed to all generators except for the selected generator in the single-unit repair mechanism.

2.4.2 Penalty mechanism

The penalty mechanism transforms a constrained optimization problem into an unconstrained problem, and penalty functions are introduced to the problem's objective function for evaluating the constraint violation of infeasible solutions. Some studies [73], [86], [64] repaired an infeasible solution to satisfy boundary constraints and utilized the penalty mechanism to deal with the power balance constraint, where the fitness was the sum of the operating cost and the penalty. One difficulty of the penalty mechanism is the penalty setting. The metaheuristic algorithms might loss exploration ability if the penalty is too large; in contrast, the algorithm might not find any feasible solution if the penalty is too small. Moreover, different ED test cases might require different penalty settings [86]. Kumar et al. [147] overcame the mentioned drawback by using an adaptive penalty function; the penalty was changed dynamically according to the violation degree of each infeasible solution.

3. Proposed Algorithm

This section describes our proposed L-HMDE in detail. Algorithm 1 shows the pseudo code and Fig. 3 demonstrates the flow chart of L-HMDE. The encoding scheme and the solution initialization procedure are explained in subsection 3.1. The solution reproduction process is described in subsection 3.2. Subsection 3.3 presents the environmental selection mechanism and the linear population size reduction mechanism. The constraint handling mechanisms is given in the subsection 3.4. The last subsection provides the time and space complexity analysis of L-HMDE.

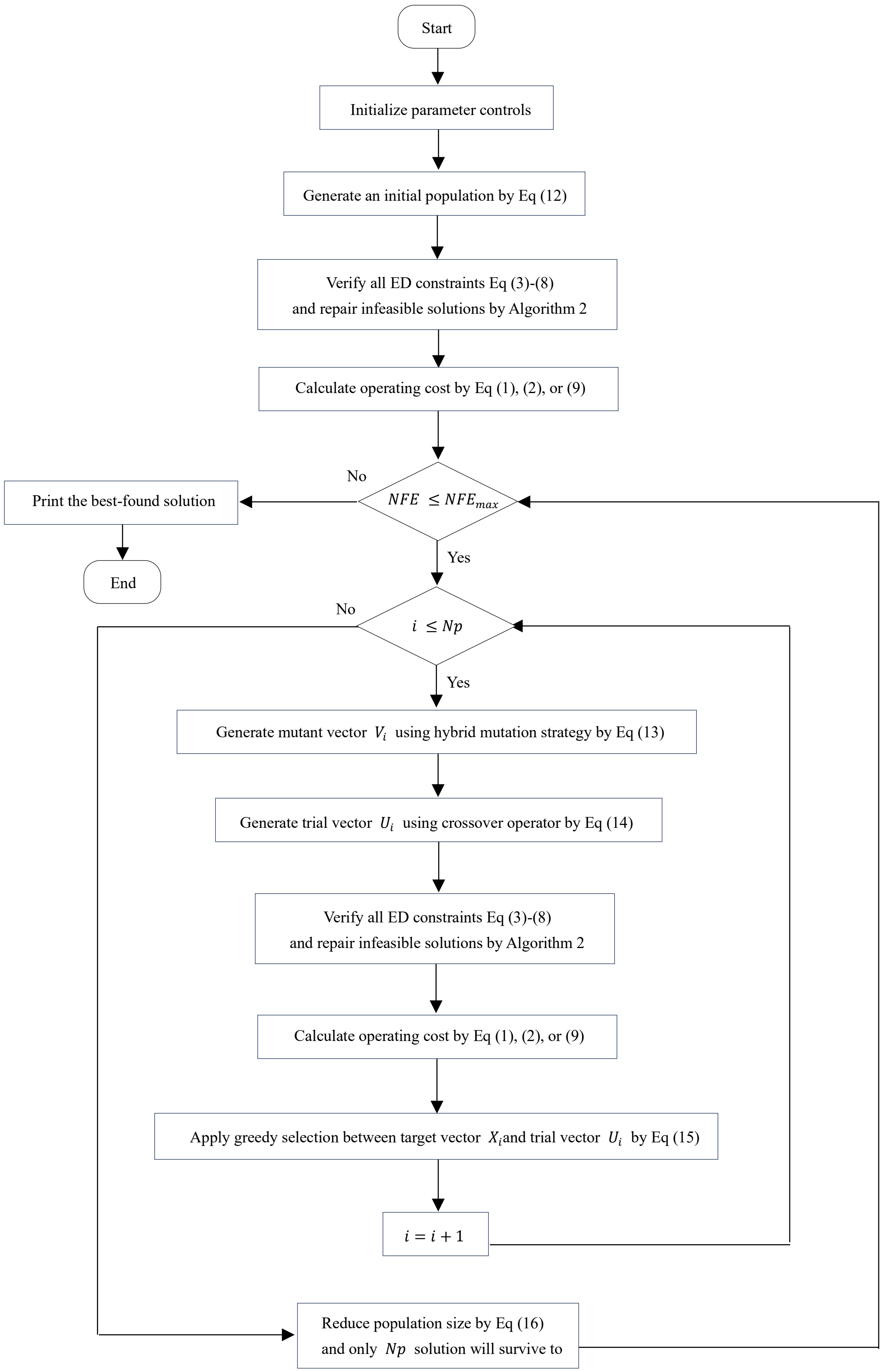

Fig 3 Flowchart of the proposed L-HMDE

Algorithm 1 L-HMDE

Notations:
Pop: Population
NP: Population size
NG: The number of generators in the power system
NFE, NFE_{max}: The current and maximum number of fitness evaluations
X_i, U_i: Target and trial vectors
V_i: Mutant vector
F, *CR*: Scaling factor and crossover rate
X_{best}: The best so far solution

```
Initialize all control parameters
Pop = Initialize(NP, NG)
Pop = Repair(Pop)
NFE = NP
while NFE ≤ NFE_max do
    i = 1, U = ∅
    while i ≤ NP and NFE ≤ NFE_max do
        X_i = Pop[i]
        V_i = Mutation(X_i, Pop, F)
        U_i = Crossover(X_i, V_i, CR)
        U_i = Repair(U_i)
        NFE = NFE + 1
        i = i + 1, U = U ∪ {U_i}
    end while
    Pop = EnvironmentalSelection(Pop, U)
    Pop = Sort(Pop)
    Update NP by using linear population size reduction
    Pop = Pop[1:NP]
    Update X_best
end while
```

3.1 Initialization

The population *Pop* consists of *NP* candidate solutions, as given in (10). Each solution X_i is encoded as a real vector of length equal to the number of generators *NG* in the power system, as given in (11). The variable *i* denotes a running index of each solution. The variable $P_{i,j}$ represents the power output of the j^{th} generator of solution X_i.

$$Pop^T = \{X_1, X_2, \ldots, X_i, \ldots X_{NP}\}^T, \qquad 1 \leq i \leq NP \tag{10}$$

$$X_i = [P_{i,1}, P_{i,2}, \ldots, P_{i,j}, \ldots P_{i,NG}], \qquad 1 \leq j \leq NG \tag{11}$$

Each initial solution is generated by a uniform randomization mechanism (12). The value of each decision variable $P_{i,j}$ lies in the feasible range of power limit [P_j^{min}, P_j^{max}]. The term *rand*(0, 1) is a random function that uniformly generates a real value between zero and one.

$$P_{i,j} = P_j^{min} + rand(0,1) \cdot (P_j^{max} - P_j^{min}) \tag{12}$$

3.2 Solution reproduction

Each solution (target vector) iteratively generates a new solution by incorporating mutation and crossover operators. In the canonical DE, a mutant vector is generated from the rand/1 operator. However, we found that the rand/1 operator has a disadvantage due to parameter sensitivity in solving the ED problem. In this paper, we adopt a hybrid mutation strategy to take advantage of multiple search characteristics and reduce parameter sensitivity. The benefits of our hybrid operator are discussed in subsection 5.2.1.

A mutant vector V_i is generated by either the rand/1 or the current-to-random/1 strategy based on probabilistic selection, as shown in (13). If the random value of *rand*(0, 1) is less than or equal to a pre-specified value δ, the mutant vector will be generated by the rand/1 strategy; otherwise, it will be generated by the current-to-rand/1 strategy. Vectors X_{r1}, X_{r2}, and X_{r3} are three distinct solutions selected randomly from the population and are different from the target vector X_i. Both mutation operators utilize the same constant scaling factor F.

$$V_i = \begin{cases} X_{r1} + F \cdot (X_{r2} - X_{r3}) & \text{if } rand(0,1) \le \delta \\ X_i + F \cdot (X_{r1} - X_i) + F \cdot (X_{r2} - X_{r3}) & otherwise \end{cases} \tag{13}$$

Using the binomial crossover, a trial vector $U_i = [u_{i,1}, u_{i,2}, \ldots, u_{i,NG}]$ is generated by crossing information from the mutant vector V_i and the target vector X_i based on the crossover rate CR, as shown in (14). The j^{th} element of a trial vector will copy the value $v_{i,j}$ of the mutant vector when the value of $rand(0, 1)$ is less than or equal to the crossover rate CR or when the j^{th} element is equal to a randomly selected element j_{rand}. Otherwise, it will copy the power output $P_{i,j}$ from the target vector X_i. The purpose of the variable j_{rand} is to guarantee that the trial vector differs from its target vector.

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand(0,1) \le CR \text{ and } j = j_{\text{rand}} \\ P_{i,j} & \text{otherwise} \end{cases} \tag{14}$$

3.3 Environmental selection

The trial vector U_i will replace the target vector X_i if its operating cost is not greater than the cost of the target vector, as given in (15). After the replacement process, the population is sorted in the ascending order of cost.

$$X_i = \begin{cases} U_i & \text{if } F_c(U_i) \le F_c(X_i) \\ X_i & \text{otherwise} \end{cases} \tag{15}$$

In canonical DE, the population size (NP) is equal in all generations, which can cause slow progress in solving some problems. In L-HMDE, the population size is linearly reduced to enhance search performance using Eq. (16), and only NP solutions survive to the next generation. This mechanism was borrowed from L-SHADE [148], and it has shown positive effects in L-SHADE. The paramters NP_{intial} and NP_{final} are the values of the initial and final population size, respectively; NP_{final} equals to the minimum number of solutions required in the adopted mutation operators. Parameters NFE and NFE_{max} are the current and the maximum number of fitness evaluations. L-HMDE will continue its search until the variable NFE reaches NFE_{max}. This mechanism only adds one more parameter to our algorithm, which is the NP_{intial} parameter. The performance improvement of the proposed algorithm with/without the linear population size reduction is discussed in subsection 5.2.3.

$$NP = round\left(\left(\frac{NP_{\text{final}} - NP_{\text{initial}}}{NFE_{\text{Max}}} \cdot NFE\right) + NP_{\text{initial}}\right) \tag{16}$$

3.4 Constraint handling

Infeasible solutions might be generated during the initialization and the reproduction processes, and they cannot be used as final solutions for the ED problem. In L-HMDE, we proposed an improved single-unit repair mechanism to fix these infeasible solutions.

3.4.1 Repair for handling boundary constraints

The violation of the power limit constraint (6) is handled by the truncating mechanism. The power output $P_{i,j}$ with an infeasible value is fixed to the closest power boundary P_j^{min} or P_j^{max}, as defined in (17). If the ramping rate constraint (7) is included in the problem, we also take the boundary values in (7) into consideration. The violation of the prohibited zone constraint (8) is handled by (18). The power output $P_{i,j}$ in the prohibited zone is fixed to the closest boundary. If it is at the middle point of the prohibited zone, it is fixed to the boundary P^{l}.

$$P_{i,j} = \begin{cases} P_j^{\text{min}}, & if\ P_{i,j} < P_j^{\text{min}} \\ P_j^{\text{max}}, & if\ P_{i,j} > P_j^{\text{max}} \end{cases} \tag{17}$$

$$P_{i,j} = \begin{cases} P^{\text{l}}, & if\left|P^{\text{l}} - P_{i,j}\right| \le \left|P^{\text{u}} - P_{i,j}\right| \\ P^{\text{u}}, & \text{otherwise} \end{cases} \tag{18}$$

Algorithm 2 Improved single-unit repair mechanism

Notations:
X_i: The infeasible solution to be repaired
P_D, P_L: The power demand and the transmission loss of the power system
Diff: The deviation to the sum of power demand and loss of an infeasible solution
$P_{i,j}$: The power output of the j^{th} generator of the solution X_i
ε: The tolerance error
T, T_{max}: The current and the maximum number of repair trials
NG: The number of generators in the power system
S: The set of selected generators to repair

Repair all variables of X_i to meet all boundary constraints by Eq (17) and (18)
Calculate the transmission loss P_L by Eq (4)
$Diff = P_D + P_L - \sum_{j=1}^{NG} P_{i,j}$
$T = 1$, $T_{max} = 30$, $S = \varnothing$
while $T \leq T_{max}$ and abs($Diff$) $\geq \varepsilon$ **do**
    for $j = 1$ to NG **do** (preliminary checking)
        if $P_j^{min} \leq (P_{i,j} + Diff) \leq P_j^{max}$
            $S \leftarrow S \cup \{j\}$
        endif
    end for
    if $S \neq \varnothing$
        Randomly select j^* from S
    Else
        Randomly select j^* from $\{1, 2, \ldots, NG\}$
    endif
    $P_{i,j^*} = P_{i,j^*} + Diff$
    Repair to meet all boundary constraints by Eq (17) and (18)
    Recalculate the transmission loss P_L by Eq (4)
    $Diff = P_D + P_L - \sum_{j=1}^{NG} P_{i,j}$
    $T = T + 1$
end while

3.4.2 *Repair for handling the power balance constraint*

Following many studies, we relax the power balance constraint (3) as an inequality constraint (19). A solution is regarded as a feasible solution when its error is less than or equal to a pre-specified tolerance error ε, which should be a very small value. The transmission loss P_L will be set to zero if it is not considered in the problem model. Fixing infeasible solutions to meet the power balance constraint usually requires many trials of repair when the transmission loss is considered.

$$\varepsilon \geq \left| P_D + P_L - \sum_{j=1}^{NG} P_{i,j} \right| \tag{19}$$

An infeasible solution X_i that violates the power balance constraint (3) is fixed by our improved single-unit repair mechanism. Each infeasible solution is allowed to be repaired for at most 30 trials; if an infeasible solution is not fixed to be feasible within 30 trials, it will not survive to the next generation. The procedure of the improved single-unit repair mechanism is presented in Algorithm 2.

First, we apply equations (17) and (18) to fix all infeasible power outputs $P_{i,j}$ that do not satisfy the boundary constraints. Then, we calculate the transmission loss. Next, we calculate the difference *diff* between the total power output and the sum of the power demand P_D and the transmission loss P_L. We want to absorb the difference *diff* by adjusting the power output of some generators. Before adjusting the solution, we do a preliminary checking: we find the generators j that do not violate the power limitation constraint (6) if we add the *diff* value to their power output $P_{i,j}$. Let S denote the set of the indices of these generators. If S is not empty, we select one generator randomly from S; otherwise, we select one generator randomly from all generators. We add the *diff* value to the selected generator. After adjusting the solution, we check the constraints again. If the solution can be regarded as feasible (i.e. the *diff* is not greater than the tolerance error) or the maximum number of trials is reached, the repair procedure stops; otherwise, we repeat the above steps.

The key difference between our improved mechanism and the standard single-unit repair mechanism (SR) [48], [130] is the step of preliminary checking. The preliminary checking helps to repair the solution successfully within fewer trials and to keep the modifications of the solution smaller. Fig. 4 is an example. In this example, we need to repair an infeasible solution and the initial *diff* value is 490. There are six generators that do not reach the boundary output values, and the standard single-unit repair mechanism selects one at a time from these generators to adjust the power output to absorb the *diff* power. In the example, the standard mechanism absorbs the *diff* by adjusting three generators. As for our improved mechanism, it first finds the two generators that can individually absorb the *diff* power. In the example, it selects the sixth generator, adjusts its output, and fixes the solution within just one trial.

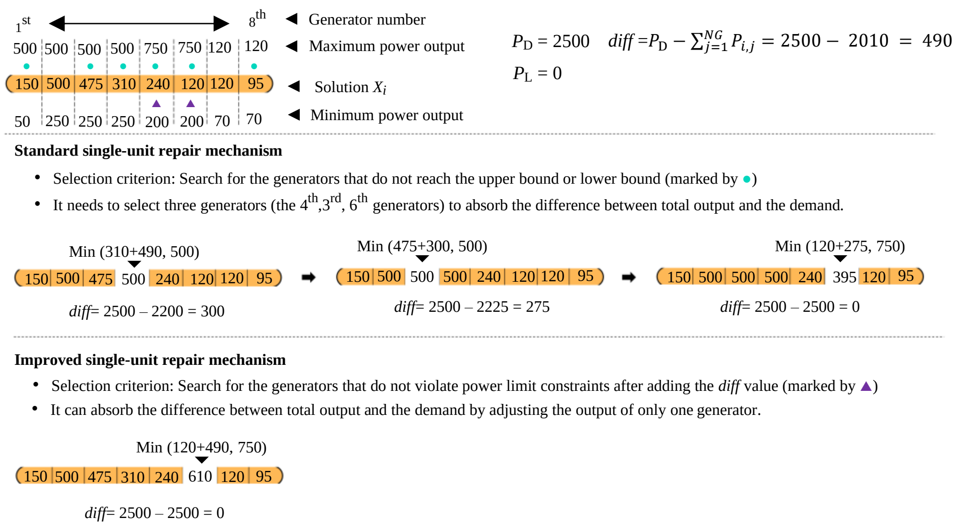

Fig 4 The difference between the standard and the improved single-unit repair mechanisms

3.5 Time and space complexity analysis

In this sub-section, we analyze the time and space complexity of the proposed L-HMDE. Our algorithm only needs space to store the population, and thus the space complexity is $O(NP_{\text{initial}} \cdot NG)$, where NP_{initial} is the initial population size and NG is the number of generators in the power system. Let $NP(t)$ denote the population size at generation t, T denote the number of generations, R denote the maximum number of trials in the repair operator, and E denote the maximum number of fitness evaluations. The time complexity of our algorithm is derived in the following.

There are six main operators in our L-HMDE: initialization, evaluation, repair, mutation, crossover, and environmental selection. The time complexity of applying each of the initialization, evaluation, mutation, and crossover operators to a single solution is $O(NG)$, and applying each of them to a population leads to the time complexity $O(NP(t) \cdot NG)$. The time complexity of repairing one solution is $O(R \cdot NG)$ and of repairing a population is $O(NP(t) \cdot R \cdot NG)$. The environmental selection operator consists of evaluation of the population, which takes complexity $O(NP(t) \cdot NG)$, and sorting of the population, which has complexity $O(NP(t) \cdot \log NP(t))$. Since $\log NP(t)$ is usually smaller than NG, the time complexity of the environmental selection is approximately $O(NP(t) \cdot NG)$. In each generation, the time complexity of all operators is bounded by $O(NP(t) \cdot R \cdot NG)$. Repeating all the operators for T generations, the time complexity is $O(\sum_{t=1}^{T} NP(t) \cdot R \cdot NG) = O(E \cdot R \cdot NG)$. In summary, the time complexity of our L-HMDE is controlled by the maximum number of fitness evaluations (E), the maximum number of repair trials (R), and the problem dimension (NG), i.e. the number of generators in the power system.

4. ED Test Cases Review

In the literature on the ED problem, many test cases were used to verify the performance of algorithms. We collect 13 test cases and introduce them briefly in this section. Table 2 gives a summary of them. Data of the model coefficients of these test cases are given in the appendix. One important thing worth noting is that some test cases have several versions and these sub-cases are very similar and different only in the values of very few coefficients. Different versions of each test case have different optimal solutions, and comparing experimental results across versions will be misleading. Researchers should be careful when they compare algorithm performance by using these test cases.

- Test case 1 [6] is a small-scale power system with six generators considering the transmission loss, the ramping rate, and prohibited zones. The system's power demand is 1263 MW. Note that the loss coefficient B_{00} must be changed from 0.056 to 0.0056 according to the notification of the data set owner in [149]. The loss coefficients of the test case are presented with the 100-MVA base capacity and must be transformed into the actual values by (5) before loss calculation. The data set of test case 1 is given in Appendix A.1.
- Test cases 2 and 3 [9] are small-scale power systems with 10 generators, requiring multiple fuel types for different power levels. The power demand of both test cases is set to 2700 MW, and only test case 3 considers the valve-point effect. The data set of the test cases is given in Appendix B.1.
- Test case 4 is a small-scale power system with 13 power generators, considering the valve-point effect. We found two versions of the cost coefficients, which were published in [4] and [82], respectively. Test case 4 has two widely used power demands, which are 1800 and 2520 MW. In this paper, we call the two cases using the cost coefficients in [4] with the power demand of 1800 and 2520 MW test cases 4.1 and 4.2, respectively. The other two cases using cost coefficients in [82] with the power demand of 1800 and 2520 MW are called test cases 4.3 and 4.4, respectively. The coefficient values of all versions are given in Appendix C.1–2.
- Test case 5 is a small-scale power system with 13 power generators, considering the valve-point effect and the transmission loss. We found four versions of this test case that used different coefficient values. Test cases 5.1 to 5.3 use the cost coefficients from [4], and test case 5.4 uses the cost coefficients from [82]. All versions use the transmission loss coefficients from [62] with some modifications. Test case 5.1 uses the same loss coefficients from [62]. Test case 5.2 sets the loss coefficient $B_{0,11}$ by 0.0017, and test cases 5.3 and 5.4 set the loss coefficient $B_{1,10}$ by 0.0005 and B_{00} by 0.000055, respectively. The loss coefficients of this test case are presented with the 100-MVA base capacity and must be transformed into the actual values by (5) before loss calculation. The power demand of all versions is 2520 MW. The data set of all versions are given in Appendix D.1–5..
- Test case 6 is a small-scale power system with 15 generators, considering the transmission loss, the ramping rate, and prohibited zones. The system's power demand is 2630 MW. We found two versions of this test case that use different previous power outputs P_j^0. Test case 6.1 uses the coefficient data set from [6]. Test case 6.2 modifies the previous power output P_2^0 to 360 and P_5^0 to 190 according to the notification of the data set owner in [149]. Both versions use the same transmission loss data set from [6]. Note that the loss coefficient $B_{1,10}$ must be changed to -0.0005 to make the loss coefficient matrix symmetrical due to the notification in [149]. The loss coefficients of this test case are presented with the 100-MVA base capacity and must be transformed into the actual values by (5) before loss calculation. The data set of all versions are given in Appendix E.1–3.
- Test case 7 [5] is a medium-scale power system with 20 generators considering the transmission loss. The system's power demand is 2500 MW. The data set of the test case is provided in Appendix F.1.
- Test case 8 is a medium-scale power system with 40 power generators considering the valve-point effect. Test case 8.1 was published in [4]. Test cases 8.2 and 8.3 use the data set from [4] with some modifications of cost coefficients. Test case 8.2 sets the cost coefficient a_7 by 278.71. Test case 8.3 sets cost coefficients a_{15} and a_{16} by 1760.4, b_{15} and b_{16} by 8.84, and c_{15} and c_{16} by 0.00752. The data set of all versions can be reviewed in Appendix G.1–3.
- Test case 9 [112] is a large-scale power system with 110 power generators, and the system's power demand is 15000 MW. The data set of the test case can be reviewed in Appendix H.1.
- Test cases 10–12 [91] is a large-scale power system with 140 generators and a power demand of 49342 MW. These test cases consider different problem characteristics. Test case 10 considers the valve-point effect, the ramping rate, and prohibited zones. Test case 11 ignores the ramping rate, and test case 12 ignores the valve-point effect. These three test cases use the same data set provided in Appendix I.1.
- Test case 13 is the largest power system in this study, which consists of 160 generators and requires different fuel types for different power levels. The test case is built up by replicating the test case 3 for 16 times. The system's power demand is 43200 MW.

Table 2 Summary of ED test cases

Test case	Model characteristics							
	Number of generators	Power demand (MW.)	Transmission loss	Valve-point effect	Ramping rate	Prohibited zones	Multiple fuel types	MVA base capacity
1	6	1263	✓		✓	✓		✓
2	10	2700					✓	
3	10	2700		✓			✓	
4*	13	1800/2520		✓				
5*	13	2520	✓	✓				✓
6*	15	2630	✓		✓	✓		✓
7	20	2500	✓					
8*	40	10500		✓				
9	110	15000						
10	140	49342		✓	✓	✓		
11	140	49342		✓		✓		
12	140	49342			✓	✓		
13	160	43200		✓			✓	

* We found more than one version of these test cases.

5. Experiments and results

We carried out experiments to verify the effects of mechanisms of our algorithm and to compare the performance of the algorithm with existing studies. The parameter setting and the computing environment of our experiments are given in sub-section 5.1. The effects of the mechanisms of our L-HMDE are presented in sub-section 5.2. Performance comparison results are presented in sub-sections 5.3 and 5.4.

5.1 Parameter setting

The parameter settings are presented in Tables 3 and 4. The initial population size NP_{intial} was 15, and the final population size NP_{final} was 4, which meets the minimal required number in the two mutation operators of L-HMDE. The probabilistic selection parameter δ of the hybrid mutation strategy was set to 0.7, which means L-HMDE selects the rand/1 mutation with probability 0.7 and the current-to-rand/1 mutation with probability 0.3. The experimental results on tuning of NP_{initial} and δ are provided later in this sub-section. The scaling factor F and the crossover rate CR were fixed as constant values at 0.5 and 0.1, respectively. The experimental results on tuning of F and CR will be presented in Section 5.2.1. The maximum number of repair trials $T_{\max}$ for each infeasible solution was 30. The acceptable tolerance error ε was 10^{-8} to maintain the accuracy of solutions; this value is much smaller than the error of most solutions in the literature. The maximum number of fitness evaluations $NFE_{\max}$ (termination criterion) is listed in Table 4. Note that $NFE_{\max}$ is the only parameter with values dependent on the test cases. We used the same parameter setting for L-HMDE to solve all 13 test cases when we compared its performance with existing algorithms. We implemented L-HMDE by the Matlab programming language (R2021a). Experiments were carried out on a computer with an Intel i7-10700 2.90GHz processor and 8 GB RAM. Each test case was solved for 100 times by each tested algorithm variant.

Table 3 Parameter setting for L-HMDE

Parameter	NP_{intial}	NP_{final}	δ	F	CR	$T_{\max}$	ε
Value	15	4	0.7	0.5	0.1	30	10^{-8}

Table 4 Maximum number of fitness evaluations for each test case

Test case	1	2	3	4	5	6	7
$NFE_{\max}$	1500	2000	10000	25000	25000	5000	5000
Test case	8	9	10	11	12	13	
$NFE_{\max}$	50000	50000	50000	50000	50000	150000	

We determined the appropriate values of the initial population size NP_{intial} and the probabilistic selection parameter δ by tuning each parameter separately. The effectiveness of parameter values was evaluated by the overall average of normalized cost E_{cost}, as defined in (20). We selected four test cases

(3, 4.1, 8.1, and 10) that cover different model characteristics. Each variant of L-HMDE with a specific parameter value solved each of the four selected test cases for 100 times and the average cost was recorded. Let $AvgCost_m(j)$ denote the average cost obtained by an algorithm variant j using a specific parameter value; $AvgCost_m^{max}$ and $AvgCost_m^{min}$ denote the maximum and minimum of the average costs obtained by all algorithm variants in the test case m, respectively. The smaller $E_{cost}(j)$ is, the better performance the algorithm variant j is. The performance results of the algorithm variants using different values of the initial population size and of the mutation probabilistic selection parameter are presented in Tables 5 and 6, respectively. Each cell contains the normalized average cost (in parentheses) and the original average cost. The last row presents the sum of normalized average cost over four test cases.

$$E_{cost}(j) = \sum_{m=1}^{4} \frac{AvgCost_m(j) - AvgCost_m^{min}}{AvgCost_m^{max} - AvgCost_m^{min}} \tag{20}$$

Initial population size: Nine values were examined for the initial population size. The values of F, CR, and δ were set by 0.5. We can infer from the results of Table 5 that L-HMDE tends to perform better with smaller initial population sizes. We set the initial population size by 15 due to its lowest E_{cost} value.

Mutation probabilistic selection parameter: Seven values were examined for the mutation probabilistic selection parameter. In this step, the initial population size was 15. Both extreme parameters (0 and 1) can negatively impact the L-HMDE in some test cases. We set the parameter by 0.7, which lead to the lowest overall average cost.

Table 5 Performance comparison of values of the initial population size

Test case	Initial population size NP_{intial}								
	5	10	15	20	30	40	50	100	200
10	(0.05) 623.84	***(0.00)*** ***623.83***	(0.05) 623.84	(0.08) 623.85	(0.15) 623.86	(0.20) 623.87	(0.26) 623.88	(0.50) 623.91	(1.00) 623.99
13	(1.00) 18045.73	(0.20) 18001.03	(0.02) 17990.75	***(0.00)*** ***17989.67***	(0.01) 17989.99	(0.00) 17989.90	(0.01) 17990.44	(0.03) 17991.20	(0.17) 17999.47
40	(0.65) 121698.86	(0.06) 121480.17	***(0.00)*** ***121457.88***	(0.32) 121576.69	(0.49) 121639.26	(0.50) 121643.66	(0.51) 121649.53	(0.75) 121738.54	(1.00) 121830.08
140	(1.00) 1658078.94	(0.13) 1657979.97	(0.07) 1657972.96	(0.02) 1657968.04	***(0.00)*** ***1657965.38***	(0.00) 1657965.77	(0.02) 1657968.03	(0.21) 1657988.82	(0.79) 1658055.01
E_{score}	(2.70)	(0.39)	***(0.14)***	(0.43)	(0.64)	(0.71)	(0.81)	(1.49)	(2.96)

Table 6 Performance comparison of values of the mutation probabilistic selection parameter

Test case	Mutation probabilistic selection parameter δ						
	0	0.1	0.3	0.5	0.7	0.9	1
10	(1.00) 623.85	(0.85) 623.85	(0.57) 623.84	(0.59) 623.84	(0.11) 623.84	***(0.00)*** ***623.84***	(0.18) 623.84
13	(1.00) 18007.99	(0.81) 18001.32	(0.48) 17990.28	(0.50) 17990.75	(0.00) 17973.77	***(0.00)*** ***17973.76***	(0.02) 17974.30
40	(0.70) 121481.14	(0.39) 121469.84	***(0.00)*** ***121455.28***	(0.07) 121457.88	(0.08) 121458.14	(0.75) 121482.89	(1.00) 121492.27
140	(0.94) 1657985.74	(0.27) 1657972.98	(0.40) 1657975.48	(0.08) 1657972.96	***(0.00)*** ***1657968.82***	(0.17) 1657971.23	(1.00) 1657986.86
E_{score}	(3.64)	(2.31)	(1.45)	(1.24)	***(0.19)***	(0.92)	(2.19)

5.2 Impact of the proposed mechanisms in L-HMDE

5.2.1 The effect of F/CR parameters and the advantage of hybrid mutation strategy

In the first experiment, we intended to examine the effect of the hybrid mutation strategy. We replaced the hybrid mutation strategy in L-HMDE by using only the current-to-rand/1 mutation or only the rand/1 mutation to make two other versions of algorithms. For each of the three algorithms, we tested 81 (9×9) variants with the values of F and CR in {0.1, 0.2, …, 0.9}. Each algorithm variants solved each test case for 100 runs. We compared their solution quality in terms of the average cost, the success rate, and the result of the Wilcoxon rank-sum test. When an algorithm variant is able to find a solution with the cost the same as the best solution in the literature in a run, we say that run is successful. The success rate is the ratio of the number of successful runs to 100. We tested whether the difference between L-HMDE and the two other algorithms is statistically different by the Wilcoxon rank-sum test at a significance level of 5%.

For each test case, we visualize the experimental results by three sets of heat maps. Taking the top three sets of heat maps for test case 4.1 in Fig. 5 as an example. The leftmost set of heat maps consist of three heat maps, each of which shows the average cost over 100 runs for one algorithm using the specified mutation operator (current-to-rand/1, rand/1, or hybrid). One heat map consists of 9×9 = 81 cells, and each cell represents the average cost of an algorithm variant using *F* and *CR* with specified values. The middle set of heat maps is similar to the first set, and the difference is in that each cell of a heat map represents the success rate of an algorithm variant over 100 runs. In these two sets of heat maps, the darker color means better performance (lower cost or higher success rate). The rightmost set of heat maps consists of only two heat maps. Each heat map illustrates the results of the Wilcoxon rank-sum tests on the solution quality of the version using the hybrid mutation strategy and one of the versions using a single mutation operator. In this set of heat maps, the dark color (+), light color (≈), and white (-) color represent that the version using the hybrid mutation strategy is statistically better than, equal to, or worse than the compared version, respectively. Based on our observations, we found that the hybrid mutation strategy has positive impact on solving seven out of 13 test cases and negative impact on only two test cases. The impact is not obvious in the remaining four test cases. We present the results in the following.

The hybrid mutation strategy positively impacts the algorithm performance in solving seven test cases. We show the heat maps of some selected test cases in Fig. 5. (Not all test cases are shown due to the limitation of space.) We can see that each version of algorithm performs well with some parameter settings but not all parameter settings. It reveals that the parameter setting is influential. The good settings of the versions using the current-to-rand/1 mutation and the rand/1 mutation are different; taking test case 8.1 as an example, current-to-rand/1 prefers medium-to-large *F* values but rand/1 prefers small-to-medium *F* values. Besides, the good settings could change when different test cases are solved; taking test cases 8.1 and 9 as examples, rand/1 prefers small *F* values when solving test case 8.1 but prefers large *F* values when solving test case 9. By using the hybrid mutation strategy, the algorithm can perform well under a larger number of parameter settings. Taking test case 9 as an example, the two algorithms using a single mutation operator perform well under around 1/3 of 81 parameter settings; by hybridizing the two mutation operators, the algorithm performs well under almost all parameter settings. We can observe the same positive impact in the middle set of heat maps about the success rate. In the rightmost set of heat maps, we can see that the algorithm using the hybrid mutation strategy significantly outperforms the two other algorithms under many parameter settings and is outperformed under very few parameter settings.

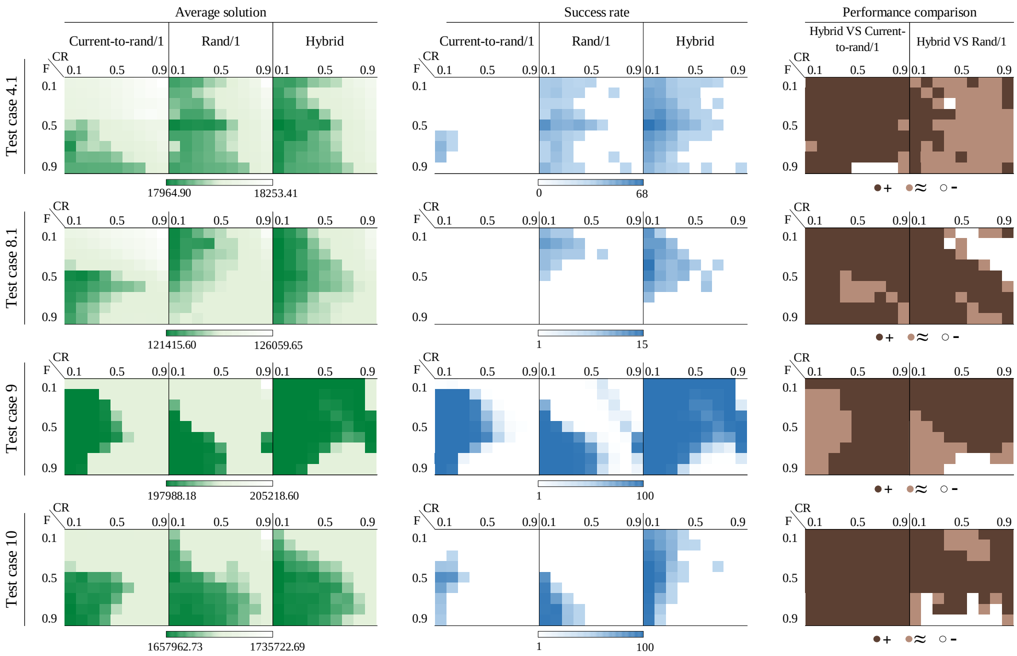

Fig 5 Heat maps of experimental results of test cases (only some are shown) that show positive impact of the hybrid mutation strategy

The hybrid mutation strategy does not show obvious effect when test cases 1, 2, 6, and 7 are solved. This is because that these test cases are relatively easy to solve. We show the heat maps of test cases 1 and 2 in Fig. 6. We can see that the algorithm using only the rand/1 mutation already solved these test cases very well, and these is little space for performance improvement.

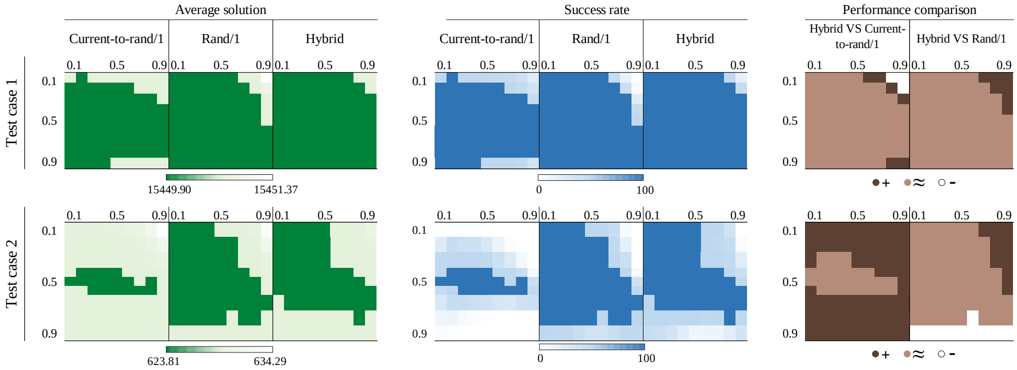

Fig 6 Heat maps of experimental results of test cases that are easy to solve by algorithms using a single mutation or hybrid mutation

The hybrid mutation strategy negatively impacts the proposed algorithm in only two test cases, case 3 and case 13, as shown in Fig. 7. These two test cases are related; actually, test case 13 are an enlarged version of test case 3. We can find that the rand/1 mutation performs much better than the current-to-rand/1 mutation does, and thus hybridizing them does not bring positive effects.

Based on our observation, the proposed algorithm with the hybrid mutation strategy performs quite well in most test cases with *CR* around 0.1 to 0.3 and *F* less than 0.6. Therefore, we set the *CR* parameter to 0.1 and the *F* parameter to 0.5 for our algorithm in all following experiments.

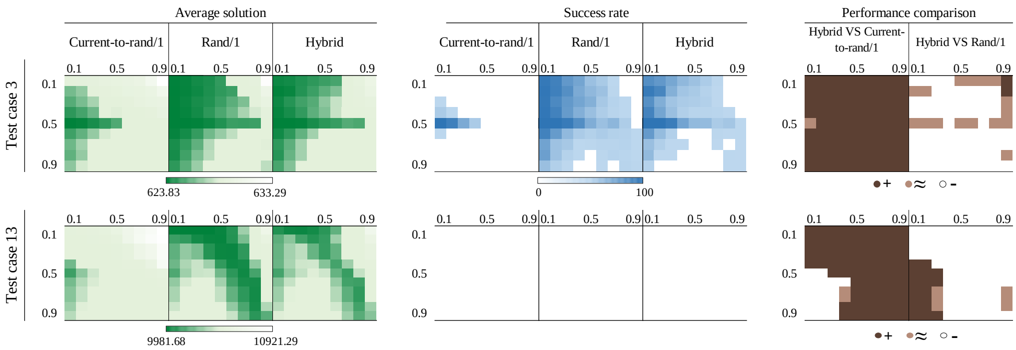

Fig 7 Heat maps of experimental results of test cases that show negative impact of the hybrid mutation strategy

5.2.2 *Performance comparison of repair mechanisms*

This experiment was conducted to observe the effect of the improved single-unit repair mechanism (ISR) in ours L-HMDE. Four repair mechanisms were compared with our mechanism: single-unit repair mechanism (SR) [48], [130], multiple-unit repair mechanism (MR) [74], multiple-unit repair mechanism with proportional adjustment (MRPA) [62], and quadratic formula (QDT) [119], [128]. These mechanisms were briefly described in subsection 2.4.

We replaced the ISR in L-HMDE by the SR, MR, and MRPA respectively to solve all test cases and by the QDT to solve only test cases 1, 5, and 6 as the QDT was specially designed for the transmission loss. The performance of each mechanism was evaluated in terms of the average cost. Table 7 presents the comparison results; the abbreviation Avg and Std stand for the average and standard deviation of the cost over 100 runs, respectively. All parameters of L-HMDE were set following Tables 3 and 4. The comparison shows that ISR significantly outperforms SR in 16 test cases, MR in 22 test cases, MRPA in 20 test cases, and QDT in 5 (out of 8) test cases. (Hereafter we regard each version of a test case as an individual test case, and thus now we have 22 test cases in total.) MR and MRPA cannot help the algorithm to find high-quality solutions in solving large-scale test cases (test case 9–13); SR, MR, and MRPA cannot solve test case 8 well, either. ISR performs worse than other repair mechanisms in only few test cases; it is significantly worse than MRPA in 2 test cases and QDT in 1 test case.

We further tested the four versions of L-HMDE using ISR, SR, MR, and MRPA with higher computational budget (i.e. larger $NFE_{\max}$). The average cost obtained by the four algorithms consuming different NFE in solving test cases 8.1, 9, and 10 is presented in Table 8. The results showed that using the existing repair mechanisms SR, MR, and MRPA leads to a slow convergence progress. Although the algorithms using those repair mechanisms still improve the solution quality gradually as the NFE increases, they could not find the solution as good as the solution by the algorithm using our ISR even they consumed three times of NFE. In contrast, our ISR helps to find high-quality solutions effectively and efficiently. As we mentioned in subsection 3.4, ISR aims to fix the infeasible solutions with smaller modification and within fewer trials. These could help to keep the search direction, obtain feasible solutions, and hence improve the final performance of the algorithm.

Table 7 Result comparison between our repair mechanism and four widely used repair mechanisms in ED problems

Test case	ISR	SR	MR	MRPA	QDT
	Avg (Std)	Avg (Std)	Avg (Std)	Avg (Std)	Avg (Std)
1	15449.90 ($3.24 \cdot 10^{-7}$)	15449.90 ($4.33 \cdot 10^{-7}$) ≈	15449.90 ($7.86 \cdot 10^{-4}$) +	15449.90 ($7.09 \cdot 10^{-7}$) -	15449.90 ($8.35 \cdot 10^{-7}$) +
2	623.81 ($2.00 \cdot 10^{-4}$)	623.81 ($2.60 \cdot 10^{-4}$) +	623.95 (0.24) +	623.81 ($3.08 \cdot 10^{-5}$) -	
3	623.83 ($6.38 \cdot 10^{-4}$)	623.83 ($7.47 \cdot 10^{-4}$) ≈	624.04 (0.03) +	623.95 (0.02) +	
4.1	17964.90 (2.98)	17971.13 (3.56) +	18030.78 (33.61) +	17998.53 (25.61) +	
4.2	24169.96 (0.42)	24170.14 (1.32) +	24198.59 (31.54) +	24181.09 (17.93) +	
4.3	17961.22 (2.44)	17967.66 (3.51) +	18028.15 (32.74) +	17994.13 (23.92) +	
4.2	24164.15 (0.67)	24171.09 (24.00) +	24198.77 (36.63) +	24177.55 (16.97) +	
5.1	24514.88 ($1.56 \cdot 10^{-4}$)	24519.76 (19.49) +	24559.03 (52.42) +	24522.57 (22.17) +	24519.56 (17.39) +
5.2	24516.28 (7.97)	24521.53 (25.96) +	24556.84 (54.09) +	24521.04 (15.47) +	24518.47 (14.69) ≈
5.3	24514.82 ($7.02 \cdot 10^{-5}$)	24520.86 (23.09) +	24562.36 (55.09) +	24530.21 (36.57) +	24519.30 (16.78) +
5.4	24513.04 (6.09)	24521.35 (24.78) +	24558.01 (54.10) +	24522.05 (18.75) +	24515.82 (14.98) +
6.1	32704.45 ($1.65 \cdot 10^{-5}$)	32704.45 ($3.62 \cdot 10^{-4}$) +	32708.51 (3.78) +	32710.65 (3.60) +	32704.45 ($9.85 \cdot 10^{-7}$) -
6.2	32588.92 ($1.83 \cdot 10^{-7}$)	32588.92 ($5.63 \cdot 10^{-5}$) +	32589.17 (1.15) +	32592.30 (3.72) +	32588.92 ($5.10 \cdot 10^{-8}$) ≈
7	62456.63 ($3.33 \cdot 10^{-5}$)	62456.63 ($4.28 \cdot 10^{-5}$) ≈	62456.63 ($9.11 \cdot 10^{-4}$) +	62456.63 ($1.13 \cdot 10^{-4}$) +	62456.63 ($6.43 \cdot 10^{-5}$) +
8.1	121417.94 (3.81)	121477.36 (32.23) +	122185.22 (141.48) +	121481.52 (41.40) +	
8.2	121409.43 (4.51)	121465.44 (32.28) +	122150.01 (112.47) +	121460.05 (37.69 +	
8.3	121375.89 (4.85)	121431.46 (28.20) +	122149.48 (139.16) +	121447.89 (35.97) +	
9	197988.18 ($8.80 \cdot 10^{-8}$)	197988.18 ($5.80 \cdot 10^{-8}$) ≈	204654.59 (968.01) +	201486.84 (538.30) +	
10	1657962.73 ($1.07 \cdot 10^{-3}$)	1657965.61 (14.88) +	1753592.79 (9105.74) +	1733826.76 (6753.14) +	
11	1559708.45 ($2.69 \cdot 10^{-4}$)	1559719.55 (27.81) +	1642539.95 (10393.88) +	1612998.26 (7338.46) +	
12	1655679.43 ($1.82 \cdot 10^{-4}$)	1655679.43 ($7.07 \cdot 10^{-5}$) ≈	1706658.04 (5372.19) +	1701636.37 (5442.47) +	
13	9983.69 (0.12)	9983.71 (0.15) ≈	10081.49 (12.37) +	9996.82 (5.26) +	
	+/≈/-	16/6/0	22/0/0	20/0/2	5/2/1

Table 8 Comparison of the average solution quality at different periods of NFE of our repair mechanism, SR, MR, and MRPA

Test case	Repair mechanism	Avg				
		NFE = 50000	NFE =75000	NFE =100000	NFE =125000	NFE =150000
8.1	ISR	121417.94	121416.37	121415.96	121415.80	121415.63
	SR	121477.36	121450.65	121442.71	121439.90	121438.78
	MR	122185.22	121690.33	121530.18	121478.85	121464.09
	MRPA	121481.52	121462.86	121456.43	121452.54	121451.02
9	ISR	197988.18	197988.18	197988.18	197988.18	197988.18
	SR	197988.18	197988.18	197988.18	197988.18	197988.18
	MR	204654.59	200987.63	199719.35	199237.87	198947.79
	MRPA	201486.84	199725.33	198896.74	198504.30	198310.56
10	ISR	1657962.73	1657962.73	1657962.73	1657962.73	1657962.73
	SR	1657965.61	1657962.73	1657962.73	1657962.73	1657962.73
	MR	1753592.79	1709679.50	1694655.14	1687742.45	1682462.64
	MRPA	1733826.76	1705306.47	1688840.50	1678453.38	1671656.69

5.2.3 *The effect of the linear population size reduction mechanism*

The impact of the linear population size reduction mechanism on the proposed algorithm was investigated in this experiment. The solution quality of the two versions of algorithms with and without linear population size reduction (hereafter called L-HMDE and HMDE) was compared. The population size of HMDE was set by 15 and remained the same in the whole search process. On the other hand, the population size of L-HMDE was initially set by 15 and linearly reduced to four. The other parameters of both algorithms were set as the values in Tables 3 and 4. The Wilcoxon rank-sum test was utilized to check the difference between L-HMDE and HMDE at a significant level of 5%.

In Table 9, the abbreviation Avg and Std stand for the average and standard deviation of the cost obtained over 100 runs. L-HMDE significantly outperforms HMDE in solving 15 out of 22 test cases and is outperformed in no test case. The linear population size reduction mechanism is particularly useful when medium- and large-scale test cases (test case 8–13) are solved.

Table 9 Result comparison of the proposed algorithm with/without linear population size reduction mechanism

Test case	L-HMDE Avg (Std)	HMDE Avg (Std)	Test case	L-HMDE Avg (Std)	HMDE Avg (Std)
1	15449.90 ($3.24\cdot10^{-7}$)	15449.90 ($2.05\cdot10^{-7}$) ≈	6.1	32704.45 ($1.65\cdot10^{-5}$)	32704.45 ($5.14\cdot10^{-4}$) +
2	623.81 ($2.00\cdot10^{-4}$)	623.81 ($1.08\cdot10^{-3}$) +	6.2	32588.92 ($1.83\cdot10^{-7}$)	32588.92 ($1.53\cdot10^{-4}$) +
3	623.83 ($6.38\cdot10^{-4}$)	623.83 ($1.47\cdot10^{-3}$) +	7	62456.63 ($3.33\cdot10^{-5}$)	62456.63 ($3.37\cdot10^{-5}$) ≈
4.1	17964.90 (2.98)	17965.32 (2.93) +	8.1	121417.94 (3.81)	121420.13 (3.77) +
4.2	24169.96 (0.42)	24170.01 (0.66) ≈	8.2	121409.43 (4.51)	121411.86 (6.99) +
4.3	17961.22 (2.44)	17962.47 (3.65) +	8.3	121375.89 (4.85)	121376.08 (4.52) ≈
4.2	24164.15 (0.67)	24164.46 (3.64) ≈	9	197988.18 ($8.80\cdot10^{-8}$)	197988.20 (0.01) +
5.1	24514.88 ($1.56\cdot10^{-4}$)	24515.55 (6.76) +	10	1657962.73 ($1.07\cdot10^{-3}$)	1657984.46 (16.01) +
5.2	24516.28 (7.97)	24517.45 (16.26) ≈	11	1559708.45 ($2.69\cdot10^{-4}$)	1559728.53 (13.57) +
5.3	24514.82 ($7.02\cdot10^{-5}$)	24514.82 ($6.40\cdot10^{-4}$) +	12	1655679.43 ($1.82\cdot10^{-4}$)	1655688.07 (1.87) +
5.4	24513.04 (6.09)	24512.43 ($1.45\cdot10^{-4}$) ≈	13	9983.69 (0.12)	9984.41 (0.13) +
				+/≈/-	15/7/0

5.3 Performance comparison with algorithms for the ED problem

The performance comparison between L-HMDE and existing algorithms is discussed in this subsection. For each test case, we present the statistical results of the best 15 algorithms (including our L-HMDE) in the literature. (There could be fewer than 15 algorithms when a test case is not widely studied.) The statistical results include the minimum (Min), the maximum (Max), the average (Avg), and the standard deviation (Std) of the cost of the solutions obtained by an algorithm over multiple runs. An algorithm is considered only when the detailed solution was reported in the paper and the cost of the solution was confirmed to be the same as the cost reported in the paper. We also estimated the NFE of these algorithms by the product of the population size and the number of generations/iterations reported in the paper. In the following tables, algorithms are ranked in the hierarchical order of Min, Avg, and NFE. The detailed solutions obtained by L-HMDE are given in the section of Appendix for reference. The power outputs P_j of solutions are presented with eight decimals to maintain the solution accuracy.

5.3.1 *Test case 1: the system with six generators with the transmission loss*

Table 10 gives the performance results of L-HMDE and the other 14 effective algorithms in solving test case 1. The best solution obtained by L-HMDE is presented in Appendix A.2. In this test case, some papers provided solutions with smaller cost than the solutions in Table 10; however, their solutions had inaccurate transmission loss or high error with respect to the power balance constraint. The concern about solution accuracy and error in test case 1 was also mentioned in [7], [83], [107], [133]. Therefore, these results are not included in our comparison.

In Table 10, our proposed L-HMDE is the fourth place of the top 15 algorithms. Although RCBA and LM found lower costs (15449.61 and 15449.80 respectively) than L-HMDE did (15449.90), we found that their solution had a relatively high error ($8.66\cdot10^{-2}$ and $2.90\cdot10^{-3}$ respectively) with respect to the power balance constraint than the solution of L-HMDE did ($3.42\cdot10^{-9}$). When we ran L-HMDE with a larger tolerance error ε as $8\cdot10^{-2}$, L-HMDE could obtain a solution with cost 15449.62, which is very close to that of those two algorithms. The detailed information is provided in Appendix A.2. ST-IRDPSO found a solution with a slightly lower cost than that of the solution of L-HMDE; however, it consumed much more NFE and its performance was not stable, as shown by a large Std value. As for the remaining nine algorithms, L-HMDE could achieve better solution quality using fewer computational efforts.

Table 10 Performance comparison for test case 1

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	RCBA [133]	2018	15449.61	-	-	-	10000
2	LM [34]	2022	15449.80	-	-	-	-
3	ST-IRDPSO [83]	2017	15449.89	15450.70	-	1.42	4000
4	**L-HMDE**		**15449.90**	**15449.90**	**15449.90**	**$3.24 \cdot 10^{-7}$**	**1500**
5	MABC [107]	2015	15449.90	15449.90	15449.90	$6.04 \cdot 10^{-8}$	2000
6	MCSA [128]	2018	15449.90	15449.90	15449.90	$1.64 \cdot 10^{-11}$	5000
7	MHS [102]	2014	15449.90	15449.90	15449.90	$1.76 \cdot 10^{-7}$	8000
8	CMFA [56]	2018	15449.90	15449.90	15449.90	$8.96 \cdot 10^{-6}$	10000
9	BSA [46]	2016	15449.90	15449.90	15449.91	$1.00 \cdot 10^{-3}$	3000
10	DHS [55]	2013	15449.90	15449.93	15449.99	$2.04 \cdot 10^{-2}$	3000
11	MSSA [137]	2016	15449.90	15449.94	15453.55	$3.65 \cdot 10^{-1}$	12000
12	MPSO-TVAC [73]	2014	15449.91	15450.17	15451.57	$3.70 \cdot 10^{-1}$	15000
13	EPSO [86]	2013	15449.94	15450.35	15452.00	-	-
14	NPSO-LRS [70]	2007	15450.00	15450.50	15452.00	-	-
15	PSO [6]	2003	15450.00	15454.00	15492.00	$2.00 \cdot 10^{-4}$	20000

5.3.2 *Test case 2-3: the system with ten generators considering multiple types of fuel*

Test cases 2 and 3 are systems with ten generators that use the same coefficient values. The difference between these two test cases is that only test case 3 considers the valve-point effect. The performance comparison between L-HMDE and the other effective algorithms are given in Tables 11 and 12, respectively. The best solutions achieved by L-HMDE for test cases 2 and 3 are listed in Appendix B.2.

Our literature review found 12 studies that applied their algorithms to solve test case 2. Their results are listed in Table 11. L-HMDE is the first place. We can see that test case 2 is an easy problem to solve; top nine algorithms could find the minimal cost in the best case, and top five algorithms could even find the minimal cost in the worst case. L-HMDE offered good solution quality and consumed the fewest NFE among all 12 algorithms.

Table 11 Performance comparison for test case 2

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	**L-HMDE**		**623.81**	**623.81**	**623.81**	**$2.00 \cdot 10^{-4}$**	**2000**
2	IPSO [98]	2013	623.81	623.81	623.81	$1.35 \cdot 10^{-5}$	3000
3	ICDEDP [57]	2008	623.81	623.81	623.81	-	4000
4	SDE [60]	2013	623.81	623.81	623.81	-	9000
5	DE [63]	2008	623.81	623.81	623.81	-	12000
6	ALHN [40]	2013	623.81	625.94	626.25	$8.26 \cdot 10^{-1}$	-
7	PPSO [90]	2019	623.81	-	-	-	20000
8	IGA-MU [9]	2005	623.81	-	-	-	-
9	MPSO [97]	2005	623.81	-	-	-	-
10	HM [10]	1984	625.18	-	-	-	-
11	MHNN [135]	1993	626.12	-	-	-	-
12	AHNN [134]	1998	626.24	-	-	-	-

For test case 3, the comparison results are presented in Table 12. Although some studies reported lower cost values than the results in Table 12, the cost values re-calculated from the detailed solutions in these papers did not match their reported cost values, as discussed in [7], [83], [56]. Thus, those results were excluded in our comparison. Test case 3 is also an easy problem. All top 15 algorithms could find the minimal cost in the best case. SDE is the best algorithm, and our L-HMDE is the second place. CCPSO is the only algorithm that reported a lower Std value than L-HMDE did, but it consumed 30 times of NFE of L-HMDE.

Table 12 Performance comparison for test case 3

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	SDE [60]	2013	623.83	623.83	623.83	-	9000
2	**L-HMDE**		**623.83**	**623.83**	**623.83**	**$6.38 \cdot 10^{-4}$**	**10000**
3	IODPSO-L [75]	2017	623.83	623.83	623.83	0.00	15000
4	DHS [55]	2013	623.83	623.83	623.83	-	50000
5	CQGSO [136]	2012	623.83	623.83	623.85	-	120000
6	CCPSO [91]	2010	623.83	623.83	623.83	$5.00 \cdot 10^{-4}$	300000
7	DPSOEP [84]	2017	623.83	623.84	623.85	-	60000
8	ARCGA [100]	2010	623.83	623.84	623.86	-	-
9	TFWO [145]	2020	623.83	623.85	-	$9.80 \cdot 10^{-3}$	8000
10	PPSO [90]	2019	623.83	623.85	-	$9.80 \cdot 10^{-3}$	20000
11	RCGA [99]	2009	623.83	623.85	623.88	-	1000
12	CCEDE [61]	2016	623.83	623.86	623.89	$7.60 \cdot 10^{-3}$	7000
13	CMFA [56]	2018	623.83	623.87	623.91	$1.89 \cdot 10^{-2}$	10000
14	CACO-LD-AP [147]	2022	623.83	623.89	624.02	$2.95 \cdot 10^{-2}$	-
15	DEPSO [51]	2013	623.83	623.90	624.08	-	25000

5.3.3 *Test case 4: the system with 13 generators with the valve-point effect*

The solution results of L-HMDE and existing algorithms in solving test cases 4.1 to 4.4 are presented in Tables 13 to 16, respectively. The best solutions obtained by L-HMDE for these test cases are given in Appendix C.3.

Table 13 shows the results of solving test case 4.1. Our L-HMDE is the seventh place. Among the 13 algorithms that could find the minimal cost, L-HMDE consumed the fifth fewest NFE. It took fewer NFE to achieve lower average cost than two algorithms (DE and ORCSA). In general, we can observe a trade-off between computational effort (NFE) and performance stability (Std).

Table 14 shows the results of solving test case 4.2. We can separate the top six algorithms into two groups: the top two algorithms could find the minimal cost (24169.91) but consumed large NFE and provided unstable performance; the next four algorithms found a slightly higher cost (24169.92) but provided stable performance. Our L-HMDE is in the second group, and it consumed the fewest NFE among the top six algorithms.

Table 13 Performance comparison for test case 4.1

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	MABC [107]	2015	17963.83	17963.83	17963.83	$2.26 \cdot 10^{-4}$	216000
2	MPDE [62]	2019	17963.83	17963.83	17963.83	0.00	1080000
3	ESSA [144]	2020	17963.83	17963.92	17964.41	$1.05 \cdot 10^{-1}$	800000
4	HAAA [142]	2018	17963.83	17963.84	17963.93	$1.90 \cdot 10^{-2}$	1187106
5	MsEBBO [53]	2013	17963.83	17964.05	17969.03	1.92	80000
6	FV-ICLPSO [76]	2022	17963.83	17964.09	17969.22	1.0397	100000
7	**L-HMDE**		**17963.83**	**17964.90**	**17978.14**	**2.98**	**25000**
8	θ-PSO [87]	2013	17963.83	17965.21	17980.20	-	4500
9	DE [63]	2008	17963.83	17965.48	17975.36	-	93600
10	CBA [132]	2016	17963.83	17965.49	17995.23	6.85	12000
11	GSO [141]	2017	17963.83	17968.46	17982.41	3.63	-
12	ORCSA [119]	2015	17963.83	17985.41	18028.56	21.95	200000
13	FMILP [37]	2020	17963.83	-	-	-	-
14	PSO-TVAC [88]	2009	17963.88	18154.56	18358.31	-	6250
15	HQPSO [82]	2008	17963.96	18273.86	18633.04	123.22	16000

Table 14 Performance comparison for test case 4.2

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	Jaya-SML [117]	2019	24169.91	24217.09	24285.89	52.91	90000
2	Ijaya [115]	2020	24169.91	24220.57	24277.55	50.34	250000
3	DE [63]	2008	24169.92	24169.92	24169.92	$4.45 \cdot 10^{-5}$	78000
4	MABC [107]	2015	24169.92	24169.92	24169.92	$5.77 \cdot 10^{-7}$	180000
5	MCSA [128]	2018	24169.92	24169.92	24169.92	$5.86 \cdot 10^{-5}$	25000
6	**L-HMDE**		**24169.92**	**24169.96**	**24174.08**	**$4.20 \cdot 10^{-1}$**	**25000**
7	ORCSA [119]	2015	24169.92	24182.21	24271.92	21.99	200000
8	CPSO-SQP [96]	2012	24190.97	-	-	-	-
9	PSO-SQP [95]	2004	24261.05	-	-	-	10000
10	Interior Point [35]	2019	24383.46	-	-	-	-

Table 15 shows the results of solving test case 4.3. L-HMDE is the fourth place. We can see that this test case may have a challenging landscape for metaheuristics since most algorithms have large Std values. Although 11 algorithms could achieve the minimal cost, large Std values reveal that these algorithms sometimes got stuck at local optimal solutions of high cost. L-HMDE offered the fourth smallest Std value, which shows its robustness of performance.

Table 16 shows the results of solving test case 4.4. L-HMDE is the third place. Only six algorithms could achieve the minimal cost, and L-HMDE is one of them. In addition, L-HMDE consumed the fewest NFE, and its Std value is smaller than DHS, ECSA, and RQEA, which consumed much more NFE.

Table 15 Performance comparison for test case 4.3

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	ESAHJ [150]	2021	17960.36	-	-	-	130000
2	MPDE [62]	2019	17960.37	17960.37	17960.50	$2.70 \cdot 10^{-2}$	1080000
3	DHS [55]	2013	17960.37	17961.12	17968.36	1.92	60000
4	**L-HMDE**		**17960.37**	**17961.22**	**17970.39**	**2.44**	**25000**
5	IDE [48]	2016	17960.37	17961.47	17969.49	2.65	120000
6	IHS [130]	2009	17960.37	17965.42	17971.65	16.95	22500
7	DEL [67]	2014	17960.37	17966.13	17975.41	4.72	24000
8	HAAA [142]	2018	17960.37	17967.56	17990.92	6.79	1187106
9	THS [126]	2016	17960.37	17977.60	-	17.06	50000000
10	NTHS [127]	2018	17960.37	17987.10	-	-	50000000
11	SDE [59]	2013	17960.37	-	-	-	18000
12	SCA- βHC [123]	2023	17960.39	-	17960.96	$5.45 \cdot 10^{-1}$	30000
13	C-GRASP-SaDE [50]	2017	17960.39	17966.11	17968.87	2.70	24000
14	MDE [44]	2010	17960.39	17967.19	17969.09	-	280000
15	CDEMD [47]	2009	17961.94	17974.69	18061.41	20.31	25000

Table 16 Performance comparison for test case 4.4

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	IDE [48]	2016	24164.05	24164.05	24164.05	$2.55 \cdot 10^{-8}$	120000
2	IODPSO-G [75]	2017	24164.05	24164.13	24164.79	$2.30 \cdot 10^{-1}$	60000
3	**L-HMDE**		**24164.05**	**24164.15**	**24168.81**	$\mathbf{6.70 \cdot 10^{-1}}$	**25000**
4	DHS [55]	2013	24164.05	24164.53	24168.81	1.14	40000
5	ECSA [120]	2023	24164.05	24168.61	-	15.959	10000000
6	RQEA [101]	2008	24164.05	-	-	-	50000
7	NRHS [127]	2018	24164.06	24185.61	-	-	50000000
8	THS [126]	2016	24164.06	24195.21	-	30.21	50000000
9	ESAHJ [150]	2021	24164.06	-	-	-	130000
10	SCA- βHC [123]	2023	24164.09	24164.38	-	$2.84 \cdot 10^{-1}$	30000
11	ADE-MMS [49]	2019	24164.12	24168.97	24255.61	23.67	8000
12	ABC [105]	2014	24166.22	-	-	-	100000
13	SDE [59]	2013	24169.92	-	-	-	18000

5.3.4 *Test case 5: the system with 13 generators with the valve-point effect and the transmission loss*

The solution results of L-HMDE and existing algorithms in solving test cases 5.1 to 5.4 are presented in Tables 17 to 20, respectively. The best solutions obtained by L-HMDE for these test cases are given in Appendix D.6. The 13-unit test cases were not widely studied in the literature. Thus, we only listed eight algorithms in Table 17, four in Table 18, and three in Tables 19 and 20.

Table 17 shows the results of solving test case 5.1. Among the eight algorithms, only half of them could achieve the minimal cost stably. L-HMDE is the second place and consumed the fewest NFE among the top four algorithms. MCSA consumed the same number of NFE and achieved even lower Std value than L-HMDE. We will have a deeper investigation of its design and consider integrating its feature in our algorithm in the future.

Table 17 Performance comparison for test case 5.1

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	MCSA [128]	2018	24514.88	24514.88	24514.88	$3.12 \cdot 10^{-7}$	25000
2	**L-HMDE**		**24514.88**	**24514.88**	**24514.88**	$\mathbf{1.56 \cdot 10^{-4}}$	**25000**
3	MABC [107]	2015	24514.88	24514.88	24514.88	$3.50 \cdot 10^{-7}$	180000
4	MPDE [62]	2019	24514.88	24514.88	24514.88	0.00	900000
5	SDE [59]	2013	24514.88	24516.31	-	-	18000
6	DSOS [143]	2020	24514.88	-	-	-	15000
7	Self-tuning HDE [66]	2007	24560.08	24706.63	24872.44	-	12500
8	MHSA [131]	2014	24585.36	24638.37	24711.30	-	45000

Tables 18–20 show the results of solving test cases 5.2–5.4. Few studies considered these three test cases. Our L-HMDE and MPDE are the only two algorithms that could achieve the minimal cost for these three cases. The advantage of L-HMDE is that it required less than 3% of NFE of MPDE.

Table 18 Performance comparison for test case 5.2

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	MPDE [62]	2019	24515.23	24515.23	24515.23	0.00	900000
2	**L-HMDE**		**24515.23**	**24516.28**	**24588.31**	**7.97**	**25000**
3	FMILP [37]	2020	24515.23	-	-	-	-
4	FPSOGSA [78]	2015	24515.36	24516.68	-	-	100000

Table 19 Performance comparison for test case 5.3

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	**L-HMDE**		**24514.82**	**24514.82**	**24514.82**	$\mathbf{7.02 \cdot 10^{-5}}$	**25000**
2	MPDE [62]	2019	24514.82	24514.82	24514.82	0.00	900000
3	OIWO [112]	2016	24514.83	24514.83	24514.83	-	-

Table 20 Performance comparison for test case 5.4

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	MPDE [62]	2019	24512.43	24512.43	24512.43	-	900000
2	**L-HMDE**		**24512.43**	**24513.04**	**24573.30**	**6.09**	**25000**
3	OGWO [111]	2018	24512.72	24512.85	24513.09	$9.83 \cdot 10^{-2}$	5000

5.3.5 *Test case 6: the system with 15 generators with the transmission loss*

The solution results of L-HMDE and existing algorithms in solving test cases 6.1 and 6.2 are presented in Tables 21 and 22, respectively. The best solutions obtained by L-HMDE for these test cases are given in Appendix E.4. Test case 6.2 was less popularly examined in the literature, and thus only six algorithms are listed in Table 22.

Table 21 lists the results of top 15 algorithms in solving test case 6.1. Even though ESSA obtained a smaller cost than L-HMDE, it still had a higher error ($2.70 \cdot 10^{-1}$) than ours ($2.99 \cdot 10^{-9}$). L-HMDE outperforms ten algorithms in terms of both solution quality and computational efficiency. It is outperformed only by CLCS-CLM, which consumed slightly fewer NFE and achieved slightly lower Std value than L-HMDE.

Table 22 lists the results of six algorithms in solving test case 6.2. The top three algorithms achieved better solutions than L-HMDE. However, DEPSO and DE consumed eight to nine times of NFE of L-HMDE, and their performance is not stable. Moreover, the errors of their solutions (DEPSO = $1.00 \cdot 10^{-2}$, DE = $7.00 \cdot 10^{-3}$) with respect to the power balance constraints are much larger than that of L-HMDE ($1.35 \cdot 10^{-10}$). L-HMDE is the fourth place. It offers good solution quality stably and efficiently.

Table 21 Performance comparison for test case 6.1

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	ESSA [144]	2020	32701.21	32701.22	32701.22	$5.00 \cdot 10^{-3}$	80000
2	CLCS-CLM [129]	2020	32704.45	32704.45	32704.45	$8.79 \cdot 10^{-6}$	4500
3	**L-HMDE**		**32704.45**	**32704.45**	**32704.45**	$\mathbf{1.65 \cdot 10^{-5}}$	**5000**
4	CTPSO [91]	2010	32704.45	32704.45	32704.45	0.00	300000
5	BSA [46]	2016	32704.45	32704.47	32704.58	$2.80 \cdot 10^{-2}$	5000
6	WCA [140]	2017	32704.45	32704.51	32704.52	$4.51 \cdot 10^{-5}$	60000
7	SWT-PSO [92]	2013	32704.45	-	-	-	9000
8	MPSO-TVAC [73]	2014	32704.47	32705.80	32728.99	3.51	75000
9	EPSO [86]	2013	32704.83	32725.37	32762.01	-	50000
10	MDE [44]	2010	32704.90	32708.10	32711.50	-	160000
11	Jaya-SML [117]	2019	32706.36	32706.68	32707.29	2.32	150000
12	CACO-LD-AP [147]	2022	32706.38	32712.47	32728.28	5.41	-
13	Ijaya [115]	2020	32706.62	32707.24	32708.59	3.08	500000
14	CSO [118]	2015	32706.66	-	-	-	25000
15	IPSO [98]	2013	32706.66	-	-	-	-

Table 22 Performance comparison for test case 6.2

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	DEPSO [51]	2013	32588.81	32588.99	32591.49	-	40000
2	DE [63]	2008	32588.87	32609.85	32641.42	-	45000
3	**L-HMDE**		**32588.92**	**32588.92**	**32588.92**	$\mathbf{1.83 \cdot 10^{-7}}$	**5000**
4	DHS [55]	2013	32588.92	32588.92	32588.93	$3.47 \cdot 10^{-3}$	24000
5	IDP [38]	2008	32590.00	-	-	-	-
6	PSO [149]	2003	33020.00	-	-	-	20000

5.3.6 *Test case 7: the system with 20 generators with the transmission loss*

Based on our literature review, we listed the results of 12 algorithms in solving test case 7 in Table 23. DSOS [143] achieved a better solution than all listed algorithms did, but it is not included since its solution has a large error (larger than 0.2) with respect to the power balance constraint. The best solution obtained by L-HMDE is presented in Appendix F2.

ADE-MMS is the only algorithm that could achieve the minimal cost. A disadvantage of ADE-MMS is that its performance is less stable than other algorithms. ORCSA and our L-HMDE are the second and third place, respectively. They offered very similar solution quality and consumed the same NFE. MCSA and CQGSO could also achieve good and robust solution quality, but they required much more NFE.

Table 23 Performance comparison for test case 7

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	ADE-MMS [49]	2019	62456.51	62456.64	62457.06	$1.32\cdot10^{-1}$	8000
2	ORCSA [119]	2015	62456.63	62456.63	62456.63	$3.00\cdot10^{-5}$	5000
3	**L-HMDE**		**62456.63**	**62456.63**	**62456.63**	$\mathbf{3.33\cdot10^{-5}}$	**5000**
4	MCSA [128]	2018	62456.63	62456.63	62456.63	$1.21\cdot10^{-11}$	40000
5	CQGSO [136]	2012	62456.63	62456.63	62456.63	-	120000
6	CBA [132]	2016	62456.63	62456.63	62501.67	$3.88\cdot10^{-1}$	12000
7	GABC [106]	2014	62456.63	62456.69	62456.72	$1.70\cdot10^{-2}$	5000
8	CACO-LD-AP [147]	2022	62456.63	62513.52	62554.37	20.80	-
9	HNN [5]	2000	62456.63	-	-	-	-
10	λ-Logic Based [39]	2009	62456.63	-	-	-	-
11	FMILP [37]	2020	62456.63	-	-	-	-
12	BSA [45]	2014	62456.69	62457.15	62458.13	-	400000
13	BBO [54]	2010	62456.79	62456.79	62456.79	-	20000

5.3.7 *Test case 8: the system with 40 generators with the valve-point effect*

The solution results of L-HMDE and existing algorithms in solving test cases 8.1−8.3 are presented in Tables 24–26, respectively. The best solutions obtained by L-HMDE for these test cases are given in Appendix G.4. Test cases 8.2 and 8.3 were less popularly examined in the literature, and thus only nine and five algorithms are listed in Tables 25 and 26, respectively.

Table 24 Performance comparison for test case 8.1

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	ESSA [144]	2020	121412.50	121450.60	121517.00	31.02	1600000
2	C-MIMO-CSO [139]	2019	121412.50	121454.20	121517.80	28.81	1400000
3	MsEBBO [53]	2013	121412.53	121417.19	121450.00	5.80	80000
4	CACO-LD-AP [147]	2022	121412.53	121428.16	121439.76	9.26	-
5	GSK-DE [58]	2023	121412.53	121451.19	121506.66	28.1149	400000
6	PPSO [90]	2019	121412.54	121412.59	121413.95	$5.63\cdot10^{-2}$	120000
7	MPDE [62]	2019	121412.54	121412.62	121414.62	$4.09\cdot10^{-1}$	2400000
8	CLCS-CLM [129]	2020	121412.54	121412.99	121414.67	$7.59\cdot10^{-1}$	90000
9	CCEDE [61]	2016	121412.54	121413.00	121414.69	$9.74\cdot10^{-2}$	70000
10	FPSOGSA [78]	2015	121412.54	121413.56	121414.98	-	300000
11	MCSA [128]	2018	121412.54	121414.16	121421.12	2.75	80000
12	SDE [59]	2013	121412.54	121415.72	121418.58	-	60000
13	**L-HMDE**		**121412.54**	**121417.94**	**121426.34**	**3.81**	**50000**
14	FV-ICLPSO [76]	2022	121412.54	121419.66	121424.27	3.2791	200000
15	DCPSO [74]	2014	121412.54	121423.13	121516.89	-	250000

Table 24 lists the results of top 15 algorithms in solving test case 8.1. We can separate the top nine algorithms into three groups. The algorithms of ranks 1, 2, and 5 consumed large NFE but still had unstable performance (large Std values). The algorithms ESSA and C-MIMO-CSO obtained the lowest cost, but their solutions have relative larger errors (9.90e-3 and 3.7e-3 respectively) with respect to the problem constraints. The algorithms MsEBBO and CACO-LD-AP achieved a slightly higher cost (121412.53) more stably using fewer NFE. The last four algorithms provided high quality solutions quite stably; among them, CCEDE consumed the fewest NFE. Test case 8.1 is the most challenging case to L-HMDE. In fact, it is one of the only two test cases that L-HMDE is not among the top six algorithms. L-HMDE could achieve the same cost in the best case as other ten algorithms did by using the fewest NFE. However, we still need to think of how to improve its solution quality without increasing too more NFE.

Tables 25 and 26 present the results of solving test cases 8.2 and 8.3, respectively. Not many studies solved these two test cases. L-HMDE is ranked second and first place, respectively. Regarding test case 8.2, L-HMDE outperforms all algorithms except MPDE and DEC-SQP in terms of solution quality, stability, and computational efficiency simultaneously. MPDE achieved more stable solution quality than

L-HMDE but meanwhile consumed much more NFE (48 times). DEC-SQP consumed much fewer NFE than all others, but its solution quality is much worse. As for test case 8.3, L-HMDE outperforms IDE in terms of solution quality and efficiency. The results of the other three algorithms show the trade-off between solution quality and computational effort.

Table 25 Performance comparison for test case 8.2

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	MPDE [62]	2019	121403.54	121403.66	121405.62	$4.95 \cdot 10^{-1}$	2400000
2	**L-HMDE**		**121403.54**	**121409.43**	**121429.09**	**4.51**	**50000**
3	DHS [55]	2013	121403.54	121410.60	121417.23	4.80	240000
4	CCPSO [91]	2010	121403.54	121445.33	121525.49	32.49	300000
5	HAAA [142]	2018	121403.70	121425.56	121428.90	5.25	1947546*
6	HcSCA [124]	2021	121403.87	121537.00	121913.32	105.98	867030*
7	IDE [48]	2016	121411.49	121429.04	121468.73	16.83	160000
8	DEC-SQP [64]	2006	121741.98	122295.13	122839.29	386.18	18000
9	Interior Point [35]	2023	122264.88	-	-	-	-

* Ref. [142], [124] only presented the maximum NFE over 30 runs.

Table 26 Performance comparison for test case 8.3

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	**L-HMDE**		**121369.08**	**121375.89**	**121403.13**	**4.85**	**50000**
2	IDE [65]	2014	121370.13	121372.28	121376.01	-	60000
3	ADE-MMS [49]	2019	121370.82	121428.65	121539.50	38.87	24000
4	MPSO [93]	2015	121379.43	121384.43	121391.07	-	20000
5	FCEP [114]	2017	121393.00	121394.00	121395.00	-	30000

5.3.8 *Test case 9: the system with 110 generators*

Test case 9 is a large-scale problem. It was provided in [112] in 2015, and hence there are still not many studies working on it. We listed the results of nine algorithms in Table 27. The best solution obtained by L-HMDE is presented in Appendix H.2. L-HMDE is ranked second place. It outperforms all algorithms except HcSCA in terms of solution quality, stability, and computational efficiency simultaneously. HcSCA found a solution with a slightly lower cost, but it consumed more than 20 times of NFE of L-HMDE.

Table 27 Performance comparison for test case 9

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	HcSCA [124]	2021	197988.17	197988.17	197988.17	$8.79 \cdot 10^{-4}$	1052020*
2	**L-HMDE**		**197988.18**	**197988.18**	**197988.18**	$\mathbf{8.80 \cdot 10^{-8}}$	**50000**
3	GSK-DE [58]	2023	197988.18	197988.18	197988.18	$1.17 \cdot 10^{-4}$	110000
4	TFWO [145]	2020	197988.18	197988.18	197988.19	$6.80 \cdot 10^{-3}$	160000
5	HIWO [116]	2019	197988.19	197988.20	197988.20	$2.50 \cdot 10^{-3}$	60000
6	OIWO [112]	2015	197989.14	197989.41	197989.93	-	-
7	DSOS [143]	2020	198007.60	-	-	-	500000
8	ISMA [146]	2021	198565.90	198782.10	198949.10	153.465	5000000
9	EBWO [113]	2023	199417.20	201729.10	205262.60	1869.996	1500000

* Ref. [124] only presented the maximum NFE over 30 runs.

5.3.9 *Test case 10−12: the system with 140 generators*

Test cases 10 to 12 are large-scale systems with 140 generators that take the same coefficient values but consider different problem constraints. Test case 10 considers the valve-point effect, the ramping rate, and prohibited zones. Test case 11 ignores the ramping rate, and test case 12 ignores the valve-point effect. The solution results of L-HMDE and existing algorithms in solving test cases 10−12 are presented in Tables 28–30, respectively. The best solutions obtained by L-HMDE for these test cases are given in Appendix I.2. We found that in the literature some studies compared experimental results across test cases. This should be avoided since these test cases have different problem characteristics and optimal solutions.

Table 28 lists the results of eight algorithms in solving test case 10. Although L-HMDE is the fifth place, the top four algorithms got a very small reduction of cost by using at least 4.5 times of NFE of L-HMDE. In addition to high computational efficiency, L-HMDE is also good for stability. It provides the second lowest average cost and the smallest Std value among all algorithms.

Test case 11 is the most popular one among the three 140-unit test cases. Table 29 lists the results of the top 15 algorithms in solving this case. L-HMDE is the fourth place. It outperforms nine algorithms in terms of solution quality, stability, and computational efficiency. CLCS-CLM is slightly more stable than L-HMDE by using 3.6 times of NFE of L-HMDE, while C-MIMO-CSO achieved slightly lower cost by using 30 times of NFE. ESSA obtained the minimal cost by using 60 times of NFE of L-HMDE; besides, its solution has a considerable error ($7 \cdot 10^{-2}$) with respect to the power balance constraint.

We only found five algorithms that solved test case 12. Table 30 lists the results. Again, L-HMDE shows its advantage in terms of solution quality, stability, and computational efficiency. There is only one algorithm (HHE) that can achieve lower cost than L-HMDE, but HHE consumed 170 times of NFE of L-HMDE.

Table 28 Performance comparison for test case 10

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	CCEDE [61]	2016	1657962.70	1657963.05	1657965.18	1.15	400000
2	HHE [52]	2014	1657962.71	-	-	-	9500000
3	DCPSO [74]	2015	1657962.72	1657962.72	1657962.72	-	500000
4	DEL [67]	2014	1657962.72	1658001.70	1651518.67	57.98	225000
5	**L-HMDE**		**1657962.73**	**1657962.73**	**1657962.73**	**$1.07 \cdot 10^{-3}$**	**50000**
6	CQGSO [136]	2012	1657962.73	1657962.74	1657962.78	-	120000
7	FMILP [37]	2020	1657964.71	-	-	-	-
8	WCA [140]	2017	1658006.70	1658029.91	1658116.01	37.15	1500000

Table 29 Performance comparison for test case 11

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	ESSA [144]	2020	1559705.00	1559706.00	1559707.00	0.84	3000000
2	C-MIMO-CSO [139]	2019	1559708.00	1559709.54	1559725.00	4.43	1500000
3	CLCS-CLM [129]	2020	1559708.44	1559708.44	1559708.44	$4.23 \cdot 10^{-6}$	180000
4	**L-HMDE**		**1559708.45**	**1559708.45**	**1559708.45**	**$2.69 \cdot 10^{-4}$**	**50000**
5	HcSCA [124]	2021	1559708.47	1559709.98	1559714.50	1.51	2058030
6	MPDE [62]	2019	1559708.81	1559709.06	1559709.43	$3.06 \cdot 10^{-1}$	4800000
7	WCA [140]	2017	1559709.42	-	-	-	1500000
8	HIWO [116]	2019	1559709.53	1559709.70	1559709.90	$8.56 \cdot 10^{-2}$	60000
9	OGWO [111]	2018	1559709.97	1559713.26	1559743.47	$9.36 \cdot 10^{-2}$	5000
10	HAAA [142]	2018	1559710.00	1559712.87	1559731.00	4.14	5236471
11	MSSA [137]	2016	1559708.70	1559708.82	1559709.21	$1.10 \cdot 10^{-1}$	160000
12	GWO [110]	2016	1559953.18	1560132.93	1560228.40	1.02	5000
13	SDE [59]	2013	1560236.85	-	-	-	120000
14	MPSO [93]	2015	1560436.00	1560445.00	1560462.00	-	60000
15	IDE [65]	2014	1564648.66	1564663.54	1564682.73	-	250000

* Ref. [142], [124] only presented the maximum NFE over 30 runs.

Table 30 Performance comparison for test case 12

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	HHE [52]	2014	1655679.41	-	-	-	8500000
2	**L-HMDE**		**1655679.43**	**1655679.43**	**1655679.43**	**$1.82 \cdot 10^{-4}$**	**50000**
3	CQGSO [136]	2012	1655679.43	1655679.43	1655679.43	-	120000
4	PPSO [90]	2019	1655679.89	1655680.97	1655681.81	1.27	440000
5	WCA [140]	2017	1655686.57	-	-	-	1500000

5.3.10 *Test case 13: the system with 160 generators with multiple types of fuels and the valve-point effect*

The solution results of L-HMDE and existing algorithms in solving test case 13 are presented in Table 31. The best solution obtained by L-HMDE is given in Appendix J.1. Test case 13 is the second hardest problem for L-HMDE in our experiments. L-HMDE consumed 150000 NFE to achieve a cost close to the minimal cost obtained by FV-ICLPSO. We observed the solution quality of L-HMDE with different numbers of NFE. Although L-HMDE consumed more NFE than the following five algorithms, it could find solutions with cost less than 10000 with 20000 NFE and solutions with cost less than 9990 with 30000 NFE in all runs. These observations showed that L-HMDE could offer competitive solution quality using the same level of NFE when compared with the algorithms ranked fifth to ninth.

Table 31 Performance comparison for test case 13

Rank	Algorithms	Publication year	Min ($/h)	Avg ($/h)	Max ($/h)	Std	NFE
1	FV-ICLPSO [76]	2022	9981.59	9981.78	9981.92	$7.16 \cdot 10^{-2}$	200000
2	HIWO [116]	2019	9981.79	9982.00	9982.19	$9.00 \cdot 10^{-2}$	60000
3	OIWO [112]	2015	9981.98	9982.99	9984.00	-	-
4	**L-HMDE**		**9983.35**	**9983.69**	**9983.91**	$\mathbf{1.20 \cdot 10^{-1}}$	**150000**
5	CSO [138]	2016	9984.24	9984.92	9986.36	$4.00 \cdot 10^{-1}$	100000
6	CMFA [56]	2018	9985.60	9987.55	9996.94	2.52	25000
7	ORCSA [119]	2015	9989.94	9992.05	9996.83	1.41	96000
8	CBA [132]	2016	10002.86	10006.33	10045.23	9.58	20000
9	BSA [46]	2016	10014.09	10035.40	10060.93	9.04	30000

5.3.11 Summary of performance comparison

In Section 5.3 we compared the performance of our L-HMDE with more than 90 existing algorithms in solving 22 test cases. We comprehensively collected experimental results in the literature and carefully verified their solutions. Then, we compared these algorithms from three aspects: solution quality, stability, and computational efficiency. We count the number of test cases each algorithm is ranked among the top six algorithms as an overall performance indicator. L-HMDE is among the top six for 20 out of 22 test cases. The next four algorithms are MPDE [62], DHS [55], ESSA [144], and CQGSO [136], which are among the top six for only eight, five, four, and four test cases, respectively. This result shows that our L-HMDE can solve a wide set of ED test cases of different scale and with different model characteristics very well. Note that L-HMDE used the same parameter setting (except NFE) to solve all test cases.

By looking into the design of the above five algorithms, we found two important design concepts in common. First, all these algorithms adopted more than one operator to produce new solutions. For example, MPDE used three mutation operators, DHS hybridized DE and HS operators, and CQGSO applied two kinds of operators for two kinds of sub-populations. Second, most of these algorithms adopted some kind of parameter control mechanisms. For example, MPDE used nonlinear decrement method to adjust the scaling factor, and ESSA used the exponential function to control the moving trajectory of the population. The design of our L-HMDE catches these two important concepts. We adopt a hybrid mutation strategy and a linear population size reduction mechanism. They are useful for balancing the exploitation and the exploration, which significantly affects the search ability of metaheuristics. Through quantitative and qualitative analysis, we suggest that researchers who are interested in solving the ED problem may put focus on the research topics of multi-operators and parameter control in the future.

5.4 Performance comparison with general-purpose algorithms

In the previous section, we verified the good performance of our proposed L-HMDE by comparing it with the existing algorithms designed for the ED problem. In this section, we want to compare L-HMDE with three general-purpose algorithms in solving not only the 22 ED test cases but also the benchmark functions of the CEC 2020 competition on single objective bound constrained numerical optimization (hereafter called CEC 2020 benchmark functions) [151]. On one hand, performance comparison between L-HMDE and general-purpose algorithms using the ED test cases can help us to understand whether the ED test cases are really challenging. On the other hand, comparison between these algorithms using the CEC benchmark functions can examine the general problem solving ability of our L-HMDE.

The three general-purpose algorithms to be compared are Success-History-based Adaptive DE (SHADE) [152], L-SHADE [148], and improved multi-operator DE (IMODE) [153]. SHADE is an adaptive DE that controls the values of F and CR based on the history of successfully generating better offspring solutions. L-SHADE extends SHADE by the linear population size reduction mechanism. It was the winner of the CEC 2014 competition on real-parameter single objective optimization. IMODE uses multiple operators to generate new solutions and selects the operator based on the population diversity. It was the winner of the CEC 2020 competition on single objective bound constrained numerical optimization. We used the implementation of SHADE and L-SHADE in the PlatEMO software package [154]. As for IMODE, we used the source code provided by the competition organizers[1].

[1] https://github.com/P-N-Suganthan/2020-Bound-Constrained-Opt-Benchmark

5.4.1 *CEC 2020 benchmark functions*

The CEC 2020 benchmark function set consists of 10 test functions. Detailed definitions and function characteristics are referred to [151]. In our experiments, we set the problem dimension to 15. We ran each algorithm to solve each function for ten times. The performance of each algorithm was assessed by the error between the best-found solution and the global optimum. Parameter settings of the compared algorithms followed the original papers, as shown in Table 32. Experimental results are presented in Table 33. We used the Wilcoxon rank-sum test to check the significance of the difference with the significance level of 0.5.

The experimental result demonstrates that our L-HMDE performs significantly better than SHADE and L-SHADE in 7 and 5 out of 10 functions, respectively. It does not perform significantly worse than these two algorithms in any function. However, L-HMDE outperforms IMODE only in two functions and is outperformed in five functions. Since our L-HMDE is designed specifically to solve the ED problems, it is not surprising that L-HMDE does not perform as well as IMODE, which is a top algorithm designed for general purpose.

Table 32 Parameter setting of the four compared algorithms in solving CEC 2020 benchmark functions

Algorithm	Parameter setting
SHADE [152]	NP = 100, |H| = NP, |A |= NP, $M_{CR}^{initial}$ = 0.5, $M_F^{initial}$ = 0.5, P_i^{best} = rand[2, 0.2·NP], σ =0.1
L-SHADE [148]	$NP^{initial}$ = 100, NP^{final} = 4, |H| = 5, |A|= 2·NP, $M_{CR}^{initial}$ = 0.5, $M_F^{initial}$ = 0.5, P_i^{best} = rand[2, 0.1·NP], σ =0.1, $\perp$ = 0
IMODE [153]	$NP^{initial}$ = 6D², NP^{final} = 4, |H| = 20D, |A|= 2.6·NP, $M_{CR}^{initial}$ = 0.2, $M_F^{initial}$ = 0.2, Ø = [1,0.1·NP], σ =0.1, FFE_{LS} = 0.85FFE^{max}
L-HMDE	$NP^{initial}$ = 100, NP^{final} = 4, CR = 0.1, F = 0.5, δ= 0.7

Table 33 Fitness errors of four algorithms in solving the CEC 2020 benchmark functions

		L-HMDE	IMODE		L-SHADE		SHADE	
F01	Best	***0.0000E+00***	***0.0000E+00***		***0.0000E+00***		***0.0000E+00***	
	Mean	***0.0000E+00***	***0.0000E+00***	≈	***0.0000E+00***	≈	***0.0000E+00***	≈
	Std.	***0.0000E+00***	***0.0000E+00***		***0.0000E+00***		***0.0000E+00***	
F02	Best	***8.3466E-02***	1.6655E-01		2.1631E+01		2.6826E+01	
	Mean	***3.6855E-01***	1.5201E+00	+	1.3273E+02	+	1.1523E+02	+
	Std.	***6.6883E-01***	1.5582E+00		2.2857E+02		5.5339E+01	
F03	Best	***1.5567E+01***	1.5646E+01		1.6877E+01		1.6469E+01	
	Mean	***1.5567E+01***	1.6179E+01	+	2.0548E+01	+	1.8783E+01	+
	Std.	***2.6335E-09***	3.4593E-01		5.2264E+00		1.3302E+00	
F04	Best	4.5641E-01	***0.0000E+00***		4.2913E-01		5.4999E-01	
	Mean	5.1998E-01	***0.0000E+00***	-	1.4363E+00	≈	8.0101E-01	+
	Std.	3.8885E-02	***0.0000E+00***		9.4091E-01		1.8822E-01	
F05	Best	1.8462E+01	***1.9899E+00***		1.4634E+00		1.1569E+01	
	Mean	3.3623E+01	***7.9374E+00***	-	4.6097E+01	≈	8.3787E+01	+
	Std.	1.7117E+01	***4.4735E+00***		5.7897E+01		5.1084E+01	
F06	Best	7.2800E-01	***2.8711E-01***		1.0241E+00		7.1795E+00	
	Mean	2.4889E+00	***6.3976E-01***	-	9.8031E+00	+	2.7904E+01	+
	Std.	2.8371E+00	***3.1415E-01***		7.5910E+00		3.3658E+01	
F07	Best	6.0916E-01	***2.3534E-01***		3.9396E-01		2.3203E-01	
	Mean	7.7913E-01	***6.5993E-01***	≈	1.3896E+01	≈	4.0791E+00	≈
	Std.	9.9590E-02	***3.4034E-01***		3.9651E+01		5.0802E+00	
F08	Best	2.4895E+01	***0.0000E+00***		1.1000E+02		1.1000E+02	
	Mean	6.0700E+01	***5.0504E+00***	-	1.1000E+02	+	1.1000E+02	+
	Std.	3.3810E+01	***1.0843E+01***		1.4211E-14		1.4211E-14	
F09	Best	1.0956E+02	***1.0000E+02***		3.9065E+02		3.9039E+02	
	Mean	3.5033E+02	***1.0000E+02***	-	3.9163E+02	+	3.9155E+02	+
	Std.	8.2238E+01	***0.0000E+00***		5.5792E-01		6.3702E-01	
F10	Best	***4.0000E+02***	***4.0000E+02***		***4.0000E+02***		***4.0000E+02***	
	Mean	***4.0000E+02***	***4.0000E+02***	≈	***4.0000E+02***	≈	***4.0000E+02***	≈
	Std.	***0.0000E+00***	***0.0000E+00***		***0.0000E+00***		***0.0000E+00***	
		+/≈/-	2/3/5		5/5/0		7/3/0	

Fig. 8 shows the convergence curves of the four compared algorithms in solving functions F2, F3, F6, and F7. We can see that SHADE and L-SHADE converge quickly and get stuck at the early stage of the search process (note that x-axis is plotted with a logarithmic scale), but L-HMDE and IMODE keep improving the solutions for a longer period.

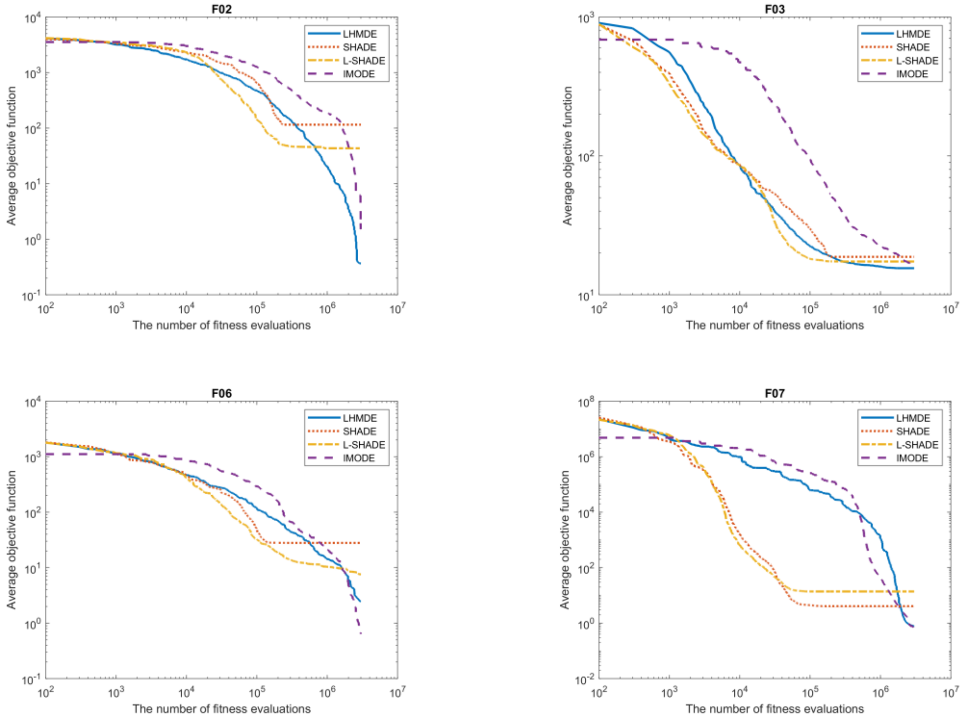

Fig 8 Convergence curves of four algorithms in solving CEC 2020 benchmark functions (F2, F3, F6, and F7)

5.4.2 Test cases of the economic dispatch problem

In this experiment, we tested the performance of L-HMDE and the three general-purpose algorithms in solving the ED test cases. Since the general-purpose algorithms do not consider the problem constraints, we incorporated our ISR repair mechanism into these algorithms. All compared algorithms used the same parameter setting as they used in solving CEC benchmark, except for the initial population size that was set to 15. The (initial) population sizes of all algorithms were set by 15. Each algorithm solved each test case for 100 runs. Table 34 presents the results.

Table 34 Solution cost of four algorithms in solving 22 ED test cases

Test case	L-HMDE	IMODE		L-SHADE		SHADE	
	Avg (Std)	Avg (Std)		Avg (Std)		Avg (Std)	
1	15449.90 ($3.24\cdot10^{-7}$)	15449.9 ($5.90\cdot10^{-7}$)	-	15449.90 ($5.68\cdot10^{-8}$)	-	15449.90 ($4.18\cdot10^{-7}$)	-
2	623.81 ($2.00\cdot10^{-4}$)	623.81 ($1.73\cdot10^{-3}$)	≈	623.81 ($5.49\cdot10^{-7}$)	-	623.81 ($1.41\cdot10^{-5}$)	-
3	623.83 ($6.38\cdot10^{-4}$)	623.84 ($6.45\cdot10^{-3}$)	+	623.83 ($7.30\cdot10^{-3}$)	+	623.85 ($1.13\cdot10^{-2}$)	+
4.1	17964.90 (2.98)	17973.15 (1.77)	+	17973.52 (15.71)	+	18015.43 (47.57)	+
4.2	24169.96 (0.42)	24185.3 (31.92)	+	24180.97 (33.25)	+	24239.76 (67.79)	+
4.3	17961.22 (2.44)	17969.98 (6.45)	+	17974.65 (22.59)	+	18011.34 (42.44)	+
4.2	24164.15 (0.67)	24187.30 (33.87)	+	24180.19 (43.14)	+	24245.70 (57.19)	+
5.1	24514.88 ($1.56\cdot10^{-4}$)	24555.52 (54.79)	+	24545.04 (61.55)	+	24623.78 (73.68)	+
5.2	24516.28 (7.97)	24542.28 (44.93)	≈	24539.07 (46.59)	+	24626.51 (76.48)	+
5.3	24514.82 ($7.02\cdot10^{-5}$)	24553.05 (52.00)	+	24551.87 (67.25)	+	24622.19 (70.05)	+
5.4	24513.04 (6.09)	24565.65 (50.00)	+	24552.87 (57.57)	+	24629.84 (72.68)	+
6.1	32704.45 ($1.65\cdot10^{-5}$)	32704.45 ($1.94\cdot10^{-4}$)	+	32704.45 ($1.05\cdot10^{-7}$)	-	32704.45 ($6.77\cdot10^{-8}$)	-
6.2	32588.92 ($1.83\cdot10^{-7}$)	32588.92 ($2.51\cdot10^{-5}$)	+	32588.92 ($3.31\cdot10^{-8}$)	≈	32588.92 ($2.23\cdot10^{-8}$)	≈
7	62456.63 ($3.33\cdot10^{-5}$)	62456.63 ($2.47\cdot10^{-4}$)	≈	62456.63 ($1.18\cdot10^{-6}$)	-	62456.63 ($2.78\cdot10^{-6}$)	-
8.1	121417.94 (3.81)	121456.90 (32.56)	+	121676.25 (207.54)	+	121657.25 (158.69)	+
8.2	121409.43 (4.51)	121448.16 (33.82)	+	121689.17 (263.78)	+	121642.19 (158.01)	+
8.3	121375.89 (4.85)	121412.31 (31.68)	+	121608.81 (232.75)	+	121606.02 (158.21)	+
9	197988.18 ($8.80\cdot10^{-8}$)	197988.18 ($3.29\cdot10^{-4}$)	+	197988.18 ($3.11\cdot10^{-5}$)	≈	197988.18 ($7.79\cdot10^{-6}$)	+
10	1657962.73 ($1.07\cdot10^{-3}$)	1658044.78 (86.05)	+	1658081.42 (120.92)	+	1658048.47 (99.82)	+
11	1559708.45 ($2.69\cdot10^{-4}$)	1559805.63 (89.26)	+	1559885.51 (232.47)	+	1559807.34 (108.47)	+
12	1655679.43 ($1.82\cdot10^{-4}$)	1655683.95 (4.90)	+	1655680.91 (3.53)	+	1655679.59 (0.92)	+
13	9983.69 (0.12)	9986.36 (4.50)	+	10000.22 (11.99)	+	9997.85 (9.72)	+
	+/≈/-	18/3/1		16/2/4		17/1/4	

The results of Wilcoxon rank-sum test show that our L-HMDE outperforms the three general-purpose algorithms in at least 16 out of 22 test cases (more than 70%). It is outperformed by IMODE in only one test case and is outperformed by L-SHADE and SHADE in four test cases. For some test cases such as cases 1, 2, 3, 6, 7, and 9, all algorithms found solutions with almost the same cost. These test cases seem to be easy and solvable by general-purpose algorithms. However, there are still many ED test cases that need tailored algorithms like our L-HMDE to solve it effectively.

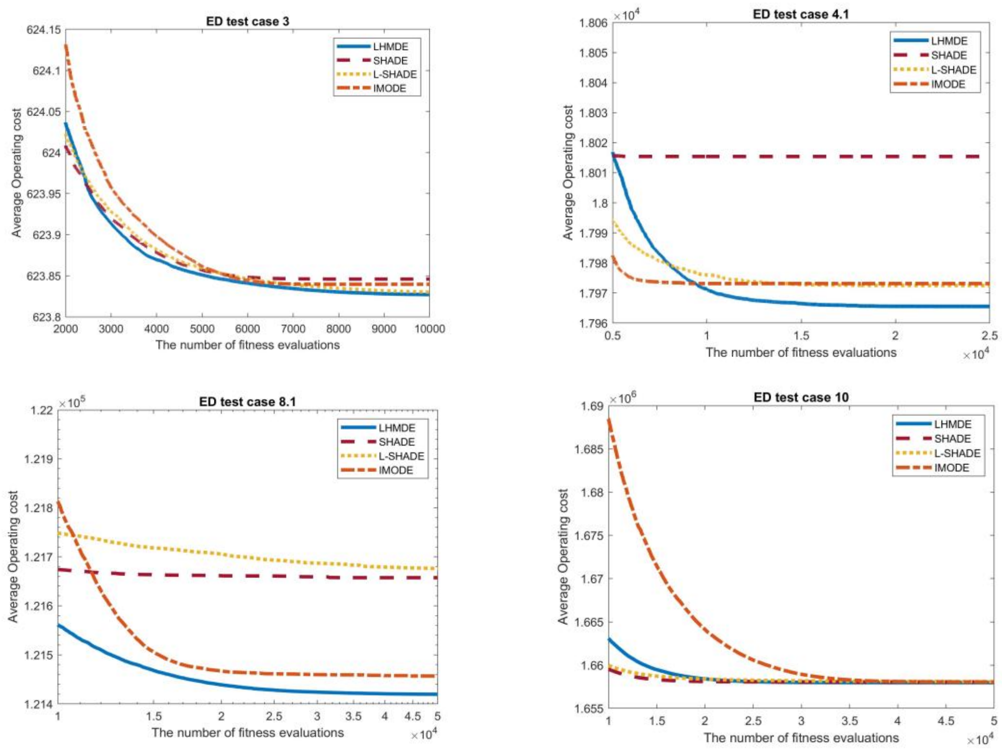

Fig 9 Convergence curves of four algorithms in solving ED test cases (case 3, 4.1, 8.1, and 10)

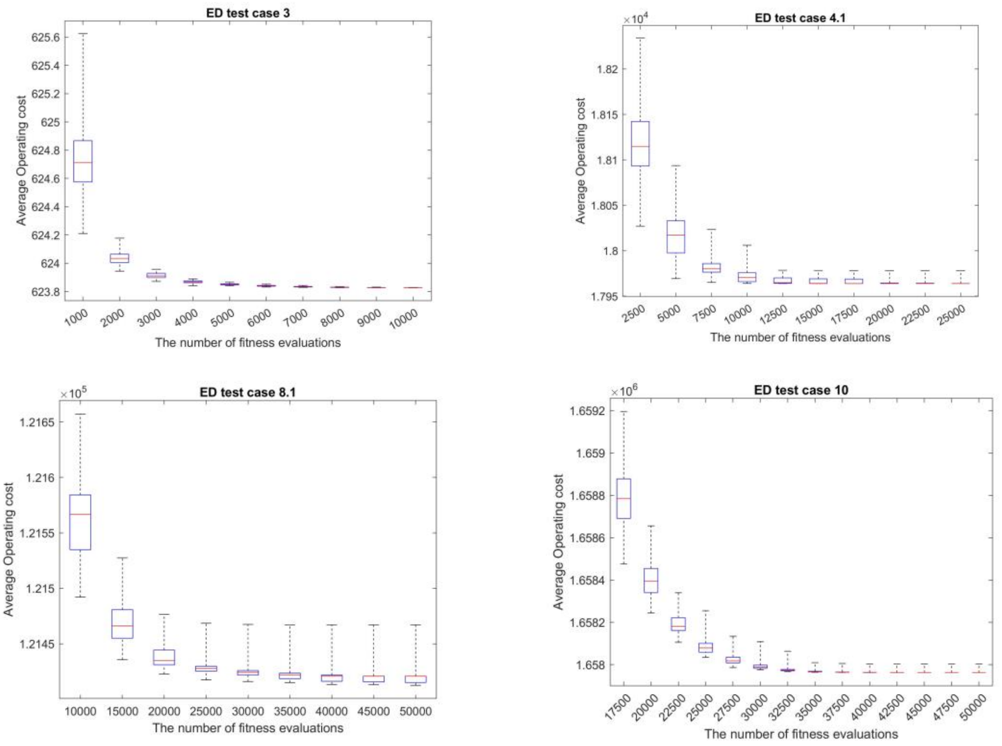

Fig 10 Population diversity of L-HMDE in solving ED test cases (case 3, 4.1, 8.1, and 10)

The convergence curves of L-HMDE and three compared algorithms are given in Fig 9. Similar to what we observed in Fig. 8, SHADE and L-SHADE converge faster and may get stuck early. In contrast, L-HMDE converges slower but keeps the ability of improving solutions, leading to better final solution in the end. Fig. 10 shows the population diversity of L-HMDE by the box plots of the objective values of solutions in the population at different generations. We can see that the population diversity is high at the early stage and gets lower with a smooth trend as the evolutionary process goes. Table 35 presents the running time consumed by the four algorithms to solve the ED test cases. Note that in our experiments all four algorithms were implemented by Matlab, and thus the impact of the programming language on the running time was reduced. According to the results in Table 35, L-HMDE requires similar running time as other three do. All of them can solve ED test cases within several seconds.

Table 35 Running time of L-HMDE and three adaptive algorithms in solving the ED problem

Test case	L-HMDE	IMODE	L-SHADE	SHADE
	Avg (Std)	Avg (Std)	Avg (Std)	Avg (Std)
1	1.38e-1 (5.03e-3)	1.83e-1 (2.73e-2)	1.10e-1 (9.62e-3)	1.05e-1 (1.58e-3)
2	1.10e-1 (1.89e-3)	1.59e-1 (5.98e-3)	8.63e-2 (1.98e-3)	7.76e-2 (1.52e-3)
3	4.63e-1 (2.12e-3)	6.72e-1 (6.50e-3)	4.11e-1 (6.16e-3)	3.00e-1 (6.82e-3)
4.1	1.11e+0 (4.89e-2)	1.56e+0 (5.49e-2)	8.11e-1 (2.04e-2)	6.86e-1 (2.33e-2)
4.2	1.11e+0 (6.18e-3)	1.87e+0 (1.02e-2)	1.01e+0 (2.31e-2)	6.89e-1 (1.01e-2)
4.3	1.10e+0 (7.64e-3)	1.78e+0 (1.03e-2)	9.85e-1 (2.55e-2)	6.77e-1 (7.81e-3)
4.2	1.10e+0 (4.56e-3)	1.88e+0 (1.07e-2)	1.00e+0 (2.56e-2)	6.84e-1 (8.74e-3)
5.1	1.46e+0 (3.07e-2)	2.15e+0 (2.89e-2)	1.30e+0 (1.40e-1)	8.18e-1 (2.88e-2)
5.2	1.45e+0 (5.70e-2)	2.16e+0 (2.29e-2)	1.31e+0 (1.32e-1)	8.18e-1 (2.97e-2)
5.3	1.46e+0 (3.07e-2)	2.17e+0 (3.91e-2)	1.29e+0 (1.40e-1)	8.21e-1 (2.77e-2)
5.4	1.46e+0 (3.46e-2)	2.16e+0 (2.48e-2)	1.32e+0 (1.42e-1)	8.73e-1 (2.11e-2)
6.1	3.95e-1 (2.91e-3)	5.59e-1 (4.61e-3)	3.26e-1 (4.96e-3)	2.93e-1 (6.15e-3)
6.2	3.98e-1 (3.78e-3)	5.62e-1 (5.03e-3)	3.33e-1 (5.61e-3)	2.99e-1 (6.92e-3)
7	3.95e-1 (2.20e-3)	5.45e-1 (3.90e-3)	3.22e-1 (5.70e-3)	2.85e-1 (3.83e-3)
8.1	2.49e+0 (1.16e-2)	3.89e+0 (7.32e-2)	2.12e+0 (3.42e-2)	1.57e+0 (3.57e-2)
8.2	2.46e+0 (1.83e-2)	4.24e+0 (7.02e-2)	2.13e+0 (2.85e-2)	1.69e+0 (3.71e-2)
8.3	3.24e+0 (1.38e-2)	4.42e+0 (3.82e-2)	2.25e+0 (3.44e-2)	1.82e+0 (4.27e-2)
9	3.62e+0 (1.85e-2)	4.71e+0 (2.49e-2)	2.43e+0 (1.53e-2)	2.00e+0 (7.84e-3)
10	1.06e+1 (4.92e-1)	6.16e+0 (1.51e-1)	4.00e+0 (3.37e-1)	3.68e+0 (5.92e-2)
11	4.76e+0 (5.32e-2)	5.81e+0 (3.56e-2)	3.42e+0 (4.35e-2	3.01e+0 (1.40e-2)
12	4.75e+0 (3.88e-2)	5.84e+0 (5.43e-2)	3.46e+0 (3.21e-2)	3.02e+0 (2.33e-2)
13	1.43e+1 (8.59e-2)	1.63e+1 (8.46e-2)	9.37e+0 (4.35e-2)	7.74e+0 (3.69e-2)

6. Conclusions

The objective of this paper is twofold: to serve as a comprehensive reference and to propose an effective solver for the economic dispatch problem. In the capacity of a valuable reference, we reviewed over 100 papers and extracted the features of various algorithms for further research exploration. Moreover, we made a compilation of 22 diverse test cases and carefully checked the details of model coefficients and the correctness of solutions. This dataset will serve as a trustful reference for experimental benchmarks in this domain. For the problem solver, we proposed L-HMDE, whose advantage is simple, effective, robust, and efficient. Based on the framework of DE, we incorporated a hybrid mutation strategy, a linear population size reduction mechanism, and an improved repair mechanism. The hybrid mutation strategy enhances the solution quality and reduces the sensitivity to the parameter setting. The linear population size reduction mechanism prolongs the evolutionary process and focuses on the promising areas, leading to better solution quality, especially for medium- and large-scale test cases. The improved repair mechanism fixes infeasible solutions more effectively, and thus helps the whole algorithm to find high-quality solutions more efficiently. We not only confirmed the positive effects of the above algorithmic components through experiments, we also compared the proposed L-HMDE with more than 90 existing algorithms. L-HMDE is ranked among the top six algorithms for 20 out of 22 test cases. It also outperforms three general-purpose algorithms in solving at least 16 out of 22 ED test cases.

There remains a scope for further refinement to the L-HMDE. First, the current version of L-HMDE requires a parameter tuning process. Although it could solve a variety of test cases quite well with a single and fixed parameter configuration, we will continue to equip it with adaptive parameter control mechanisms. This research direction also aligns with the insight extracted from the literature review. It is worth noting that adaptive control is not a trivial topic and demands careful investigations. While many existing algorithms incorporate adaptive control mechanisms, they do not perform better than L-HMDE. Second, we want to enhance the search ability of L-HMDE by the niching methods. The linear population size reduction mechanism can help to allocate computing resources effectively to promising areas in the search space, but it may sometimes overlook potential areas. We expect the use of niching methods to improve the performance of L-HMDE further. Third, we will apply L-HMDE to other extended economic dispatch problems. These problems will bring new challenges. For example, we need to put in concepts such as dominance and Pareto optimality to deal with multiple objectives in the economic emission dispatch problems. In summary, adaptive parameter control, niching, and multiobjective optimization are the three main topics with which we will continue in our future work.

Acknowledgments

This research is supported by the Ministry of Science and Technology, Taiwan, R.O.C. under Grant no. 109-2221-E-003-025 and 110-2221-E-003-017.

Reference

[1] J.I. Arachchi, S. Managi, Preferences for energy sustainability: Different effects of gender on knowledge and importance, Renewable Sustainable Energy Rev., 141 (2021) 1-13.

[2] M.S. Bakare, A. Abdulkarim, M. Zeeshan, A.N. Shuaibu, A comprehensive overview on demand side energy management towards smart grids: challenges, solutions, and future direction, Energy Inform., 6 (4) (2023) 1-59.

[3] S.O. Orero, M.R. Irving, Large scale unit commitment using a hybrid genetic algorithm, Int. J. Electr. Power Energy Syst., 19 (1) (1997) 45–55.

[4] N. Sinha, R. Chakrabati, P.K. Chattopadhyay, Evolutionary programming techniques for economic load dispatch, IEEE Trans. Evol. Comput., 7 (1) (2003) 83–94.

[5] C.T. Su, C.T. Lin, New approach with a hopfield modeling framework to economic dispatch, IEEE Trans. Power Syst., 15 (2) (2000) 541–545.

[6] Z.L. Gaing, Particle swarm optimization to solving the economic dispatch considering the generator constraints, IEEE Trans. Power Syst., 18 (3) (2003) 1187–1195.

[7] W.T. Elsayed, Y.G. Hegazy, F.M. Bendary, M.S. El-Bages, A review on accuracy issues related to solving the non-convex economic dispatch problem, Electr. Power Syst. Res., 141 (2016) 325-332.

[8] H. Saadat, Power system analysis, 2nd ed., Mc-Graw-Hill, New York, USA, 2002, pp. 291–295.

[9] C.L. Chiang, Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels, IEEE Trans. Power Syst., 20 (4) (2005) 1690–1699.

[10] C.E. Lin, G.L. Viviani, Hierarchical economic dispatch for piecewise quadratic cost functions, IEEE Trans. Power Appar. Syst., PAS-103 (1984) 1170–1175.

[11] A.B. Kunya, A.S. Abubakar, S.S. Yusuf, Review of economic dispatch in multi-area power system: state-of-the-art and future prospective, Electr. Power Syst. Res., 217 (2023) 1-16.

[12] M. Ghasemi, J. Aghaei, E. Akbari, S. Ghavidel, L. Li, A differential evolution particle swarm optimizer for various types of multi-area economic dispatch problems, Energy, 107 (2016) 182-195.

[13] M. Mohammadian, A. Lorestani, M. M. Ardehali, Optimization of single and multi-areas economic dispatch problems based on evolutionary particle swarm optimization algorithm, Energy, 161 (2018) 710-724.

[14] M. Nazari-Heris, B. Mohammadi-Ivatloo, G.B. Gharehpetian, A comprehensive review of heuristic optimization algorithms for optimal combined heat and power dispatch from economic and environmental perspectives, Renew. Sustain. Energy Rev., 81 (2) (2018) 2128-2143.

[15] X. Chen, K. Li, B. Xu, Z. Yang, Biogeography-based learning particle swarm optimization for combined heat and power economic dispatch problem, Knowl.-Based Syst., 208 (2020) 1-19.

[16] A. Srivastava, D.K. Das, A new Kho-Kho optimization Algorithm: An application to solve combined emission economic dispatch and combined heat and power economic dispatch problem. Eng. Appl. Artif. Intell., 94 (2020) 1-18.

[17] J. Sun, J. Deng, Y. Li, Indicator & crowding distance-based evolutionary algorithm for combined heat and power economic emission dispatch, Appl. Soft Comput., 90 (2020) 1-15.

[18] X. Xia, A. Elaiw, Optimal dynamic economic dispatch of generation: A review, Electr. Power Syst. Res., 80 (8) (2010) 975-986.

[19] W. Yang, Z. Peng, Z. Yang, Y. Guo, X. Chen, An enhanced exploratory whale optimization algorithm for dynamic economic

dispatch, Energy Rep., 7 (2021) 7015-7029.
[20] J. C. S. Chavez, A. Zamora-Mendez, M. R. A. Paternina, J. F. Y. Heredia, R. Cardenas-Javier, A hybrid optimization framework for the non-convex economic dispatch problem via meta-heuristic algorithms, Electr. Power Syst. Res., 177 (2019) 1-10.
[21] B. Y. Qu, Y. S. Zhu, Y. C. Jiao, M. Y. Wu, P. N. Suganthan, J. J. Liang, A survey on multi-objective evolutionary algorithms for the solution of the environmental/ economic dispatch problems, Swarm Evol. Comput., 38 (2018) 1-11.
[22] T. Liu, L. Jiao, W. Ma, J. Ma, R. Shang, Cultural quantum-behaved particle swarm optimization for environmental/economic dispatch, Appl. Soft Comput., 48 (2016) 597-611.
[23] Z. Xin-gang, L. Ji, M. Jin, Z. Ying, An improved quantum particle swarm optimization algorithm for environmental economic dispatch, Expert Syst. Appl., 152 (2020) 1-14.
[24] S. Mondal, A. Bhattacharya, S. H. Nee Dey, Multi-objective economic emission load dispatch solution using gravitational search algorithm and considering wind power penetration, Int. J. Electr. Power Energy Syst., 44 (1) 282-292.
[25] J. Mir, S. Kasim, H. Mahdin, R. R. Saedudin, R. Hassan, R. Ramlan, A proposed formulation for multi-objective renewable economic load dispatch, J. Ambient Intell. Humaniz. Comput., 14 (2023) 10299-10320.
[26] G. Abbas, J. Gu, U. Farooq, M. U. Asad and M. El-Hawary, Solution of an economic dispatch problem through particle swarm optimization: A detailed survey - part I, IEEE Access, 5 (2017) 15105-15141.
[27] G. Abbas, J. Gu, U. Farooq, A. Raza, M. U. Asad and M. E. El-Hawary, Solution of an economic dispatch problem through particle swarm optimization: A detailed survey – part II, IEEE Access, 5 (2017) 24426-24445.
[28] L. Jebaraj, C.Venkatesan, I. Soubache, C.C.A. Rajan, Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: A review, Renew. Sustain. Energy Rev., 77 (2017) 1206-1220.
[29] P.R. Lolla, S.K. Rangu, K.R. Dhenuvakonda, A.R. Singh, A comprehensive review of soft computing algorithms for optimal generation scheduling, Int. J. Energy Res., 45 (2) (2020) 1170–1189.
[30] S. Fanshel, E.S. Lynes, Economic power generation using linear programming, IEEE Trans. Power Appar. Syst., 83 (4) (1964) 347–356.
[31] A.A. El-Keib, H. Ma, J.L. Hart, Environmentally constrained economic dispatch using the lagrangian relaxation method, IEEE Trans. Power Syst., 9 (4) (1994) 1723–1729
[32] P. Lowery, Generating unit commitment by dynamic programming, IEEE Trans. Power Appar. Syst., PAS-85 (5) (1966) 422–426.
[33] M. Sydulu, A very fast and effective noniterative “λ logic based” algorithm for economic dispatch of thermal units, in: Proceedings of the IEEE region 10 conference TENCON, Cheju, South Korea, 1999, 1434–1437.
[34] M.O.F. Goni, M. Nahiduzzaman, M. S. Anower, I. Kamwa, S.M. Muyeen, Integration of machine learning with economic energy scheduling. Int. J. Electr. Power Energy Syst., 142 (2022) 1-9.
[35] G. Abbas, I.A. Khan, N. Ashraf, M.T. Raza, M. Rashad, R. Muzzammel, On employing a constrained nonlinear optimizer to constrained economic dispatch rroblems, Sustainability, 15(13) (2023) 1-23.
[36] D.C. Walters, G.B. Sheble, Genetic algorithm solution of economic dispatch with valve point loading, IEEE Trans. Power Syst., 8 (3) (1993) 1325–1332.
[37] S. Pan, J. Jian, H. Chen, L. Yang, A full mixed-integer linear programming formulation for economic dispatch with valve-point effects, transmission loss and prohibited operating zones, Electr. Power Syst. Res., 180 (2020) 1–12.
[38] R. Balamurugan, S. Subramanian, An improved dynamic programming approach to economic power dispatch with generator constraints and transmission losses, J. Electr. Eng. Technol., 3 (3) (2008) 320–330.
[39] T. Adhinarayanan, M. Sydulu, Efficient lambda logic based optimisation procedure to solve the large scale generator constrained economic dispatch problem, J. Electr. Eng. Technol., 4 (3) (2009) 301–309.
[40] V.N. Dieu, W. Ongsakul, J. Polprasert, The augmented lagrange hopfield network for economic dispatch with multiple fuel options, Math. Comput. Model., 57 (2013) 30–39.
[41] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, Australia, 1995, 1942–1948.
[42] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, 39–43.
[43] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim., 11 (1997) 341–359.
[44] N. Amjady, H. Sharifzadeh, Solution of non-convex economic dispatch problem considering valve loading effect by a new modified differential evolution algorithm, Int. J. Electr. Power Energy Syst., 32 (8) (2010) 893–903.
[45] M. Modiri-Delshad, N.A. Rahim, Solving non-convex economic dispatch problem via backtracking search algorithm, Energy, 77 (2014) 372–381.
[46] M. Modiri-Delshad, S.H.A. Kaboli, E. Taslimi-Renani, N. Abd Rahim, Backtracking search algorithm for solving economic dispatch problems with valve-point effects and multiple fuel options, Energy, 116 (2016) 637–649.
[47] L.S. Coelho, R.C. Thom Souza, V.C. Mariani, Improved differential evolution approach based on cultural algorithm and diversity measure applied to solve economic load dispatch problems, Math. Comput. Simul., 79 (10) (2009) 3136–3147.

[48] D. Zou, S. Li, G.G. Wang, Z. Li, H. Ouyang, An improved differential evolution algorithm for the economic load dispatch problems with or without valve-point effects, Appl. Energy, 181 (2016) 375–390.

[49] Q. Zhang, D. Zou, N. Duan, X. Shen, An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem, Appl. Soft Comput., 78 (2019) 641–669.

[50] J.X. Neto, G. Reynoso-Meza, T.H. Ruppel, V.C. Mariani, L. dosSantos Coelho, Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution, Int. J. Electr. Power Energy Syst., 84 (2017) 13–24.

[51] S. Sayah, A. Hamouda, A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems, Appl. Soft Comput., 13 (4) (2013) 1608–1619.

[52] M. Pandit, L. Srivastava, M. Sharma, H.M. Dubey, B.K. Panigrahi, Large-scale multi-zone optimal power dispatch using hybrid hierarchical evolution technique, J. Eng., 2014 (3) (2014) 71–80.

[53] G. Xiong, D. Shi, X. Duan, Multi-strategy ensemble biogeography-based optimization for economic dispatch problems, Appl. Energy, 111 (2013) 801–811.

[54] A. Bhattacharya, P.K. Chattopadhyay, Biogeography-based optimization for different economic load dispatch problems, IEEE Trans. Power Syst., 25 (2) (2009) 1064–1077.

[55] L. Wang, L. Li, An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems, Int. J. Electr. Power Energy Syst., 44 (1) (2013) 832–843.

[56] Y. Yang, B. Wei, H. Liu, Y. Zhang, J. Zhao, E. Manla, Chaos firefly algorithm with self-adaptation mutation mechanism for solving large-scale economic dispatch with valve-point effects and multiple fuel options, IEEE Access, 6 (2018) 45907–45922.

[57] R. Balamurugan, S. Subramanian, Hybrid integer coded differential evolution – dynamic programming approach for economic load dispatch with multiple fuel options, Energy Convers. Manag., 49 (4) (2008) 608–14.

[58] Q. Liu, G. Xiong, X. Fu, A. W. Mohamed, J. Zhang, M. A. Al-Betar, H. Chen, J. Chen, S. Xu, Hybridizing gaining–sharing knowledge and differential evolution for large-scale power system economic dispatch problems, J. Comput. Des. Eng., 10(2) (2023) 615-631.

[59] A.S. Reddy, K. Vaisakh, Shuffled differential evolution for large scale economic dispatch, Electr. Power Syst. Res., 96 (2013) 237–245.

[60] K. Vaisakh, A.S. Reddy, MSFLA/GHS/SFLA-GHS/SDE algorithms for economic dispatch problem considering multiple fuels and valve point loadings, Appl. Soft Comput., 13 (11) (2013) 4281-4291.

[61] M. Ghasemi, M. Taghizadeh, S. Ghavidel, A. Abbasian, Colonial competitive differential evolution: an experimental study for optimal economic load dispatch, Appl. Soft Comput., 40 (2016) 342–363.

[62] X. Li, H. Zhang, Z. Lu, A differential evolution algorithm based on multi-population for economic dispatch problems with valve-point effects, IEEE Access, 7 (2019) 95585–95609.

[63] N. Noman, H. Iba, Differential evolution for economic load dispatch problems, Electr. Power Syst. Res., 78 (8) (2008) 1322–1331.

[64] L.S. Coelho, V.C. Mariani, Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect, IEEE Trans. Power Syst., 21 (2) (2006) 989-996.

[65] M. Basu, Improved differential evolution for economic dispatch, Int. J. Electr. Power Energy Syst., 63 (2014) 855–861.

[66] S.K. Wang, J.P. Chiou, C.W. Liu, Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm, IET Gener. Transm. Distrib., 1 (5) (2007) 793–803.

[67] L.S. Coelho, T.C. Bora, V.C. Mariani, Differential evolution based on truncated lévy-type flights and population diversity measure to solve economic load dispatch problems, Int. J. Electr. Power Energy Syst., 57 (2014) 178–188.

[68] T. Visutarrom, T. C. Chiang, A. Konak, and S. Kulturel-Konak, Reinforcement learning-based differential evolution for solving economic dispatch problems, in: Proceedings of IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 2020, 913–917.

[69] T. C. Chiang, T. Visutarrom, A. Konak, and S. Kulturel-Konak, An adaptive multiobjective evolutionary algorithm for economic emission dispatch, in: Proceedings of 2022 IEEE Congress on Evolutionary Computation, Padua, Italy, 2022, 1–8.

[70] A.I. Selvakumar, K. Thanushkodi, A new particle swarm optimization solution to nonconvex economic dispatch problems, IEEE Trans. Power Syst., 22 (1) (2007) 42–51.

[71] A.I. Selvakumar, K. Thanushkodi, Anti-predatory particle swarm optimization: solution to nonconvex economic dispatch problems, Electr. Power Syst. Res., 78 (1) (2008) 2–10.

[72] V.K. Jadoun, N. Gupta, A. Swarnkar, K.R. Niazi, Non-convex economic load dispatch using particle swarm optimization with elevated search and addressed operators, in: Proceedings of 2015 International Conference on Recent Developments in Control, Automation and Power Engineering, Noida, India, 2015, 113–118.

[73] M.N. Abdullah, A.H.A. Bakar, N.A. Rahim, H. Mokhlis, H.A. Illias, J.J. Jamian, Modified particle swarm optimization with time varying acceleration coefficients for economic load dispatch with generator constraints, J. Electr. Eng. Technol., 9 (1) (2014) 15–26.

[74] V.K. Jadoun, N. Gupta, K.R. Niazi, A. Swarnkar, Dynamically controlled particle swarm optimization for large-scale nonconvex economic dispatch problems, Int. Trans. Elect. Energy Syst., 25 (11) (2014) 3060–3074.
[75] Q. Qin, S. Cheng, X. Chu, X. Lei, Y. Shi, Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization, Appl. Soft Comput., 59 (2017) 229–242.
[76] S. Xu, G. Xiong, A. W. Mohamed, H. R.E.H. Bouchekara, Forgetting velocity based improved comprehensive learning particle swarm optimization for non-convex economic dispatch problems with valve-point effects and multi-fuel options, Energy, 256 (2022) 1-26.
[77] N. Singh, T. Chakrabarti, P. Chakrabarti, M. Margala, A. Gupta, S. P. Praveen, S.B. Krishnan, B. Unhelkar, Novel Heuristic Optimization Technique to Solve Economic Load Dispatch and Economic Emission Load Dispatch Problems. Electronics, 12 (13) (2023) 1-16.
[78] S. Duman, N. Yorukeren, I.H. Altas, 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., 64 (2015) 121–135.
[79] M. Ellahi, G. Abbas, G. B. Satrya, M. R. Usman, J. Gu, A Modified Hybrid Particle Swarm Optimization with Bat Algorithm Parameter Inspired Acceleration Coefficients for Solving Eco-Friendly and Economic Dispatch Problems, IEEE Access, 9 (2021) 82169-82187.
[80] A. Gacem, D. Benattous, Hybrid genetic algorithm and particle swarm for optimal power flow with non-smooth fuel cost functions, Int. J. Syst. Assur. Eng. Manag. 8(S1) (2017) 146–153.
[81] A.Y. Saber, Economic dispatch using particle swarm optimization with bacterial foraging effect, Int. J. Electr. Power Energy Syst., 34 (2012) 38–46.
[82] L.D.S. Coelho, V.C. Mariani, Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects, Energy Convers. Manag., 49 (11) (2008) 3080–3085.
[83] W.T. Elsayed, Y.G. Hegazy, M.S. El-bages, F.M. Bendary, Improved random drift particle swarm optimization with self-adaptive mechanism for solving the power economic dispatch problem, IEEE Trans. Industr. Inform., 13 (3) (2017) 1017–1026.
[84] J. Chen, J. Zheng, P. Wu, L. Zhang, Q. Wu, Dynamic particle swarm optimizer with escaping prey for solving constrained non-convex and piecewise optimization problems, Expert Syst. Appl., 86 (2017) 208–223.
[85] R. Kumar, D. Sharma, A. Sadu, A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch, Int. J. Electr. Power Energy Syst., 33 (1) (2011) 115–123.
[86] M.N. Abdullah, A.H.A. Bakar, N.A. Rahim, H. Moklis, Economic load dispatch with nonsmooth cost functions using evolutionary particle swarm optimization, IEEJ Trans. Electr. Electron. Eng., 8 (S1) (2013) S30–S37.
[87] V. Hosseinnezhad, E. Babaei, Economic load dispatch using θ-PSO, Int. J. Electr. Power Energy Syst., 49 (2013) 160–169.
[88] K.T. Chaturvedi, M. Pandit, L. Srivastava, Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch, Int. J. Electr. Power Energy Syst., 31 (6) (2009) 249–57.
[89] L. dos S. Coelho, V. C. Mariani, A novel chaotic particle swarm optimization approach using Hénon map and implicit filtering local search for economic load dispatch, Chaos, Solitons & Fractals, 39 (2) (2009) 510–518.
[90] M. Gholamghasemi, E. Akbari, M.B. Asadpoor, M. Ghasemi, A new solution to the non-convex economic load dispatch problems using phasor particle swarm optimization, Appl. Soft Comput., 79 (2019) 111–124.
[91] J.-B. Park, Y.-W. Jeong, J.-R. Shin, K.Y. Lee, An improved particle swarm optimization for nonconvex economic dispatch problems, IEEE Trans. Power Syst., 25 (1) (2010) 156–166.
[92] S. Chalermchaiarbha W. Ongsakul, Stochastic weight trade-off particle swarm optimization for nonconvex economic dispatch, Energy Convers. Manag., 70 (2013) 66–75.
[93] M. Basu, Modified particle swarm optimization for nonconvex economic dispatch problems, Int. J. Electr. Power Energy Syst., 69 (2015) 304–312.
[94] C. Li, J. Sun, V. Palade, L.W. Li, Diversity collaboratively guided random drift particle swarm optimization, Int. J. Mach. Learn. Cybern., 12 (2021) 2617–2638.
[95] T.A.A. Victoire, A.E. Jeyakumar, Hybrid PSO-SQP for economic dispatch with valve-point effect, Electr. Power Syst. Res., 71 (1) (2004) 51–59.
[96] J. Cai, Q. Li, L. Li, H. Peng, Y. Yang, A hybrid CPSO-SQP method for economic dispatch considering the valve-point effects, Energy Convers. Manag., 53 (1) (2012) 175–181.
[97] J.B. Park, K.S. Lee, J.R. Shin, K.Y. Lee, A particle swarm optimization for economic dispatch with nonsmooth cost functions, IEEE Trans. Power Syst., 20 (1) (2005) 34–42.
[98] A.K. Barisal, Dynamic search space squeezing strategy based intelligent algorithm solutions to economic dispatch with multiple fuels, Int. J. Electr. Power Energy Syst., 45 (1) (2013) 50–59.
[99] N. Amjady, H. Nasiri-Rad, Nonconvex economic dispatch with ac constraints by a new real coded genetic algorithm, IEEE Trans. Power Syst., 24 (3) (2009) 1489–1502.
[100] N. Amjady, H. Nasiri-Rad, Solution of nonconvex and nonsmooth economic dispatch by a new adaptive real coded genetic algorithm, Expert Syst. Appl., 37 (7) (2010) 5239–5245.
[101] G.S. Babu, D.B. Das, C. Patvardhan, Real-parameter quantum evolutionary algorithm for economic load dispatch, IET

Gener. Transm. Distrib., 2 (1) (2008) 22–31.
[102] D.C. Secui, G. Bendea, S. Dzitac, C. Bendea, C. Hora, A modified harmony search algorithm for the economic dispatch problem, Stud. Inform. Control., 23 (2) (2014) 143–152.
[103] D. Aydin, S. Ozyon, Solution to non-convex economic dispatch problem with valve point effects by incremental artificial bee colony with local search, Appl. Soft Comput., 13 (5) (2013) 2456–2466.
[104] S. Ozyon, D. Aydin, Incremental artificial bee colony with local search to economic dispatch problem with ramp rate limits and prohibited operating zones, Energy Convers. Manag., 65 (2013) 397–407.
[105] Y. Labbi, D.B. Attous, B. Mahdad, Artificial bee colony optimization for economic dispatch with valve point effect, Front. Energy, 8 (4) (2014) 449-458.
[106] H.T. Jadhav, R. Roy, Effect of turbine wake on optimal generation schedule and transmission losses in wind integrated power system, Sustain. Energy Technol. Assess., 7 (2014) 123–135.
[107] D.C. Secui, A new modified artificial bee colony algorithm for the economic dispatch problem, Energy Convers. Manag., 89 (2015) 43–62.
[108] M.A. Awadallah, M.A. Al-Betar, A.L.a. Bolaji, E.M. Alsukhni, H. Al-Zoubi, Natural selection methods for artificial bee colony with new versions of onlooker bee, Soft Comput., 23 (2019) 6455-6494.
[109] W.T. El-Sayed, E.F. El-Saadany, H.H. Zeineldin, A. Al-Durra, M.S. El-Moursi, Deterministic-like solution to the non-convex economic dispatch problem, IET Gener. Transm. Distrib., 15 (3) (2020) 420–435.
[110] M. Pradhan, P.K. Roy, T. Pal, Grey wolf optimization applied to economic load dispatch problems, Int. J. Electr. Power Energy Syst., 83 (2016) 325–334.
[111] M. Pradhan, P.K. Roy, T. Pal, Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system, Ain Shams Eng. J., 9 (4) (2018) 2015–2025.
[112] A.K. Barisal, R.C. Prusty, Large scale economic dispatch of power systems using oppositional invasive weed optimization, Appl. Soft Comput., 29 (2015) 122–137.
[113] M. H. Hassan, S. Kamel, F. Jurado, M. Ebeed, M. F. Elnaggar, Economic load dispatch solution of large-scale power systems using an enhanced beluga whale optimizer, Alex. Eng. J., 72, (2023) 573-591.
[114] M. Basu, Fast convergence evolutionary programming for economic dispatch problems, IET Gener. Transm. Distrib., 11 (16) (2017) 4009-4017.
[115] C. Chen, D. Zou, C. Li, Improved jaya algorithm for economic dispatch considering valve-point effect and multi-fuel options, IEEE Access, 8 (2020) 84981–84995.
[116] Z.X. Zheng, J.Q. Li, H.Y. Sang, A hybrid invasive weed optimization algorithm for the economic load dispatch problem in power systems, Math. Biosci. Eng., 16 (4) (2019) 2775–2794.
[117] J. Yu, C. Kim, A. Wadood, T. Khurshaid, S. Rhee, Jaya algorithm with self-adaptive multi-population and lévy flights for solving economic load dispatch problems, IEEE Access, 7 (2019) 21372–21384.
[118] S. Sahoo, K.M. Dash, R.C. Prusty, A.K. Barisal, Comparative analysis of optimal load dispatch through evolutionary algorithms, Ain Shams Eng. J., 6 (2015) 107–120.
[119] T.T. Nguyen, D.N. Vo, The application of one rank cuckoo search algorithm for solving economic load dispatch problems, Appl. Soft Comput., 37 (2015) 763–773.
[120] M. S. Braik, M. A. Awadallah, M. A. Al-Betar, A. I. Hammouri, R. A. Zitar, A non-convex economic load dispatch problem using chameleon swarm algorithm with roulette wheel and Levy flight methods, Appl. Intell., 53 (2023) 17508–17547.
[121] S. Chansareewittaya, Hybrid BA/ATS for Economic Dispatch Problem, in: 2018 22nd International Computer Science and Engineering Conference (ICSEC), Chiang Mai, Thailand, 2018, 1-4.
[122] C. Takeang, A. Aurasopon, Multiple of Hybrid Lambda Iteration and Simulated Annealing Algorithm to Solve Economic Dispatch Problem with Ramp Rate Limit and Prohibited Operating Zones, J. Electr. Eng. Technol., 14 (2019) 111–120.
[123] M. A. Al-Betar, M. A. Awadallah, R. A. Zitar, K. Assaleh, Economic load dispatch using memetic sine cosine algorithm, J. Ambient Intell. Human. Comput., 14 (2023) 11685–11713.
[124] G. Kaur, J.S. Dhillon, Economic power generation scheduling exploiting hill-climbed Sine-Cosine algorithm, Appl. Soft Comput., 111 (2021) 1–20.
[125] S. Basak, B. Dey, B. Bhattacharyya, Uncertainty-based dynamic economic dispatch for diverse load and wind profiles using a novel hybrid algorithm, Environ. Dev. Sustain., 25 (2023) 4723-4763.
[126] M.A. Al-Betar, M.A. Awadallah, A.T. Khader, A.L. Bolaji, Tournament-based harmony search algorithm for non-convex economic load dispatch problem, Appl. Soft Comput., 47 (2016) 449–459.
[127] M.A. Al-Betar, M.A. Awadallah, A.T. Khader, Bolaji ALa, A. Almomani, Economic load dispatch problems with valve-point loading using natural updated harmony search, Neural. Comput. Appl., 29 (2018) 767–781.
[128] J. Zhao, S. Liu, M. Zhou, X. Guo, L. Qi, Modified cuckoo search algorithm to solve economic power dispatch optimization problems, IEEE/CAA J. Autom. Sin., 5 (4) (2018) 794–806.
[129] Z. Huang, J. Zhao, L. Qi, Z. Gao, H. Duan, Comprehensive learning cuckoo search with chaos-lambda method for solving economic dispatch problems, Appl. Intell., 50 (2020) 2779–2799.
[130] L.S. Coelho, V.C. Mariani, An improved harmony search algorithm for power economic load dispatch, Energy Convers.

Manag., 50 (10) (2009) 2522–2526.
[131] B. Jeddi, V. Vahidinasab, A modified harmony search method for environmental/economic load dispatch of real-world power systems, Energy Convers. Manag., 78 (2014), 661–675.
[132] B.R. Adarsh, T. Raghunathan, T. Jayabarathi, X.S. Yang, Economic dispatch using chaotic bat algorithm, Energy, 96 (2016) 666–675.
[133] H. Liang, Y. Liu, Y. Shen, F. Li, Y. Man, A hybrid bat algorithm for economic dispatch with random wind power, IEEE Trans. Power Syst., 33 (5) (2018) 5052–5061.
[134] K.Y. Lee, A. Sode-Yome, J.H. Park, Adaptive hopfield neural networks for economic load dispatch, IEEE Trans. Power Syst., 13 (2) (1998) 519–525.
[135] J.H. Park, Y.S. Kim, I.K. Eom, K.Y. Lee, Economic load dispatch for piecewise quadratic cost function using hopfield neural network, IEEE Trans. Power Syst., 8 (3) (1993) 1030–1038.
[136] M. Moradi-Dalvand, B. Mohammadi-Ivatloo, A. Najafi, A. Rabiee, Continuous quick group search optimizer for solving non-convex economic dispatch problems, Electr. Power Syst. Res., 93 (2012) 93–105.
[137] W.T. Elsayed, Y.G. Hegazy, F.M. Bendary, M.S. El-bages, Modified social spider algorithm for solving the economic dispatch problem, Eng. Sci. Technol. Int. J., 19 (4) (2016) 1672-1681.
[138] A. Meng, J. Li, H. Yin, An efficient crisscross optimization solution to large-scale non-convex economic load dispatch with multiple fuel types and valve-point effects, Energy, 113 (2016) 1147–1161.
[139] M. Kumar, J.S. Dhillon, A conglomerated ion-motion and crisscross search optimizer for electric power load dispatch, Appl. Soft Comput., 83 (2019) 1–29.
[140] M.A. Elhameed, A.A. El-Fergany, Water cycle algorithm-based economic dispatcher for sequential and simultaneous objectives including practical constraints, Appl. Soft Comput., 58 (2017) 145–154.
[141] N. Rajput, V. Chaudhary, H.M. Dubey, M. Pandit, Optimal generation scheduling of thermal System using biologically inspired grasshopper algorithm, in: Proceedings of 2017 2nd International Conference on Telecommunication and Networks, Noida, India, 2017, 1–6.
[142] M. Kumar, J.S. Dhillon, Hybrid artificial algae algorithm for economic load dispatch, Appl. Soft Comput., 71 (2018) 89–109.
[143] B. Vedik, P. Naveen, C.K. Shiva, A novel disruption based symbiotic organisms search to solve economic dispatch, Evol. Intell., 15 (2022) 255–290.
[144] V. Kansal, J.S. Dhillon, Emended salp swarm algorithm for multiobjective electric power dispatch problem, Appl. Soft Comput., 90 (2020) 1–26.
[145] M. Ghasemi, I.F. Davoudkhani, E. Akbari, A. Rahimnejad, S. Ghavidel, L. Li, A novel and effective optimization algorithm for global optimization and its engineering applications: turbulent flow of water-based optimization (TFWO), Eng. Appl. Artif. Intell., 92 (2020) 1–14.
[146] M.H. Hassan, S. Kamel, L. Abualigah, A. Eid, Development and application of slime mould algorithm for optimal economic emission dispatch, Expert Syst. Appl., 182 (2021) 1–28.
[147] A. Kumara, M. Thakur, G. Mittal, Planning optimal power dispatch schedule using constrained ant colony optimization, Appl. Soft Comput., 115 (2022) 1–18.
[148] R. Tanabe, A. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: Proceedings of 2014 IEEE Congress on Evolutionary Computation, Beijing, China, 2014, 1658–1665.
[149] Z.L. Gaing, Closure to 'discussion of 'particle swarm optimization to solving the economic dispatch considering the generator constraints, IEEE Trans. Power Syst., 19 (4) (2004) 2122–2123.
[150] M. F. Tabassum, M. Saeed, N. A. Chaudhry, J. Ali, M. Farman, S. Akram, Evolutionary simplex adaptive Hooke-Jeeves algorithm for economic load dispatch problem considering valve point loading effects, Ain Shams Eng. J., 12 (2021) 1001-1015.
[151] P-N-Suganthan, 2020-Bound-Constrained-Opt-Benchmark https://github.com/P-N-Suganthan/2020-Bound-Constrained-Opt-Benchmark, 2023 (accessed 18 August 2023).
[152] R. Tanabe, A. Fukunaga, Evaluating the performance of SHADE on CEC 2013 benchmark problems, in: Proceedings of 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 2013, 1952–1959.
[153] K. M. Sallam, S. M. Elsayed, R. K. Chakrabortty and M. J. Ryan, Improved Multi-operator Differential Evolution Algorithm for Solving Unconstrained Problems, in: Proceedings of 2020 IEEE Congress on Evolutionary Computation, Glasgow, UK, 2020, 1-8.
[154] Y. Tian, R. Cheng, X. Zhang and Y. Jin, PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum], IEEE Computational Intelligence Magazine, 12 (4) (2017).

Appendix. Data set and minimum solution obtained by L-HMDE in each test case

A.1 Cost coefficient and loss coefficient of test case 1 (1263 MW)

Unit	*Min*	*Max*	a_j	b_j	c_j	UR_j	DR_j	P^0	*prohibited zones*
1	100	500	240	7	0.007	80	120	440	[210-240] [350-380]
2	50	200	200	10	0.0095	50	90	170	[90-110] [140-160]
3	80	300	220	8.5	0.009	65	100	200	[150-170] [210-240]
4	50	150	200	11	0.009	50	90	150	[80-90] [110-120]
5	50	200	220	10.5	0.008	50	90	190	[90-110] [140-150]
6	50	120	190	12	0.0075	50	90	110	[75-85] [100-105]

$$B_{gh\ (p.u)} = \begin{bmatrix} 0.0017 & 0.0012 & 0.0007 & -0.0001 & -0.0005 & -0.0002 \\ 0.0012 & 0.0014 & 0.0009 & 0.0001 & -0.0006 & -0.0001 \\ 0.0007 & 0.0009 & 0.0031 & 0 & -0.001 & -0.0006 \\ -0.0001 & 0.0001 & 0 & 0.0024 & -0.0006 & -0.0008 \\ -0.0005 & -0.0006 & -0.001 & -0.0006 & 0.0129 & -0.0002 \\ -0.0002 & -0.0001 & -0.0006 & -0.0008 & -0.0002 & 0.015 \end{bmatrix}$$

$$B_{0g\ (p.u)} = \begin{bmatrix} -0.3908 & -0.1297 & 0.7047 & 0.0591 & 0.2161 & -0.6635 \end{bmatrix}$$

$$B_{00\ (p.u)} = 0.0056$$

A.2 Best solution obtained by L-HMDE for test case 1

Units	L-HMDE	
	P_j (ε=10^{-8})	P_j (ε=$8\cdot10^{-2}$)
1	447.49989636	449.69602973
2	173.31229599	174.84824042
3	263.47513172	263.88420773
4	139.07083031	131.26923245
5	165.46617476	167.17283819
6	87.13378571	89.16294951
Total power output (MW)	1275.96	1276.03
Transmission loss (MW)	12.96	13.11
Error (MW)	$3.42\cdot10^{-9}$	$7.23\cdot10^{-2}$
Operating Cost ($/h)	15449.90	15449.62

B.1 Cost coefficient of test cases 2 and 3 (2700 MW)

Unit	*Fuel types*	*Lower bound*	*Upper bound*	a_j	b_j	c_j	e_j	f_j
1	1	100	196	2.697E+01	-3.975E-01	2.176E-03	2.697E-02	-3.975E+00
	2	196	250	2.113E+01	-3.059E-01	1.861E-03	2.113E-02	-3.059E+00
2	2	50	114	1.865E+00	-3.988E-02	1.138E-03	1.865E-03	-3.988E-01
	3	114	157	1.365E+01	-1.980E-01	1.620E-03	1.365E-02	-1.980E+00
	1	157	230	1.184E+02	-1.269E+00	4.194E-03	1.184E-01	-1.269E+01
3	1	200	332	3.979E+01	-3.116E-01	1.457E-03	3.979E-02	-3.116E+00
	3	332	388	-2.875E+00	3.389E-02	8.035E-04	-2.876E-03	3.389E-01
	2	388	500	-5.914E+01	4.864E-01	1.176E-05	-5.914E-02	4.864E+00
4	1	99	138	1.983E+00	-3.114E-02	1.049E-03	1.983E-03	-3.114E-01
	2	138	200	5.285E+01	-6.348E-01	2.758E-03	5.285E-02	-6.348E+00
	3	200	265	2.668E+02	-2.338E+00	5.935E-03	2.668E-01	-2.338E+01
5	1	190	338	1.392E+01	-8.733E-02	1.066E-03	1.392E-02	-8.733E-01
	2	338	407	9.976E+01	-5.206E-01	1.597E-03	9.976E-02	-5.206E+00
	3	407	490	-5.399E+01	4.462E-01	1.498E-04	-5.399E-02	4.462E+00
6	2	85	138	1.983E+00	-3.114E-02	1.049E-03	1.983E-03	-3.114E-01
	1	138	200	5.285E+01	-6.348E-01	2.758E-03	5.285E-02	-6.348E+00
	3	200	265	2.668E+02	-2.338E+00	5.935E-03	2.668E-01	-2.338E+01
7	1	200	331	1.893E+01	-1.325E-01	1.107E-03	1.893E-02	-1.325E+00
	2	331	391	4.377E+01	-2.267E-01	1.165E-03	4.377E-02	-2.267E+00
	3	391	500	-4.335E+01	3.559E-01	2.454E-04	-4.335E-02	3.559E+00
8	1	99	138	1.983E+00	-3.114E-02	1.049E-03	1.983E-03	-3.114E-01
	2	138	200	5.285E+01	-6.348E-01	2.758E-03	5.285E-02	-6.348E+00
	3	200	265	2.668E+02	-2.338E+00	5.935E-03	2.668E-01	-2.338E+01
9	3	130	213	1.423E+01	-1.817E-02	6.121E-04	1.423E-02	-1.817E-01
	1	213	370	8.853E+01	-5.675E-01	1.554E-03	8.853E-02	-5.675E+00
	3	370	440	1.423E+01	-1.817E-02	6.121E-04	1.423E-02	-1.817E-01
10	1	200	362	1.397E+01	-9.938E-02	1.102E-03	1.397E-02	-9.938E-01
	3	362	407	4.671E+01	-2.024E-01	1.137E-03	4.671E-02	-2.024E+00
	2	407	490	-6.113E+01	5.084E-01	4.164E-05	-6.113E-02	5.084E+00

B.2 Best solution obtained by L-HMDE for test cases 2 and 3

Units	Test case 2		Test case 3	
	P_j	Fuel type	P_j	Fuel type
1	218.26855296	2	218.59397122	2
2	211.66085162	1	211.71173943	1
3	280.74415997	1	280.65707656	1
4	239.62565815	3	239.63942993	3
5	278.45951026	1	279.93462813	1
6	239.65624687	3	239.63942815	3
7	288.59537752	1	287.72730808	1
8	239.63664448	3	239.50505679	3
9	428.54030695	3	426.72348627	3
10	274.81269122	1	275.86787544	1
Total Power output (MW)	2700.00		2700.00	
Error (MW)	0.00		0.00	
Operating Cost ($/h)	623.81		623.83	

C.1 Cost coefficient of test cases 4.1 (1800 MW) and 4.2 (2520 MW)

Units	*min*	*max*	a_j	b_j	c_j	e_j	f_j
1	0	680	550	8.1	0.00028	300	0.035
2	0	360	309	8.1	0.00056	200	0.042
3	0	360	307	8.1	0.00056	**200***	0.042
4	60	180	240	7.74	0.00324	150	0.063
5	60	180	240	7.74	0.00324	150	0.063
6	60	180	240	7.74	0.00324	150	0.063
7	60	180	240	7.74	0.00324	150	0.063
8	60	180	240	7.74	0.00324	150	0.063
9	60	180	240	7.74	0.00324	150	0.063
10	40	120	126	8.6	0.00284	100	0.084
11	40	120	126	8.6	0.00284	100	0.084
12	55	120	126	8.6	0.00284	100	0.084
13	55	120	126	8.6	0.00284	100	0.084

C.2 Cost coefficient of test cases 4.3 (1800 MW) and 4.4 (2520 MW)

Units	*min*	*max*	a_j	b_j	c_j	e_j	f_j
1	0	680	550	8.1	0.00028	300	0.035
2	0	360	309	8.1	0.00056	200	0.042
3	0	360	307	8.1	0.00056	**150***	0.042
4	60	180	240	7.74	0.00324	150	0.063
5	60	180	240	7.74	0.00324	150	0.063
6	60	180	240	7.74	0.00324	150	0.063
7	60	180	240	7.74	0.00324	150	0.063
8	60	180	240	7.74	0.00324	150	0.063
9	60	180	240	7.74	0.00324	150	0.063
10	40	120	126	8.6	0.00284	100	0.084
11	40	120	126	8.6	0.00284	100	0.084
12	55	120	126	8.6	0.00284	100	0.084
13	55	120	126	8.6	0.00284	100	0.084

C.3 Best solution obtained by L-HMDE for test case 4

Units	Test case 4.1 P_j	Test case 4.2 P_j	Test case 4.3 P_j	Test case 4.4 P_j
1	628.31853069	628.31853071	628.31853071	628.31853071
2	222.74908724	299.19930034	149.59965017	299.19930033
3	149.59963314	299.19930028	222.74906889	294.48391833
4	109.86654990	159.73310011	109.86655005	159.73310011
5	109.86655005	159.73310011	60	159.73310011
6	109.86654924	159.73310011	109.86655005	159.73310011
7	109.86654978	159.73310011	109.86655003	159.73310011
8	60	159.73310011	109.86655005	159.73310011
9	109.86654996	159.73310010	109.86655005	159.73310011
10	40	77.39991241	40	77.39991252
11	40	77.39991235	40	77.39991252
12	55	92.39990679	55	92.39991247
13	55	87.68453647	55	92.39991246
Total power output (MW)	17963.83	24169.92	17960.37	24164.05
Error (MW)	0.00	0.00	0.00	0.00
Operating Cost ($/h)	1800.00	2520.00	1800.00	2520.00

D.1 Cost coefficient of test cases 5.1 to 5.3 (2520 MW)

Units	*min*	*max*	a_j	b_j	c_j	e_j	f_j
1	0	680	550	8.1	0.00028	300	0.035
2	0	360	309	8.1	0.00056	200	0.042
3	0	360	307	8.1	0.00056	**200***	0.042
4	60	180	240	7.74	0.00324	150	0.063
5	60	180	240	7.74	0.00324	150	0.063
6	60	180	240	7.74	0.00324	150	0.063
7	60	180	240	7.74	0.00324	150	0.063
8	60	180	240	7.74	0.00324	150	0.063
9	60	180	240	7.74	0.00324	150	0.063
10	40	120	126	8.6	0.00284	100	0.084
11	40	120	126	8.6	0.00284	100	0.084
12	55	120	126	8.6	0.00284	100	0.084
13	55	120	126	8.6	0.00284	100	0.084

D.2 Cost coefficient of test case 5.4 (2520 MW)

Units	*min*	*max*	a_j	b_j	c_j	e_j	f_j
1	0	680	550	8.1	0.00028	300	0.035
2	0	360	309	8.1	0.00056	200	0.042
3	0	360	307	8.1	0.00056	**150***	0.042
4	60	180	240	7.74	0.00324	150	0.063
5	60	180	240	7.74	0.00324	150	0.063
6	60	180	240	7.74	0.00324	150	0.063
7	60	180	240	7.74	0.00324	150	0.063
8	60	180	240	7.74	0.00324	150	0.063
9	60	180	240	7.74	0.00324	150	0.063
10	40	120	126	8.6	0.00284	100	0.084
11	40	120	126	8.6	0.00284	100	0.084
12	55	120	126	8.6	0.00284	100	0.084
13	55	120	126	8.6	0.00284	100	0.084

D.3 loss coefficient of test case 5.1

$$B_{gh\,(p.u)} = \begin{bmatrix}
0.0014 & 0.0012 & 0.0007 & -0.0001 & -0.0003 & -0.0001 & -0.0001 & -0.0001 & -0.0003 & -0.0005 & -0.0003 & -0.0002 & 0.0004 \\
0.0012 & 0.0015 & 0.0013 & 0 & -0.0005 & -0.0002 & 0 & 0.0001 & -0.0002 & -0.0004 & -0.0004 & 0 & 0.0004 \\
0.0007 & 0.0013 & 0.0076 & -0.0001 & -0.0013 & -0.0009 & -0.0001 & 0 & -0.0008 & -0.0012 & -0.0017 & 0 & -0.0026 \\
-0.0001 & 0 & -0.0001 & 0.0034 & -0.0007 & -0.0004 & 0.0011 & 0.005 & 0.0029 & 0.0032 & -0.0011 & 0 & 0.0001 \\
-0.0003 & -0.0005 & -0.0013 & -0.0007 & 0.009 & 0.0014 & -0.0003 & -0.0012 & -0.001 & -0.0013 & 0.0007 & -0.0002 & -0.0002 \\
-0.0001 & -0.0002 & -0.0009 & -0.0004 & 0.0014 & 0.0016 & 0 & -0.0006 & -0.0005 & -0.0008 & 0.0011 & -0.0001 & -0.0002 \\
-0.0001 & 0 & -0.0001 & 0.0011 & -0.0003 & 0 & 0.0015 & 0.0017 & 0.0015 & 0.0009 & -0.0005 & 0.0007 & 0 \\
-0.0001 & 0.0001 & 0 & 0.005 & -0.0012 & -0.0006 & 0.0017 & 0.0168 & 0.0082 & 0.0079 & -0.0023 & -0.0036 & 0.0001 \\
-0.0003 & -0.0002 & -0.0008 & 0.0029 & -0.001 & -0.0005 & 0.0015 & 0.0082 & 0.0129 & 0.0116 & -0.0021 & -0.0025 & 0.0007 \\
-0.0005 & -0.0004 & -0.0012 & 0.0032 & -0.0013 & -0.0008 & 0.0009 & 0.0079 & 0.0116 & 0.02 & -0.0027 & -0.0034 & 0.0009 \\
-0.0003 & -0.0004 & -0.0017 & -0.0011 & 0.0007 & 0.0011 & -0.0005 & -0.0023 & -0.0021 & -0.0027 & 0.014 & 0.0001 & 0.0004 \\
-0.0002 & 0 & 0 & 0 & -0.0002 & -0.0001 & 0.0007 & -0.0036 & -0.0025 & -0.0034 & 0.0001 & 0.0054 & -0.0001 \\
0.0004 & 0.0004 & -0.0026 & 0.0001 & -0.0002 & -0.0002 & 0 & 0.0001 & 0.0007 & 0.0009 & 0.0004 & -0.0001 & 0.0103
\end{bmatrix}$$

$$B_{0g\,(p.u)} = \begin{bmatrix} -0.0001 & -0.0002 & 0.0028 & -0.0001 & 0.0001 & -0.0003 & -0.0002 & -0.0002 & 0.0006 & 0.0039 & -0.0017 & 0 & -0.0032 \end{bmatrix}$$

$$B_{00\,(p.u)} = 0.0055$$

D.4 loss coefficient of test case 5.2

$$B_{gh\,(p.u)} = \begin{bmatrix}
0.0014 & 0.0012 & 0.0007 & -0.0001 & -0.0003 & -0.0001 & -0.0001 & -0.0001 & -0.0003 & -0.0005 & -0.0003 & -0.0002 & 0.0004 \\
0.0012 & 0.0015 & 0.0013 & 0 & -0.0005 & -0.0002 & 0 & 0.0001 & -0.0002 & -0.0004 & -0.0004 & 0 & 0.0004 \\
0.0007 & 0.0013 & 0.0076 & -0.0001 & -0.0013 & -0.0009 & -0.0001 & 0 & -0.0008 & -0.0012 & -0.0017 & 0 & -0.0026 \\
-0.0001 & 0 & -0.0001 & 0.0034 & -0.0007 & -0.0004 & 0.0011 & 0.005 & 0.0029 & 0.0032 & -0.0011 & 0 & 0.0001 \\
-0.0003 & -0.0005 & -0.0013 & -0.0007 & 0.009 & 0.0014 & -0.0003 & -0.0012 & -0.001 & -0.0013 & 0.0007 & -0.0002 & -0.0002 \\
-0.0001 & -0.0002 & -0.0009 & -0.0004 & 0.0014 & 0.0016 & 0 & -0.0006 & -0.0005 & -0.0008 & 0.0011 & -0.0001 & -0.0002 \\
-0.0001 & 0 & -0.0001 & 0.0011 & -0.0003 & 0 & 0.0015 & 0.0017 & 0.0015 & 0.0009 & -0.0005 & 0.0007 & 0 \\
-0.0001 & 0.0001 & 0 & 0.005 & -0.0012 & -0.0006 & 0.0017 & 0.0168 & 0.0082 & 0.0079 & -0.0023 & -0.0036 & 0.0001 \\
-0.0003 & -0.0002 & -0.0008 & 0.0029 & -0.001 & -0.0005 & 0.0015 & 0.0082 & 0.0129 & 0.0116 & -0.0021 & -0.0025 & 0.0007 \\
-0.0005 & -0.0004 & -0.0012 & 0.0032 & -0.0013 & -0.0008 & 0.0009 & 0.0079 & 0.0116 & 0.02 & -0.0027 & -0.0034 & 0.0009 \\
-0.0003 & -0.0004 & -0.0017 & -0.0011 & 0.0007 & 0.0011 & -0.0005 & -0.0023 & -0.0021 & -0.0027 & 0.014 & 0.0001 & 0.0004 \\
-0.0002 & 0 & 0 & 0 & -0.0002 & -0.0001 & 0.0007 & -0.0036 & -0.0025 & -0.0034 & 0.0001 & 0.0054 & -0.0001 \\
0.0004 & 0.0004 & -0.0026 & 0.0001 & -0.0002 & -0.0002 & 0 & 0.0001 & 0.0007 & 0.0009 & 0.0004 & -0.0001 & 0.0103
\end{bmatrix}$$

$$B_{0g\,(p.u)} = \begin{bmatrix} -0.0001 & -0.0002 & 0.0028 & -0.0001 & 0.0001 & -0.0003 & -0.0002 & -0.0002 & 0.0006 & 0.0039 & \mathbf{0.0017*} & 0 & -0.0032 \end{bmatrix}$$

$$B_{00\,(p.u)} = 0.0055$$

D.5 loss coefficient of test case 5.3 to 5.4

$$B_{gh\,(p.u)} = \begin{bmatrix}
0.0014 & 0.0012 & 0.0007 & -0.0001 & -0.0003 & -0.0001 & -0.0001 & -0.0001 & -0.0003 & \boldsymbol{0.0005*} & -0.0003 & -0.0002 & 0.0004 \\
0.0012 & 0.0015 & 0.0013 & 0 & -0.0005 & -0.0002 & 0 & 0.0001 & -0.0002 & -0.0004 & -0.0004 & 0 & 0.0004 \\
0.0007 & 0.0013 & 0.0076 & -0.0001 & -0.0013 & -0.0009 & -0.0001 & 0 & -0.0008 & -0.0012 & -0.0017 & 0 & -0.0026 \\
-0.0001 & 0 & -0.0001 & 0.0034 & -0.0007 & -0.0004 & 0.0011 & 0.005 & 0.0029 & 0.0032 & -0.0011 & 0 & 0.0001 \\
-0.0003 & -0.0005 & -0.0013 & -0.0007 & 0.009 & 0.0014 & -0.0003 & -0.0012 & -0.001 & -0.0013 & 0.0007 & -0.0002 & -0.0002 \\
-0.0001 & -0.0002 & -0.0009 & -0.0004 & 0.0014 & 0.0016 & 0 & -0.0006 & -0.0005 & -0.0008 & 0.0011 & -0.0001 & -0.0002 \\
-0.0001 & 0 & -0.0001 & 0.0011 & -0.0003 & 0 & 0.0015 & 0.0017 & 0.0015 & 0.0009 & -0.0005 & 0.0007 & 0 \\
-0.0001 & 0.0001 & 0 & 0.005 & -0.0012 & -0.0006 & 0.0017 & 0.0168 & 0.0082 & 0.0079 & -0.0023 & -0.0036 & 0.0001 \\
-0.0003 & -0.0002 & -0.0008 & 0.0029 & -0.001 & -0.0005 & 0.0015 & 0.0082 & 0.0129 & 0.0116 & -0.0021 & -0.0025 & 0.0007 \\
-0.0005 & -0.0004 & -0.0012 & 0.0032 & -0.0013 & -0.0008 & 0.0009 & 0.0079 & 0.0116 & 0.02 & -0.0027 & -0.0034 & 0.0009 \\
-0.0003 & -0.0004 & -0.0017 & -0.0011 & 0.0007 & 0.0011 & -0.0005 & -0.0023 & -0.0021 & -0.0027 & 0.014 & 0.0001 & 0.0004 \\
-0.0002 & 0 & 0 & 0 & -0.0002 & -0.0001 & 0.0007 & -0.0036 & -0.0025 & -0.0034 & 0.0001 & 0.0054 & -0.0001 \\
0.0004 & 0.0004 & -0.0026 & 0.0001 & -0.0002 & -0.0002 & 0 & 0.0001 & 0.0007 & 0.0009 & 0.0004 & -0.0001 & 0.0103
\end{bmatrix}$$

$$B_{0g\,(p.u)} = \begin{bmatrix} -0.0001 & -0.0002 & 0.0028 & -0.0001 & 0.0001 & -0.0003 & -0.0002 & -0.0002 & 0.0006 & 0.0039 & -0.0017 & 0 & -0.0032 \end{bmatrix}$$

$$B_{00\,(p.u)} = \mathbf{0.000055*}$$

D.6 Best solution obtained by L-HMDE for test cases 5

Units	Test case 5.1 P_j	Test case 5.2 P_j	Test case 5.3 P_j	Test case 5.4 P_j
1	628.31853071	628.31853071	628.31853071	628.31853071
2	299.19930034	299.19930034	299.19930034	299.19930034
3	299.19930031	299.19930034	299.19930034	297.36723637
4	159.73310011	159.73310011	159.73310011	159.73310011
5	159.73310011	159.73310011	159.73310011	159.73310011
6	159.73310011	159.73310011	159.73310011	159.73310011
7	159.73310011	159.73310011	159.73310011	159.73310011
8	159.73310011	159.73310011	159.73310011	159.73310011
9	159.73310011	159.73310011	159.73310011	159.73310011
10	77.39991254	77.39991230	77.39991246	77.39991254
11	113.11115204	113.49588879	113.05314527	114.79982508
12	92.39991240	92.39991220	92.39991243	92.39991254
13	92.39991089	92.39991234	92.39991240	92.39991254
Total power output (MW)	2560.43	2560.81	2560.37	2560.28
Transmission loss (MW)	40.43	40.81	40.37	40.28
Error (MW)	$2.67\cdot10^{-9}$	$8.46\cdot10^{-10}$	$2.44\text{E}\cdot10^{-9}$	$3.25\text{E}\cdot10^{-10}$
Operating Cost ($/h)	24514.88	24515.23	24514.82	24512.43

E.1 Cost coefficient of test case 6.1 (2630 MW)

Unit	*Min*	*Max*	a_j	b_j	c_j	UR_j	DR_j	P^0	*prohibited zones*
1	150	455	671	10.1	0.000299	80	120	400	
2	150	455	574	10.2	0.000183	80	120	**300***	[185, 225] [305, 335] [420, 450]
3	20	130	374	8.8	0.001126	130	130	105	
4	20	130	374	8.8	0.001126	130	130	100	
5	150	470	461	10.4	0.000205	80	120	**90***	[180, 200] [305, 335] [390, 420]
6	135	460	630	10.1	0.000301	80	120	400	[230, 255] [365, 395] [430, 455]
7	135	465	548	9.8	0.000364	80	120	350	
8	60	300	227	11.2	0.000338	65	100	95	
9	25	162	173	11.2	0.000807	60	100	105	
10	25	160	175	10.7	0.001203	60	100	110	
11	20	80	186	10.2	0.003586	80	80	60	
12	20	80	230	9.9	0.005513	80	80	40	[30, 40] [55, 65]
13	25	85	225	13.1	0.000371	80	80	30	
14	15	55	309	12.1	0.001929	55	55	20	
15	15	55	323	12.4	0.004447	55	55	20	

E.2 Cost coefficient of test case 6.2 (2630 MW)

Unit	*Min*	*Max*	a_j	b_j	c_j	UR_j	DR_j	P^0	*prohibited zones*
1	150	455	671	10.1	0.000299	80	120	400	
2	150	455	574	10.2	0.000183	80	120	**360***	[185, 225] [305, 335] [420, 450]
3	20	130	374	8.8	0.001126	130	130	105	
4	20	130	374	8.8	0.001126	130	130	100	
5	150	470	461	10.4	0.000205	80	120	**190***	[180, 200] [305, 335] [390, 420]
6	135	460	630	10.1	0.000301	80	120	400	[230, 255] [365, 395] [430, 455]
7	135	465	548	9.8	0.000364	80	120	350	
8	60	300	227	11.2	0.000338	65	100	95	
9	25	162	173	11.2	0.000807	60	100	105	
10	25	160	175	10.7	0.001203	60	100	110	
11	20	80	186	10.2	0.003586	80	80	60	
12	20	80	230	9.9	0.005513	80	80	40	[30, 40] [55, 65]
13	25	85	225	13.1	0.000371	80	80	30	
14	15	55	309	12.1	0.001929	55	55	20	
15	15	55	323	12.4	0.004447	55	55	20	

E.3 loss coefficient of test cases 6

$$B_{gh\,(p.u)} = \begin{bmatrix} 0.0014 & 0.0012 & 0.0007 & -0.0001 & -0.0003 & -0.0001 & -0.0001 & -0.0001 & -0.0003 & -0.0005 & -0.0003 & -0.0002 & 0.0004 & 0.0003 & -0.0001 \\ 0.0012 & 0.0015 & 0.0013 & 0 & -0.0005 & -0.0002 & 0 & 0.0001 & -0.0002 & -0.0004 & -0.0004 & 0 & 0.0004 & 0.001 & -0.0002 \\ 0.0007 & 0.0013 & 0.0076 & -0.0001 & -0.0013 & -0.0009 & -0.0001 & 0 & -0.0008 & -0.0012 & -0.0017 & 0 & -0.0026 & 0.0111 & -0.0028 \\ -0.0001 & 0 & -0.0001 & 0.0034 & -0.0007 & -0.0004 & 0.0011 & 0.005 & 0.0029 & 0.0032 & -0.0011 & 0 & 0.0001 & 0.0001 & -0.0026 \\ -0.0003 & -0.0005 & -0.0013 & -0.0007 & 0.009 & 0.0014 & -0.0003 & -0.0012 & -0.001 & -0.0013 & 0.0007 & -0.0002 & -0.0002 & -0.0024 & -0.0003 \\ -0.0001 & -0.0002 & -0.0009 & -0.0004 & 0.0014 & 0.0016 & 0 & -0.0006 & -0.0005 & -0.0008 & 0.0011 & -0.0001 & -0.0002 & -0.0017 & 0.0003 \\ -0.0001 & 0 & -0.0001 & 0.0011 & -0.0003 & 0 & 0.0015 & 0.0017 & 0.0015 & 0.0009 & -0.0005 & 0.0007 & 0 & -0.0002 & -0.0008 \\ -0.0001 & 0.0001 & 0 & 0.005 & -0.0012 & -0.0006 & 0.0017 & 0.0168 & 0.0082 & 0.0079 & -0.0023 & -0.0036 & 0.0001 & 0.0005 & -0.0078 \\ -0.0003 & -0.0002 & -0.0008 & 0.0029 & -0.001 & -0.0005 & 0.0015 & 0.0082 & 0.0129 & 0.0116 & -0.0021 & -0.0025 & 0.0007 & -0.0012 & -0.0072 \\ -0.0005 & -0.0004 & -0.0012 & 0.0032 & -0.0013 & -0.0008 & 0.0009 & 0.0079 & 0.0116 & 0.02 & -0.0027 & -0.0034 & 0.0009 & -0.0011 & -0.0088 \\ -0.0003 & -0.0004 & -0.0017 & -0.0011 & 0.0007 & 0.0011 & -0.0005 & -0.0023 & -0.0021 & -0.0027 & 0.014 & 0.0001 & 0.0004 & -0.0038 & 0.0168 \\ -0.0002 & 0 & 0 & 0 & -0.0002 & -0.0001 & 0.0007 & -0.0036 & -0.0025 & -0.0034 & 0.0001 & 0.0054 & -0.0001 & -0.0004 & 0.0028 \\ 0.0004 & 0.0004 & -0.0026 & 0.0001 & -0.0002 & -0.0002 & 0 & 0.0001 & 0.0007 & 0.0009 & 0.0004 & -0.0001 & 0.0103 & -0.0101 & 0.0028 \\ 0.0003 & 0.001 & 0.0111 & 0.0001 & -0.0024 & -0.0017 & -0.0002 & 0.0005 & -0.0012 & -0.0011 & -0.0038 & -0.0004 & -0.0101 & 0.0578 & -0.0094 \\ -0.0001 & -0.0002 & -0.0028 & -0.0026 & -0.0003 & 0.0003 & -0.0008 & -0.0078 & -0.0072 & -0.0088 & 0.0168 & 0.0028 & 0.0028 & -0.0094 & 0.1283 \end{bmatrix}$$

$$B_{0g\,(p.u)} = \begin{bmatrix} -0.0001 & -0.0002 & 0.0028 & -0.0001 & 0.0001 & -0.0003 & -0.0002 & -0.0002 & 0.0006 & 0.0039 & -0.0017 & 0 & -0.0032 & 0.0067 & -0.0064 \end{bmatrix}$$

$$B_{00\,(p.u)} = 0.0055$$

E.4 Best solution obtained by L-HMDE for test case 6

Units	Test case 6.1 P_j	Test case 6.2 P_j
1	455	455
2	380	420
3	130	130
4	130	130
5	170	269.99999981
6	460	460
7	430	430
8	71.74668697	60
9	58.91474709	25.00000001
10	160	62.97623465
11	80	79.99999997
12	80	80
13	25	25
14	15	15
15	15	15
Total power output (MW)	2660.66	2657.98
Transmission loss (MW)	30.66	27.98
Error (MW)	$2.99 \cdot 10^{-9}$	$2.88 \cdot 10^{-9}$
Operating Cost ($/h)	32704.45	32588.92

F.1 Cost coefficient and loss coefficient of test case 7 (2500 MW)

Unit	*Min*	*Max*	a_j	b_j	c_j
1	150	600	1000	18.19	0.00068
2	50	200	970	19.26	0.00071
3	50	200	600	19.8	0.0065
4	50	200	700	19.1	0.005
5	50	160	420	18.1	0.00738
6	20	100	360	19.26	0.00612
7	25	125	490	17.14	0.0079
8	50	150	660	18.92	0.00813
9	50	200	765	18.27	0.00522
10	30	150	770	18.92	0.00573
11	100	300	800	16.69	0.0048
12	150	500	970	16.76	0.0031
13	40	160	900	17.36	0.0085
14	20	130	700	18.7	0.00511
15	25	185	450	18.7	0.00398
16	20	80	370	14.26	0.0712
17	30	85	480	19.14	0.0089
18	30	120	680	18.92	0.00713
19	40	120	700	18.47	0.00622
20	30	100	850	19.79	0.00773

B_{gh}=

0.00870	0.00043	-0.00461	0.00036	0.00032	-0.00066	0.00096	-0.00160	0.00080	-0.00010	0.00360	0.00064	0.00079	0.00210	0.00170	0.00080	-0.00320	0.00070	0.00048	-0.00070
0.00043	0.00830	-0.00097	0.00022	0.00075	-0.00028	0.00504	0.00170	0.00054	0.00720	-0.00028	0.00098	-0.00046	0.00130	0.00080	-0.00020	0.00052	-0.00170	0.00080	0.00020
-0.00461	-0.00097	0.00900	-0.00200	0.00063	0.00300	0.00170	-0.00430	0.00310	-0.00200	0.00070	-0.00077	0.00093	0.00460	-0.00030	0.00420	0.00038	0.00070	-0.00200	0.00360
0.00036	0.00022	-0.00200	0.00530	0.00047	0.00262	-0.00196	0.00210	0.00067	0.00180	-0.00045	0.00092	0.00240	0.00760	-0.00020	0.00070	-0.00100	0.00086	0.00160	0.00087
0.00032	0.00075	0.00063	0.00047	0.00860	-0.00080	0.00037	0.00072	-0.00090	0.00069	0.00180	0.00430	-0.00280	-0.00070	0.00230	0.00360	0.00080	0.00020	-0.00300	0.00050
-0.00066	-0.00028	0.00300	0.00262	-0.00080	0.01180	-0.00490	0.00030	0.00300	-0.00300	0.00040	0.00078	0.00640	0.00260	-0.00020	0.00210	-0.00040	0.00230	0.00160	-0.00210
0.00096	0.00504	0.00170	-0.00196	0.00037	-0.00490	0.00824	-0.00090	0.00590	-0.00060	0.00850	-0.00083	0.00720	0.00480	-0.00090	-0.00010	0.00130	0.00076	0.00190	0.00130
-0.00160	0.00170	-0.00430	0.00210	0.00072	0.00030	-0.00090	0.00120	-0.00096	0.00056	0.00160	0.00080	-0.00040	0.00023	0.00075	-0.00056	0.00080	-0.00030	0.00530	0.00080
0.00080	0.00054	0.00310	0.00067	-0.00090	0.00300	0.00590	-0.00096	0.00093	-0.00030	0.00650	0.00230	0.00260	0.00058	-0.00010	0.00023	-0.00030	0.00150	0.00074	0.00070
-0.00010	0.00720	-0.00200	0.00180	0.00069	-0.00300	-0.00060	0.00056	-0.00030	0.00099	-0.00660	0.00390	0.00230	-0.00030	0.00280	-0.00080	0.00038	0.00190	0.00047	-0.00026
0.00360	-0.00028	0.00070	-0.00045	0.00180	0.00040	0.00850	0.00160	0.00650	-0.00660	0.01070	0.00530	-0.00060	0.00070	0.00190	-0.00260	0.00093	-0.00060	0.00380	-0.00150
0.00064	0.00098	-0.00077	0.00092	0.00430	0.00078	-0.00083	0.00080	0.00230	0.00390	0.00530	0.00800	0.00090	0.00210	-0.00070	0.00570	0.00540	0.00150	0.00070	0.00010
0.00079	-0.00046	0.00093	0.00240	-0.00280	0.00640	0.00720	-0.00040	0.00260	0.00230	-0.00060	0.00090	0.01100	0.00087	-0.00100	0.00360	0.00046	-0.00090	0.00060	0.00150
0.00210	0.00130	0.00460	0.00760	-0.00070	0.00260	0.00480	0.00023	0.00058	-0.00030	0.00070	0.00210	0.00087	0.00380	0.00050	-0.00070	0.00190	0.00230	-0.00097	0.00090
0.00170	0.00080	-0.00030	-0.00020	0.00230	-0.00020	-0.00090	0.00075	-0.00010	0.00280	0.00190	-0.00070	-0.00100	0.00050	0.01100	0.00190	-0.00080	0.00260	0.00230	-0.00010
0.00080	-0.00020	0.00420	0.00070	0.00360	0.00210	-0.00010	-0.00056	0.00023	-0.00080	-0.00260	0.00570	0.00360	-0.00070	0.00190	0.01080	0.00250	-0.00180	0.00090	-0.00260
-0.00320	0.00052	0.00038	-0.00100	0.00080	-0.00040	0.00130	0.00080	-0.00030	0.00038	0.00093	0.00540	0.00046	0.00190	-0.00080	0.00250	0.00870	0.00420	-0.00030	0.00068
0.00070	-0.00170	0.00070	0.00086	0.00020	0.00230	0.00076	-0.00030	0.00150	0.00190	-0.00060	0.00150	-0.00090	0.00230	0.00260	-0.00180	0.00420	0.00220	0.00016	-0.00030
0.00048	0.00080	-0.00200	0.00160	-0.00300	0.00160	0.00190	0.00530	0.00074	0.00047	0.00380	0.00070	0.00060	-0.00097	0.00230	0.00090	-0.00030	0.00016	0.00760	0.00069
-0.00070	0.00020	0.00360	0.00087	0.00050	-0.00210	0.00130	0.00080	0.00070	-0.00026	-0.00150	0.00010	0.00150	0.00090	-0.00010	-0.00260	0.00068	-0.00030	0.00069	0.00700

B_{0g}=

0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

$B_{00} = 0$

F.2 Best solution obtained by L-HMDE for test case 7

Units	P_j	Units	P_j
1	512.79388086	11	150.23692195
2	169.09254956	12	292.76503676
3	126.88055794	13	119.11325013
4	102.88511492	14	30.82309280
5	113.69343302	15	115.80670089
6	73.56109941	16	36.25350691
7	115.28727588	17	66.85580240
8	116.40621683	18	87.97706880
9	100.41370515	19	100.78890765
10	106.02272829	20	54.31001287
		Total power output (MW)	2591.97
		Transmission loss (MW)	91.97
		Error (MW)	$1.08 \cdot 10^{-9}$
		Operating Cost ($/h)	62456.63

G.1 Cost coefficient of test case 8.1 (10500 MW)

Unit	Min	Max	a_i	b_i	c_i	e_i	f_i
1	36	114	94.705	6.73	0.0069	100	0.084
2	36	114	94.705	6.73	0.0069	100	0.084
3	60	120	309.54	7.07	0.02028	100	0.084
4	80	190	369.03	8.18	0.00942	150	0.063
5	47	97	148.89	5.35	0.0114	120	0.077
6	68	140	222.33	8.05	0.01142	100	0.084
7	110	300	287.71	8.03	0.00357	200	0.042
8	135	300	391.98	6.99	0.00492	200	0.042
9	135	300	455.76	6.6	0.00573	200	0.042
10	130	300	722.82	12.9	0.00605	200	0.042
11	94	375	635.2	12.9	0.00515	200	0.042
12	94	375	654.69	12.8	0.00569	200	0.042
13	125	500	913.4	12.5	0.00421	300	0.035
14	125	500	1760.4	8.84	0.00752	300	0.035
15	125	500	1728.3	9.15	0.00708	300	0.035
16	125	500	1728.3	9.15	0.00708	300	0.035
17	220	500	647.85	7.97	0.00313	300	0.035
18	220	500	649.69	7.95	0.00313	300	0.035
19	242	550	647.83	7.97	0.00313	300	0.035
20	242	550	647.81	7.97	0.00313	300	0.035
21	254	550	785.96	6.63	0.00298	300	0.035
22	254	550	785.96	6.63	0.00298	300	0.035
23	254	550	794.53	6.66	0.00284	300	0.035
24	254	550	794.53	6.66	0.00284	300	0.035
25	254	550	801.32	7.1	0.00277	300	0.035
26	254	550	801.32	7.1	0.00277	300	0.035
27	10	150	1055.1	3.33	0.52124	120	0.077
28	10	150	1055.1	3.33	0.52124	120	0.077
29	10	150	1055.1	3.33	0.52124	120	0.077
30	47	97	148.89	5.35	0.0114	120	0.077
31	60	190	222.92	6.43	0.0016	150	0.063
32	60	190	222.92	6.43	0.0016	150	0.063
33	60	190	222.92	6.43	0.0016	150	0.063
34	90	200	107.87	8.95	0.0001	200	0.042
35	90	200	116.58	8.62	0.0001	200	0.042
36	90	200	116.58	8.62	0.0001	200	0.042
37	25	110	307.45	5.88	0.0161	80	0.098
38	25	110	307.45	5.88	0.0161	80	0.098
39	25	110	307.45	5.88	0.0161	80	0.098
40	242	550	647.83	7.97	0.00313	300	0.035

G.2 Cost coefficient of test case 8.2 (10500 MW)

Unit	Min	Max	a_i	b_i	c_i	e_i	f_i
1	36	114	94.705	6.73	0.0069	100	0.084
2	36	114	94.705	6.73	0.0069	100	0.084
3	60	120	309.54	7.07	0.02028	100	0.084
4	80	190	369.03	8.18	0.00942	150	0.063
5	47	97	148.89	5.35	0.0114	120	0.077
6	68	140	222.33	8.05	0.01142	100	0.084
7	110	300	**278.71***	8.03	0.00357	200	0.042
8	135	300	391.98	6.99	0.00492	200	0.042
9	135	300	455.76	6.6	0.00573	200	0.042
10	130	300	722.82	12.9	0.00605	200	0.042
11	94	375	635.2	12.9	0.00515	200	0.042
12	94	375	654.69	12.8	0.00569	200	0.042
13	125	500	913.4	12.5	0.00421	300	0.035
14	125	500	1760.4	8.84	0.00752	300	0.035
15	125	500	1728.3	9.15	0.00708	300	0.035
16	125	500	1728.3	9.15	0.00708	300	0.035
17	220	500	647.85	7.97	0.00313	300	0.035
18	220	500	649.69	7.95	0.00313	300	0.035
19	242	550	647.83	7.97	0.00313	300	0.035
20	242	550	647.81	7.97	0.00313	300	0.035
21	254	550	785.96	6.63	0.00298	300	0.035
22	254	550	785.96	6.63	0.00298	300	0.035
23	254	550	794.53	6.66	0.00284	300	0.035
24	254	550	794.53	6.66	0.00284	300	0.035
25	254	550	801.32	7.1	0.00277	300	0.035
26	254	550	801.32	7.1	0.00277	300	0.035
27	10	150	1055.1	3.33	0.52124	120	0.077
28	10	150	1055.1	3.33	0.52124	120	0.077
29	10	150	1055.1	3.33	0.52124	120	0.077
30	47	97	148.89	5.35	0.0114	120	0.077
31	60	190	222.92	6.43	0.0016	150	0.063
32	60	190	222.92	6.43	0.0016	150	0.063
33	60	190	222.92	6.43	0.0016	150	0.063
34	90	200	107.87	8.95	0.0001	200	0.042
35	90	200	116.58	8.62	0.0001	200	0.042
36	90	200	116.58	8.62	0.0001	200	0.042
37	25	110	307.45	5.88	0.0161	80	0.098
38	25	110	307.45	5.88	0.0161	80	0.098
39	25	110	307.45	5.88	0.0161	80	0.098
40	242	550	647.83	7.97	0.00313	300	0.035

G.3 Cost coefficient of test case 8.3 (10500 MW)

Unit	*Min*	*Max*	a_j	b_j	c_j	e_j	f_j
1	36	114	94.705	6.73	0.0069	100	0.084
2	36	114	94.705	6.73	0.0069	100	0.084
3	60	120	309.54	7.07	0.02028	100	0.084
4	80	190	369.03	8.18	0.00942	150	0.063
5	47	97	148.89	5.35	0.0114	120	0.077
6	68	140	222.33	8.05	0.01142	100	0.084
7	110	300	287.71	8.03	0.00357	200	0.042
8	135	300	391.98	6.99	0.00492	200	0.042
9	135	300	455.76	6.6	0.00573	200	0.042
10	130	300	722.82	12.9	0.00605	200	0.042
11	94	375	635.2	12.9	0.00515	200	0.042
12	94	375	654.69	12.8	0.00569	200	0.042
13	125	500	913.4	12.5	0.00421	300	0.035
14	125	500	1760.4	8.84	0.00752	300	0.035
15	125	500	**1760.4***	**8.84***	**0.00752***	300	0.035
16	125	500	**1760.4***	**8.84***	**0.00752***	300	0.035
17	220	500	647.85	7.97	0.00313	300	0.035
18	220	500	649.69	7.95	0.00313	300	0.035
19	242	550	647.83	7.97	0.00313	300	0.035
20	242	550	647.81	7.97	0.00313	300	0.035
21	254	550	785.96	6.63	0.00298	300	0.035
22	254	550	785.96	6.63	0.00298	300	0.035
23	254	550	794.53	6.66	0.00284	300	0.035
24	254	550	794.53	6.66	0.00284	300	0.035
25	254	550	801.32	7.1	0.00277	300	0.035
26	254	550	801.32	7.1	0.00277	300	0.035
27	10	150	1055.1	3.33	0.52124	120	0.077
28	10	150	1055.1	3.33	0.52124	120	0.077
29	10	150	1055.1	3.33	0.52124	120	0.077
30	47	97	148.89	5.35	0.0114	120	0.077
31	60	190	222.92	6.43	0.0016	150	0.063
32	60	190	222.92	6.43	0.0016	150	0.063
33	60	190	222.92	6.43	0.0016	150	0.063
34	90	200	107.87	8.95	0.0001	200	0.042
35	90	200	116.58	8.62	0.0001	200	0.042
36	90	200	116.58	8.62	0.0001	200	0.042
37	25	110	307.45	5.88	0.0161	80	0.098
38	25	110	307.45	5.88	0.0161	80	0.098
39	25	110	307.45	5.88	0.0161	80	0.098
40	242	550	647.83	7.97	0.00313	300	0.035

G.4 Best solution obtained by L-HMDE for test case 8

Units	Test case 8.1 P_j	Test case 8.2 P_j	Test case 8.3 P_j	Units	Test case 8.1 P_j	Test case 8.2 P_j	Test case 8.3 P_j
1	110.79982538	110.79982705	110.79982513	21	523.27937032	523.27937036	523.27937022
2	110.79982538	110.79982633	110.79982434	22	523.27937036	523.27937032	523.27937029
3	97.39991259	97.39991258	97.39991252	23	523.27937033	523.27937056	523.27937045
4	179.73310014	179.73310021	179.73310005	24	523.27937032	523.27937059	523.27937068
5	87.79990442	87.79990610	87.79990471	25	523.27937041	523.27937055	523.27937033
6	140	140.00000000	140	26	523.27937030	523.27937062	523.27937029
7	259.59965029	259.59965024	259.59965009	27	10	10.00000000	10
8	284.59965030	284.59965083	284.59964982	28	10.00000001	10.00000000	10.00000001
9	284.59965023	284.59965118	284.59965071	29	10.00000001	10.00000000	10
10	130	130.00000003	130.00000002	30	87.79990475	87.79990486	87.79990617
11	94	94.00000000	94.00000025	31	190	190.00000000	190
12	94.00000001	94.00000000	94	32	190	190.00000000	189.99999911
13	214.75979011	214.75979012	214.75978952	33	190	190.00000000	190
14	394.27937031	394.27937031	394.27937021	34	164.79982528	164.79982711	164.79982529
15	394.27937031	394.27937029	394.27937023	35	194.39777649	194.39776735	194.39777880
16	394.27937031	394.27937031	394.27936974	36	200	199.99999996	200
17	489.27937034	489.27937080	489.27937049	37	110	110.00000000	109.99999928
18	489.27937032	489.27937031	489.27937023	38	110	109.99999998	110
19	511.27937034	511.27937032	511.27937029	39	110	110.00000000	110
20	511.27937033	511.27937038	511.27937040	40	511.27937031	511.27937035	511.27937033
Total power output (MW)					10500.00	10500.00	10500.00
Error (MW)					$1.82 \cdot 10^{-12}$	0.00	$1.82 \cdot 10^{-12}$
Operating Cost ($/h)					121412.54	121403.54	121369.08

H.1 Cost coefficient of test case 9 (15000 MW)

Unit	Min	Max	a_j	b_j	c_j
1	2.4	12	24.389	25.547	0.0253
2	2.4	12	24.411	25.675	0.0265
3	2.4	12	24.638	25.803	0.028
4	2.4	12	24.76	25.932	0.0284
5	2.4	12	24.888	26.061	0.0286
6	4	20	117.755	37.551	0.012
7	4	20	118.108	37.664	0.0126
8	4	20	118.458	37.777	0.0136
9	4	20	118.821	37.89	0.0143
10	15.2	76	81.136	13.327	0.0088
11	15.2	76	81.298	13.354	0.0089
12	15.2	76	81.464	13.8	0.0091
13	15.2	76	81.626	13.407	0.0093
14	25	100	217.895	18	0.0062
15	25	100	218.335	18.1	0.0061
16	25	100	218.775	18.2	0.006
17	54.3	155	142.735	10.694	0.0046
18	54.3	155	143.029	10.715	0.0047
19	54.3	155	143.318	10.737	0.0048
20	54.3	155	143.597	10.758	0.0049
21	68.9	197	259.131	23	0.0026
22	68.9	197	259.649	23.1	0.0026
23	68.9	197	260.176	23.2	0.0026
24	140	350	177.057	10.862	0.0015
25	100	400	210.002	7.492	0.0019
26	100	400	211.91	7.503	0.0019
27	140	500	210	12	0.0014
28	140	500	180	12.1	0.0013
29	50	200	240	12.2	0.0026
30	25	100	220	12.5	0.0039
31	10	50	60	23	0.0051
32	5	20	50	13.5	0.005
33	20	80	200	13.2	0.0078
34	75	250	140	12.4	0.0012
35	110	360	120	10.3	0.0038
36	130	400	90	9.9	0.0043
37	10	40	80	13.4	0.0011
38	20	70	70	13.3	0.0023
39	25	100	115	12.9	0.0034
40	20	120	150	12.8	0.0067
41	40	180	40	12.7	0.0056
42	50	220	300	12.6	0.0023
43	120	440	250	7.4	0.0012
44	160	560	100	6.6	0.0045
45	150	660	160	6.5	0.0022
46	200	700	130	6.2	0.0067
47	5.4	32	34.389	26.547	0.0353
48	5.4	32	34.411	26.675	0.0365
49	8.4	52	34.638	26.803	0.038
50	8.4	52	34.761	26.932	0.0384
51	8.4	52	34.888	17.061	0.0386
52	12	60	127.755	38.551	0.032
53	12	60	128.108	36.664	0.0326
54	12	60	128.458	38.777	0.0236
55	12	60	128.821	38.89	0.0243
56	25.2	96	82.136	14.327	0.0098
57	25.2	96	82.298	14.354	0.0099
58	35	100	82.464	14.38	0.0092
59	35	100	82.626	14.407	0.0094
60	45	120	218.895	19	0.0072
61	45	120	219.335	19.1	0.0071
62	45	120	219.775	19.2	0.007
63	54.3	185	143.735	11.694	0.0066
64	54.3	185	144.029	11.715	0.0057
65	54.3	185	144.318	11.737	0.0058
66	54.3	185	144.597	11.758	0.0059
67	70	197	269.131	24	0.0036
68	70	197	269.649	24.1	0.0036
69	70	197	270.176	24.2	0.0036
70	150	360	187.057	11.862	0.0025
71	160	400	320.002	8.492	0.0029
72	160	400	321.91	8.503	0.003
73	60	300	52.136	13.327	0.0054
74	50	250	42.298	12.354	0.0055
75	30	90	32.464	11.38	0.0099
76	12	50	23.626	9.407	0.0031
77	160	450	220	14	0.0024
78	150	600	190	13.1	0.0023
79	50	200	250	13.2	0.0036
80	20	120	230	13.5	0.0049
81	10	55	70	24	0.0061
82	12	40	60	14.5	0.007
83	20	80	210	14.2	0.0088
84	50	200	150	13.4	0.0022
85	80	325	130	11.3	0.0048
86	120	440	80	8.9	0.0053
87	10	35	90	14.4	0.0021
88	20	55	80	14.3	0.0033
89	20	100	125	13.9	0.0034
90	40	220	160	13.8	0.0037
91	30	140	50	13.7	0.0066
92	40	100	400	13.6	0.0043
93	100	440	260	8.4	0.0022
94	100	500	110	7.6	0.0055
95	100	600	170	7.5	0.0032
96	200	700	140	7.2	0.0077
97	3.6	15	26.389	26.547	0.0353
98	3.6	15	25.411	26.675	0.0365
99	4.4	22	25.638	26.803	0.038
100	4.4	22	25.76	26.932	0.0384
101	10	60	65	15.3	0.021
102	10	80	82	16	0.023
103	20	100	86	20.2	0.024
104	20	120	84	20.2	0.035
105	40	150	75	25.6	0.034
106	40	280	56	30.5	0.037
107	50	520	67	32.5	0.039
108	30	150	68	26	0.035
109	40	320	69	25.8	0.028
110	20	200	72	27	0.026

H.2 Best solution obtained by L-HMDE for test cases 9

Units	P_j	Units	P_j	Units	P_j	Units	P_j	Units	P_j
1	2.4	23	68.9	45	660	67	70	89	82.42490232
2	2.4	24	350	46	616.45382762	68	70	90	89.25591921
3	2.4	25	400	47	5.4	69	70	91	57.61102329
4	2.4	26	400	48	5.4	70	360	92	100
5	2.4	27	500	49	8.4	71	400	93	440
6	4	28	500	50	8.4	72	400	94	500
7	4	29	200	51	8.4	73	104.95347226	95	600
8	4	30	100	52	12	74	191.49810138	96	471.45972504
9	4	31	10	53	12	75	90	97	3.6
10	64.40368790	32	20	54	12	76	50	98	3.6
11	62.16272511	33	80	55	12	77	160	99	4.4
12	36.28933659	34	250	56	25.2	78	295.75788666	100	4.4
13	56.63872123	35	360	57	25.2	79	175.06917239	101	10
14	25	36	400	58	35	80	98.01263890	102	10
15	25	37	40	59	35	81	10	103	20
16	25	38	70	60	45	82	12	104	20
17	155	39	100	61	45	83	20	105	40
18	155	40	120	62	45	84	200	106	40
19	155	41	157.18535542	63	184.99999997	85	324.99999999	107	50
20	155	42	220	64	185	86	440	108	30
21	68.9	43	440	65	185	87	14.40761971	109	40
22	68.9	44	560	66	185	88	24.31588500	110	20
						Total power output (MW)			15000.00
						Error (MW)			$1.00 \cdot 10^{-8}$
						Operating Cost ($/h)			197988.18

I.1 Cost coefficient of test cases 10 to 12 (49342 MW)

Unit	a_j	b_j	c_j	*Min*	*Max*	UR_j	DR_j	P^0	e_j	f_j	*prohibited zones*
1	1220.645	61.242	0.032888	71	119	30	120	98.4	-	-	-
2	1315.118	41.095	0.008280	120	189	30	120	134.0	-	-	-
3	874.288	46.310	0.003849	125	190	60	60	141.5	-	-	-
4	874.288	46.310	0.003849	125	190	60	60	183.3	-	-	-
5	1976.469	54.242	0.042468	90	190	150	150	125.0	700	0.080	-
6	1338.087	61.215	0.014992	90	190	150	150	91.3	-	-	-
7	1818.299	11.791	0.007039	280	490	180	300	401.1	-	-	-
8	1133.978	15.055	0.003079	280	490	180	300	329.5	-	-	[250, 280] [305, 335] [420, 450]
9	1320.636	13.226	0.005063	260	496	300	510	386.1	-	-	-
10	1320.636	13.226	0.005063	260	496	300	510	427.3	600	0.055	-
11	1320.636	13.226	0.005063	260	496	300	510	412.2	-	-	-
12	1106.539	14.498	0.003552	260	496	300	510	370.1	-	-	-
13	1176.504	14.651	0.003901	260	506	600	600	301.8	-	-	-
14	1176.504	14.651	0.003901	260	509	600	600	368.0	-	-	-
15	1176.504	14.651	0.003901	260	506	600	600	301.9	800	0.060	-
16	1176.504	14.651	0.003901	260	505	600	600	476.4	-	-	-
17	1017.406	15.669	0.002393	260	506	600	600	283.1	-	-	-
18	1017.406	15.669	0.002393	260	506	600	600	414.1	-	-	-
19	1229.131	14.656	0.003684	260	505	600	600	328.0	-	-	-
20	1229.131	14.656	0.003684	260	505	600	600	389.4	-	-	-
21	1229.131	14.656	0.003684	260	505	600	600	354.7	-	-	-
22	1229.131	14.656	0.003684	260	505	600	600	262.0	600	0.050	-
23	1267.894	14.378	0.004004	260	505	600	600	461.5	-	-	-
24	1229.131	14.656	0.003684	260	505	600	600	371.6	-	-	-
25	975.926	16.261	0.001619	280	537	300	300	462.6	-	-	-
26	1532.093	13.362	0.005093	280	537	300	300	379.2	-	-	-
27	641.989	17.203	0.000993	280	549	360	360	530.8	-	-	-
28	641.989	17.203	0.000993	280	549	360	360	391.9	-	-	-
29	911.533	15.274	0.002473	260	501	180	180	480.1	-	-	-
30	910.533	15.212	0.002547	260	501	180	180	319.0	-	-	-
31	1074.810	15.033	0.003542	260	506	600	600	329.5	-	-	-
32	1074.810	15.033	0.003542	260	506	600	600	333.8	-	-	[220, 250] [320, 350] [390, 420]
33	1074.810	15.033	0.003542	260	506	600	600	390.0	600	0.043	-
34	1074.810	15.033	0.003542	260	506	600	600	432.0	-	-	-
35	1278.460	13.992	0.003132	260	500	660	660	402.0	-	-	-
36	861.742	15.679	0.001323	260	500	900	900	428.0	-	-	-
37	408.834	16.542	0.002950	120	241	180	180	178.4	-	-	-
38	408.834	16.542	0.002950	120	241	180	180	194.1	-	-	-
39	1288.815	16.518	0.000991	423	774	600	600	474.0	-	-	-
40	1436.251	15.815	0.001581	423	769	600	600	609.8	600	0.043	-
41	669.988	75.464	0.902360	3	19	210	210	17.8	-	-	-
42	134.544	129.544	0.110295	3	28	366	366	6.9	-	-	-
43	3427.912	56.613	0.024493	160	250	702	702	224.3	-	-	-
44	3751.772	54.451	0.029156	160	250	702	702	210.0	-	-	-
45	3918.780	54.736	0.024667	160	250	702	702	212.0	-	-	-
46	3379.580	58.034	0.016517	160	250	702	702	200.8	-	-	-
47	3345.296	55.981	0.026584	160	250	702	702	220.0	-	-	-
48	3138.754	61.520	0.007540	160	250	702	702	232.9	-	-	-
49	3453.050	58.635	0.016430	160	250	702	702	168.0	-	-	-
50	5119.300	44.647	0.045934	160	250	702	702	208.4	-	-	-
51	1898.415	71.584	0.000044	165	504	1350	1350	443.9	-	-	-
52	1898.415	71.584	0.000044	165	504	1350	1350	426.0	1100	0.043	-
53	1898.415	71.584	0.000044	165	504	1350	1350	434.1	-	-	-
54	1898.415	71.584	0.000044	165	504	1350	1350	402.5	-	-	-
55	2473.390	85.120	0.002528	180	471	1350	1350	357.4	-	-	-
56	2781.705	87.682	0.000131	180	561	720	720	423.0	-	-	-
57	5515.508	69.532	0.010372	103	341	720	720	220.0	-	-	-
58	3478.300	78.339	0.007627	198	617	2700	2700	369.4	-	-	-
59	6240.909	58.172	0.012464	100	312	1500	1500	273.5	-	-	-
60	9960.110	46.636	0.039441	153	471	1656	1656	336.0	-	-	-
61	3671.997	76.947	0.007278	163	500	2160	2160	432.0	-	-	-
62	1837.383	80.761	0.000044	95	302	900	900	220.0	-	-	-
63	3108.395	70.136	0.000044	160	511	1200	1200	410.6	-	-	-
64	3108.395	70.136	0.000044	160	511	1200	1200	422.7	-	-	-
65	7095.484	49.840	0.018827	196	490	1014	1014	351.0	-	-	-
66	3392.732	65.404	0.010852	196	490	1014	1014	296.0	-	-	-
67	7095.484	49.840	0.018827	196	490	1014	1014	411.1	-	-	-
68	7095.484	49.840	0.018827	196	490	1014	1014	263.2	-	-	-
69	4288.320	66.465	0.034560	130	432	1350	1350	370.3	-	-	-
70	13813.001	22.941	0.081540	130	432	1350	1350	418.7	1200	0.030	-
71	4435.493	64.314	0.023534	137	455	1350	1350	409.6	-	-	-
72	9750.750	45.017	0.035475	137	455	1350	1350	412.0	1000	0.050	-
73	1042.366	70.644	0.000915	195	541	780	780	423.2	-	-	-
74	1159.895	70.959	0.000044	175	536	1650	1650	428.0	-	-	[230, 255] [365, 395] [430, 455]
75	1159.895	70.959	0.000044	175	540	1650	1650	436.0	-	-	-
76	1303.990	70.302	0.001307	175	538	1650	1650	428.0	-	-	-
77	1156.193	70.662	0.000392	175	540	1650	1650	425.0	-	-	-
78	2118.968	71.101	0.000087	330	574	1620	1620	497.2	-	-	-
79	779.519	37.854	0.000521	160	531	1482	1482	510.0	-	-	-
80	829.888	37.768	0.000498	160	531	1482	1482	470.0	-	-	-
81	2333.690	67.983	0.001046	200	542	1668	1668	464.1	-	-	-
82	2028.954	77.838	0.132050	56	132	120	120	118.1	-	-	-
83	4412.017	63.671	0.096968	115	245	180	180	141.3	-	-	-
84	2982.219	79.458	0.054868	115	245	120	180	132.0	1000	0.050	-
85	2982.219	79.458	0.054868	115	245	120	180	135.0	-	-	-
86	3174.939	93.966	0.014382	207	307	120	180	252.0	-	-	-
87	3218.359	94.723	0.013161	207	307	120	180	221.0	-	-	-
88	3723.822	66.919	0.016033	175	345	318	318	245.9	-	-	-
89	3551.405	68.185	0.013653	175	345	318	318	247.9	-	-	-
90	4322.615	60.821	0.028148	175	345	318	318	183.6	-	-	-
91	3493.739	68.551	0.013470	175	345	318	318	288.0	-	-	-
92	226.799	2.842	0.000064	360	580	18	18	557.4	-	-	-
93	382.932	2.946	0.000252	415	645	18	18	529.5	-	-	-
94	156.987	3.096	0.000022	795	984	36	36	800.8	-	-	-

I.1 Cost coefficient of test cases 10 to 12 (continue)

Unit	a_j	b_j	c_j	*Min*	*Max*	UR_j	DR_j	P^0	e_j	f_j	*prohibited zones*
95	154.484	3.040	0.000022	795	978	36	36	801.5	-	-	-
96	332.834	1.709	0.000203	578	682	138	204	582.7	-	-	-
97	326.599	1.668	0.000198	615	720	144	216	680.7	-	-	-
98	345.306	1.789	0.000215	612	718	144	216	670.7	-	-	-
99	350.372	1.815	0.000218	612	720	144	216	651.7	-	-	-
100	370.377	2.726	0.000193	758	964	48	48	921.0	-	-	-
101	367.067	2.732	0.000197	755	958	48	48	916.8	-	-	-
102	124.875	2.651	0.000324	750	1007	36	54	911.9	-	-	-
103	130.785	2.798	0.000344	750	1006	36	54	898.0	-	-	-
104	878.746	1.595	0.000690	713	1013	30	30	905.0	-	-	-
105	827.959	1.503	0.000650	718	1020	30	30	846.5	-	-	-
106	432.007	2.425	0.000233	791	954	30	30	850.9	-	-	-
107	445.606	2.499	0.000239	786	952	30	30	843.7	-	-	-
108	467.223	2.674	0.000261	795	1006	36	36	841.4	-	-	-
109	475.940	2.692	0.000259	795	1013	36	36	835.7	-	-	-
110	899.462	1.633	0.000707	795	1021	36	36	828.8	-	-	-
111	1000.367	1.816	0.000786	795	1015	36	36	846.0	-	-	-
112	1269.132	89.830	0.014355	94	203	120	120	179.0	-	-	-
113	1269.132	89.830	0.014355	94	203	120	120	120.8	-	-	-
114	1269.132	89.830	0.014355	94	203	120	120	121.0	-	-	-
115	4965.124	64.125	0.030266	244	379	480	480	317.4	-	-	-
116	4965.124	64.125	0.030266	244	379	480	480	318.4	-	-	-
117	4965.124	64.125	0.030266	244	379	480	480	335.8	-	-	-
118	2243.185	76.129	0.024027	95	190	240	240	151.0	-	-	-
119	2290.381	81.805	0.001580	95	189	240	240	129.5	600	0.070	-
120	1681.533	81.140	0.022095	116	194	120	120	130.0	-	-	-
121	6743.302	46.665	0.076810	175	321	180	180	218.9	1200	0.043	-
122	394.398	78.412	0.953443	2	19	90	90	5.4	-	-	-
123	1243.165	112.088	0.000044	4	59	90	90	45.0	-	-	-
124	1454.740	90.871	0.072468	15	83	300	300	20.0	-	-	-
125	1011.051	97.116	0.000448	9	53	162	162	16.3	-	-	-
126	909.269	83.244	0.599112	12	37	114	114	20.0	-	-	-
127	689.378	95.665	0.244706	10	34	120	120	22.1	-	-	-
128	1443.792	91.202	0.000042	112	373	1080	1080	125.0	-	-	-
129	535.553	104.501	0.085145	4	20	60	60	10.0	-	-	-
130	617.734	83.015	0.524718	5	38	66	66	13.0	-	-	-
131	90.966	127.795	0.176515	5	19	12	6	7.5	-	-	-
132	974.447	77.929	0.063414	50	98	300	300	53.2	-	-	-
133	263.810	92.779	2.740485	5	10	6	6	6.4	-	-	-
134	1335.594	80.950	0.112438	42	74	60	60	69.1	-	-	-
135	1033.871	89.073	0.041529	42	74	60	60	49.9	-	-	-
136	1391.325	161.288	0.000911	41	105	528	528	91.0	-	-	[50, 75] [85, 95]
137	4477.110	161.829	0.005245	17	51	300	300	41.0	-	-	-
138	57.794	84.972	0.234787	7	19	18	30	13.7	-	-	-
139	57.794	84.972	0.234787	7	19	18	30	7.4	-	-	-
140	1258.437	16.087	1.111878	26	40	72	120	28.6	-	-	-

I.2 Best solution obtained by L-HMDE for test cases 10 to 12

Units	Test case 10 P_j	Test case 11 P_j	Test case 12 P_j	Units	Test case 10 P_j	Test case 11 P_j	Test case 12 P_j
1	118.99999989	115.14972113	119	71	141.53455011	137	141.58735214
2	164	189	164	72	388.32741228	325.49555919	365.91162972
3	190	190	190	73	195.00003202	195	195.00000093
4	190	190	190	74	196.39681466	175	217.69222848
5	168.53981636	168.53981634	189.9999999	75	196.16845947	175.00000169	217.33482512
6	190	190	190	76	257.95272338	175	258.68449517
7	490	490	490	77	400.84087678	175	403.28753976
8	490	490	490	78	330.00000028	330	330.00000309
9	496	496	496	79	531	531	531
10	495.99999996	496	496	80	531	531	531
11	496	496	496	81	541.99999997	398.04628436	542
12	496	496	496	82	56	56	56
13	506	506	506	83	115	115	115
14	509	509	508.99999999	84	115	115	115
15	506	506	506	85	115	115	115
16	505	505	505	86	207.0000001	207	207
17	506	506	506	87	207	207	207
18	506	506	506	88	175.00000062	175	175
19	505	505	505	89	175.00000001	175	175
20	505	505	505	90	180.38665746	175.00000524	180.42194708
21	505	505	505	91	175	175	175
22	505	505	505	92	575.4	580	575.4
23	505	505	505	93	547.5	645	547.5
24	505	505	505	94	836.8	984	836.8
25	537	537	537	95	837.5	978	837.5
26	537	537	537	96	682	682	682
27	549	549	549	97	720	720	720
28	549	549	549	98	718	718	718
29	501	501	501	99	720	720	720
30	498.99999997	501	499	100	964	964	964
31	506	506	506	101	958	957.99999999	958
32	506	505.99999997	506	102	947.89999999	1007	947.9
33	506	506	506	103	934	1006	934
34	506	506	506	104	935	1013	935
35	500	500	500	105	876.5	1020	876.5
36	500	500	500	106	880.9	954	880.9
37	241	241	241	107	873.7	952	873.7
38	241	241	241	108	877.4	1006	877.4
39	774	774	774	109	871.7	1013	871.7
40	769	769	768.99999999	110	864.8	1021	864.8
41	3.00000001	3	3	111	882	1015	882
42	3	3	3	112	94	94	94
43	250	249.10439642	250	113	94	94	94
44	250	246.36018119	250	114	94	94	94
45	249.99999992	250	250	115	244	244	244
46	250	250	250	116	244	244	244
47	250	241.40764330	249.99999998	117	244	244	244
48	250	249.99999989	250	118	95	95	95.00000002
49	250	250	250	119	95	95	95
50	249.9999997	249.99999976	249.99999997	120	116	116	116
51	165.00000032	165.00000001	165.00000006	121	175	175	175
52	165.00000001	165.00000001	165	122	2	2	2
53	165.00000011	165.00000001	165	123	4	4	4
54	165.00000017	165.00000022	165.00000196	124	15	15	15
55	180	180	180	125	9	9	9
56	180	180	180	126	12	12	12
57	103.00000036	103	103	127	10	10	10
58	198	198	198	128	112.0000000500	112	112
59	312	312	312	129	4	4	4
60	308.56994756	281.17663849	308.59076625	130	5	5.00000002	5
61	163	163	163	131	5	5	5
62	95	95	95	132	50	50	50
63	511	160.00000056	510.99999997	133	5	5	5
64	511	160	510.99999997	134	42	42	42
65	490	489.99999926	490	135	42.00000001	42	42
66	256.74320067	196	256.827941	136	41	41	41
67	489.99999729	490	489.99999999	137	17	17	17
68	490	489.9999978	490	138	7	7	7
69	130	130	130	139	7	7	7
70	339.43951027	234.71975515	294.56126946	140	26.00000024	26	26
				Total power output (MW)	49342.00	49342.00	49342.00
				Error (MW)	0.00	0.00	0.00
				Operating Cost ($/h)	1657962.73	1559708.45	1655679.43

J.1 Best solution obtained by L-HMDE for test cases 13 (43200 MW)

Units	P_j	Units	P_j	Units	P_j	Units	P_j	Units	P_j
1	217.44689923	33	278.71598059	65	278.83944863	97	287.89776782	129	424.24659375
2	212.19787520	34	240.44970827	66	240.31301064	98	240.44464970	130	272.43674026
3	281.61086120	35	279.06289000	67	286.66232663	99	431.63179954	131	216.36034759
4	237.89270764	36	240.17776566	68	240.04635432	100	274.60740859	132	210.96202226
5	275.84755977	37	288.81313400	69	423.12461093	101	219.62385488	133	281.55035117
6	240.44087023	38	241.39149546	70	273.68671573	102	211.22601586	134	239.50570195
7	292.95451782	39	421.56673093	71	216.56724320	103	282.73793258	135	276.46780292
8	239.64184145	40	272.72158598	72	211.71059590	104	239.37098223	136	240.44408821
9	423.70206593	41	218.16860851	73	282.64302347	105	277.30010716	137	289.83913000
10	276.12974354	42	211.96764349	74	238.96744012	106	239.50629559	138	238.83235014
11	217.30979861	43	280.63730121	75	280.45091069	107	291.81524379	139	426.57917727
12	212.71785352	44	239.50641501	76	240.17759150	108	240.44182133	140	275.68946229
13	279.69865195	45	279.14819642	77	291.71573687	109	430.56977057	141	218.30889440
14	239.23867833	46	239.23356552	78	239.90777301	110	275.55311062	142	212.94446692
15	280.68648543	47	290.34777988	79	428.34924504	111	217.92749486	143	280.74032226
16	239.50462259	48	238.69180339	80	273.97292630	112	212.44341410	144	238.83151132
17	289.97301302	49	429.42489574	81	219.86910612	113	279.40061515	145	275.41956341
18	239.49725913	50	276.03807548	82	211.97530416	114	239.90856985	146	239.10205401
19	425.70340400	51	216.90563265	83	278.64666111	115	280.47084245	147	287.86144619
20	272.96740311	52	211.21483416	84	239.64107650	116	240.04487080	148	239.76989910
21	217.79845618	53	279.57469032	85	279.83458853	117	291.39553705	149	426.43086397
22	213.44637109	54	239.50408513	86	240.58129403	118	240.30874489	150	275.90337025
23	279.51328894	55	275.39545274	87	291.33188596	119	428.60272236	151	218.69082106
24	237.62139836	56	240.18062648	88	240.17807651	120	275.85572629	152	211.71270250
25	277.38925496	57	288.03323509	89	427.34995080	121	215.58581251	153	283.85981539
26	239.23996369	58	239.77483265	90	276.48278217	122	212.69596702	154	237.89348052
27	292.07263888	59	428.53354902	91	217.76282024	123	281.43329434	155	279.25855927
28	240.57889774	60	274.65843570	92	210.71987710	124	239.77499383	156	238.96620093
29	432.22246834	61	217.21963369	93	283.68525408	125	274.96759583	157	288.42020655
30	273.97545602	62	212.22241714	94	239.91253332	126	238.69835538	158	240.85010960
31	220.77753775	63	279.57012561	95	278.85545055	127	292.79185733	159	430.35166766
32	210.20589036	64	239.77529777	96	240.44907093	128	239.36839768	160	274.32786211
								Total power output (MW)	9983.35
								Error (MW)	0.00
								Operating Cost ($/h)	43200.00