

Enhanced Differential Evolution: A Novel Strategy for Global Optimization and Its Application to Optimal Microgrid Planning

Jin-Yang Lin, Xiang-Yu Shi, Xi Zhu, Jun-De Lin

Research Center for Microelectronics Technology
Fujian University of Technology, Fuzhou 350118, China
lin1299217@163.com, 540749317@qq.com, 1315058354@qq.com, 2865031369@qq.com

Trong-The Nguyen*

Multimedia Communications Lab.
VNU-HCM, University of Information Technology, Vietnam
Vietnam National University, Ho Chi Minh City 700000, Vietnam
thent@uit.edu.vn

*Corresponding author: Trong-The Nguyen

Received November 2, 2023, revised January 9, 2024, accepted April 2, 2024.

ABSTRACT. *This study proposes a novel strategy called Enhanced Crossover and Mutation Differential Evolution (ECMDE) to address the limitations of the Differential Evolution (DE) algorithm. DE often suffers from slow convergence speed, low accuracy, and vulnerability to local optima. The ECMDE algorithm aims to overcome these issues by employing adaptive crossover and mutation techniques. A chaotic mapping approach is used to initialize the population in the initialization stage, ensuring that the initial population is well-distributed across the search space. In the improved mutation operation, a new mutation strategy is proposed using a dynamic adaptive factor instead of F to address the late convergence issue. Additionally, an adaptive crossover operator is employed. To evaluate the effectiveness of the ECMDE algorithm, it is compared with other algorithms using the CEC2013 benchmark function. The results of the comparative study demonstrate that the ECMDE algorithm exhibits superior global convergence ability and faster convergence speed compared to the other algorithms tested. Finally, the ECMDE algorithm is applied to optimal microgrid planning issues, providing effective optimization performance and reducing the operating cost of the microgrid scheme.*

Keywords: Microgrid planning; Swarm intelligence algorithm; Energy storage optimization; Differential Evolution; Crossover and mutation strategy.

1. **Introduction.** Microgrids typically consist of multiple energy sources such as solar photovoltaic, wind power, and energy storage systems, allowing them to independently produce, store independently, and supply electricity [1]. In addition, microgrids can also be connected to the traditional power grid as a part of its distributed energy system [2]. Microgrids can be divided into three types: stand-alone, interconnected, and islanded, each suitable for different application scenarios [3]. Microgrids have a wide range of applications, including residential communities, commercial buildings, industrial parks, and rural areas. Microgrids play a crucial role in addressing energy supply shortages, promoting energy transition, and reducing carbon emissions [4]. With the continuous development of clean and distributed energy technologies, microgrids have a broadening

application scope and market prospects. In the future of energy development, microgrids are expected to become an essential component, contributing to the promotion of energy transition and carbon emissions reduction [5]. Microgrids have several advantages compared to traditional power grids [6], including:

- High reliability: Microgrids can operate independently, which reduces the risk of large-scale power outages.
- Good sustainability: Microgrids can use renewable energy, such as solar and wind energy, to achieve the utilization of clean energy.
- Excellent economy: Microgrids can adopt distributed power supply, which reduces transmission loss and energy costs.

The operation of a microgrid system is a complex task that requires a sophisticated control system as distributed power sources are subject to uncertainty. Weather, light, and wind speed can affect the output power of these sources, making the output power unpredictable [7]. To optimize a microgrid, various constraints such as power supply reliability, voltage stability, frequency stability, and power balance must be considered comprehensively [8, 9]. This increases the complexity of microgrid optimization, making it essential for maximizing economic benefits and reducing operating costs [10, 11].

However, microgrid optimization involves multi-objective, multi-energy, and dynamic uncertain load demand, making it a challenging optimization problem [12]. One solution is to apply metaheuristic algorithms. In recent years, the development and widespread use of computer technology has led to a rapid growth of meta-heuristic algorithms [13]. These algorithms aim at optimization and achieve excellent optimization performance through combining and improving existing algorithms [14, 15]. The main advantage of meta-heuristic algorithms is their high flexibility and adaptability, allowing them to solve various optimization problems quickly [16, 17]. The flexibility of these algorithms is demonstrated by their ability to design and adjust customized algorithms for different problem structures and constraints [18]. Meanwhile, their adaptability is highlighted by their ability to continuously optimize the performance and efficiency of algorithms through mechanisms such as learning and evolution [19][20].

Classic meta-heuristic algorithms such as particle swarm optimization (PSO) [21], differential evolution (DE) [22], Artificial bees colony (ABC)[23], Seagull optimization algorithm (SOA) [24] and grey wolf algorithm (GWO)[25] have been widely used and verified for their optimization effects. Additionally, meta-heuristic algorithms based on deep learning and reinforcement learning have rapidly developed and applied in recent years. The significance of metaheuristic algorithms is not only in the academic community but also in practical applications [26]. In manufacturing, transportation, energy management, and other fields, meta-heuristic algorithms have been widely used to solve optimization problems and have achieved significant economic and social benefits [27]. For optimizing the microgrid, this article will use the DE algorithm as the optimization method [28]. The search process of the algorithm can be controlled by setting appropriate parameters such as population size, crossover rate, mutation rate, etc [29]. To enhance the search ability of the algorithm and reduce the risk of falling into local optima, improvements will be made to improve its performance.

This article is structured as follows: section 2 introduces the microgrid model and the differential evolution algorithm. Section 3 presents the improved differential evolution algorithm (ECMDE) and compare it with other algorithms on the CEC2013 test function. In Section 4 applies the ECMDE algorithm to optimize the microgrid and evaluate the optimization results. Finally, section 5 provides a summary of the article.

2. Related work.

2.1. Microgrid Model.

The distributed power generation in a microgrid mainly includes photovoltaic (PV), wind turbines (WT), energy storage (ES), microturbine (MT), and other sources [3][30]. The mathematical models for these components will be introduced below.

Wind turbine (WT) electricity production is affected by wind speed v . When the wind speed falls below the cut-in wind speed, the WT produces no electricity. The power production rises as the wind speed grows within the range between the cut-in wind speed and the specified wind speed. When the wind speed is between the specified and cut-out wind speeds, the generator produces electricity at its rated capacity [4][8]. When the wind speed exceeds the cut-out wind speed, the generator will engage a self-protection state and shut down. As a result, the wind machine can only function properly when the wind speed is within a certain range. The common fan output power function is:

$$P_w = \begin{cases} 0 & v < v_{in}, v > v_{out} \\ v^3(P_{wr}/(v_r^3 - v_{in}^3)) - P_{wr}((v_{in}^3)/(v_r^3 - v_{in}^3)) & v_{in} \leq v < v_r \\ P_{wr} & v_r \leq v < v_{out} \end{cases} \quad (1)$$

We define P_w as a wind turbine's electricity production, P_{wr} as its rated output power, v as the real-time wind speed during each time interval, v_r , v_{in} and v_{out} as the wind speeds at the rated capacity, cut-in threshold, and cut-out limit.

The microturbine (MT) has a power range of 25-300 kW and is known for its high efficiency, low emissions, and compact size [31]. The amount of fuel consumed is directly related to the electricity generated by the microturbine. The relationship between generation efficiency and output power can be expressed as follows:

$$\eta_{MT} = 0.4174\left(\frac{P_{MT}}{65}\right) - 0.3095\left(\frac{P_{MT}}{65}\right)^2 + 0.0753\left(\frac{P_{MT}}{65}\right)^3 + 0.1068 \quad (2)$$

The formula of microturbine fuel cost is as follows:

$$C_{MT}(t) = \frac{C_p}{LHV} \times \frac{P_{MT}}{\eta_{MT}} \quad (3)$$

The fuel price is denoted by C_p , while the lower calorific value of gas is denoted by LHV.

Diesel Generators have high reliability and mature operation and maintenance technology [32]. When the other units' power generation is insufficient, Diesel can be supplemented in the microgrid, and the fuel cost of DE using the quadratic function expression is:

$$C_{DE} = \alpha + \beta P_{DE} + \gamma P_{DE}^2 \quad (4)$$

We use C_{DE} is Diesel Generators to denote the fuel cost, P_{DE} to denote the output power, α , β and γ to denote the fuel cost coefficient.

Photovoltaic (PV) power generation is a process that utilizes photovoltaic cells, which are semiconductor devices that exhibit the photovoltaic effect, to convert incident light energy into electrical energy through the generation of an electric current, and its output power is proportional to the intensity of the light and the ambient temperature. To determine the total power of a solar power production device, use the following formula:

$$P_{pv} = P_{pvr} \times [1 + K_t(0.0256G + T - T_{tef})] \times \frac{G}{G_{ref}} \quad (5)$$

Where P_{pv} is the photovoltaic power generation system's output power; P_{pvr} is the rated power under reference conditions; G indicates the actual light intensity, Light intensity under reference conditions: $G_{ref}=1000$ W/m²; $K_t=-3.7 \times 10^{-3}$ (1/°C) is the temperature

coefficient of photovoltaic panel; T is the ambient temperature; $T_{ref}=25\text{ }^\circ\text{C}$ is the reference temperature.

Energy storage (ES) systems can effectively track changes in wind and solar energy output to facilitate charging and discharging, serving as a buffer for the power grid and enhancing the reliability and continuity of power supply. Specifically, when the total output power of distributed generation exceeds the total load, the ES system will be charged, while it will be discharged when the total output power is less than the total load. The battery's charge and discharge status can be expressed as:

$$E_S B(t) = \begin{cases} E_S B(t-1) + ((P_{load}(t) - P_{load}(t)/\eta_{inv})\eta_{sb}\Delta t, & \text{Charge} \\ E_S B(t-1) - (\frac{P_{load}(t)}{\eta_{inv}} - P_{total}(t))\eta_{sb}\Delta t, & \text{Discharge} \end{cases} \quad (6)$$

At any given time t , $P_{total}(t)$ indicates the micro power supply's overall capacity, while $P_{load}(t)$ indicates the system's overall load. Additionally, η_{inv} and η_{sb} respectively denote the working efficiency of the inverter and the energy storage (ES) system's filling and discharging effectiveness [33].

In this paper, the system still needs to meet some inequality constraints during operation, among which the output power of distributed generation needs to meet its upper and lower limits, which can be stated as follows:

$$P_i^{min}(t) \leq P_i^t(t) \leq P_i^{max}(t) \quad (7)$$

In this formula, P_i^t , P_i^{min} , P_i^{max} represent the actual power, minimum power and maximum power output by the micro-power i when the system is running.

The battery in the system satisfies its own upper and lower limit constraints which can be expressed as:

$$P_{bt}^{min} \leq P_{bt}(t) \leq P_{bt}^{max} \quad (8)$$

In the formula, P_{bt}^{min} are the minimum active power output by the battery when working; P_{bt}^{max} is the maximum active power output by the battery when working.

The limitations of power exchange with the external power grid should also be taken into account while the system is in operation. can be said to be:

$$P_{EX}^{min} \leq P_{grid} \leq P_{EX}^{max} \quad (9)$$

The highest power the system can interchange with the outside power grid is P_{EX}^{max} , and the least power it can exchange with the outside power grid is P_{EX}^{min} . In this paper, the equality constraints mainly include power balance constraints and primary charge and discharge constraints of batteries, and their corresponding formulas are as follows:

$$\sum_{i=1}^M P_i(t) + P_{grid}(t) - P_{load}(t) = 0, t = 1, 2, \dots, M, \quad (10)$$

Where m represents the type of distributed power supply; $P_i(t)$ is the active power output by micro-power supply i in t period; when the load consumption exceeds the system's electricity production, the value of $P_{grid}(t)$, which represents the power transferred between the system and the external power grid during the specified time period, is "+", and "-" when it is not; $P_{load}(t)$ is the amount of power used by the load side over the course of t .

2.2. Differential evolution (DE) algorithm.

The DE algorithm is a type of stochastic optimization method commonly used for solving global optimization problems [22]. It mimics the principles of evolution, where candidate solutions undergo random mutations and selection processes. At each iteration,

DE generates a new population of candidate solutions by recombining and mutating the existing solutions and then selects the best ones to form the next population. This mutation and recombination process are performed using the difference between randomly candidate chosen solutions. The mutated candidate solutions are then evaluated using a fitness function, and the best-performing solutions are selected for the next iteration. It is an efficient and effective optimization technique.

2.2.1. Algorithm initialization.

In an optimization problem, each individual in a population represents a potential solution, and their position information is used to determine the candidate solutions [22]. Before optimization begins, the position of population members must be initialized to ensure an even distribution throughout the D -dimensional optimization space, which typically corresponds to a D -dimensional space. A random method is often employed to generate the initial population position information distribution. Specifically, the population size is denoted as NP and the initial population position distribution is calculated using the following formula:

$$x_{i,j}(0) = x_{i,j}^L + rand(0,1)(x_{i,j}^L - x_{i,j}^M) \quad (11)$$

The i -th individual in a population indicates a potential answer to an optimization issue, and the j -th decision variable of that individual is denoted by $x_{i,j}(0)$. The i -th and j -th range from 1 to NP and 1 to D , respectively. Meanwhile, the function $rand(0,1)$ creates a random integer with a uniform distribution within the range $[0,1]$. This formula is utilized to initialize the position information of the population.

2.2.2. Population mutation strategy.

Mutation is a critical process in the DE algorithm that allows the algorithm to explore new regions in the search space [22][34]. This process creates new candidate solutions by calculating the differences between two randomly selected individuals from the current population [22]. Common mutation operations include the following:

DE/rand/1/bin:

$$V_i^{t+1} = x_{r1}^t + F * (x_{r2}^t - x_{r3}^t) \quad (12)$$

DE/rand/2/bin:

$$V_i^{t+1} = x_{r1}^t + F * (x_{r2}^t - x_{r3}^t) + F * (x_{r4}^t - x_{r5}^t) \quad (13)$$

DE/current-to-best/1/bin

$$V_i^{t+1} = x_i^t + F_1 * (x_{best}^t - x_i^t) + F_2 * (x_{r1}^t - x_{r2}^t) \quad (14)$$

In the formula, V_i^{t+1} is the experimental individual i -th in the generation $t+1$ population after mutation, $i \in [1, N]$, and the population size is denoted by n ; $x_{r1}^t, x_{r2}^t, x_{r3}^t, x_{r4}^t, x_{r5}^t$ are three individuals randomly selected in the T generation population, and r_1, r_2, r_3, r_4 and r_5 represent the identification numbers of different individuals in the same generation population; x_{best}^t denotes the best person in the population's g -th generation. F is the variation probability and the value is between 0 and 1.

2.2.3. Population crossover.

The crossover operation in Differential Evolution (DE) algorithm combines information from multiple candidate solutions [35] to create new solutions with higher optimization potential using a binomial crossover operator.

$$U_{ij}^{t+1} = \begin{cases} v_{ij}^{t+1}, & rand(j) \leq CR \text{ or } j = rand(i) \\ x_{ij}^{t+1}, & rand(j) > CR \text{ or } j \neq rand(i) \end{cases} \quad (15)$$

In the formula, U_{ij}^{t+1} represents an updated individual obtained by crossing the j gene of the test individual; $rand(j)$ is a random integer with a homogeneous distribution, with a number ranging from 0 to 1, and j -th represents the j -th gene; CR is the cross probability, and its value is between 0 and 1; $rand(i)$ is the generated random integer, i takes the value in $[1, D]$, and d represents the D -dimensional parameter (number of decision variables); x_{ij}^t represents the individual of t generation population without mutation operation; v_{ij}^{t+1} represents the individuals of t generation population after mutation operation. The value of CR in this paper is 0.9.

2.2.4. Select new species.

The individuals after crossover operation and other individuals in the population are selected using a greedy algorithm, and the $t + 1$ generation individuals are selected by comparing the fitness to obtain a new population.

$$x_{ij}^{t+1} = \begin{cases} u_i^{t+1}, & f(u_i^{t+1}) \leq f(x_i^t) \\ x_i^t, & f(u_i^{t+1}) > f(x_i^t) \end{cases} \quad (16)$$

Where $f(u_i^{t+1})$ represent that fitness of the test individual through crossover; $f(x_i^t)$ indicates the fitness of the target individual.

3. Enhanced crossover and mutation differential evolution algorithm (ECMDE).

3.1. Algorithm improvement.

Improving certain variables is necessary to enhance the optimization capability of the DE algorithm and overcome the problem of being susceptible to local optima. The key parameters for enhancing the DE algorithm's performance are the initialization population, mutation factor, and crossover probability.

3.1.1. Chaotic sequence initialization.

The efficiency of most current intelligent optimization algorithms is greatly influenced by population initialization. The uniformly distributed population can appropriately broaden the algorithm's search scope, improving the algorithm's convergence speed and solution accuracy. By using chaos mapping to initialize the population, individuals can be distributed as evenly as possible in the search space. This feature can be used to improve the algorithm's performance. The primary concept is to map variables into the value range of the chaotic variable space using the properties of chaos, and then to convert the result into the ideal variable space linearly [27]. There are currently many different chaotic maps in the optimization field, most notably the Tent, Circle, and Logistic maps. In this work, the Circle chaotic map is used to create the starting population [36, 44]. The definition of a circle map is as follows:

$$x_{i+1} = mod(x_i + 0.2 - \left(\frac{0.5}{2\pi}\right) \sin(2\pi x_i), 1) \quad (17)$$

When generating the original population, the circular mapping technique produces a more uniform spread of population locations than randomly dispersed populations. This expands the ECMDE algorithm's search area and broadens the population locations, thereby addressing the issue of local optima and improving the algorithm's optimization efficiency.

3.1.2. Improved DE/rand/2/bin mutation strategy.

In large part, mutation determines how well DE works. Using a fixed mutation operator F can lead to insufficient convergence performance in the later stages of the algorithm. This paper proposes a new mutation strategy that uses a dynamic adaptive factor to replace F to solve the problem of insufficient convergence performance in the later stages of the algorithm [37]. The following is the formula:

$$V_i^{t+1} = x_{r5}^t + \gamma * (x_{r1}^t - x_{r2}^t) + F * (x_{r3}^t - x_{r4}^t) \quad (18)$$

$$\gamma = \gamma_{min} + (\gamma_{max} - \gamma_{min}) * e^{-((\frac{t}{T})^2)^{(1/3)}} \quad (19)$$

Where T represents the maximum iteration count, and t represents the current iteration count. The variation factors have upper and lower bounds, denoted by max and min respectively, In this article, $\gamma_{max}=0.9$, $\gamma_{min}=0.2$; $r1 \neq r2 \neq r3 \neq r4 \neq r5$.

During the initial phase of evolution, the population explores a wider range of possibilities to discover the optimal solution, and a high value of F is preferable at this stage. As the evolution process advances, F should be gradually reduced to improve the population's ability to conduct accurate and focused local searches.

3.1.3. Chaotic sequence initialization.

The a value in the crossover operator determines the proportion of genetic information from the mutant or parents in the new individual. A large a value favors mutants and improves convergence speed, while a small a value favors parents and enhances global optimization. However, the standard DE algorithm uses a fixed a value, which neglects the trade-off between global and local search. Therefore, an adaptive monotone-decreasing crossover operator is introduced to address this issue. The formula is as follows:

$$CR = Cr * (1 - (\frac{t}{T})^2 * \sin(\frac{\pi}{2} * \frac{t}{T})) \quad (20)$$

Where t represents the present stage of the iteration process, T is the maximum iteration and $Cr = 0.9$.

When the algorithm begins execution, the operator is initialized with a larger value. This adjustment has the potential to enhance the algorithm's early convergence speed while simultaneously decreasing the operator's value. Furthermore, in the refined formula, the ratio of t/T is squared, resulting in an accelerated rate of decrement for the function operator during subsequent iterations. Consequently, this modification facilitates more rapid exploration of the solution space, thereby promoting global optimization and mitigating the risk of the algorithm becoming trapped in a local optimum.

Table 1 shows an enhanced crossover and mutation differential evolution algorithm (ECMDE) pseudo-code.

3.2. Experimental results on mathematic test functions.

Twenty eight benchmark functions in CEC2013 [38] are used to simulate the ECMDE in order to demonstrate the effectiveness of the revised algorithm ECMDE. Test functions 6 to 20 are Fundamental Multimodal Functions, and 1 to 5 are Unimodal Functions. The composition functions from 21 to 28. In addition, to reduce the influence of chance in the experiment, all algorithms were tested 20 times, aiming to enhance the objectivity of the experimental results. The compared algorithms and their parameters can be found in Table 1.

The results of the algorithm on the CEC2013 test functions are presented in the following tables and graphs. Each test function was run 20 times. The "MEAN" column in the tables represents the average value across the 20 runs, while the "BEST" column represents the best result obtained in the 20 runs. The highlighted data in the tables

TABLE 1. Pseudo code of ECMDE

Pseudo code of ECMDE

1.	Initialization: Initiate the population using circle mapping $P^G, G=1$ and evaluate the fitness of each individual in the initial population.
2.	While $t < T$ do
3.	For $i = 1:N_p$ do
4.	(1) Mutation operator
5.	Random selection: $r1 \neq r2 \neq r3 \neq r4 \neq r5$;
6.	$\gamma = \gamma_{min} + (\gamma_{max} - \gamma_{min}) * e^{-(\frac{t}{T})^2(1/3)}$;
7.	$V_i^{t+1} = x_{r5}^t + \gamma * (x_{r1}^t - x_{r2}^t) + F * (x_{r3}^t - x_{r4}^t)$
8.	(2) Crossover operator
9.	$j_{rand} = rndint[1, D]$
10.	For $j=1$ to D do
11.	$CR = Cr * (1 - (t/T)^2 * \sin(\pi/2 * t/T))$;
12.	If $rand[0,1] < CR$ or $j = j_{rand}$
13.	$U_i(g) = V_i(g)$
14.	Else
15.	$U_i(g) = X_i(g)$
16.	(3) Selection operator
17.	If $f(U_i(g)) < f(X_i(g))$ then
18.	$X_i(g) = U_i(g)$
19.	End if
20.	End for
21.	$G = G + 1$
22.	End While
23.	Output: The best object for the entire population.

TABLE 2. Parameters settings for the algorithms, e.g., DE, GWO, PSO, ABC, and SOA

Algorithms	Parameters initial settings
DE	NP=60 Cr=0.6 F=0.6, iteration=1000
GWO	NP=60 $A \in [-1, 1], C \in [0, 2], r1 \in rand[0, 1], r2 \in rand[0, 1], iteration = 1000$
PSO	NP=60 $c1=2 c2=2 w=0.8 Vmax=1 Vmin=1, iteration=1000$
ECMDE	NP=60 $\gamma_{min} = 0.4, \gamma_{max} = 0.9, Cr \in [0.9, 0], iteration = 1000$
SOA	NP=60, $fc=2, u=1, v=1$

indicates the winners in the comparisons. Each table concludes with a summary that includes variables for victory, failure, and draw. In terms of the graphs, we have chosen two unimodal functions, two multimodal functions, and two hybrid functions for comparison.

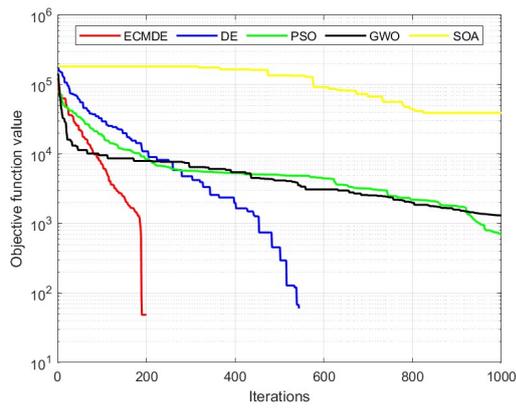
Table 3 present the results of ECMDE, comparing the performance of traditional optimization algorithms PSO and DE in 30-dimensional and 50-dimensional test functions. Similarly, Table 4 display the results of ECMDE, comparing the performance of new optimization algorithms GWO and SOA in the same test functions but with different dimensions. The final outcomes demonstrate that the ECMDE algorithm performs exceptionally well when compared to both traditional and new optimization algorithms across different dimensions. This indicates that our modifications to the original differential evolution algorithm have been highly effective.

Next, we chose six sample test functions, CEC1 through CEC5, CEC10 through CEC19, CEC21 and CEC22, to evaluate the convergence behavior of the ECMDE algorithm [39].

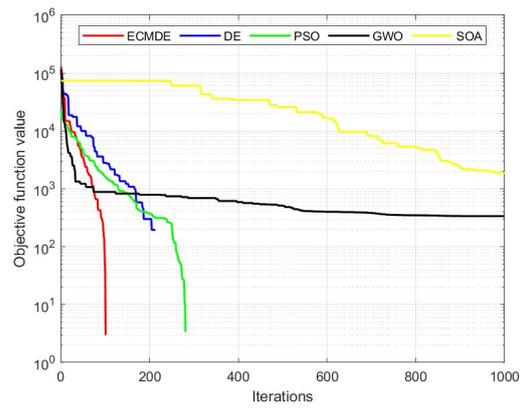
TABLE 3. PSO, DE, ECMDE performance on 30-dimension CEC2013 test function

30D	PSO		DE		ECMDE	
	BEST	MEAN	BEST	MEAN	BEST	MEAN
CEC1	-1.37E+03	2.12E+02	-1.40E+03	-1.40E+03	-1.40E+03	-1.39E+03
CEC2	1.04E+07	3.74E+07	1.46E+08	1.93E+08	1.86E+06	7.02E+06
CEC3	1.65E+10	3.01E+13	1.22E+09	2.09E+09	5.68E+07	2.35E+08
CEC4	1.12E+04	3.98E+04	7.70E+04	1.01E+05	5.87E+04	1.11E+05
CEC5	-8.51E+02	7.72E+02	-1.00E+03	-9.98E+02	-1.00E+03	-1.00E+03
CEC6	-8.48E+02	-6.56E+02	-8.62E+02	-8.56E+02	-8.84E+02	-8.58E+02
CEC7	-6.55E+02	-3.76E+02	-7.34E+02	-7.21E+02	-7.58E+02	-7.14E+02
CEC8	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02
CEC9	-5.71E+02	-5.65E+02	-5.63E+02	-5.60E+02	-5.82E+02	-5.73E+02
CEC10	-4.85E+02	-1.03E+02	-2.92E+02	-1.94E+02	-5.00E+02	-4.68E+02
CEC11	-2.10E+02	-1.02E+02	-2.59E+02	-2.33E+02	-3.61E+02	-3.39E+02
CEC12	-9.88E+01	2.87E+00	-8.43E+01	-6.34E+01	-2.66E+02	-2.00E+02
CEC13	4.31E+01	1.70E+02	-3.05E+00	3.58E+01	-8.79E+01	-1.82E+01
CEC14	2.89E+03	4.73E+03	4.46E+03	4.93E+03	2.07E+03	3.77E+03
CEC15	3.36E+03	5.12E+03	7.56E+03	7.84E+03	4.33E+03	7.15E+03
CEC16	2.01E+02	2.03E+02	2.02E+02	2.03E+02	2.02E+02	2.04E+02
CEC17	5.21E+02	6.73E+02	4.87E+02	5.06E+02	3.68E+02	4.35E+02
CEC18	6.58E+02	7.57E+02	6.37E+02	6.61E+02	5.03E+02	6.09E+02
CEC19	5.19E+02	1.01E+03	5.18E+02	5.19E+02	5.04E+02	5.24E+02
CEC20	6.12E+02	6.14E+02	6.13E+02	6.13E+02	6.12E+02	6.13E+02
CEC21	9.43E+02	1.13E+03	1.00E+03	1.03E+03	9.00E+02	9.82E+02
CEC22	5.00E+03	6.18E+03	6.14E+03	6.89E+03	2.90E+03	3.84E+03
CEC23	5.53E+03	7.02E+03	8.56E+03	8.99E+03	5.33E+03	8.08E+03
CEC24	1.30E+03	1.34E+03	1.30E+03	1.30E+03	1.25E+03	1.26E+03
CEC25	1.40E+03	1.44E+03	1.40E+03	1.41E+03	1.36E+03	1.38E+03
CEC26	1.58E+03	1.60E+03	1.41E+03	1.46E+03	1.40E+03	1.55E+03
CEC27	2.43E+03	2.67E+03	2.59E+03	2.64E+03	2.04E+03	2.23E+03
CEC28	2.44E+03	4.21E+03	1.71E+03	1.74E+03	1.70E+03	1.96E+03
win	3	3	1	6	20	16
lose	23	25	24	21	4	10
draw	2	1	3	2	4	2

The operation results are shown in Figure 1 to 3. These six pictures that the ECMDE algorithm has a faster rate of convergence than the other four algorithms. The CEC1 and CEC5 unimodal functions are straightforward, and the ECMDE can easily discover the ideal value. The ECMDE iterates much more quickly than previous algorithms for the somewhat complicated Basic Multimodal Functions CEC10. In CEC19, inflection points emerge in all six algorithms, showing that the algorithm is caught in the local optimal, but the ECMDE has the fastest convergence speed and maximum convergence accuracy after jumping out the local optimal. The convergence curve of the method is clearly better than that of the comparison algorithm in the most complex Composition Functions, indicating that the ECMDE algorithm has greater global optimization capacity than other comparison algorithms. Some algorithms slowdown in late evolution and fall into local optimization. Only the ECMDE algorithm has the highest convergence accuracy, demonstrating its capacity to solve complicated optimization problems.

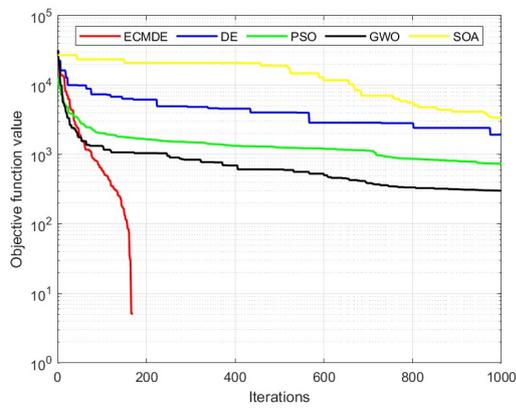


(A) CEC1

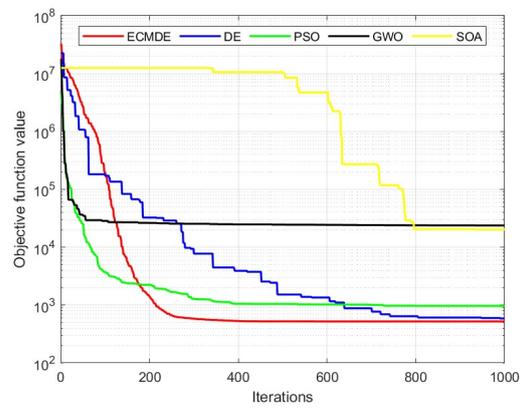


(B) CEC5

FIGURE 1. Unimodal Functions

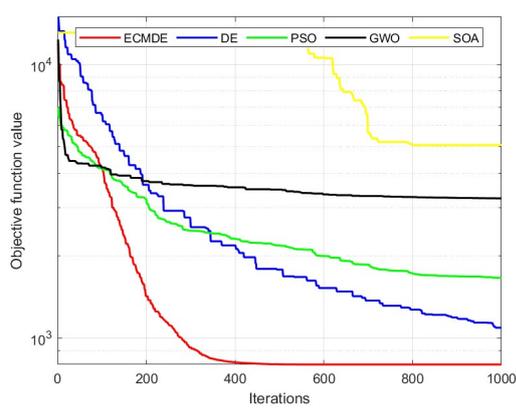


(A) CEC10

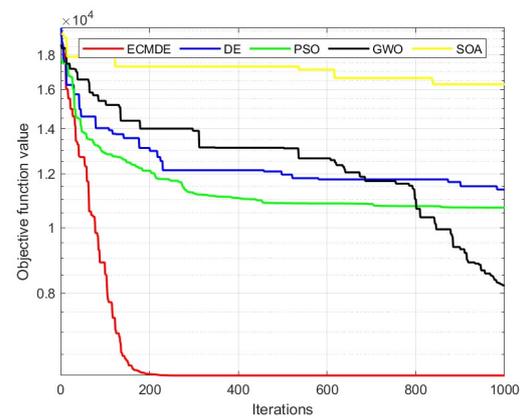


(B) CEC19

FIGURE 2. Multimodal Functions



(A) CEC21



(B) CEC22

FIGURE 3. Composition Functions

TABLE 4. GWO, SOA, ECMDE performance on 30-dimension cec2013 test function.

30D	PSO		DE		ECMDE	
	BEST	MEAN	BEST	MEAN	BEST	MEAN
CEC1	-1.25E+03	1.87E+02	7.07E+03	1.39E+04	-1.40E+03	-1.39E+03
CEC2	4.32E+06	2.46E+07	8.28E+07	1.14E+08	1.86E+06	7.02E+06
CEC3	1.57E+09	4.74E+09	3.70E+10	5.47E+10	5.68E+07	2.35E+08
CEC4	2.13E+04	3.86E+04	6.42E+04	9.62E+04	5.87E+04	1.11E+05
CEC5	-8.61E+02	-1.93E+02	5.20E+02	2.89E+03	-1.00E+03	-1.00E+03
CEC6	-8.10E+02	-7.73E+02	-5.39E+02	-1.05E+02	-8.84E+02	-8.58E+02
CEC7	-7.56E+02	-7.31E+02	-6.52E+02	-6.17E+02	-7.58E+02	-7.14E+02
CEC8	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02
CEC9	-5.87E+02	-5.80E+02	-5.71E+02	-5.67E+02	-5.82E+02	-5.73E+02
CEC10	-4.16E+02	-2.19E+02	6.66E+02	1.35E+03	-5.00E+02	-4.68E+02
CEC11	-3.49E+02	-3.05E+02	-4.48E-01	5.98E+01	-3.61E+02	-3.39E+02
CEC12	-2.25E+02	-1.59E+02	1.09E+02	1.67E+02	-2.66E+02	-2.00E+02
CEC13	-8.90E+01	1.95E+01	2.12E+02	2.72E+02	-8.79E+01	-1.82E+01
CEC14	1.80E+03	3.58E+03	6.35E+03	7.15E+03	2.07E+03	3.77E+03
CEC15	2.69E+03	5.20E+03	5.61E+03	6.77E+03	4.33E+03	7.15E+03
CEC16	2.02E+02	2.04E+02	2.02E+02	2.03E+02	2.02E+02	2.04E+02
CEC17	4.03E+02	4.77E+02	8.76E+02	9.33E+02	3.68E+02	4.35E+02
CEC18	6.34E+02	6.67E+02	9.77E+02	1.04E+03	5.03E+02	6.09E+02
CEC19	5.05E+02	6.00E+02	2.35E+03	7.14E+03	5.04E+02	5.24E+02
CEC20	6.12E+02	6.13E+02	6.14E+02	6.14E+02	6.12E+02	6.13E+02
CEC21	1.14E+03	1.67E+03	2.65E+03	2.79E+03	9.00E+02	9.82E+02
CEC22	2.29E+03	3.95E+03	7.48E+03	8.38E+03	2.90E+03	3.84E+03
CEC23	3.35E+03	5.68E+03	6.79E+03	8.05E+03	5.33E+03	8.08E+03
CEC24	1.24E+03	1.27E+03	1.28E+03	1.30E+03	1.25E+03	1.26E+03
CEC25	1.36E+03	1.39E+03	1.41E+03	1.43E+03	1.36E+03	1.38E+03
CEC26	1.40E+03	1.54E+03	1.40E+03	1.43E+03	1.40E+03	1.55E+03
CEC27	2.06E+03	2.14E+03	2.43E+03	2.52E+03	2.04E+03	2.13E+03
CEC28	2.10E+03	2.63E+03	4.12E+03	4.51E+03	1.70E+03	1.96E+03
win	8	6	0	0	15	17
lose	15	20	25	25	8	9
draw	5	2	3	3	5	2

4. Application of ECMDE in Microgrid Optimization.

4.1. Microgrid operating cost function.

The total economic expense associated with generating power from micro-grids consists of the expenses incurred for burning fuel and maintaining and operating micro-power sources, such as micro-fuel engines, batteries, and photovoltaic systems. Additionally, there are costs involved in coordinating with the larger power grid. The specific formula is as follows:

$$F(t) = \sum_{t=1}^T [C_{PV}(t) + C_{WT}(t) + C_{MT}(t) + C_{ES}(t) + C_{Grid}(t)] \quad (21)$$

In the given formula, $C_i(t)$ represents the expenses associated with a micro-power supply, including operating costs, maintenance costs, penalty costs, etc., The letter "F" in the formula represents the total expenses in running a microgrid.

The state of charge of a battery can have an impact on its service life during actual usage. Overcharging or undercharging a battery can reduce its overall lifespan due to its impact on its charging and discharging cycles. Therefore, set the battery charge status warning area as shown in the figure below. When the battery is in a non-safe state, a fine will be charged, the formula is as follows:

$$C_{ES}(t) = C_{ES'}(t) + C_{loss} \quad (22)$$

C_{loss} is the penalty cost when the battery is in an unsafe state, $C_{ES'}(t)$ is the cost of battery operation, and $C_{ES}(t)$ is the total cost.

4.2. ECMDE Process for Microgrid Optimization.

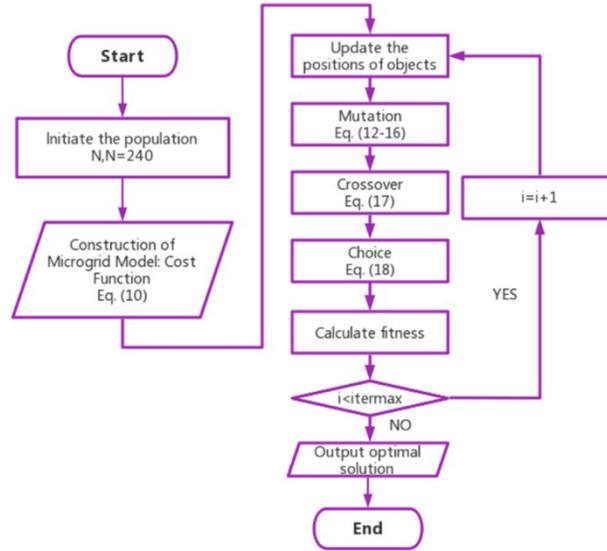


FIGURE 4. The ECMDE Flowchart for Microgrid Optimization

Figure 4 is the flowchart of ECMDE in microgrid optimization. The main optimization process steps for applying ECMDE to micro grid optimal scheduling problems are described as follows:

Step 1: Enter the microgrid model specifications and the daily load requirement Load (t), upper and lower limits of various power supplies, costs, and other parameters (shown in Table 7), time-of-use electricity price (shown in Table 6), and pollution control costs (shown in Table 8).

Step 2: Set the parameters of the ECMDE and compare the parameters of the algorithm to determine the algorithm's dimension population and particle number, and generate initial particles within the upper and lower limits of the microgrid power supply.

Step 3: Rank the generated particles by fitness to select the best particles

Step 4: Perform mutation crossover operations on particles, sort the generated individuals, compare the particles with the best sorted particles before, and save better particles to the next generation.

Step 5: Verify the end condition, then continue steps 2 through 4 until the maximum number of iterations is achieved, and finally, output the optimal position and optimal global value.

TABLE 5. Electricity meter

Periods of times	Interval hours	Price/(\$/KW·h)
Normal period	07:00–10:00	0.49
	15:00–18:00	
	21:00–23:00	
Peak period	10:00–15:00	0.83
	18:00–21:00	
Valley period	23:00–07:00	0.17

TABLE 6. Parameters and operating costs of various power supplies

Power sources	KW-power capacity		Maintenance Cost	Power generation price(\$/KW)
	Upper	Lower		
PV	300	0	0.01	0
WT	200	0	0.0296	0
MT	25	1	0.05	0.5
ES	150	-100	0.005	0.08
Grid	200	10	0.023	0.17-0.83

TABLE 7. Pollution Remediation Costs

The emissions		CO_2	SO_2	NO_2
Discharge coefficient/(g/kW)	MT	706.5×10^{-3}	2.8×10^{-6}	0.77×10^{-3}
	GRID	889×10^{-3}	1.9×10^{-5}	1.5×10^{-6}
Related operations costs/(\$/kg)		0.03	2.1	8.9

TABLE 8. Cost of two schemes

Scheme	Scheme 1	Scheme 2
Cost	1804.23	1838.24

4.3. Microgrid Data Analysis and Discussion.

The research employs a time-of-use pricing scheme for electricity, which divides a 24-hour day into three separate periods: peak, flat, and valley. The cost of purchasing and selling electricity during each period is itemized in a table. Given the precondition of ensuring stable operation of the microgrid, we can procure electricity from the microgrid during periods of low pricing while meeting the load demand and storing any excess electricity in batteries. When the electricity price is high or the load is heavy, the batteries can output electricity to reduce operating costs.

The data of a typical day in a certain place is used as the input quantity of the calculation example. The output curves and load demand curves of renewable energy photovoltaics and wind turbines are shown in Figure 5:

During 3:00-11:00 and 17:00-18:00, the load's demand for electric energy gradually increases, while in other periods the demand continues to decrease. Two load peaks in a day occur at 11:00-12:00 and 17:00-18:00, the peak power demand occurs at 11:00, which is 592kW, and the valley value of power demand occurs at 3:00 at night. It is 210kW, and this figure can reflect some details of electricity consumption of residents' production and life as load [46]. The above illustration shows that distributed energy generation has volatility, randomness and intermittent nature, The volatility of distributed energy

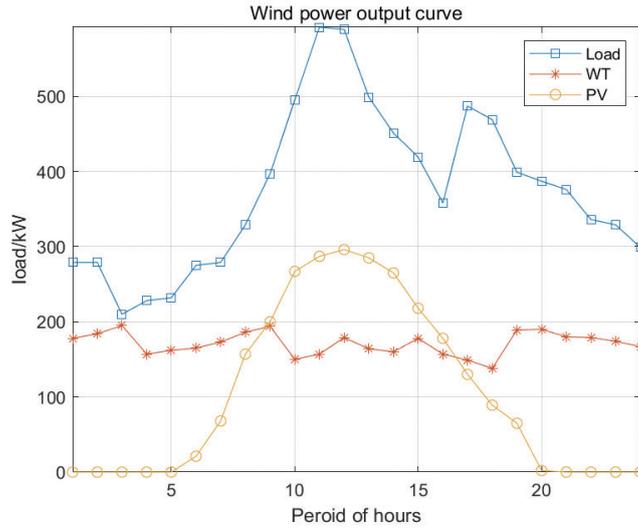


FIGURE 5. Microgrid wind power, photovoltaic, load output

generation can have an impact on the stable operation of the power grid when it is integrated. To ensure the safe and reliable operation of the microgrid and improve its economic performance, scheduling optimization of the microgrid is necessary.

4.4. Results Analysis and Discussion. The parameters of the ECMDE and other algorithms are presented in detail in Table 1, where the algorithm's maximum number of iterations is 1000 and the population size is 240. Additionally, Table 6 provides information on the upper and lower limits of output, maintenance costs, and power generation costs for different micro-power sources. Table 7 lists the pollution costs related to emissions. Figure 5 shows the 24-hour load demand and photovoltaic wind power output curve of a typical micro-grid. The battery can store a maximum charge of 500kWh.

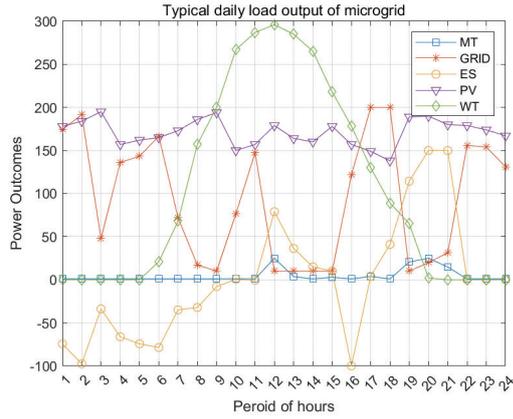
To demonstrate the efficacy of the ECMDE algorithm, two microgrid scheduling schemes are employed. The first scheme takes into account the depth of energy storage discharge, while the second scheme does not. The microgrid model used in this article is solved together with the original DE algorithm, SOA algorithm [38], and PSO algorithm [41], and the results are compared.

Scheduling scheme 1: Scheduling without considering the depth of discharge of the energy storage device.

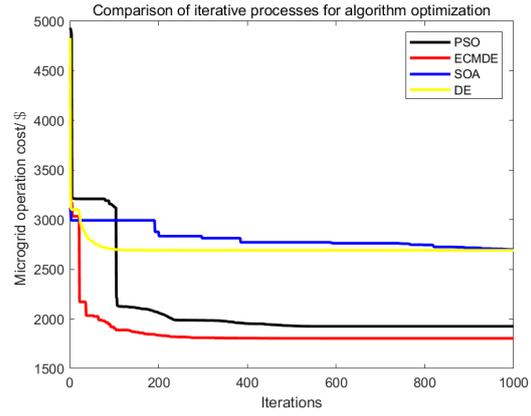
Figure 6(A) illustrates the output of each micro power supply when scheduling without considering the depth of battery discharge. The optimization situation is shown in Figure 6(B).

Scheduling scheme 2: scheduling considering the depth of discharge of energy storage devices. When scheduling considering the depth of battery discharge, the output of each micro power supply is shown in Figure 7(A). The optimization situation is shown in Figure 7(B).

As shown in Figure 6, during the period from 1:00 to 7:00, due to low load and continuous wind turbine output, and the current electricity prices are low, the micro grid purchases a large amount of electricity. At 8:00 to 10:00, photovoltaic and wind turbines simultaneously output, and the battery absorbs excess electricity during this process, making the charge in the battery reach a peak; At 11:00 to 13:00, the peak power consumption is at this time, and the battery fan, photovoltaic micro fuel engine, simultaneously outputs to meet the power load demand. During the period from 14:00 to 17:00, due to a

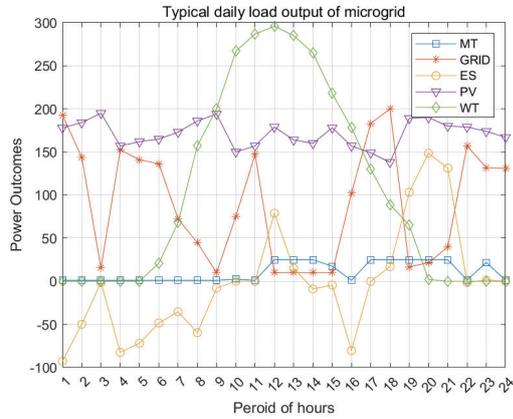


(A) Micro power supply output chart

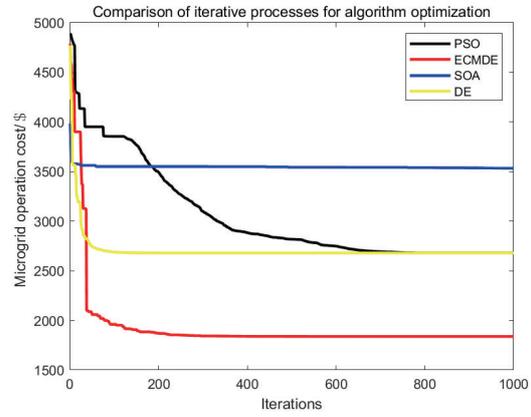


(B) Algorithm optimization comparison

FIGURE 6. Results of scheduling scheme 1



(A) Micro power supply output chart



(B) Algorithm optimization comparison

FIGURE 7. Results of scheduling scheme 2

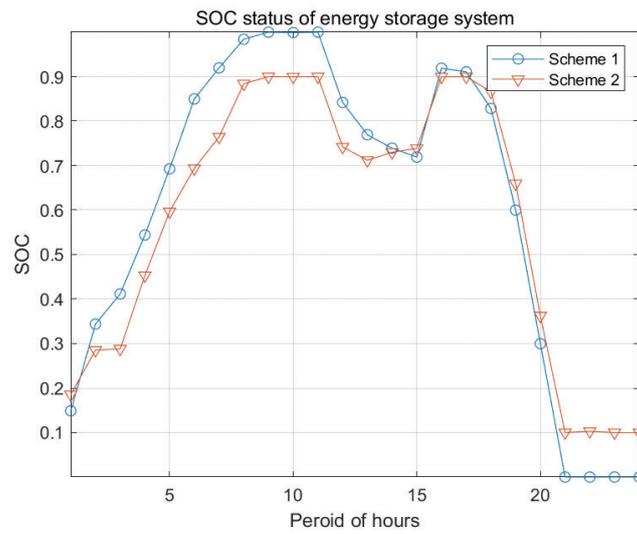


FIGURE 8. Battery state of charge(SOC)

decrease in load, the battery is charged; During the period from 18:00 to 21:00, the second peak period of electricity consumption comes. Due to the inability of photovoltaic power to provide sufficient electricity at this time, and the high price of time-of-use electricity prices, it is necessary to reduce the purchase of electricity from the large power grid and discharge a large amount of batteries to meet the load demand, which can reduce the cost of operation. Micro fuel engines and wind power also continue to work at this time; The battery discharge from 22:00 to 24:00 allows the charge to be controlled between 5% and 10% of the total electricity at the last moment of the day, preventing excessive discharge, extending the battery usage time, and completing the day's electricity scheduling. During this scheduling process, the battery plays an important role in storing electricity when the load is low or the electricity price is cheap.

Figure 8 shows the battery SOC diagram for two scheduling schemes. Generally speaking, if the battery operates for an extended period, the battery's lifespan decreases with an increase in discharge depth. In contrast, shallow discharge depth leads to extended battery life. Shallow cycling the battery offers two major benefits: firstly, the battery has a longer cycle life, and secondly, it can retain more spare ampere-hour capacity, thereby improving the power supply assurance rate of distributed generation systems. Therefore, Scheme 2 considers the impact of battery discharge depth on battery life, and the energy storage system and the large power grid have the same priority. Although the optimization results of the two scheduling schemes have approximately the same total cost (Table 8), Scheme 2 considers the impact of discharge depth on battery life and maintenance. Compared to Scheme 1, Scheme 2 can maintain the battery below a lower cycle discharge, reduce battery maintenance costs, and improve battery life. It can effectively cooperate with the operation and scheduling optimization of the microgrid trading market.

By comparing the optimization results depicted in Figure 5 and Figure 7, it is evident that the ECMDE algorithm outperforms DE, SOA, PSO, and other algorithms in terms of convergence speed and accuracy under both scheduling conditions. This confirms the effectiveness of the improved algorithm.

5. Conclusion. This article proposes an enhanced version of the Differential Evolution Algorithm (ECMDE) for optimal scheduling in micro power grids. The original DE algorithm suffers from issues such as slow convergence and local optimization. To address these issues, the proposed algorithm incorporates chaos initialization, dynamic adaptive factors for improving mutation and linear crossover factors to increase the diversity of the population and ability to jump out of local optimization, thereby improving the optimization performance of DE. Comparative studies were conducted with DE, GWO, PSO, and SOA algorithms using the CEC 2013 test set, and the results indicate that ECMDE performs well. The proposed algorithm offers excellent performance and stability for microgrid systems in terms of optimal scheduling. The ECMDE will be applied further for applications of hydro reservoirs and wind-solar power planning optimization in future work.

Acknowledgment. This work is partially supported by Marine Research Special Foundation of Fujian University of Technology (GY-Z23086).

REFERENCES

- [1] M.-L. Tuballa and M.-L. Abundo, "A review of the development of Smart Grid technologies," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 710-725, 2016.
- [2] T.-T. Nguyen, T.-G. Ngo, T.-K. Dao, and T.-T.-T. Nguyen, "Microgrid Operations Planning Based on Improving the Flying Sparrow Search Algorithm," *Symmetry*, vol. 14, no. 1. 2022, doi: 10.3390/sym14010168.

- [3] T.-K. Dao, T.-T. Nguyen, V.-T. Nguyen, and T.-D. Nguyen, "A Hybridized Flower Pollination Algorithm and Its Application on Microgrid Operations Planning," *Applied Sciences*, vol. 12, no. 13, 6487, 2022.
- [4] S. Sen and V. Kumar, "Microgrid control: A comprehensive survey," *Annual Reviews in control*. vol. 45, pp. 118-151, 2018.
- [5] A. K. Raji and D. N. Luta, "Modeling and optimization of a community microgrid components," *Annual Reviews in control*. vol. 45, pp. 118-151, 2018.
- [6] H. Farhangi and G. Joós, *Microgrid planning and design: a concise guide*, John Wiley Sons. 2019.
- [7] S.-A. Amamra, H. Ahmed, and R. A. El-Sehiemy, "Firefly algorithm optimized robust protection scheme for DC microgrid," *Electric Power Components and Systems*. vol. 45, no. 10, pp. 1141-1151, 2017.
- [8] T.-T. Nguyen, T.-K. Dao, T.-T.-T. Nguyen, and T.-D. Nguyen, "An Optimal Microgrid Operations Planning Using Improved Archimedes Optimization Algorithm," *IEEE Access*. vol. 10, pp. 67940-67957, 2022, doi: 10.1109/ACCESS.2022.3185737.
- [9] T.-Y. Wu, A. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm," *Mathematics*, vol. 11, no. 10, 2339, 2023.
- [10] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, 1977, 2023.
- [11] T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *Journal of Ambient Intelligence and Humanized Computing*, 2021, doi: 10.1007/s12652-020-02740-2.
- [12] T. T. Nguyen, T. G. Ngo, T. K. Dao, and T. T. T. Nguyen, "Microgrid Operations Planning Based on Improving the Flying Sparrow Search Algorithm," *Symmetry*. vol. 14, no. 1, 168, Jan. 2022, doi: 10.3390/sym14010168.
- [13] C.-H. Wang, T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "An optimization approach for potential power generator outputs based on parallelized firefly algorithm," *Smart Innovation, Systems and Technologies*, vol. 64. 2017, Springer, Cham, doi.org/10.1007/978-3-319-50212-0_36
- [14] T.-T. Nguyen, T.-D. Nguyen, and V.-T. Nguyen, "An Optimizing Pulse Coupled Neural Network based on Golden Eagle Optimizer for Automatic Image Segmentation," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 13, no. 3, pp. 155-164, 2022.
- [15] L. Kang, R. Chen, N. Xiong, Y. Chen, Y. Hu, and C. Chen, "Selecting Hyper-Parameters of Gaussian Process Regression Based on Non-Inertial Particle Swarm Optimization in Internet of Things," *IEEE Access*, vol. 7, pp. 59504-59513, 2019, doi: 10.1109/ACCESS.2019.2913757.
- [16] T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, J.-H. Liu, and S.-W. Weng, "An Improved Slime Mold Algorithm and Its Application for Optimal Operation of Cascade Hydropower Stations," *IEEE Access*, vol. 8, p. 1, 2020, doi: 10.1109/ACCESS.2020.3045975.
- [17] C.-M. Chen, S. Lv, J. Ning, and J. M.-T. Wu, "A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem," *Symmetry*, vol. 15, no. 1, 124, 2023.
- [18] J.-S. Pan, T.-K. Dao, T.-T. Nguyen, S.-C. Chu, and T.-S. Pan, "Dynamic Diversity Population based Flower Pollination Algorithm for Multimodal Optimization," *Nguyen, N.T., Trawiński, B., Fujita, H., Hong, TP. (eds) Intelligent Information and Database Systems, ACIIDS 2016. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, vol. 9621, 2016, doi.org/10.1007/978-3-662-49381-6_42.
- [19] T. K. Dao, T. S. Pan, T. T. Nguyen, and S. C. Chu, "Evolved Bat Algorithm for solving the Economic Load Dispatch problem," *Advances in Intelligent Systems and Computing*, vol. 329, pp. 109-119, doi: 10.1007/978-3-319-12286-112, 2015.
- [20] T. Dao, J. Yu, T. Nguyen, and T. Ngo, "A Hybrid Improved MVO and FNN for Identifying Collected Data Failure in Cluster Heads in WSN," *IEEE Access*, vol. 8, pp. 124311-124322, 2020, doi: 10.1109/ACCESS.2020.3005247.
- [21] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69-73, doi: 10.1109/ICEC.1998.699146.
- [22] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997, doi: 10.1023/A:1008202821328.
- [23] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems," *Foundations of Fuzzy Logic and Soft Computing*, vol. 4529, 2007, pp. 789-798.

- [24] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowledge-Based Systems*, vol. 165, pp. 169-196, 2019.
- [25] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [26] P. Hu, J.-S. Pan, and S.-C. Chu, "Improved Binary Grey Wolf Optimizer and Its application for feature selection," *Knowledge-Based Systems*, vol. 195, p. 105746, 2020, doi: <https://doi.org/10.1016/j.knosys.2020.105746>.
- [27] Z. Meng, J. S. Pan, and K. K. Tseng, "PaDE: An enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80-99, Mar. 2019, doi: 10.1016/j.knosys.2019.01.006.
- [28] Z. Meng, J.-S. Pan, and H. Xu, "QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization," *Knowledge-Based Systems*, vol. 109, pp. 104-121, Oct. 2016, doi: 10.1016/J.KNOSYS.2016.06.029.
- [29] J.-S. Pan, Z. Meng, S.-C. Chu, and J. F. Roddick, "QUATRE Algorithm with Sort Strategy for Global Optimization in Comparison with DE and PSO," *Variants BT - Proceedings of the Fourth Euro-China Conference on Intelligent Data Analysis and Applications*, Springer Berlin Heidelberg, 2018, pp. 314-323.
- [30] M. B. Shadmand and R. S. Balog, "Multi-objective optimization and design of photovoltaic-wind hybrid system for community smart DC microgrid," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2635-2643, 2014.
- [31] S. A. Pourmousavi, M. H. Nehrir, C. M. Colson, and C. Wang, "Real-time energy management of a stand-alone hybrid wind-microturbine energy system using particle swarm optimization," *IEEE Transactions on Sustainable Energy*, vol. 1, no. 3, pp. 193-201, 2010, doi: 10.1109/TSTE.2010.2061881.
- [32] T.-T. Nguyen, M. J. Wang, J. S. Pan, T.-K. Dao, and T. G. Ngo, "A Load Economic Dispatch Based on Ion Motion Optimization Algorithm," *Smart Innovation, Systems and Technologies*, vol. 157, pp. 115-125, doi: 10.1007/978-981-13-9710-3-12, 2020.
- [33] T.-T. Nguyen, M.-J. Wang, J.-S. Pan, T. Dao, and T.-G. Ngo, "A Load Economic Dispatch Based on Ion Motion Optimization Algorithm", *BT - Advances in Intelligent Information Hiding and Multimedia Signal Processing*, 2020, pp. 115-125.
- [34] J.-S. Pan, Z. Meng, H. Xu, and X. Li, "QUasi-Affine TRansformation Evolution (QUATRE) algorithm: A new simple and accurate structure for global optimization," *Knowledge-Based Systems*, vol. 155, pp. 35-53, 2018, doi: <https://doi.org/10.1016/j.knosys.2018.04.034>.
- [35] Z. Meng and J.-S. Pan, "QUasi-Affine TRansformation Evolution with External ARchive (QUATRE-EAR): An enhanced structure for Differential Evolution," *Knowledge-Based Systems*, vol. 155, pp. 35-53, 2018, doi: <https://doi.org/10.1016/j.knosys.2018.04.034>.
- [36] P. Snaselova and F. Zboril, "Genetic Algorithm using Theory of Chaos," *Procedia Computer Science*, vol. 51, pp. 316-325, 2015, doi: <https://doi.org/10.1016/j.procs.2015.05.248>.
- [37] N. Liu, J. S. Pan, and T. T. Nguyen, "A bi-population QUasi-Affine TRansformation Evolution algorithm for global optimization and its application to dynamic deployment in wireless sensor networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2019, no. 1, Dec. 2019, doi: 10.1186/s13638-019-1481-6.
- [38] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin, "Benchmark Functions for the CEC'2013 Special Session and Competition on Large-Scale Global Optimization," *IEEE Conference Engineering Competition*, 2013.
- [39] J. J. Liang, B. Y. Qu, D. W. Gong, and C. T. Yue, "Problem Definitions and Evaluation Criteria for the CEC 2019 Special Session on Multimodal Multiobjective Optimization," *IEEE Conference Engineering Competition*, Laboratory, Zhengzhou University, 2019.
- [40] P. C. Sahu, R. C. Prusty, and S. Panda, "Improved-GWO designed FO based type-II fuzzy controller for frequency awareness of an AC microgrid under plug in electric vehicle," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1879-1896, 2021.
- [41] H. Bevrani, F. Habibi, P. Babahajyani, M. Watanabe, and Y. Mitani, "Intelligent frequency control in an AC microgrid: Online PSO-based fuzzy tuning approach," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1935-1944, 2012.
- [42] N. K. Paliwal, A. K. Singh, and N. K. Singh, "A day-ahead optimal energy scheduling in a remote microgrid alongwith battery storage system via global best guided ABC algorithm," *Journal of Energy Storage*, vol. 25, 100877, 2019.

- [43] P. Arumugam and V. Kuppan, "A GBDT-SOA approach for the system modelling of optimal energy management in grid-connected micro-grid system," *International Journal of Energy Research*, vol. 45, no. 5, pp. 6765-6783, 2021.
- [44] T. T. Nguyen, J. S. Pan, and T. K. Dao, "An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network," *IEEE Access*, vol. 7, pp. 75985-75998, 2019, doi: 10.1109/ACCESS.2019.2921721.
- [45] T. T. Nguyen, J. S. Pan, and T. K. Dao, "An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network," *IEEE Access*, vol. 7, pp. 75985-75998, 2019, doi: 10.1109/ACCESS.2019.2921721.
- [46] T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, T.-G. Ngo, and J. Yu, "A Scheme of Color Image Multithreshold Segmentation Based on Improved Moth-Flame Algorithm," *IEEE Access*, vol. 8, pp. 174142-174159, 2020, doi: 10.1109/ACCESS.2020.3025833.