



Multi-objective coyote optimization algorithm based on hybrid elite framework and Meta-Lamarckian learning strategy for optimal power flow problem

Jun-Hua Zhu¹ · Jie-Sheng Wang¹ · Xing-Yue Zhang¹ · Yu-Cai Wang¹ · Hao-Ming Song¹ · Yue Zheng¹ · Xun Liu¹

Published online: 15 April 2024
© The Author(s) 2024

Abstract

A multi-objective coyote optimization algorithm based on hybrid elite framework and Meta-Lamarckian learning strategy (MOCOA-ML) was proposed to solve the optimal power flow (OPF) problem. MOCOA-ML adds external archives with grid mechanism on the basis of elite non-dominated sorting. It can guarantee the diversity of the population while obtaining the Pareto solution set. When selecting elite coyotes, there is a greater probability to select the elite in sparse areas, which is conducive to the development of sparse areas. In addition, combined with Meta-Lamarckian learning strategy, based on four crossover operators (horizontal crossover operator, longitudinal crossover operator, elite crossover operator and direct crossover operator), the local search method is adaptively selected for optimization, and its convergence performance is improved. First, the simulation is carried out in 20 test functions, and compared with MODA, MOPSO, MOJAYA, NSGA-II, MOEA/D, MOAOS and MOTEQ. The experimental results showed that MOCOA-ML achieved the best inverted generational distance value and the best hypervolume value in 11 and 13 test functions, respectively. Then, MOCOA-ML is used to solve the optimal power flow problem. Taking the fuel cost, power loss and total emissions as objective functions, the tests of two-objective and three-objective benchmark problems are carried out on IEEE 30-bus system and IEEE 57-bus system. The results are compared with MOPSO, MOGWO and MSSA algorithms. The experimental results of OPF demonstrate that MOCOA-ML can find competitive solutions and ranks first in six cases. It also shows that the proposed method has obtained a satisfactory uniform Pareto front.

Keywords Optimal power flow · Coyote optimization algorithm · Multi-objective optimization · Non-dominated ranking · Grid mechanism · Meta-Lamarckian learning strategy

List of symbols

f_{cost} Objective function of fuel cost

✉ Jie-Sheng Wang
wang_jiesheng@126.com

¹ School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, People's Republic of China

f_{Ploss}	Objective function of active power loss
$f_{Emission}$	Objective function of emission
a_i, b_i, c_i	Fuel cost coefficient of the i -th generator
P_{G_i}	Active power emitted by the i -th generator
Q_{G_i}	Reactive power emitted by the i -th generator
N_G	Number of generators
G_{ij}	Conductance between the two nodes
B_{ij}	Inductance between the two nodes
V	Node voltage
δ	Phase angle corresponding to the node voltage
Nl	Number of transmission lines
$\alpha_i, \beta_i, \gamma_i, \xi_i, \lambda_i$	Emission coefficients of the i -th generator
T	Setting of the transformer tap position
Q	Reactive capacity of shunt capacitor
NT	Number of transformers
NC	Number of reactive capacitors
S_l	Power transmitted on the line
P_{D_i}	Active power demand of load
Q_{D_i}	Reactive power demand of load
$soc_c^{p,t}$	Social condition of the c -th coyote of the p -th pack at the t -th generation.
lb, ub	Lower and upper bounds
$alpha^{p,t}$	Social condition of elite coyote of the p -th pack at the t -th generation
$cul^{p,t}$	Median of the social conditions of the p -th pack at the t -th generation
$pup^{p,t}$	Social conditions of offspring of the p -th pack at the t -th generation
P_s	Scattering probability
P_a	Association probability

1 Introduction

The optimal power flow (OPF) problem is a large-scale, highly nonlinear, non-convex optimization problem (Singh et al. 2021). In 1962, Carpentier studied economic scheduling and added more constraints (Carpentier 1962). This expanded the economic dispatch problem, which is served as the foundation for the development of the OPF problem. The basic task of the power system is to operate safely and reliably to meet the power supply demand of the load side (Abdelaziz et al. 2016). The OPF flow is an effective tool, which can help experts make decisions on the planning and dispatching of power systems. Its core process is to adjust the control variables, such as active power generated by thermal power units, and obtain the power transmitted on each branch and the voltage of each node through power flow calculation. Through multiple iterations, the decision variables of the system are modified to obtain satisfactory operation status (Khunkitti et al. 2021).

The OPF problem has many constraints and multiple local optimal solutions (Davoodi et al. 2018). It means that there are a lot of infeasible solutions in the system. And it is easy to fall into the local optimal and stagnate in the process of solving. It is difficult for traditional optimization methods to find satisfactory results, such as nonlinear programming (Lavaei and Low 2011), Newton method (Santos and Costa 1995) and gradient method (Dommel and Tinney 1968). Intelligent algorithms with global search capabilities have

received widespread attention. They are widely used in optimization problems in science and engineering, and can obtain competitive solutions. Many intelligent algorithms are employed in order to tackle the OPF problems and economic dispatch problem, including Particle Swarm Optimization (PSO) (Gomez-Gonzalez et al. 2012), Cuckoo Search (CS) (Ponnusamy and Rengarajan 2014), Ant Lion Optimizer (ALO) (Ali et al. 2016), Differential Evolution (DE) (Sayah and Zehar 2008) and Mine Blast Algorithm (MBA) (Ali and Abd Elazim 2018). These algorithms employ continuous iterations until a predefined termination condition is met. In each iteration, each individual updates according to certain rules or formulas, and constantly moves to the global optimal position. Many algorithms are faced with the problems of unbalanced exploration and exploitation and difficulty in jumping out of the local optimum. Some scholars have proposed different improvement strategies for the OPF problem.

Salma et al. proposed an improved salp swarm algorithm (ISSA), which incorporates random mutation and adaptively adjusts the exploration and exploitation process. The study considers the multiple fuel costs, valve point effects and prohibited operating zones of generators in the OPF system. ISSA has been able to find competitive solutions in multiple case studies (Abd El-sattar et al. 2021). Awad et al. proposed a new differential evolution algorithm, called DEa-AR, to solve the stochastic optimal active-reactive power dispatch (OARPD) problems involving renewable energy. DEa-AR uses arithmetic compound crossover strategy and adjusts the scaling factor based on Laplacian distribution. It also added an archive to place the inferior solution for later use. The simulation results show that the proposed algorithm can effectively solve the OARPD problem containing renewable energy and provide a high-quality solution (Awad et al. 2019). Farhat et al. proposed an enhanced slime mold algorithm (ESMA) based on neighborhood dimension learning search strategy so as to enhance its exploitation capability. Its test system incorporates wind and photovoltaic generators, and its objective function incorporates a carbon tax in order to reduce emissions. The testing results show that ESMA obtains the optimal solution and show better convergence performance (Farhat et al. 2022). Bentuati proposed an Enhanced Moth Swarm Algorithm (EMSA), which combines MSA with a reverse learning strategy to maintain the diversity of the moth population. It was tested in 12 cases of three OPF testing systems, and the results showed that EMSA had better performance (Bentouati et al. 2021).

Real problems usually have multiple objective functions. If these indexes do not conflict with each other, an optimal solution can be found by using optimization techniques. However, it is more common for objective functions to conflict with each other, and the improvement of one objective function will inevitably lead to the reduction of another objective function. This problem is known as a multi-objective optimization problem (MOP), and its optimal solutions form a set called the Pareto solution set (Rizk-Allah et al. 2020). The OPF problem consists of multiple objective functions, such as thermal power unit fuel costs, active power loss and emissions, which are inherently conflicting with one another (Fonseca and Fleming 1993). Therefore, the OPF problem is regarded as a multi-objective optimization problem to balance these conflicting objective functions. In many literatures above, the OPF problem is treated as a single objective optimization problem, and its objective functions are optimized separately. However, this approach is no longer suitable at present. American ecosystem conservation organizations strongly urge power plants not only to pursue the lowest power generation cost, but also to consider the pollution index (Taher et al. 2019). So the trend in recent years is to develop a multi-objective method to solve the OPF problems. Intelligent algorithms combining multi-objective thought have achieved exciting results on this problem,

including NSGA-II (Jeyadevi et al. 2011), MOPSO (Hazra and Sinha 2011), MOEA/D (Medina et al. 2014), MOGJO (Snášel et al. 2023), etc. According to the law that there is no free lunch in the world, no perfect algorithm can have excellent performance in any problem, so the multi-objective optimization algorithm for OPF needs further research.

Shabanpour et al. proposed a modified teaching–learning-based optimization (MTLBO) based on an adaptive wavelet mutation strategy, which attached an external archive and used fuzzy clustering techniques to maintain the diversity of the external archive. It solves the multi-objective OPF problem including power generation cost and emissions, and obtains a set of Pareto solutions (Shabanpour-Haghighi et al. 2014). El-Sattar et al. used a Jaya optimization algorithm to solve the OPF problem, and solved the single-objective and multi-objective cases respectively. In the multi-objective framework, the Jaya algorithm is combined with the Pareto concept to obtain the non-dominant solution, and then the fuzzy set theory is used to obtain the optimal compromise solution. However, the solution set obtained by this method in solving multi-objective OPF problem is uneven (El-Sattar et al. 2019). Zhang proposed an improved decomposition method based on multi-objective evolutionary algorithm (MOEA/D) to deal with the competition of each index in the optimal power flow. An improved Chebyshev decomposition method is introduced to decompose each index in order to obtain uniformly distributed Pareto frontiers on each target. Simulation results show that it can find well-distributed Pareto solution sets (Zhang et al. 2016). Khan et al. proposed a multi-objective hybrid firefly and particle swarm optimization algorithm (MOHFPSO) by using a multi-objective structure based on non-dominated sorting and crowded distance methods. And MOHFPSO applied the ideal distance minimization method to select the optimal compromise solution from the Pareto optimal set. Although the Pareto solution set obtained is improved compared with the standard algorithm, its coverage rate decreases (Khan et al. 2020). Chen et al. proposed a Novel Hybrid Bat Algorithm (NHBA) to modify the local search formula and add a mutation mechanism by using a monotone random fill model (MRFME) based on extreme value. In order to obtain more feasible solutions, a non-dominated sorting method combining the Pareto fuzzy dominance (CPFD) of constraints is proposed. The results of OPF show that this method can deal with constraints better (Chen et al. 2019). Zhang et al. improved the NSGA-III algorithm named I-NSGA-III and applied it to the multi-objective OPF problem. An adaptive elimination strategy was proposed to reduce the use of selection strategies, and boundary point preservation strategy was integrated to maintain population diversity. Experimental results of OPF show that the proposed algorithm had better performance on three objectives, but not on two objectives (Zhang et al. 2019).

Multi-objective optimization algorithms often employ two strategies: population elitism and archive elitism. Population elitism algorithms (such as MOJAYA, MOHFPSO, NHBA) typically have a fixed population size. Excellent individuals may not be preserved and can be discarded during the evolutionary process. For algorithms with an archive (such as MTLBO, MOEA/D), the archive is usually used to store non-dominated solutions. However, the evolutionary process of the population is non-greedy, which does not guarantee the convergence and stability of the algorithm's search for optimal solutions. We believe that combining these two aspects can help maintain the stability of the algorithm's search for optimal solutions and find better solutions. Moreover, due to the different nature and characteristics of various problems, the same operator may perform well or poorly on different problems. For example, DE algorithm has developed many operators to adapt to different types of optimization problems. In the absence of prior knowledge, we are

committed to developing an adaptive, parameter-free local optimizer that allows the algorithm to spontaneously select the appropriate operator for position updating.

In this paper, the coyote optimization algorithm (COA) is selected for research. COA is a new optimization algorithm proposed by Pierezan and Coelho in 2018 (Pierezan and Coelho 2018). COA combines the principles of evolution and swarm intelligence and has a unique algorithm setup, which includes swarm search of sub-populations and considers the birth and death process of coyotes. The algorithm has demonstrated excellent performance and has been successfully applied in numerous fields. Souza proposed a binary version of COA, which utilizes a hyperbolic transfer function to select the best feature subset for classification and employs the naive Bayes classifier to verify the performance of COA. The results show that COA can find subsets with fewer features and achieve better classification accuracy (Souza et al. 2020). Li added a differential evolution strategy to COA and combined it with the fuzzy Kapoor entropy and fuzzy median aggregation method to utilize it in the realm of threshold image segmentation and exhibit improved image segmentation quality (Li et al. 2021). Ali applied COA to solve the Unit Commitment (UC) problem in power systems, which aims to satisfy constraints while achieving an economic minimum cost over time. During the simulation experiments, COA was employed to determine the optimal generation schedule. The obtained results demonstrated that COA outperformed the existing literature in terms of both total cost reduction and shorter CPU running time (Ali et al. 2023).

Existing multi-objective algorithms for the multi-objective OPF problem face challenges in balancing convergence and diversity simultaneously. It is necessary to provide sufficient pressure during the offspring selection process and enhance the diversity and convergence of multi-objective optimization algorithms. This will help in discovering a higher quality solution set in the MOOPF problem. In this paper, a multi-objective COA based on hybrid elite mechanism and Meta-Lamarckian learning strategy (MOCOAL-ML) is proposed for solving multi-objective OPF problems. The main contributions are as follows.

- (1) The coyote optimization algorithm was combined with non-dominant ranking. Non-dominant ranking was used to judge the dominant relationship among individuals, and the individuals equal to the population number were selected from all the individuals to enter the next iteration.
- (2) An external archive is added to retain the excellent individuals, which is similar to the archive in MOPSO and adopts the mechanism of grid. The role of the archive is to make the stored solution set more diverse and to have a greater probability of selecting the elite in the sparse area when selecting the elite coyote in COA. It is conducive to the development of the sparse area.
- (3) A local development optimizer based on the Meta-Lamarckian learning strategy is proposed to optimize the population solution. The local optimizer integrates four kinds of crossover operators, and adaptively adjusts the probability of each operator in the optimization process to achieve more efficient search.

The remaining sections of the paper are organized as follows. In Sect. 2, the mathematical model of the OPF problem is presented. In Sect. 3, the proposed MOCOAL based on hybrid elite mechanism and Meta-Lamarckian learning is introduced. In Sect. 4, the performance of the proposed algorithm is tested by benchmark functions. In Sect. 5, six cases are

selected in IEEE 30-node system and IEEE 57-node system for simulation experiments. In Sect. 6, the conclusion and future research direction are presented.

2 Mathematical model of multi-objective optimal power flow problem

2.1 Multi-objective optimization problem

Comparing solutions is a simple task in single-objective optimization since there is only one objective function to consider. For the minimization problem, the solution X is superior to Y if and only if $f(X)$ is less than $f(Y)$. However, in the field of multi-objective problems, each solution has multiple evaluation indexes, so some definitions need to be introduced.

Definition 1 Pareto domination. When a solution X is superior to a solution Y in all objectives, then the solution X dominates the solution Y , or alternatively, the solution X is dominated by the solution Y . If the solution X has at least one goal better than the solution Y , and there is some index worse than the solution Y , then the solution X and the solution Y do not dominate each other.

Definition 2 Pareto optimal solution. Solutions that are not dominated by either solution are called Pareto optimal solutions and are also called non-dominated solutions.

Definition 3 Pareto solution set. A set of non-dominant solutions is called a Pareto solution set.

Definition 4 Pareto frontier. Pareto solution sets form Pareto frontier after function mapping.

The multi-objective optimal power flow (MOOPF) problem is a constrained optimization problem. The objective is to minimize the selected objective functions under the condition of satisfying the equality constraints and inequality constraints. Since each index conflicts with each other, the answer to this problem is a Pareto solution set, which represents the optimal trade-off between multiple objectives. Mathematically, the MOOPF problem can be expressed in the following form:

$$\begin{aligned} \text{Minimize } F(x, y) &= f_1(x, y), f_2(x, y) \dots f_N(x, y) \\ \text{s.t. } g(x, y) &= 0 \\ h(x, y) &\leq 0 \end{aligned} \quad (1)$$

where, $f(x, y)$ represents the objective function of the OPF problem; $F(x, y)$ represents the set of multiple objective functions; $g(x, y)$ represents the equality constraint; The inequality constraint is represented by $h(x, y)$; x and y represent control variables and state variables respectively.

2.2 Objective function

In this experiment, a total of three objective functions are selected, which are fuel cost, active power loss and pollution emission.

2.2.1 Fuel cost

The fuel cost of each thermal power unit has a certain functional relationship with the active power. In the study, approximate fitting is performed in the form of quadratic function, which is shown in Eq. (2).

$$f_{cost} = \sum_{i=1}^{NG} a_i + b_i P_{G_i} + c_i P_{G_i}^2 \quad (2)$$

where, a_i , b_i , c_i are the fuel cost coefficient of the i -th generator, and P_{G_i} are the active power emitted by the i -th generator; NG is the total number of generators.

2.2.2 Active power loss

There are resistance and conductance with fixed parameters in transmission line. Active power loss occurs when power is transferred through the grid. The mathematical formula of active power loss is shown in Eq. (3).

$$f_{Ploss} = \sum_{i=1}^{NL} \sum_{j \neq i}^{NL} G_{ij} [V_i^2 + V_j^2 - 2V_i V_j \cos(\delta_i - \delta_j)] \quad (3)$$

where, i and j are the i -th and j -th nodes respectively, G_{ij} is the conductance between the two nodes, V is the node voltage, and δ is the phase angle corresponding to the node voltage; NL is the total number of transmission lines.

2.2.3 Emission

In the current society, environmental protection is an important topic. It is necessary to reduce the emission index of thermal power units. The total emission of air pollutants such as CO_x and NO_x produced by thermal power units can be defined as:

$$f_{Emission} = \sum_{i=1}^{NG} \gamma_i + \beta_i P_{G_i} + \alpha_i P_{G_i}^2 + \xi_i \exp(\lambda_i P_{G_i}) \quad (4)$$

where, α_i , β_i , γ_i , ξ_i and λ_i are the emission coefficients of the i -th generator.

2.3 Control variables

The control variables are the quantity that can be adjusted manually in the power system. They mainly include the active power output by the generator, the voltage of the generator bus, the tap position of the on-load tap changer and the reactive power of the shunt capacitor. The operation state of the power system can be changed by changing the control variables.

$$x = [P_{G_2}, \dots, P_{G_{NG}}, V_{G_1}, \dots, V_{G_{NG}}, T_1, \dots, T_{NT}, \dots, Q_{C_1}, \dots, Q_{C_{NC}}] \quad (5)$$

where, the active power output of the generator is $P_{G_2}, \dots, P_{G_{NG}}$; The magnitude of the generator bus voltage is $V_{G_1}, \dots, V_{G_{NG}}$; T_1, \dots, T_{NT} is the setting of the transformer tap position; $Q_{C_1}, \dots, Q_{C_{NC}}$ is the reactive capacity of shunt capacitor; NT is the number of transformers; NC is the number of reactive capacitors.

2.4 State variables

State variables are called dependent variables, which changes as the control variable changes. The state variable in the OPF problem is shown in Eq. (6). Once the control variables in the system are defined, by employing the Newton–Raphson method, the power flow of the entire system and the voltage value of each node can be determined.

$$y = [P_{G_1}, V_{L_1}, \dots, V_{L_{NL}}, Q_{G_1}, \dots, Q_{G_{NG}}, \dots, S_{l_1}, \dots, S_{l_{NL}}] \quad (6)$$

where, P_{G_1} is the active power input by the balance node (in 30-node system and 57-node system); NL respectively represent the number of load nodes (PQ nodes). $V_{L_1}, \dots, V_{L_{NL}}$ is the voltage of each load node in the power system; $Q_{G_1}, \dots, Q_{G_{NG}}$ is the reactive power generated by the generator; $S_{l_1}, \dots, S_{l_{NL}}$ is the power transmitted on the line;

2.5 Equality constraints

The power in the power system must satisfy the law of conservation of energy, which means that the power emitted is equal to the power consumed. The most typical equality constraint is the balance of active power and reactive power in the system, as shown in Eqs. (7 and 8).

$$P_{G_i} = P_{D_i} + V_i \sum_{j=1}^{NB} V_j [G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j)], i = 1, \dots, NB \quad (7)$$

$$Q_{G_i} = Q_{D_i} + V_i \sum_{j=1}^{NB} V_j [G_{ij} \sin(\delta_i - \delta_j) - B_{ij} \cos(\delta_i - \delta_j)], i = 1, \dots, NB \quad (8)$$

Equation (7) is the active power equation constraint, and P_{D_i} is the active power demand of load. Equation (8) is the constraint of reactive power equation, and Q_{D_i} is the reactive power demand of load. δ_i represents the phase Angle of the i -th bus. G_{ij} and B_{ij} are the conductance and inductance of the transmission line between the i -th bus and the j -th bus, respectively. NB indicates the number of nodes.

No additional treatment is needed for this equality constraint, because the termination condition of Newton–Raphson method can meet Eqs. (7 and 8). The successful execution of the power flow calculation program indicates that the results conform to the equation constraints.

2.6 Inequality constraints

Inequality constraints mainly restrict the safe operation of devices in the system. The following four parts are considered here: generator constraints, reactive capacitor capacity constraints, transformer constraints and safety constraints.

(1) Generator constraints

$$P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max}, i = 1, \dots, NG \quad (9)$$

$$Q_{G_i}^{min} \leq Q_{G_i} \leq Q_{G_i}^{max}, i = 1, \dots, NG \quad (10)$$

$$V_{G_i}^{min} \leq V_{G_i} \leq V_{G_i}^{max}, i = 1, \dots, NG \quad (11)$$

(2) Reactive compensation constraint

$$Q_{C_j}^{min} \leq Q_{C_j} \leq Q_{C_j}^{max}, j = 1, \dots, NC \quad (12)$$

(3) Transformer constraint

$$T_K^{min} \leq T_K \leq T_K^{max}, K = 1, \dots, NT \quad (13)$$

(4) Safety constraints

$$V_{L_m}^{min} \leq V_{L_m} \leq V_{L_m}^{max}, m = 1, \dots, NL \quad (14)$$

$$S_{l_n} \leq S_{l_n}^{max}, n = 1, \dots, NI \quad (15)$$

where, $S_{l_n}^{max}$ represents the maximum transmission power on the i -th transmission line.

Some of these inequality constraints restrict the value range of control variables, and the upper and lower limits of control variables can meet these inequality constraints. The other part is to limit the value range of the state variables and the penalty function method is selected to deal with it. The penalty function method can transform the constrained optimization problem into an unconstrained optimization problem. Equations (16 and 17) are the penalty function and the modified objective function formula respectively.

$$\begin{aligned} \text{Penalty} = & k_p \times (P_{G1} - P_{G1}^{lim})^2 + k_Q \times \sum_{i=1}^{NG} (Q_{Gi} - Q_{Gi}^{lim})^2 \\ & + k_V \times \sum_{m=1}^{NL} (V_{Lm} - V_{Lm}^{lim})^2 + k_S \times \sum_{n=1}^{NI} (S_{l_n} - S_{l_n}^{lim})^2 \end{aligned} \quad (16)$$

$$f_i = f_i + \text{penalty} \quad (17)$$

where, f_i is the i -th objective function; penalty is a penalty item; The value of k_p is set to 10^6 ; The value of k_Q is set to 10^6 ; The value of k_V is set to 10^9 ; The value of k_S is set to 10^6 . The voltage constraint of the load node is easily violated, so the maximum penalty coefficient is set for it.

2.7 Fuzzy membership function

After solving the MOOPF problem, a set of Pareto solutions are obtained. Because these solutions are in the same dominant level, the pros and cons of each solution in Pareto frontier cannot be directly judged. In MOP, the fuzzy system can be used to deal with the contradictory relations of various objective functions. The concept of fuzzy membership

function is introduced in Ref. Hazra and Sinha (2011). Membership function defined by a single objective function can be described as follows:

$$\mu_i^k = \begin{cases} 1 & f_i \leq f_i^{\min} \\ \frac{f_i^{\max} - f_i}{f_i^{\max} - f_i^{\min}} & f_i^{\min} < f_i < f_i^{\max} \\ 0 & f_i \geq f_i^{\max} \end{cases} \quad (18)$$

where, f_i^{\max} is the maximum value of the i -th objective function in Pareto solution set, and f_i^{\min} is the minimum value of the i -th objective function in Pareto solution set. The image of this function is shown in Fig. 1.

For the k -th individual in the solution set, the normalized membership function μ^k is defined as follows:

$$\mu^k = \frac{\sum_{i=1}^N \mu_i^k}{\sum_{k=1}^{Po} \sum_{i=1}^N \mu_i^k} \quad (19)$$

where, Po represents the number of Pareto solution sets, and N represents the number of objective functions.

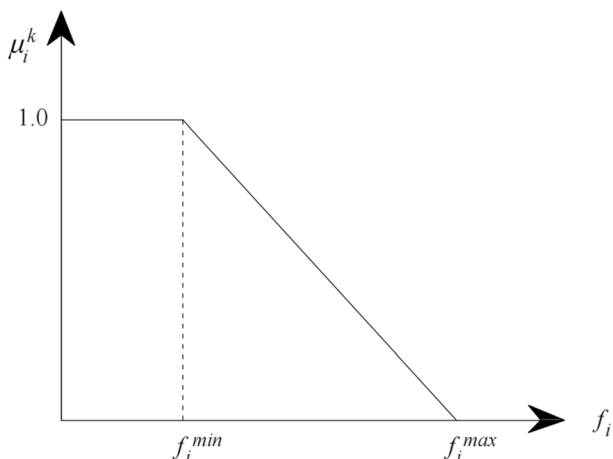
The greater the value of normalized membership function μ^k , the higher the satisfaction of the solution. The solution with the maximum membership function is the best compromise solution.

3 Multi-objective coyote optimization algorithm based on hybrid elite mechanism and Meta-Lamarckian learning strategy

3.1 Coyote optimization algorithm

COA is inspired by the behavior of coyotes and operates on a swarm-based approach. COA does not prioritize the wolf hierarchy and has a distinct algorithmic structure. The focus

Fig. 1 Fuzzy membership function



of COA is to imitate the social structure and experience-sharing aspect of coyotes. In the COA, the population of coyotes is divided into N_p packs with N_c coyotes in each pack. The number of coyotes in each pack is fixed. Therefore, the population number in this algorithm is obtained by multiplying N_p and N_c . Each coyote has a social condition attribute (a set of decision variables), and the social condition of the c -th coyote of the p -th pack is written as:

$$soc_c^{p,t} = \vec{x} = (x_1, x_2, \dots, x_D) \quad (20)$$

where, SOC represents the decision variable, D is the search space dimension. The first step is to initialize the coyote population. As a randomized algorithm, the initial social conditions for each coyote of COA are set randomly. It passes through Eq. (21) and assign a random value to the j -th dimension of the c -th coyote of the p -th pack in the searching space during the t -th iteration.

$$soc_{c,j}^{p,t} = lb_j + r_j \cdot (ub_j - lb_j) \quad (21)$$

where, lb_j and ub_j represent the lower and upper bounds of the j -th dimensional control variable respectively, r_j is a random number between $[0, 1]$. Coyotes were then assessed for their adaptation to current social conditions.

$$fit_c^{p,t} = f(soc_c^{p,t}) \quad (22)$$

where, $fit_c^{p,t}$ is the fitness value (objective function value).

There is one alpha coyote in each pack, and it is the individual with the best fitness value. In the minimization problem, the alpha of the p -th pack at time t -th is defined as:

$$alpha^{p,t} = \left\{ soc_c^{p,t} \mid arg_{c=\{1,2,\dots,N_c\}} \min f(soc_c^{p,t}) \right\} \quad (23)$$

COA assumes that coyotes have a certain amount of intelligence and organization, and each population shares social conditions that will help the population develop. Thus, the COA associates individual information from coyotes and calculates it as a cultural trend for the group.

$$cult_j^{p,t} = \begin{cases} O_{\frac{N_c+1}{2}j}^{p,t}, & N_c \text{ is odd} \\ \frac{O_{\frac{N_c}{2}j}^{p,t} + O_{\frac{N_c+1}{2}j}^{p,t}}{2}, & \text{otherwise} \end{cases} \quad (24)$$

where, $O^{p,t}$ denotes the social condition ranking of all coyotes in the p -th pack in the range $[1, D]$ at the t -th iteration. All in all, the cultural disposition of the pack was equal to the median of the social conditions of all coyotes in the pack.

For showing the social conditions of different coyotes in pack affect each other, the COA assumes that each coyote individual receives alpha effects (δ_1) and population effects (δ_2). The former represents the cultural difference between the random coyote cr_1 and the alpha coyote, while the latter represents the difference between the cultural tendency of the random coyote cr_2 and the group. δ_1 and δ_2 are shown in Eq. (25).

$$\begin{aligned}\delta_1 &= \alpha^{p,t} - soc_{cr_1}^{p,t} \\ \delta_2 &= cult^{p,t} - soc_{cr_2}^{p,t}\end{aligned}\quad (25)$$

Therefore, the new social conditions of coyotes are updated by the influence of alpha and group.

$$new_soc_c^{p,t} = soc_c^{p,t} + r_1 \cdot \delta_1 + r_2 \cdot \delta_2 \quad (26)$$

where, r_1 represents the weights affected by alpha and population. r_1 is defined as random numbers in the range [0, 1] generated with uniform distribution. r_2 decreases linearly with the number of iterations, $r_2 = 1 - it/Maxit$. The new social situation is then assessed by Eq. (27).

$$new_fit_c^{p,t} = f(new_soc_c^{p,t}) \quad (27)$$

Coyotes have the cognitive ability to judge whether new social conditions are better than old ones, which means that only when they get better social conditions, they will be updated.

$$soc_c^{p,t+1} = \begin{cases} new_soc_c^{p,t}, & new_fit_c^{p,t} < fit_c^{p,t} \\ soc_c^{p,t}, & otherwise \end{cases} \quad (28)$$

After each pack position update, coyote births and deaths are considered. The birth of a new coyote is a crossover of the social conditions of the parents (chosen at random) and then the random effects of the environment. The formula for the birth is shown in Eq. (29).

$$pup_j^{p,t} = \begin{cases} soc_{r_1j}^{p,t}, & rnd_j < P_s \text{ or } j = j_1 \\ soc_{r_2j}^{p,t}, & rnd_j \geq P_s + P_a \text{ or } j = j_2 \\ R_j, & otherwise \end{cases} \quad (29)$$

where, r_1 and r_2 are random coyote individuals in the p -th pack; j_1 and j_2 are the two random dimensions of the problem; P_s is the scattering probability, P_a is the association probability; R_j is a random number in the range of control variables for the j -th dimension; and rnd_j is a random number generated with a uniform distribution in the range [0, 1]. Scattering and association probabilities guide the cultural diversity of coyotes so that P_s and P_a can be defined as:

$$\begin{aligned}P_s &= 1/D \\ P_a &= (1 - P_s)/2\end{aligned}\quad (30)$$

where, P_a has the same effect on both parents.

After evaluating the fitness values of all coyotes, the one with the highest fitness value is chosen as the global optimal solution for the problem. The pseudo code for COA is presented in Algorithm 1.

Algorithm 1 Pseudo code of the COA

```

1: Initialize using Eq. (21)
2: Evaluate the fitness value of the population
3: While Iter < MaxIter
4:   For each  $p$  pack do
5:     find the alpha coyote of the pack (Eq. 23).
6:   Calculate the social tendency of the pack (Eq. 24).
7:   For each  $c$  coyotes of the  $p$  pack do
8:     Update the social condition (Eq. 26).
9:     Evaluate the new social condition (Eq.27).
10:    Adaptation (Eq. 28).
11:   End For
12:   Birth (Eq.29).
13: End For
14: End While
15: Select the best adapted coyote

```

3.2 Meta-Lamarckian learning

Traditionally, Meta-Lamarckian Learning is often used in memetic algorithm (MA), and a local search process is added after the iterative update of MA (Ong and Keane 2004). It is difficult for a single local search method to achieve good results in different problems. Therefore, multiple local search (LS) methods are often used in MA searches. Meta-Lamarckian learning is motivated by the desire to improve search performance and reduce the probability of using inappropriate local methods. Meta-Lamarckian learning (adaptive) strategy can be described as cooperative and/or competitive. Competition means that LS method with higher fitness improvement has a higher chance to be selected for subsequent optimization. Cooperation means that LS and their improvement rewards work together to select an LS for subsequent optimization (Konstantinidis et al. 2018). Meta-Lamarckian Learning usually uses the improvement of the fitness value of a single objective function as an indicator to establish a reward mechanism, so it is often used in single-objective optimization and multi-objective optimization based on decomposition. For being used in multi-objective optimization, the reward mechanism is defined as follows:

$$\rho_k = \frac{n_s}{n} \quad (31)$$

where, ρ_k is the reward value of the k -th LS method, n is the number of times that the LS method is used in the iteration, and n_s is the number of times that the LS method is used to generate a non-inferior solution.

The incentive mechanism is to calculate the ratio of the number of non-inferior solutions generated by using each local optimization strategy to the number of generated individuals.

In each iteration, after obtaining the reward value ρ_k of each LS method, the probability of updating the roulette LS method after normalization is used for the next iteration. In other words, if a certain LS is used to generate the highest proportion of high-quality solutions, it is more likely to be selected in the next iteration. The probability of each LS method being selected at the beginning is equal. With the progress of iteration, the method with high reward value obtains higher probability of adoption. Random roulette works as follows:

Step 1: Calculate the reward value ρ_k of each LS method.

Step 2: Standardize (normalize) the reward value of each LS method to obtain the relative reward value.

Step 3: Allocate space for each LS based on relative reward value.

Step 4: Generate a random number and select the LS method of the disk position corresponding to the random number.

The common local optimization methods include crossover, mutation, Powell method and simplex search method. In this experiment, a total of four crossover operators are adopted into the local optimizer, which are respectively called horizontal crossover operator, longitudinal crossover operator, elite crossover operator and direct crossover operator.

3.2.1 Transverse crossover operator

Inspired by the crisscross optimization algorithm (CSO) (Meng et al. 2014), the transverse crossing process is selected as the LS. As shown in Eq. (32), the function of this operator is to generate new individuals at the position between parents with a high probability and individuals at the extension line of parents with a low probability.

$$Xnew_{i,d} = r_1 \cdot X_{i_1,d} + (1 - r_1) \cdot X_{i_2,d} + a \cdot (X_{i_1,d} - X_{i_2,d}) \quad (32)$$

where, $Xnew_{i,d}$ is the i -th new individual in the d -th dimension; r_1 is a random number between $[0, 1]$; a is the random number between $[-1, 1]$; $X_{i_1,d}$ and $X_{i_2,d}$ are randomly selected parents in the cross operation.

3.2.2 Longitudinal crossover operator

Inspired by the CSO (Meng et al. 2014), the longitudinal crossover process is selected as the LS. As shown in Eq. (33), the effect of the operator is to change the value of one dimension of the individual.

$$Xnew_{i,d_1} = r_2 \cdot X_{i,d_1} + (1 - r_2) \cdot X_{i,d_2} \quad (33)$$

where, r_2 is a random number between $[0, 1]$; X_{i,d_1} and X_{i,d_2} are values of the same individual in the dimensions of d_1 and d_2 . Since solutions may have different upper and lower limits in different dimensions, the values of each dimension should be normalized.

3.2.3 Direct crossover operator

Equation (29) is selected as the LS, whose function is to generate individuals in the position of parent or parent.

3.2.4 Elite crossover operator

An elite crossover operator is proposed based on the direct crossover operator. As shown in Eq. (34), it serves to cross the position of the current coyote with that of the alpha coyote it follows.

$$X_{new_{i,d}} = \begin{cases} \alpha_{i,d}, & \text{rnd}_j < P_s \text{ or } j = j_1 \\ X_{i,d}, & \text{rnd}_j \geq P_s + P_a \text{ or } j = j_2 \\ R_j, & \text{otherwise} \end{cases} \quad (34)$$

where, $\alpha_{i,d}$ is an elite coyote that $X_{i,d}$ has followed.

3.3 Multi-objective COA based on hybrid elite framework and Meta-Lamarckian learning strategy

3.3.1 Elite non-dominant sorting

In the proposed MOCOA, NSGA-II's elite non-dominant ranking method and the crowding distance method to maintain diversity are introduced. The crowding distance is calculated to rank the populations of the same non-dominant level. First, a non-dominant ranking was used to obtain non-dominant levels of different individuals, and then the crowding distance method was used to maintain the diversity between the optimal sets.

3.3.1.1 Fast non-dominated sorting Firstly, all targets of the objective function F are evaluated for each solution obtained from the basic search method (COA) or the initially generated random population P_0 . Each solution p has two properties, n_p is the number of solutions that dominate individual p , and S_p is the set of solutions that individual p dominates.

- (1) For solutions with $n_p = 0$, the solutions are not dominated by any individual, whose non-dominated level p_{rank} is set to 1 and stored in set F_1 .
- (2) For each solution p with $n_p = 0$, access each member q in the set S_p , and its dominant count n_q decreases by 1. If the n_q count drops to zero, the corresponding solution q is stored in the second non-dominated level set F_2 , whose non-dominated level p_{rank} is set to 2.
- (3) Repeat the process for each member of the second non-dominated level to obtain the third non-dominated level, and then repeat the process until the whole population is divided into different non-dominated levels.

3.3.1.2 Determine crowding distance To ensure that Pareto optimal solutions are well-distributed in the objective space, NSGA-II utilizes a crowding distance method to assess the quality of each solution within the same front, resulting in a more evenly distributed solution set. The main goal of using the crowding distance approach is to preserve population diversity by achieving a trade-off between solutions. Specifically, it refers to the density of individuals in a single rank layer after the non-dominant sorting of a population in accordance with the dominant relationship.

The crowding degree/crowding distance is calculated as follows. For each objective function, find two solutions adjacent to the current solution and calculate the functional difference between the two solutions. To calculate the crowding distance of a given solution, the differences between the objective function values of neighboring solutions are summed. The individual crowding degree at the boundary of each non-dominant layer is directly set to infinity (Jeyadevi et al. 2011). The sum of the two sides of the rectangle in Fig. 2 is the crowding distance of the p -th individual.

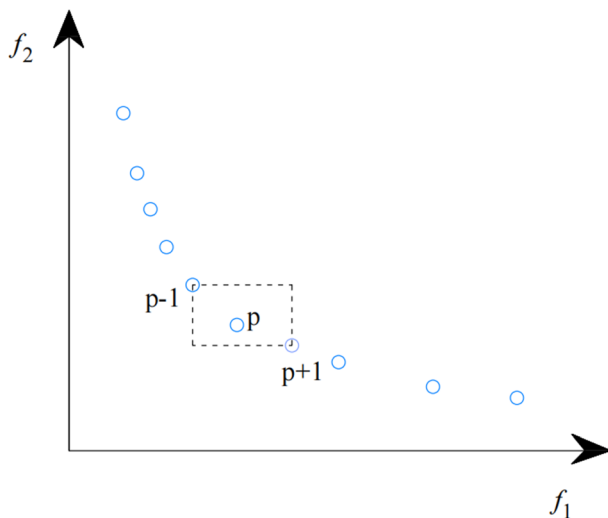
3.3.1.3 Crowding comparison operator and elite strategy After the previous fast non-dominant ranking and crowding degree calculation, the i -th individual in the population has two attributes: the non-dominant layer p_{rank} (the number of levels) and the crowding distance p_d . According to these two attributes, the crowding degree comparison operator can be defined as follows. The individual p is compared with another individual q . If any of the following conditions are true, the individual p wins.

- (1) If the non-dominated layer of individual p is better than the non-dominated layer of individual q , $p_{rank} < q_{rank}$;
- (2) If they have the same rank and the individual p has a larger crowding distance than the individual q , that is, $p_{rank} = q_{rank}$ and $p_d > q_d$.

The first condition ensures that the selected individual belongs to the superior non-inferior rank. The second condition can select the individual in the less crowded area (with a greater distance of crowding) among two individuals with the same non-inferior rank.

The elite strategy is used to select individuals to enter the next iteration. The new population P generated in the t -th iteration is combined with the old population Q . Then a series of non-dominated sets are generated by non-dominated sorting, and the degree of crowding is calculated. Set the population number to N in the iteration, and select from the first layer until enough N individuals are selected according to the crowding comparison operator. These N better individuals enter the next iteration process and continue to update according to the formula of COA. This selection process is shown in Fig. 3.

Fig. 2 Crowding distance



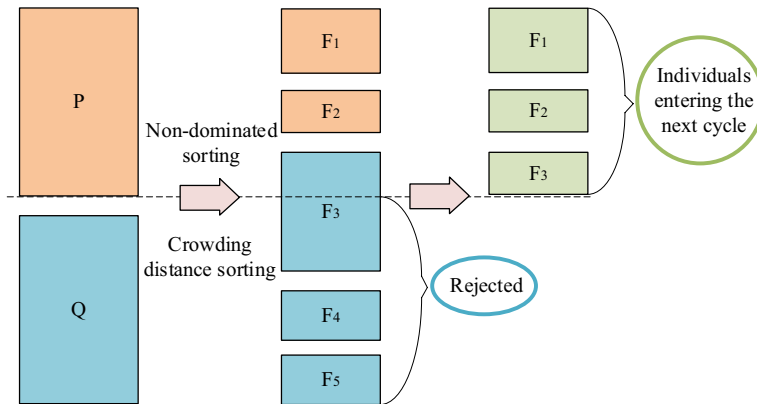


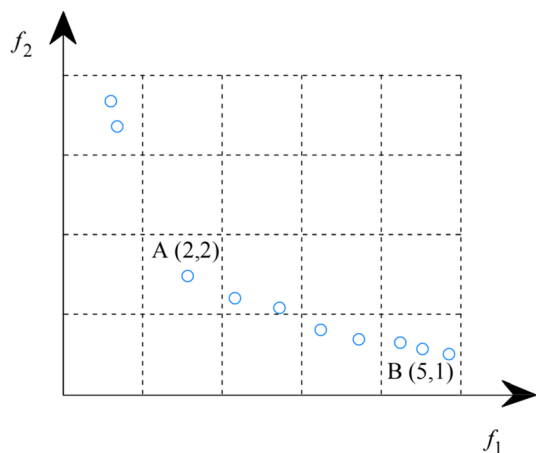
Fig. 3 Individual selection based on non-dominant ranking

3.3.2 Archives based on grid mechanism

3.3.2.1 Grid mechanism For the solution set stored in the archive, the target space is divided equally by grid, and the number of grids on each target is set manually. Figure 4 is a schematic diagram in two-dimensional space. The number of grids on each target is 5. Each grid containing the solution is given an index number. For example, the index number of grid A in Fig. 4 is (2, 2). The purpose of the grid mechanism is to distinguish the density of the archive space in order to find a more crowded or sparse area for the next operation (Coello and Lechuga 2002).

3.3.2.2 External archive An external archive is a storage unit defined as a fixed size. It can save or retrieve the non-dominant Pareto optimal solution obtained so far. The key module for archiving is an archiving controller that can control archiving when the solution wants to access the archive or when the archive is full. It is important to note that the archive has a maximum number of members. During the iteration, the non-dominant solutions obtained to date were compared to archived data. There are three different cases that can happen.

Fig. 4 Individual selection process of grid mechanism



- (1) New members are dominated by at least one archive member. In this case, the solution should not be allowed to enter the archive.
- (2) The new solution dominates one or more solutions in the archive. In this case, delete the dominant solution from the archive and allow the new solution to enter the archive.
- (3) If the new solution and archive members are not mutually dominant, the new solution should be added to the archive.

If the archive is full, the grid mechanism should first be run to rearrange the segmentation of the object space. Through the roulette selection technique, the grids are selected to remove one of the solutions and the probability of each grid being selected is shown in Eq. (35). Then, the new non-dominated solution is recorded in the archive to improve the diversity of Pareto optimal frontier. As shown in Fig. 4, when the archive is full, there is a greater probability to select B(5, 1), the most crowded area, and randomly delete one of the solutions.

$$P = \frac{n}{E} \quad (35)$$

where, E is a constant and n is the number of solutions in the grid.

3.3.2.3 Elite selection mechanism Elite is the alpha coyote in COA. Firstly, the grid mechanism is used to divide the archive. Then select a solution from the archive as alpha coyote through roulette. The probability of selection is calculated by Eq. (36).

$$P = \frac{E}{n} \quad (36)$$

where, E is a constant and n is the number of solutions in the grid.

The fewer the number of solutions in the grid, the greater the chance that the grid will be selected. As shown in Fig. 4, there is only one solution in A (2, 2), so A has the highest probability of being selected. The mechanism selects the sparser location solution as alpha coyote (elite). Alpha coyote, as the leader of the population, will guide the population to search for a more sparse solution set space, which can help it to find a more uniform Pareto front.

3.3.3 The process of MOCOA-ML

The proposed multi-objective coyote optimization algorithm (MOCOA) uses the multi-objective framework of non-dominated sorting and external files to obtain the Pareto optimal solution. There is one and only one optimal solution obtained by the single objective COA, which is the solution corresponding to the optimal fitness value. The MOCOA adopts the idea of COA to update the population position, merges the new solution set and the old solution set. Then, the non-dominant sorting and the crowding distance methods are used to get the undominated relations in the new set. After that, according to the size of the population, select the better individuals to enter the next iteration process, and other poor individuals are eliminated (dead). Archive the non-dominated solution (the first frontier individual) obtained from the non-dominated sorting. If the archive is full, use the grid mechanism to delete and add the individual. In addition, during the iteration process, alpha coyotes (leaders) are also selected from the archive according to the roulette method. After the iteration, output the Pareto solution set in the archive.

The Meta-Lamarckian learning strategy was combined with multi-objective coyote optimization algorithm, and it was named MOCOA-ML. On each loop, after the coyote position in each pack is updated, the local optimizer based on Meta-Lamarckian learning starts working, randomly picking individuals in the pack for a local search. Each time LS is selected and the scheme is selected by way of roulette according to the reward value in the last iteration. The individual generated by the local optimizer is compared with one of its parents. If the new individual dominates the parent, the new individual replaces the parent. If the new individual and the parent do not dominate, or the new individual is dominated by the parent, the new individual replaces any individual in the previous iteration. In other words, we tend to retain solutions generated by the local optimizer. This approach does not add any more computational pressure to the non-dominated sorting process, since the number of individuals participating in non-dominated sorting is still twice as large as the number of populations. The flow chart of MOCOA-ML is shown in Fig. 5. It should be noted that MOCOA-ML differs from MOCOA in whether it contains a local optimizer under the Meta-Lamarckian learning strategy.

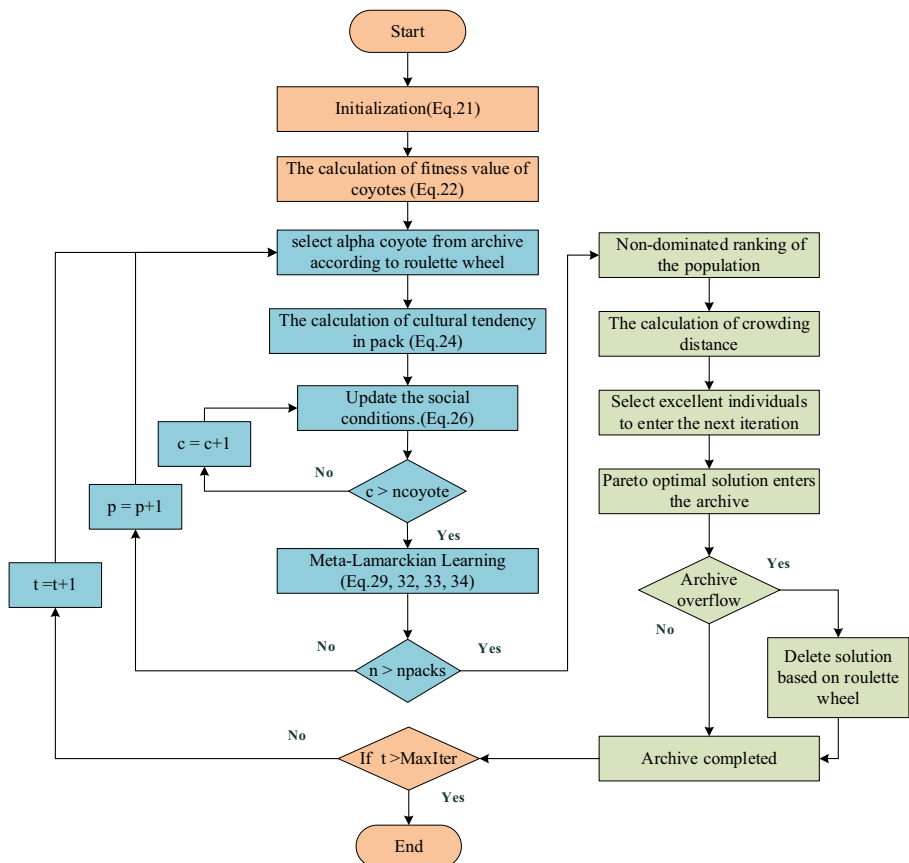


Fig. 5 Flow chart of MOCOA-ML

4 Test function simulation and result analysis

In order to verify the performance of MOCOAL-ML, several test functions were selected for testing, and the results were compared with MOCOAL, MODA, MOPSO, MOJAYA, NSGA-II, MOEA/D, MOAOS and MOTEAL. Because the algorithms used in the experiments were all random algorithms, and in order to be true and fair, when MOCOAL-ML and other multi-objective intelligent optimization algorithms were used for the test, each group of experiments were independently run 10 times. The maximum number of iterations is set to 300, the population number N is set to 100, and the size of archive is set to 100. The parameter settings of the multi-objective improved algorithm and comparison algorithms are shown in Table 1. For each algorithm, calculate the fuzzy membership function of each solution according to Eqs. (18, 19) in Sect. 2.7, and the solution with the maximum membership function μ_{max} is considered the best compromise solution.

4.1 Performance metrics

Convergence and diversity are two key points in finding an appropriate Pareto optimal solution set for a particular problem. Convergence refers to the ability of multi-objective algorithm to determine the accurate approximation of Pareto optimal solution. Diversity refers to the ability of the algorithm to find a more complete Pareto front. The ultimate goal of the multi-objective optimization algorithm is to find the most accurate approximate value of the true Pareto optimal solution (convergence) with uniform distribution (diversity) on all targets. In this part, three commonly used indicators are selected to reflect the advantages and disadvantages of Pareto solution set of each algorithm. They are inverted generational distance (IGD) (Coello and Cortés 2005) and hypervolume (HV) (Zitzler and Thiele 1999). The first indicator is a reverse indicator, and the second is a positive indicator.

Table 1 Setting of algorithm parameters

Algorithm	Parameters
MOCOAL	Number of pack (n_packs) = 10; Number of coyotes in each pack (n_coy) = 10; Number of grid = 7; Archive size = 100
MOCOAL-ML	Number of pack (n_packs) = 10; Number of coyotes in each pack (n_coy) = 10; Number of grid = 7; Archive size = 100
MODA Mirjalili (2016)	Archive size = 100; Inertia weigh(w) = $w = 0.9 - t^* / ((0.9 - 0.2) / \text{MaxIt})$
MOPSO Hazra and Sinha (2011)	Number of Grid = 7; Inertia weight(w) = 0.5; coefficient (c_1) = 1; Social coefficient (c_2) = 2; Archive size = 100
MOJAYA Berrouk et al. (2018)	Archive size = 100
NSGA-II Jeyadevi et al. (2011)	Crossover probability (Pc) = 0.7; Mutation probability (Pm) = 0.4; $\mu = 0.02$;
MOEA/D Medina et al. (2014)	Archive size = 100; gamma = 0.5; T = 0.15 * Population
MOAOS Azizi et al. (2022)	Archive size = 100; Layer number = 10; Foton rate = 0.1; Number of grid = 30;
MOTEAL Kumar et al. (2022)	$c_1 = 1.1$; $c_2 = c_1 * 2$; Crossover probability (Pc) = 0.7; Mutation probability (Pm) = 0.4; $\mu = 0.02$; Mutation Step Size = 0.1

4.1.1 Inverted generational distance

The IGD metric is used to calculate the minimum distance between an individual on the actual Pareto frontier and the set of individuals generated by the algorithm. It can be expressed as:

$$IGD = \frac{\sqrt{\sum_{p=1}^{ko} (d_i^p)^2}}{ko} \quad (37)$$

where, ko is the number of Pareto solutions. It is expressed as the Euclidean distance between the p -th real Pareto solution and the nearest obtained Pareto solution.

4.1.2 Hypervolume

The HV value is the volume of the space covered by the Pareto front. The higher the HV value, the better the diversity and convergence of the corresponding Pareto frontier.

$$HV = \delta(\bigcup_{i=1}^{|N|} c^i) \quad (38)$$

where, δ is a Lebesgue measure used to measure volume. $|N|$ represents the number of Pareto solution sets, and c^i represents the hypercube formed by the reference point and the i -th solution in the solution set.

4.2 Function optimization simulation and result analysis

4.2.1 Simulation result and analysis of benchmark test functions

In order to prove the performance of MOCOA-ML, experiments were carried out on the test functions ZDT1-ZDT4, ZDT6, DTLZ2 and DTLZ4-DTLZ7. MOCOA, MODA, MOPSO, MOJAYA, NSGA-II, MOEA/D, MOAOS and MOTE0 were selected as the comparison algorithms. Tables 2, 3 records the optimal value, average value and standard deviation of each algorithm in IGD and HV. Figure 6 shows the Pareto frontier of double-objective test functions, and Fig. 7 shows the Pareto frontier of three-objective test functions.

From the experimental results, it can be observed that MOCOA-ML has the ability to find the Pareto front of each test function and has better convergence and coverage. The average rankings obtained from the Friedman test are listed in Tables 2 and 3, and MOCOA-ML ranks first in both IGD and HV. This indicates that it outperforms MOCOA, MODA, MOPSO, MOJAYA, NSGA-II, MOEA/D, MOAOS and MOTE0 on most test functions. Additionally, MOCOA-ML performs better than MOCOA in all performance metrics, demonstrating the effectiveness of the Meta-Lamarckian learning strategy. Among the 10 test functions, MOTE0 performs well, second only to MOCOA-ML. NSGA-II and MOPSO also have good results, while MOJAYA performs the worst. MOCOA-ML exhibits better competitiveness in both bi-objective and tri-objective problems. In summary, MOCOA-ML showcases commendable performance

Table 2 Simulation results of test functions (IGD)

Function		MCOA-ML	MCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
ZDT1	Best	4.36E-04	4.20E-04	5.44E-02	5.00E-04	1.76E-02	1.65E-03	8.11E-04	3.58E-04	3.65E-04
	Ave	5.16E-04	1.80E-03	6.46E-02	1.29E-03	2.27E-02	4.55E-03	3.20E-03	4.47E-04	2.71E-03
	Std	6.54E-05	2.78E-03	7.90E-03	2.18E-03	4.32E-03	3.17E-03	2.80E-03	1.06E-04	1.89E-03
ZDT2	Best	4.49E-04	4.11E-04	2.01E-03	4.58E-04	3.27E-02	4.24E-03	1.90E-02	3.56E-04	3.27E-04
	Ave	4.77E-04	2.85E-02	3.20E-03	1.14E-02	3.44E-02	1.11E-02	4.85E-02	8.98E-03	1.21E-03
	Std	3.19E-05	2.85E-02	2.15E-03	2.15E-02	5.46E-03	3.94E-03	1.56E-02	1.41E-02	1.30E-03
ZDT3	Best	3.08E-04	2.91E-04	5.77E-03	3.17E-04	1.23E-02	1.78E-03	7.85E-03	2.83E-04	4.16E-03
	Ave	3.58E-04	1.15E-03	1.28E-02	1.58E-03	2.01E-02	5.89E-03	1.22E-02	3.28E-04	8.42E-03
	Std	3.63E-05	2.57E-03	4.35E-03	2.59E-03	4.86E-03	3.39E-03	2.40E-03	4.01E-05	2.06E-03
ZDT4	Best	3.76E-04	2.36E-02	6.72E-03	2.35E-02	4.01E-02	5.70E-02	2.98E-02	3.75E-04	1.75E-02
	Ave	4.57E-04	7.50E-02	1.87E-02	7.23E-02	6.34E-02	5.71E-02	2.90E-01	5.73E-03	7.18E-02
	Std	6.52E-05	4.16E-02	6.29E-03	3.55E-02	2.49E-02	9.90E-05	3.42E-01	1.29E-02	4.74E-02
ZDT6	Best	3.69E-04	3.77E-04	7.75E-04	4.08E-04	4.79E-03	2.80E-04	2.80E-02	4.30E-04	2.74E-04
	Ave	4.19E-04	4.28E-04	1.22E-02	5.08E-04	1.54E-02	2.93E-04	1.09E-01	1.62E-03	2.98E-04
	Std	3.54E-05	3.65E-05	3.15E-02	6.06E-05	7.36E-03	8.12E-06	7.09E-02	3.27E-03	1.99E-05
DTLZ2	Best	1.22E-03	1.27E-03	4.21E-03	1.97E-03	6.80E-03	1.01E-03	1.38E-03	6.89E-03	1.23E-03
	Ave	1.26E-03	1.35E-03	5.13E-03	2.27E-03	8.21E-03	1.09E-03	1.48E-03	7.37E-03	1.49E-03
	Std	3.31E-05	7.00E-05	6.20E-04	2.67E-04	8.63E-04	6.04E-05	1.04E-04	3.59E-04	1.38E-04
DTLZ4	Best	1.18E-03	1.22E-03	8.70E-03	1.94E-03	1.39E-02	1.01E-03	1.41E-03	1.07E-02	1.89E-03
	Ave	1.31E-03	1.37E-03	9.69E-03	3.72E-03	1.54E-02	1.16E-02	2.57E-03	1.16E-02	5.29E-03
	Std	7.73E-05	8.71E-05	8.86E-04	1.37E-03	1.87E-03	4.23E-03	2.63E-03	4.60E-04	2.34E-03
DTLZ5	Best	1.47E-04	1.87E-04	2.68E-03	7.33E-04	3.66E-03	7.03E-05	1.79E-04	4.50E-03	2.27E-04
	Ave	1.72E-04	2.34E-04	3.37E-03	1.05E-03	5.74E-03	8.60E-05	3.23E-04	5.44E-03	2.84E-04
	Std	1.74E-05	5.56E-05	6.38E-04	2.23E-04	1.02E-03	1.44E-05	1.34E-04	5.70E-04	6.11E-05
DTLZ6	Best	1.53E-04	1.57E-04	9.39E-02	1.85E-04	2.93E-03	1.11E-01	3.43E-04	1.28E-03	1.17E-04
	Ave	1.87E-04	1.27E-02	1.12E-01	2.19E-04	3.15E-02	1.38E-01	6.24E-03	6.18E-03	1.41E-04
	Std	1.80E-05	1.87E-02	1.19E-02	2.27E-05	2.37E-02	1.90E-02	1.02E-02	3.87E-03	3.21E-05

Table 2 (continued)

Function		MOCOA-ML	MOCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
DTLZ7	Best	5.67E-04	5.73E-04	1.36E-02	5.46E-04	9.21E-03	1.03E-03	3.69E-03	9.73E-04	6.34E-03
	Ave	5.98E-04	3.61E-03	1.64E-02	5.80E-04	1.00E-02	2.50E-03	8.26E-03	4.97E-03	6.42E-03
	Std	4.36E-04	4.20E-04	5.44E-02	5.00E-04	1.76E-02	1.65E-03	8.11E-04	3.58E-04	3.65E-04
Avg		1.80	4.40	6.70	4.50	8	4.35	6.40	4.65	4.20
Rank		1	4	8	5	9	3	7	6	2

Best results are bolded

Table 3 Simulation results of test functions (HV)

Function		MOCOA-ML	MOCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
ZDT1	Best	7.16E-01	7.16E-01	0.00E+00	7.14E-01	4.69E-01	6.75E-01	7.05E-01	7.18E-01	7.18E-01
	Ave	7.15E-01	6.77E-01	0.00E+00	6.90E-01	3.78E-01	6.48E-01	6.30E-01	7.17E-01	7.01E-01
	Std	4.28E-04	7.98E-02	0.00E+00	6.18E-02	9.69E-02	3.01E-02	8.07E-02	6.41E-04	1.28E-02
ZDT2	Best	4.42E-01	4.41E-01	4.16E-01	4.41E-01	9.09E-02	3.35E-01	7.57E-02	4.43E-01	4.43E-01
	Ave	4.41E-01	2.21E-01	3.89E-01	3.21E-01	8.18E-02	2.68E-01	7.57E-03	3.41E-01	4.31E-01
	Std	5.34E-04	2.31E-01	2.04E-02	1.73E-01	2.87E-02	4.13E-02	2.39E-02	1.63E-01	1.78E-02
ZDT3	Best	6.01E-01	6.01E-01	7.29E-01	6.03E-01	4.59E-01	7.65E-01	4.88E-01	6.03E-01	7.90E-01
	Ave	5.98E-01	5.84E-01	4.74E-01	5.80E-01	2.30E-01	6.69E-01	3.96E-01	6.00E-01	7.68E-01
	Std	9.51E-04	4.54E-02	1.77E-01	3.92E-02	1.51E-01	7.05E-02	6.54E-02	1.51E-03	3.10E-02
ZDT4	Best	7.18E-01	1.72E-01	5.21E-01	1.74E-01	1.05E-01	0.00E+00	8.70E-02	7.18E-01	2.83E-01
	Ave	7.17E-01	1.72E-02	3.23E-01	3.74E-02	4.69E-02	0.00E+00	1.46E-02	6.35E-01	3.32E-02
	Std	6.18E-04	5.43E-02	1.32E-01	7.19E-02	4.96E-02	0.00E+00	2.82E-02	2.01E-01	8.84E-02
ZDT6	Best	3.87E-01	3.87E-01	3.81E-01	3.86E-01	3.27E-01	3.88E-01	2.46E-02	3.86E-01	3.88E-01
	Ave	3.86E-01	3.86E-01	3.31E-01	3.85E-01	2.22E-01	3.88E-01	4.43E-03	3.71E-01	3.88E-01
	Std	3.21E-04	4.58E-04	1.18E-01	6.73E-04	6.25E-02	1.10E-04	9.42E-03	3.98E-02	2.30E-04
DTLZ2	Best	4.91E-01	4.78E-01	2.09E-01	4.01E-01	3.75E-02	5.21E-01	4.97E-01	4.02E-02	4.86E-01
	Ave	4.86E-01	4.65E-01	1.64E-01	3.59E-01	1.15E-02	5.12E-01	4.86E-01	1.98E-02	4.53E-01
	Std	5.42E-03	1.23E-02	3.64E-02	3.52E-02	1.09E-02	7.50E-03	8.13E-03	1.25E-02	1.78E-02
DTLZ4	Best	4.95E-01	4.96E-01	2.64E-02	4.66E-01	8.97E-02	5.36E-01	4.94E-01	0.00E+00	4.92E-01
	Ave	4.85E-01	4.72E-01	8.08E-03	2.71E-01	9.91E-03	1.86E-01	4.19E-01	0.00E+00	4.07E-01
	Std	9.80E-03	1.52E-02	8.53E-03	1.03E-01	2.82E-02	1.62E-01	8.47E-02	0.00E+00	5.45E-02
DTLZ5	Best	1.96E-01	1.92E-01	4.39E-02	1.44E-01	1.18E-03	1.99E-01	1.95E-01	6.83E-03	1.89E-01
	Ave	1.94E-01	1.88E-01	2.99E-02	1.17E-01	2.01E-04	1.99E-01	1.92E-01	9.68E-04	1.84E-01
	Std	1.00E-03	5.88E-03	9.56E-03	2.19E-02	4.23E-04	3.71E-04	4.62E-03	2.18E-03	4.96E-03
DTLZ6	Best	1.98E-01	1.98E-01	0.00E+00	1.98E-01	1.39E-01	0.00E+00	1.97E-01	1.56E-01	2.00E-01
	Ave	1.98E-01	9.88E-02	0.00E+00	1.97E-01	4.86E-02	0.00E+00	1.34E-01	6.54E-02	1.99E-01
	Std	3.00E-04	1.04E-01	0.00E+00	6.06E-04	6.35E-02	0.00E+00	9.24E-02	5.25E-02	9.58E-04

Table 3 (continued)

Function		MCOA-ML	MCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
DTLZ7	Best	2.73E-01	2.70E-01	0.00E+00	2.74E-01	1.10E-01	2.19E-01	8.43E-02	2.56E-01	2.02E-01
	Ave	2.68E-01	2.04E-01	0.00E+00	2.67E-01	9.29E-02	2.12E-01	1.51E-02	2.15E-01	2.01E-01
	Std	2.41E-03	1.10E-01	0.00E+00	6.69E-03	6.25E-03	5.98E-03	2.58E-02	3.47E-02	1.10E-03
Avg		8	5.25	3.15	5.30	2.40	5.50	3.85	5	6.55
Rank		1	5	8	4	9	3	7	6	2

Best results are bolded

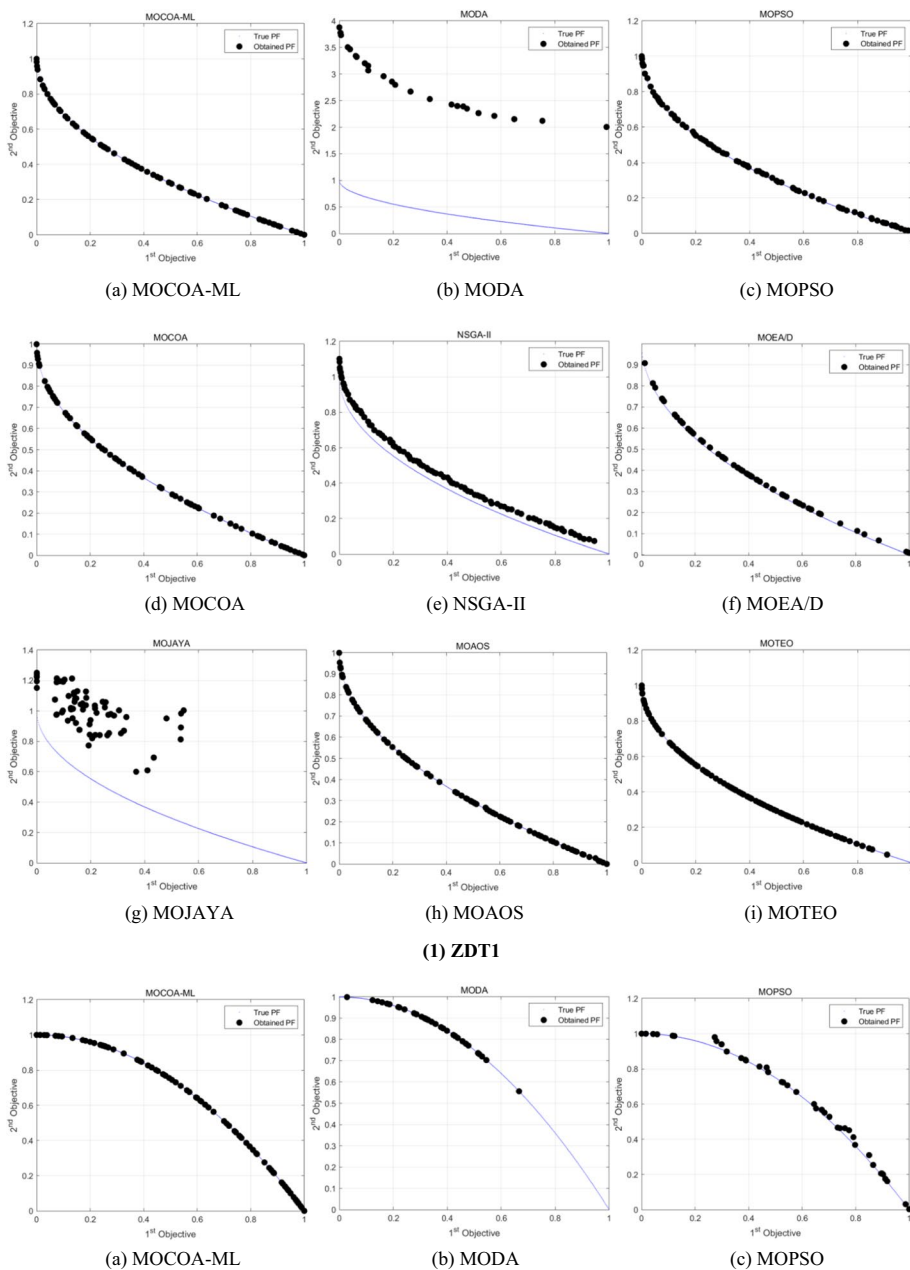


Fig. 6 Pareto frontiers obtained by each algorithm on two-objective test functions

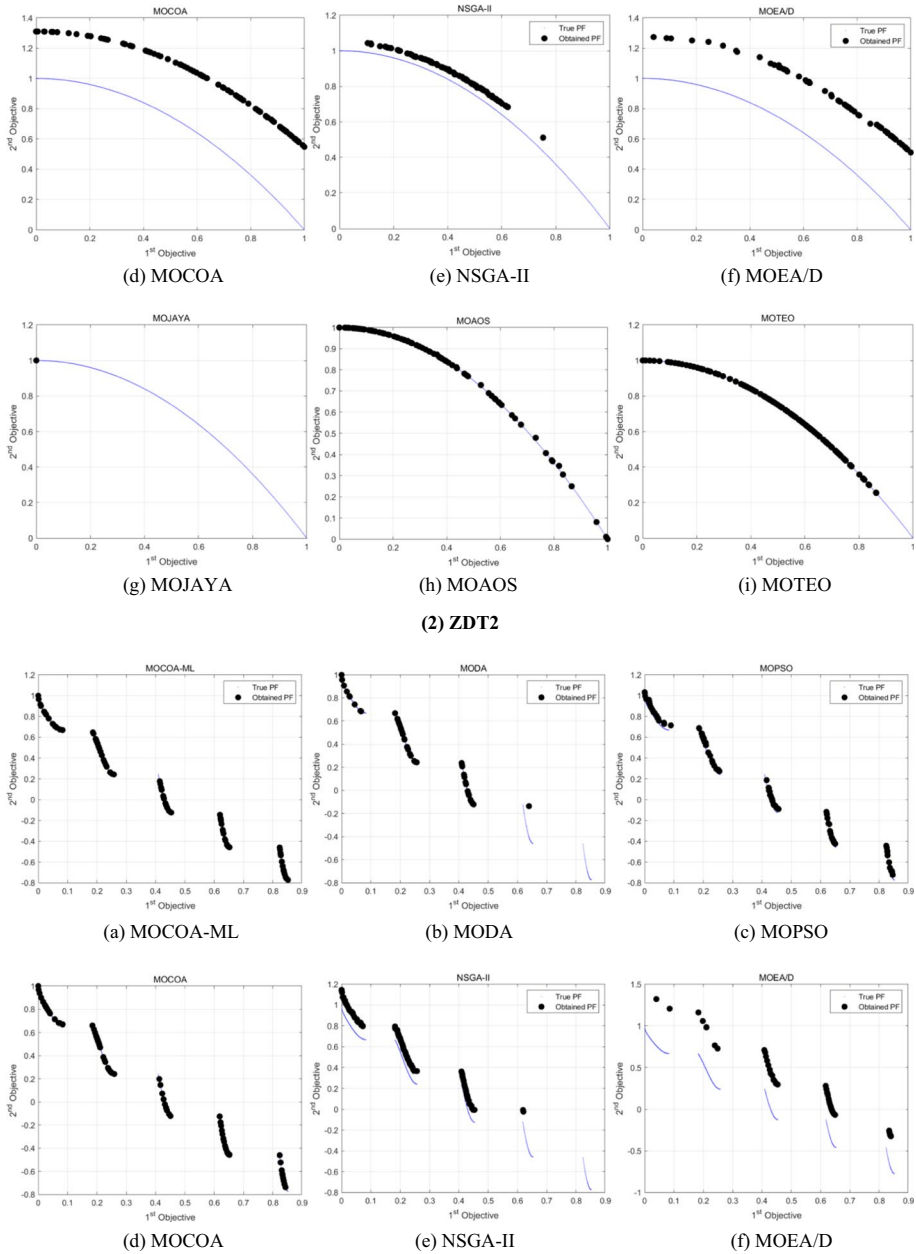


Fig. 6 (continued)

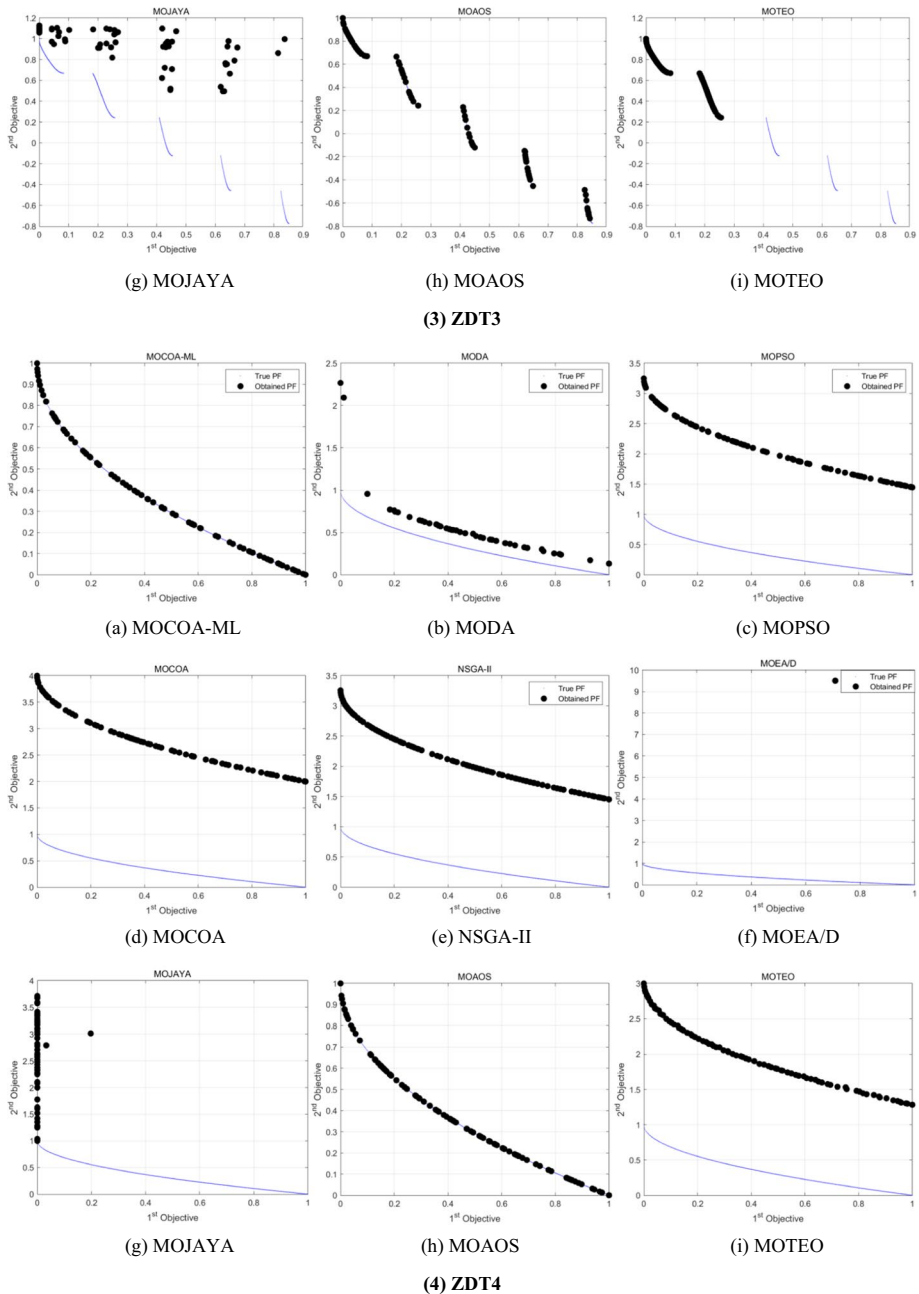


Fig. 6 (continued)

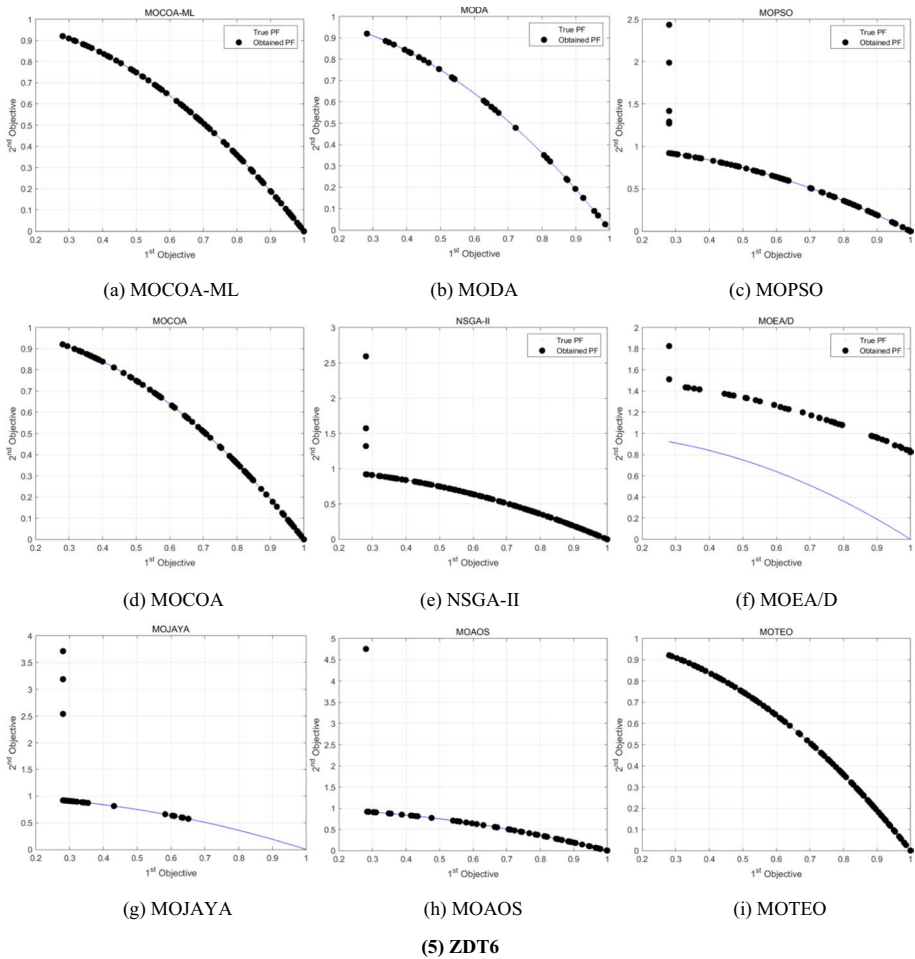


Fig. 6 (continued)

and can be considered as a viable alternative algorithm. It is important to note that in Table 3, the rankings obtained from the Friedman test are inverted due to HV being a positive indicator.

4.2.2 UF test function results and analysis

In this section, MOCO-ML was used to solve the test functions UF1-UF10. MOCO, MODA, MOJAYA, MOPSO, NSGA-II, MOEA/D, MOAOS and MOTE were selected as the comparative algorithms. The best values, average values, standard deviations of IGD and HV obtained by each algorithm, and the average rankings obtained from the Friedman test are recorded in Tables 4, 5. Figure 8 shows the best Pareto front obtained from the experiments. From Tables 4 and 5, it can be observed that MOCO-ML exhibits significantly better convergence and diversity in UF1-UF6, UF8, and UF10, while its performance is slightly worse in UF9. The average rankings obtained from the Friedman test

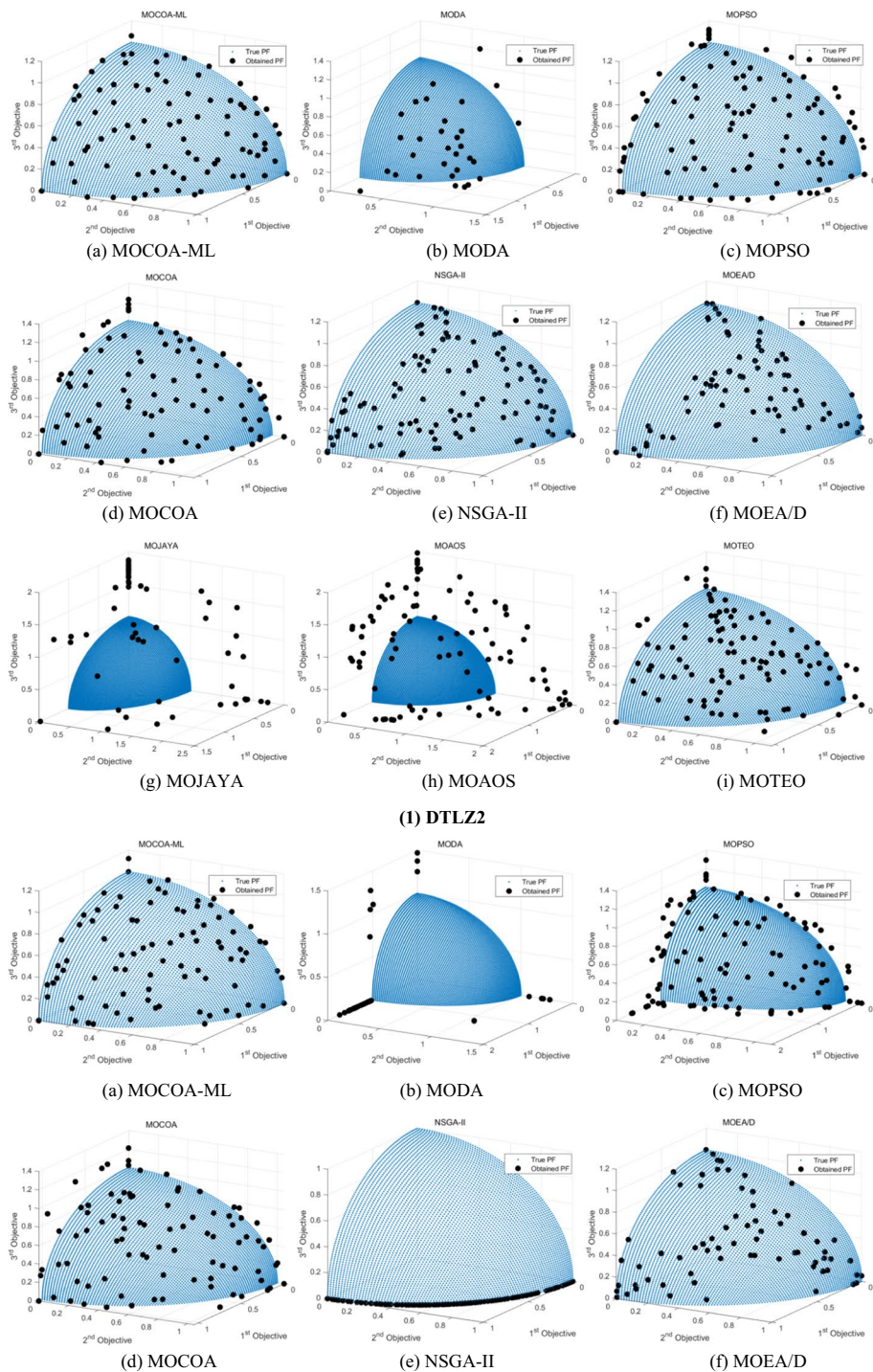


Fig. 7 Pareto frontiers obtained by each algorithm on three-objective test functions

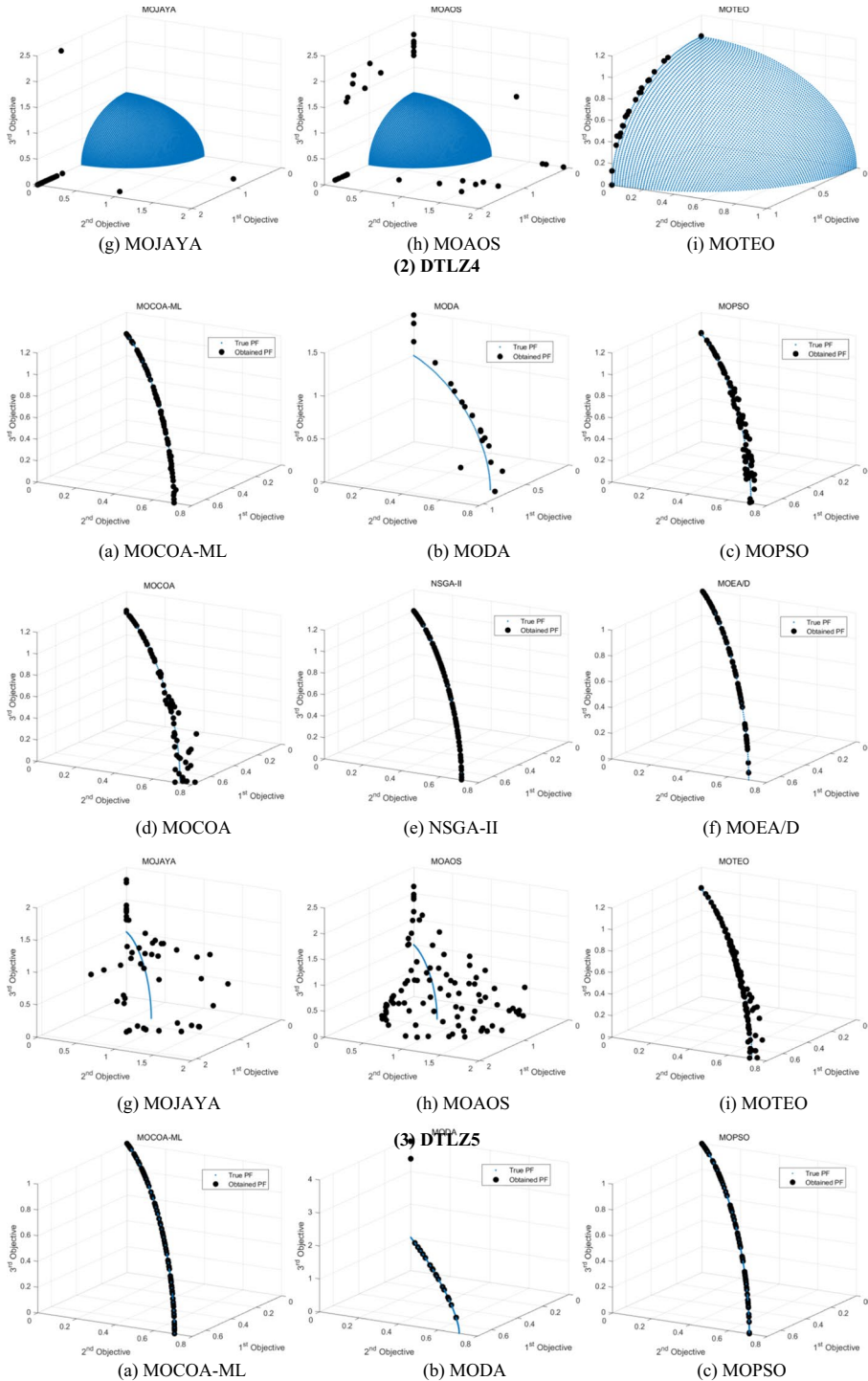


Fig. 7 (continued)

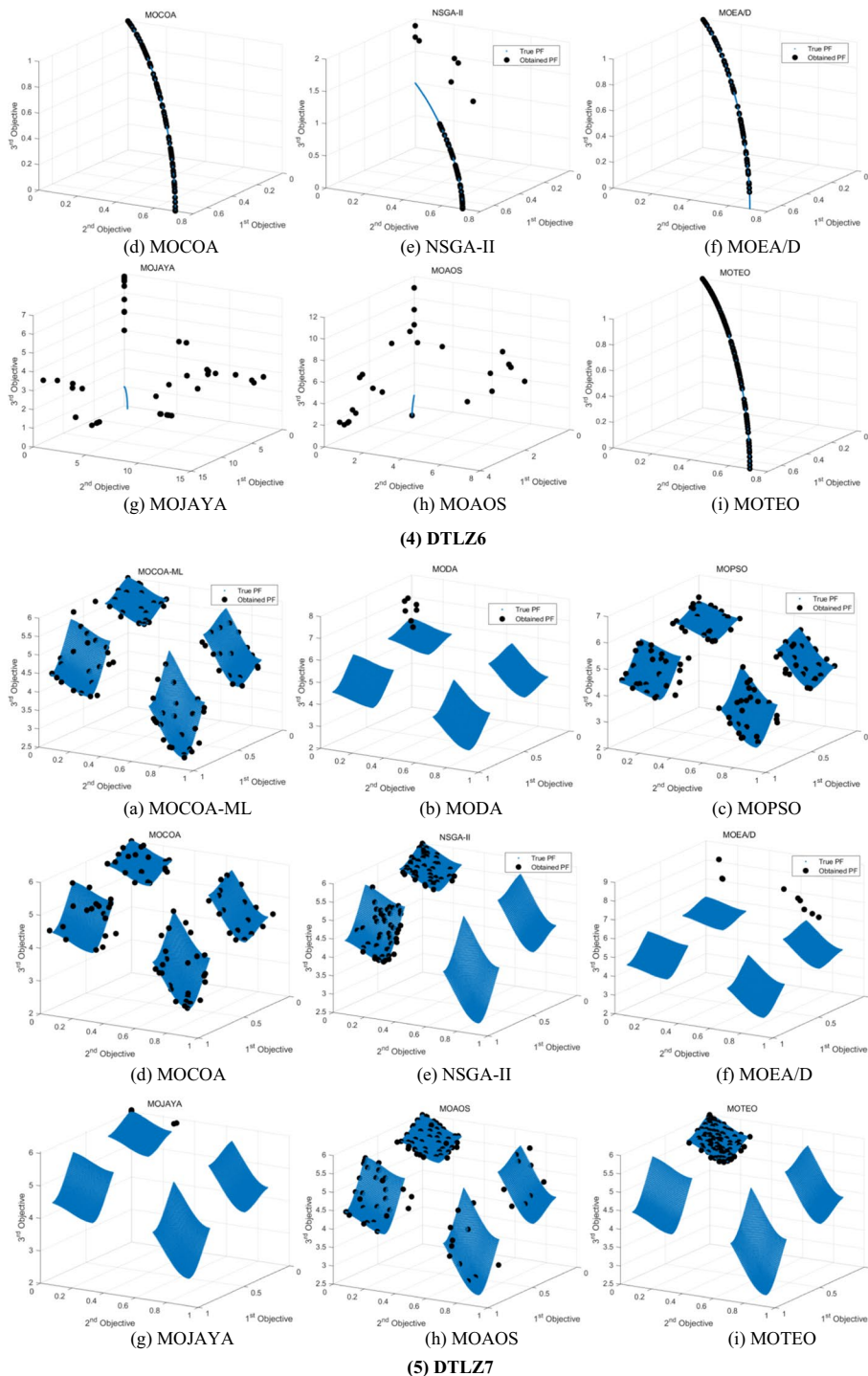


Fig. 7 (continued)

Table 4 Simulation results of test functions (IGD)

Function		MCOA-ML	MCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
UF1	Best	2.79E-03	2.97E-03	3.66E-03	3.25E-03	3.49E-03	3.77E-03	3.60E-03	3.54E-03	3.44E-03
	Ave	3.09E-03	3.22E-03	3.94E-03	3.58E-03	4.01E-03	4.98E-03	5.05E-03	3.74E-03	3.77E-03
	Std	1.47E-04	1.88E-04	2.62E-04	3.09E-04	2.55E-04	8.20E-04	1.30E-03	1.60E-04	2.48E-04
UF2	Best	1.34E-03	1.40E-03	2.34E-03	1.70E-03	2.05E-03	5.61E-03	2.32E-03	1.94E-03	1.25E-02
	Ave	1.45E-03	1.59E-03	3.16E-03	1.82E-03	2.61E-03	1.18E-02	3.57E-03	2.15E-03	1.87E-02
	Std	8.74E-05	1.28E-04	6.17E-04	1.61E-04	3.24E-04	2.89E-03	6.92E-04	1.26E-04	2.78E-03
UF3	Best	6.99E-03	1.08E-02	4.12E-03	9.69E-03	1.25E-02	6.71E-03	1.28E-02	1.36E-02	6.66E-03
	Ave	8.27E-03	1.43E-02	1.03E-02	1.68E-02	1.95E-02	1.04E-02	1.58E-02	1.40E-02	2.27E-02
	Std	8.61E-04	2.22E-03	4.90E-03	4.61E-03	3.10E-03	2.27E-03	2.64E-03	1.41E-04	1.00E-02
UF4	Best	2.71E-03	2.68E-03	3.89E-03	2.80E-03	4.41E-03	3.30E-03	3.48E-03	3.63E-03	2.72E-03
	Ave	2.90E-03	2.94E-03	4.20E-03	3.12E-03	4.68E-03	9.19E-03	3.62E-03	3.91E-03	5.10E-03
	Std	1.38E-04	1.66E-04	2.12E-04	1.35E-04	2.71E-04	4.86E-03	1.22E-04	1.62E-04	3.05E-03
UF5	Best	4.11E-02	4.15E-02	1.76E-01	5.87E-02	1.11E-01	9.07E-02	9.61E-02	1.62E-01	1.35E-01
	Ave	4.17E-02	6.37E-02	2.30E-01	1.18E-01	1.51E-01	1.41E-01	1.94E-01	2.17E-01	1.70E-01
	Std	4.76E-04	2.41E-02	4.10E-02	5.84E-02	4.17E-02	3.25E-02	5.73E-02	3.03E-02	2.55E-02
UF6	Best	3.21E-03	5.84E-03	2.49E-02	8.35E-03	2.05E-02	7.74E-03	1.45E-02	2.67E-02	6.94E-03
	Ave	7.14E-03	9.92E-03	3.16E-02	1.35E-02	2.58E-02	1.32E-02	2.49E-02	3.13E-02	1.28E-02
	Std	2.60E-03	1.77E-03	4.23E-03	5.84E-03	3.20E-03	4.82E-03	6.63E-03	2.82E-03	3.22E-03
UF7	Best	2.32E-03	2.33E-03	3.09E-03	2.34E-03	3.55E-03	1.17E-02	5.00E-03	2.95E-03	2.88E-03
	Ave	2.58E-03	2.45E-03	3.62E-03	2.73E-03	4.62E-03	1.47E-02	1.04E-02	3.24E-03	3.03E-03
	Std	1.66E-04	1.13E-04	3.71E-04	1.91E-04	1.03E-03	3.21E-03	5.04E-03	3.33E-04	8.31E-05
UF8	Best	1.87E-03	1.94E-03	2.56E-03	1.86E-03	2.97E-03	7.44E-03	3.58E-03	2.66E-03	9.03E-03
	Ave	2.08E-03	2.17E-03	3.05E-03	2.17E-03	3.54E-03	8.72E-03	4.69E-03	2.76E-03	9.74E-03
	Std	1.51E-04	2.39E-04	4.76E-04	4.30E-04	3.90E-04	9.95E-04	7.70E-04	1.13E-04	4.15E-04
UF9	Best	1.69E-03	1.72E-03	2.11E-03	1.72E-03	2.05E-03	4.91E-03	2.10E-03	1.83E-03	6.65E-03
	Ave	1.84E-03	1.87E-03	3.25E-03	1.78E-03	2.78E-03	5.13E-03	3.61E-03	2.22E-03	7.05E-03
	Std	8.86E-05	1.11E-04	6.78E-04	4.09E-05	4.15E-04	1.80E-04	9.59E-04	2.27E-04	2.80E-04

Table 4 (continued)

Function		MCOA-ML	MCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
UF10	Best	1.79E-03	2.90E-03	5.45E-03	3.06E-03	3.22E-03	4.51E-03	5.41E-03	2.77E-03	8.77E-03
	Ave	2.03E-03	3.10E-03	1.12E-02	3.30E-03	4.22E-03	7.17E-03	8.30E-03	2.94E-03	9.77E-03
	Std	1.07E-04	9.87E-05	5.17E-03	2.59E-04	6.81E-04	2.78E-03	2.26E-03	1.47E-04	4.32E-04
Avg		1.20	2.45	6.40	3.45	6.20	6.70	6.80	4.80	7
Rank		1	2	6	3	5	7	8	4	9

Best results are bolded

Table 5 Simulation results of test functions (HV)

Function		MCOA-ML	MCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
UF1	Best	6.15E-01	6.14E-01	5.71E-01	5.90E-01	5.67E-01	6.04E-01	5.83E-01	5.71E-01	6.05E-01
	Ave	6.12E-01	6.05E-01	5.53E-01	5.85E-01	5.43E-01	5.76E-01	5.49E-01	5.60E-01	5.98E-01
	Std	2.73E-03	7.71E-03	1.40E-02	3.23E-03	1.44E-02	1.60E-02	3.34E-02	9.69E-03	6.61E-03
UF2	Best	6.79E-01	6.76E-01	6.30E-01	6.70E-01	6.49E-01	5.58E-01	6.28E-01	6.60E-01	5.12E-01
	Ave	6.76E-01	6.72E-01	5.91E-01	6.64E-01	6.37E-01	4.94E-01	5.98E-01	6.54E-01	4.00E-01
	Std	1.99E-03	3.52E-03	3.63E-02	4.85E-03	6.33E-03	3.26E-02	2.07E-02	3.30E-03	5.65E-02
UF3	Best	4.60E-01	3.04E-01	5.58E-01	3.22E-01	1.75E-01	4.31E-01	2.85E-01	1.75E-01	4.47E-01
	Ave	3.92E-01	1.79E-01	3.31E-01	1.66E-01	1.01E-01	3.84E-01	1.65E-01	1.74E-01	1.88E-01
	Std	3.75E-02	5.03E-02	1.23E-01	8.72E-02	3.33E-02	3.85E-02	7.32E-02	3.29E-04	1.58E-01
UF4	Best	3.46E-01	3.38E-01	2.73E-01	3.25E-01	2.55E-01	3.10E-01	3.43E-01	2.85E-01	3.32E-01
	Ave	3.41E-01	3.30E-01	2.60E-01	3.19E-01	2.40E-01	2.10E-01	3.38E-01	2.74E-01	2.61E-01
	Std	3.49E-03	5.61E-03	8.71E-03	4.83E-03	1.09E-02	7.02E-02	2.83E-03	6.29E-03	6.86E-02
UF5	Best	3.21E-01	3.20E-01	0.00E+00	2.40E-01	4.17E-02	1.54E-01	4.76E-02	0.00E+00	2.72E-02
	Ave	3.20E-01	2.27E-01	0.00E+00	9.37E-02	1.21E-02	5.33E-02	5.10E-03	0.00E+00	6.89E-03
	Std	6.29E-04	9.05E-02	0.00E+00	8.55E-02	1.56E-02	4.90E-02	1.50E-02	0.00E+00	1.12E-02
UF6	Best	3.50E-01	2.10E-01	5.06E-03	1.07E-01	3.52E-03	2.55E-01	6.45E-02	0.00E+00	2.05E-01
	Ave	2.97E-01	8.21E-02	5.06E-04	5.24E-02	3.52E-04	1.83E-01	9.27E-03	0.00E+00	1.31E-01
	Std	8.70E-02	5.31E-02	1.60E-03	4.02E-02	1.11E-03	8.60E-02	2.10E-02	0.00E+00	5.80E-02
UF7	Best	5.14E-01	5.12E-01	4.59E-01	5.06E-01	4.36E-01	3.46E-01	4.33E-01	4.67E-01	5.10E-01
	Ave	5.12E-01	5.10E-01	4.41E-01	5.02E-01	3.93E-01	3.09E-01	3.10E-01	4.52E-01	5.05E-01
	Std	1.79E-03	1.66E-03	1.68E-02	3.17E-03	4.08E-02	5.77E-02	9.38E-02	1.60E-02	4.09E-03
UF8	Best	3.93E-01	3.82E-01	3.25E-01	3.95E-01	3.59E-01	1.55E-01	2.75E-01	3.49E-01	1.28E-01
	Ave	3.76E-01	3.64E-01	2.69E-01	3.67E-01	3.12E-01	1.12E-01	2.27E-01	3.40E-01	1.08E-01
	Std	1.27E-02	1.76E-02	3.03E-02	3.32E-02	5.09E-02	2.13E-02	2.47E-02	6.42E-03	1.04E-02
UF9	Best	5.55E-01	5.53E-01	5.00E-01	5.52E-01	5.23E-01	3.33E-01	5.16E-01	5.42E-01	1.38E-01
	Ave	5.37E-01	5.36E-01	4.06E-01	5.46E-01	4.46E-01	3.10E-01	4.27E-01	5.13E-01	1.30E-01
	Std	9.26E-03	1.30E-02	5.45E-02	5.30E-03	5.30E-02	1.71E-02	5.96E-02	1.45E-02	1.03E-02

Table 5 (continued)

Function		MCOA-ML	MCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
UF10	Best	5.53E-01	3.14E-01	8.17E-02	3.24E-01	3.31E-01	1.15E-01	9.13E-02	3.44E-01	1.37E-01
	Ave	5.35E-01	2.97E-01	1.04E-02	3.08E-01	3.14E-01	7.86E-02	1.71E-02	3.36E-01	1.08E-01
	Std	1.58E-02	1.33E-02	2.56E-02	9.11E-03	2.68E-02	2.93E-02	3.50E-02	1.19E-02	1.18E-02
Avg		8.90	6.90	3.25	6.30	3.60	3.80	3.40	4.65	4.20
Rank		1	2	9	3	7	6	8	4	5

Best results are bolded

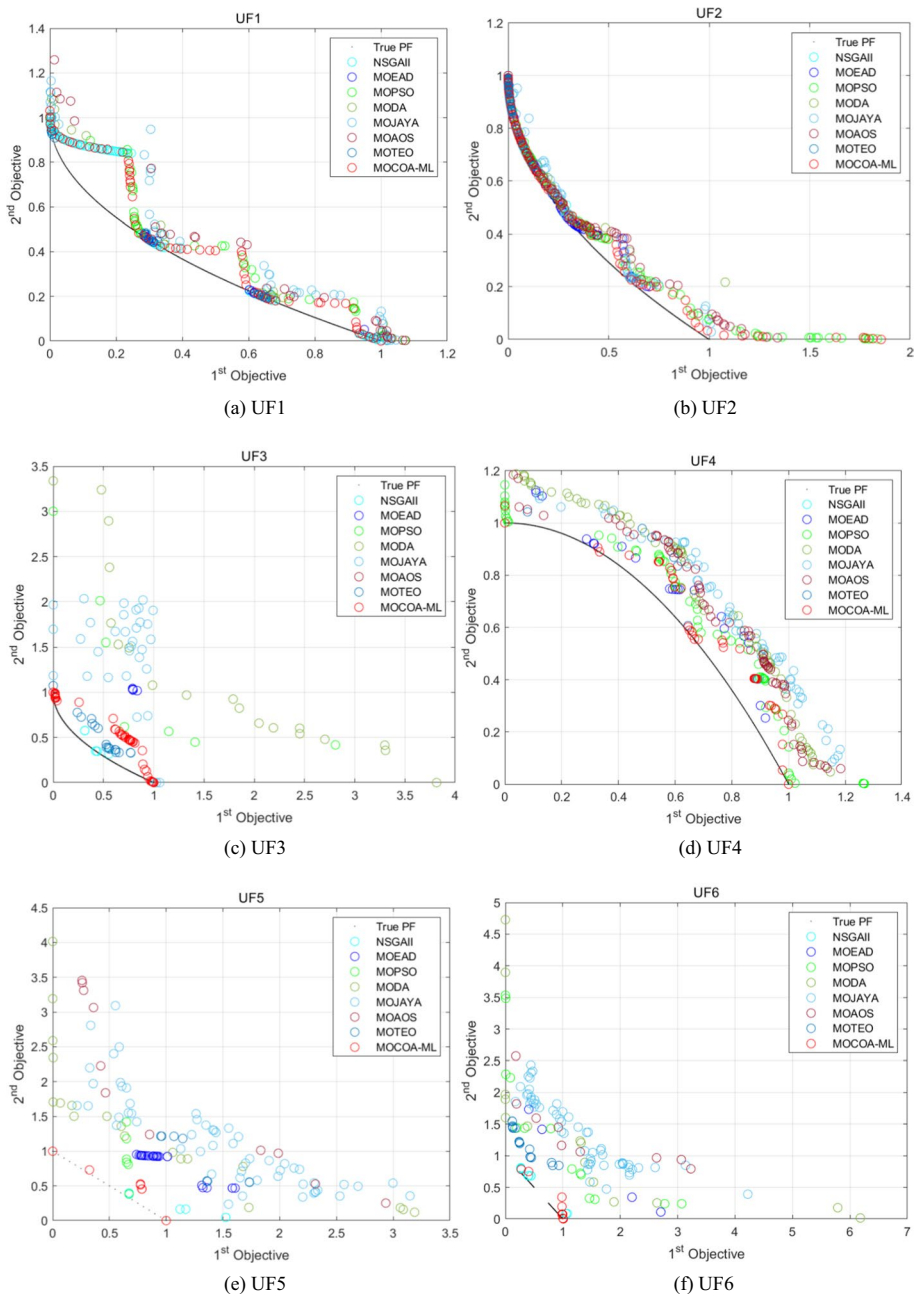


Fig. 8 Pareto frontiers obtained by each algorithm under CEC 2009

show that MOCO-ML achieves the first rank in both IGD and HV indicators among the nine algorithms. MOCO ranks second, followed by MOPSO. These results indicate that MOCO-ML has highly competitive performance on the UF series test functions.

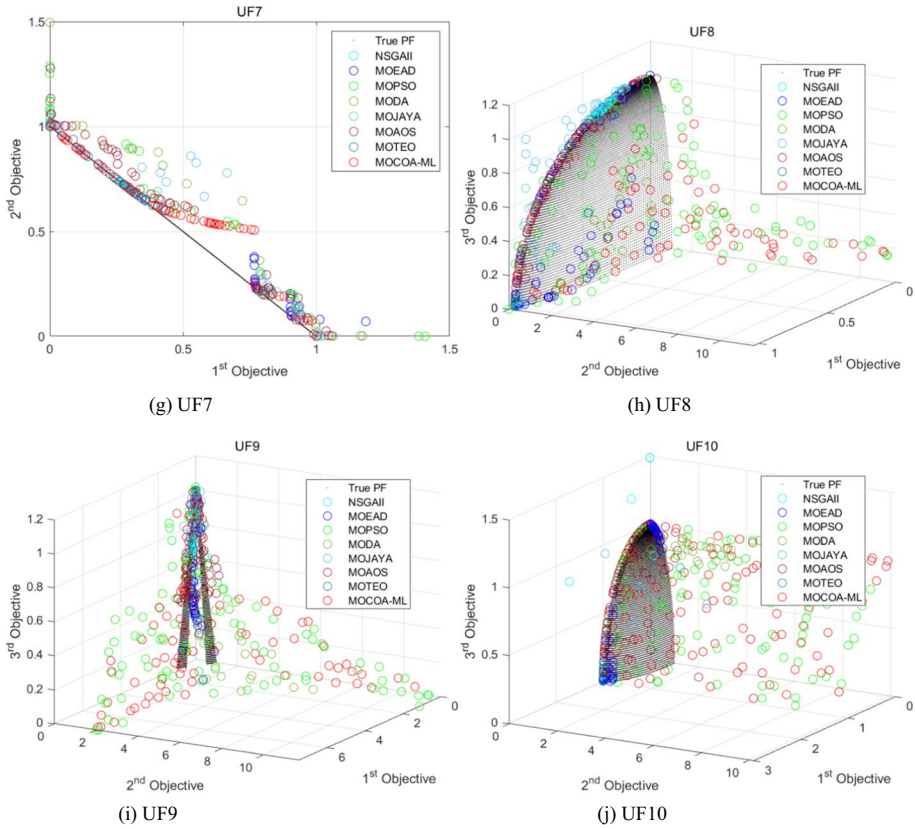


Fig. 8 (continued)

4.3 Population diversity analysis

MOCOAL-ML utilizes a grid mechanism to maintain the diversity of the archive. To validate the effectiveness of this mechanism, the diversity of the archive during the convergence process of MOCOAL-ML on different test functions was analyzed. The diversity curves for some test functions are shown in Fig. 9. The formula for calculating the population diversity div is given by Eq. (39) (Zamani et al. 2021).

$$div = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{d=1}^D (x_{id} - x_{mean,d})^2} \quad (39)$$

where, N represents the number of individuals in the archive, D represents the maximum dimension of the decision variables, x_{id} represents the value of the i -th individual on the d -th dimension, and $x_{mean,d}$ represents the mean value of all individuals in the archive on the d -th dimension.

In the curve shown in Fig. 9, smaller values indicate poorer diversity in the archive, while larger values indicate higher population dispersion. It can be observed that in ZDT1, div decreases continuously. This is because the true Pareto solution set of ZDT1 contains

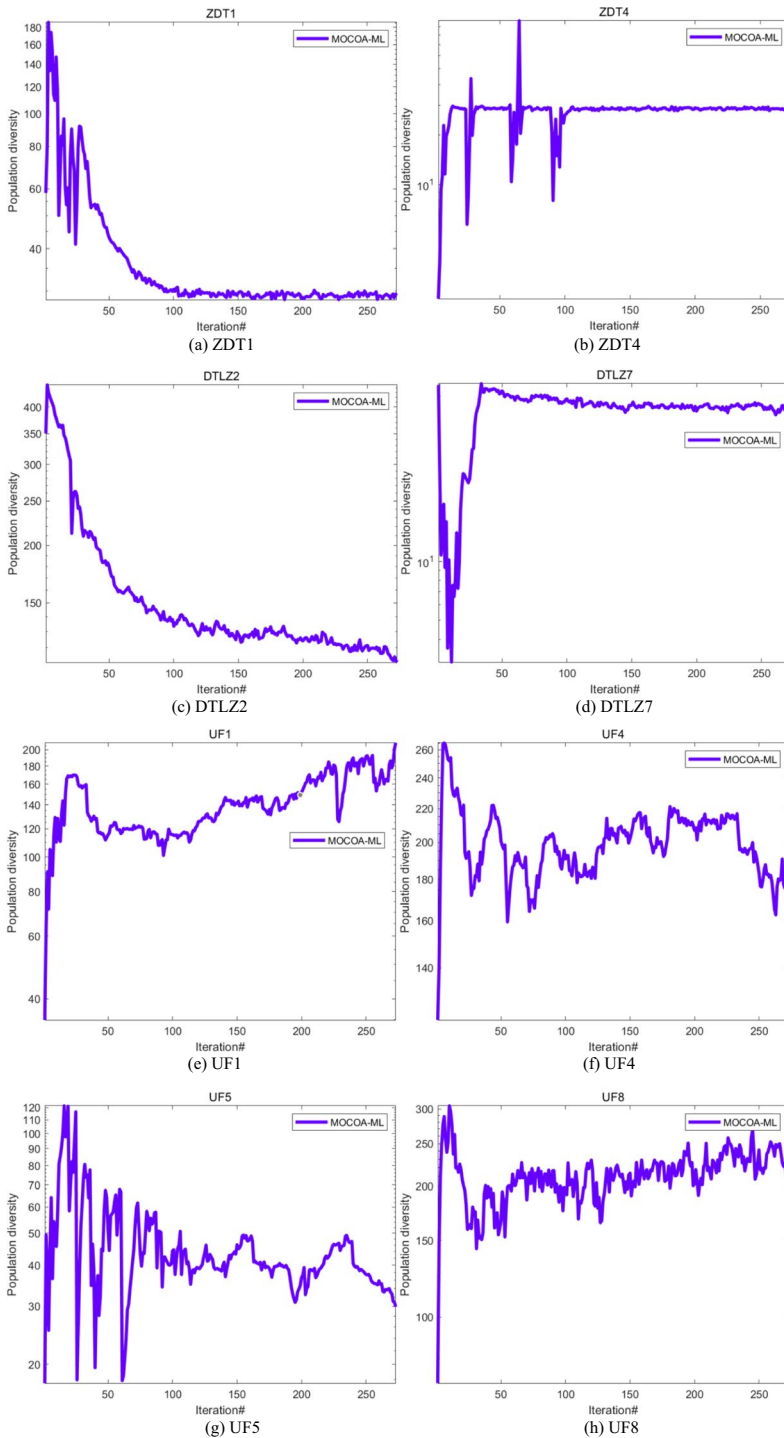


Fig. 9 Iteration curves of population diversity

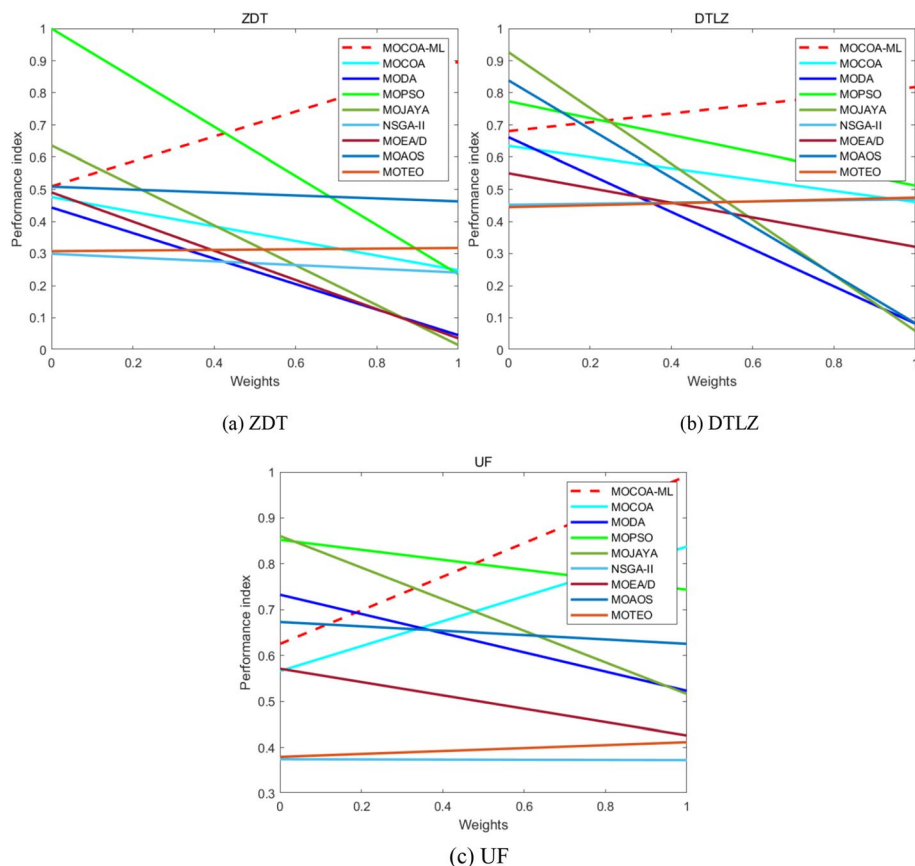


Fig. 10 PI diagram of test functions

a significant number of zeros. During the iterative process, the individuals in the archive gradually approach the true Pareto solution set, resulting in this outcome. On the other hand, in other test functions, the div values remain at a high level, indicating good diversity in the archive. This further demonstrates the effectiveness of the grid mechanism in maintaining diversity.

4.4 Performance index analysis

To compare the performance of different algorithms, they are evaluated by using the Performance Index (PI) (Deep and Thakur 2007). PI is a positive indicator that takes into account the algorithm's runtime. A higher value of PI indicates better algorithm performance. The detailed formulas for calculating PI are given by Eqs. (40 and 41).

$$Ave t_i^j = \frac{1}{H} \sum_{T=1}^H Time(T) \quad (40)$$

Table 6 Results of Wilcoxon signed-rank test on IGD

IGD	MOCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
ZDT1	6.88E-01	9.13E-05	7.01E-03	9.13E-05	9.13E-05	9.13E-05	9.91E-01	1.41E-03
ZDT2	3.78E-02	9.13E-05	1.81E-03	4.37E-05	9.13E-05	9.13E-05	8.46E-01	5.15E-01
ZDT3	7.64E-01	9.13E-05	1.88E-02	9.13E-05	9.13E-05	9.13E-05	9.62E-01	9.13E-05
ZDT4	9.13E-05	9.13E-05	9.13E-05	8.44E-05	9.13E-05	9.13E-05	5.15E-01	9.13E-05
ZDT6	2.36E-01	9.13E-05	1.41E-03	9.13E-05	1.00E+00	9.13E-05	2.91E-04	1.00E+00
DTLZ2	2.29E-03	9.13E-05	9.13E-05	9.13E-05	1.00E+00	9.13E-05	9.13E-05	1.10E-03
DTLZ4	4.45E-02	9.13E-05	9.13E-05	9.13E-05	1.41E-03	1.65E-04	9.13E-05	9.13E-05
DTLZ5	1.65E-04	9.13E-05	9.13E-05	9.13E-05	1.00E+00	2.91E-04	9.13E-05	9.13E-05
DTLZ6	2.26E-02	9.13E-05	4.55E-03	9.08E-05	9.13E-05	9.13E-05	9.13E-05	9.99E-01
DTLZ7	2.70E-02	9.13E-05	9.81E-01	8.63E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05
UF1	6.06E-02	9.13E-05	2.20E-04	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05
UF2	1.56E-02	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05
UF3	9.13E-05	1.37E-01	9.13E-05	9.13E-05	1.56E-02	9.13E-05	6.59E-05	9.73E-03
UF4	3.12E-01	9.13E-05	3.64E-03	9.13E-05	9.13E-05	9.13E-05	9.13E-05	1.41E-03
UF5	5.04E-04	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05
UF6	5.66E-03	9.13E-05	1.81E-03	9.13E-05	2.90E-03	9.13E-05	9.13E-05	5.04E-04
UF7	9.81E-01	9.13E-05	4.45E-02	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05
UF8	2.60E-01	9.13E-05	7.86E-01	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05
UF9	2.85E-01	9.13E-05	9.56E-01	9.13E-05	9.13E-05	9.13E-05	5.04E-04	9.13E-05
UF10	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05
\pm/\equiv	1/12/7	0/19/1	0/17/3	0/20/0	3/17/0	0/20/0	2/16/2	1/17/2

$$PI_i = \frac{1}{Nf} \sum_{j=1}^{Nf} \left(\alpha \times \frac{Minf^j}{Avef_i^j} + \beta \times \frac{Mint^j}{Avef_i^j} \right) \quad (41)$$

where, $Avef_i^j$ represents the average runtime of the i -th algorithm on the j -th test function, H represents the number of runs of an algorithm on a test function, $Time(T)$ represents the time taken by the algorithm in the T -th run, PI_i is the performance index of the i -th algorithm, Nf represents the total number of test functions, $Minf^j$ represents the minimum average error value obtained by all algorithms on the j -th function, $Avef_i^j$ represents the average error value obtained by the i -th algorithm on the j -th function, and $Mint^j$ represents the minimum time obtained by all algorithms on the j -th function. α and β are parameters in the range $[0, 1]$ and have a linear relationship, $\alpha + \beta = 1$.

In this section, α is set to 0, 0.2, 0.4, 0.6, 0.8, and 1. $Minf^j$ and $Avef_i^j$ are used as the average minimum IGD value and the average IGD value obtained by the i -th algorithm, respectively. The PI values of these algorithms on different series of test functions are

Table 7 Results of Wilcoxon signed-rank test on HV

HV	MOCOA	MODA	MOPSO	MOJAYA	NSGA-II	MOEA/D	MOAOS	MOTEO
ZDT1	4.85E-01	3.19E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	1.00E+00	1.41E-03
ZDT2	2.65E-02	9.13E-05	2.91E-04	4.37E-05	9.13E-05	4.37E-05	9.19E-01	5.15E-01
ZDT3	1.92E-01	7.02E-02	5.21E-02	9.13E-05	9.39E-01	9.13E-05	9.97E-01	1.00E+00
ZDT4	5.53E-05	9.13E-05	6.59E-05	7.69E-05	3.19E-05	6.59E-05	7.86E-01	6.59E-05
ZDT6	1.06E-01	9.13E-05	1.10E-03	9.13E-05	1.00E+00	5.53E-05	2.20E-04	1.00E+00
DTLZ2	2.20E-04	9.13E-05	9.13E-05	9.13E-05	1.00E+00	5.75E-01	9.13E-05	5.04E-04
DTLZ4	4.45E-02	8.63E-05	1.23E-04	5.53E-05	1.41E-03	1.81E-03	3.19E-05	5.04E-04
DTLZ5	1.23E-04	9.13E-05	9.13E-05	6.59E-05	1.00E+00	4.55E-01	6.59E-05	9.13E-05
DTLZ6	1.01E-02	3.19E-05	1.37E-01	7.47E-05	3.19E-05	8.93E-05	9.13E-05	9.99E-01
DTLZ7	9.29E-02	3.19E-05	3.67E-01	8.15E-05	9.13E-05	8.93E-05	9.13E-05	9.13E-05
UF1	1.06E-02	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05
UF2	7.01E-03	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05
UF3	9.13E-05	3.20E-02	9.08E-05	8.93E-05	5.15E-01	9.13E-05	9.13E-05	1.45E-02
UF4	2.91E-04	9.13E-05	9.13E-05	9.13E-05	9.13E-05	3.20E-02	9.13E-05	9.13E-05
UF5	1.65E-04	3.19E-05	8.93E-05	8.15E-05	9.08E-05	5.53E-05	3.19E-05	7.47E-05
UF6	8.53E-04	4.37E-05	9.13E-05	4.37E-05	4.55E-01	7.47E-05	3.19E-05	6.06E-02
UF7	1.56E-02	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	9.13E-05	2.20E-04
UF8	6.06E-02	9.13E-05	7.40E-01	2.20E-04	9.13E-05	9.13E-05	1.23E-04	9.13E-05
UF9	5.75E-01	9.13E-05	9.84E-01	9.13E-05	9.13E-05	9.13E-05	8.53E-04	9.13E-05
UF10	9.13E-05	6.59E-05	9.13E-05	9.13E-05	9.13E-05	6.59E-05	9.13E-05	9.13E-05
$\pm/ =$	0/15/5	0/19/1	1/15/4	0/20/0	3/14/3	0/18/2	0/16/4	1/16/3

Table 8 MAE results of IGD index for each algorithm

Algorithms	MAE ZDT	Rank	MAE DTLZ	Rank	MAE UF	Rank
MOCOA-ML	2.532E-03	2	6.533E-04	1	6.581E-03	1
MOCOA	5.015E-03	4	6.818E-04	2	7.412E-03	2
MODA	1.394E-02	7	2.463E-02	9	2.286E-02	9
MOPSO	5.044E-03	5	1.073E-03	3	9.348E-03	3
MOJAYA	2.149E-02	9	7.300E-03	7	1.657E-02	6
NSGA-II	1.299E-02	6	2.284E-02	8	1.464E-02	4
MOEA/D	1.709E-02	8	1.400E-03	4	1.489E-02	5
MOAOS	3.604E-04	1	4.869E-03	6	2.216E-02	8
MOTEO	4.525E-03	3	1.961E-03	5	1.946E-02	7

plotted in Fig. 10. It can be observed that the MOCOA-ML algorithm has a certain advantage in terms of PI values on the test functions in all three series.

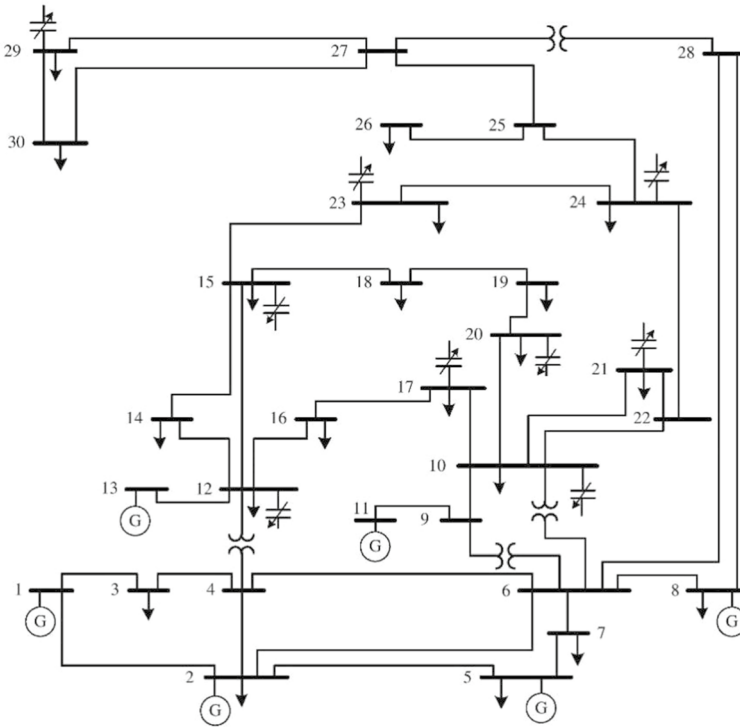


Fig. 11 Standard IEEE 30-bus system

Table 9 Coefficient values of generators of IEEE 30-bus system (Biswas et al. 2018)

Generator	Bus	a	b	c	α	β	γ	ξ	λ
G_1	1	0	2	0.00375	4.091	-5.554	6.49	0.0002	2.857
G_2	2	0	1.75	0.0175	2.543	-6.074	5.638	0.0005	3.333
G_3	5	0	1	0.0625	4.258	-5.094	4.586	0.000001	8
G_4	8	0	3.25	0.00834	5.326	-3.55	3.38	0.002	2
G_5	11	0	3	0.025	4.258	-5.094	4.586	0.000001	8
G_6	13	0	3	0.025	6.131	-5.555	5.151	0.00001	6.667

4.5 Wilcoxon signed-rank test

Wilcoxon signed-rank test is a non-parametric hypothesis test used to compare whether there is a difference in medians between two related samples (Zamani et al. 2022). It is suitable for situations where the data of the two related samples do not follow a normal distribution. In this test, the p-value with a significance level of 95% ($\alpha=0.05$) is calculated. If the p-value is less than or equal to the significance level (0.05), the null hypothesis

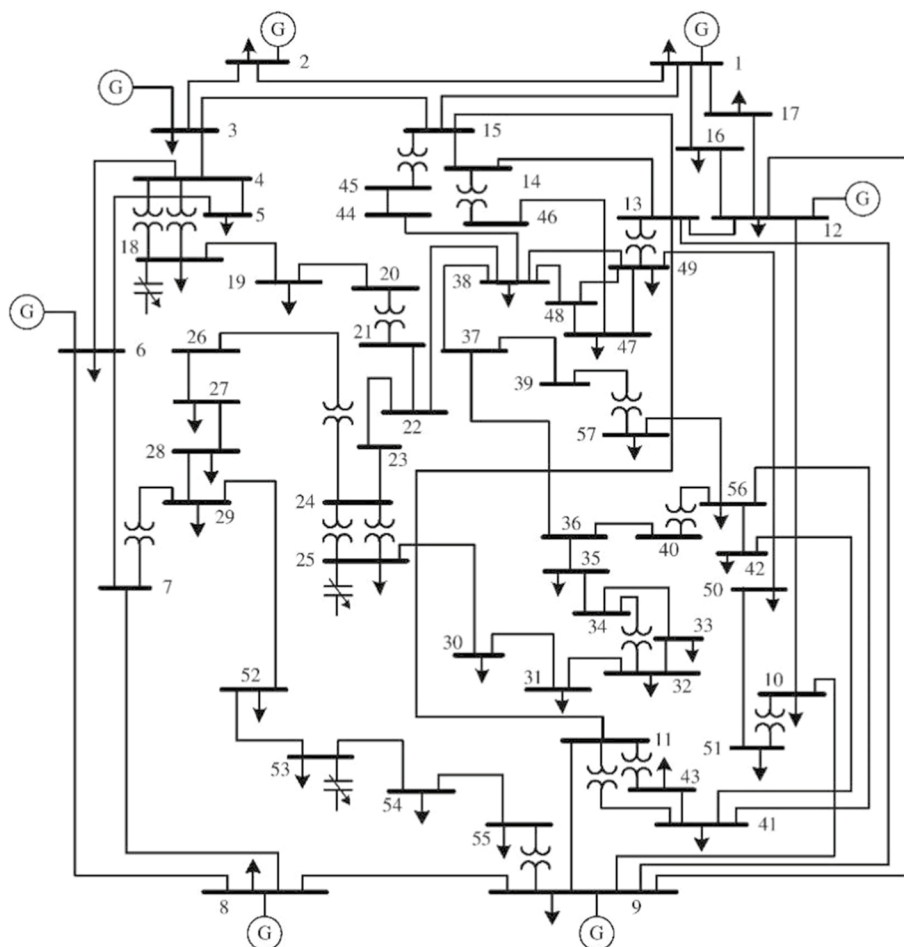


Fig. 12 Standard IEEE 57-bus system

Table 10 Coefficient values of generators of IEEE 57-bus system (Biswas et al. 2018)

Generator	Bus	a	b	c	α	β	γ	ξ	λ
G_1	1	0	20	0.0775795	4.091	- 5.554	6.49	0.0002	0.286
G_2	2	0	40	0.01	2.543	- 6.047	5.638	0.0005	0.333
G_3	3	0	20	0.25	6.131	- 5.555	5.151	0.00001	0.667
G_4	6	0	40	0.01	3.491	- 5.754	6.39	0.0003	0.266
G_5	8	0	20	0.0222222	4.258	- 5.094	4.586	0.000001	0.8
G_6	9	0	40	0.01	2.754	- 5.847	5.238	0.0004	0.288
G_7	12	0	20	0.0322581	5.326	- 3.555	3.38	0.002	0.2

Table 11 Objective function combination

Combination	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
f_{Cost}	✓	✓	✓	✓	✓	✓
f_{Ploss}	✓		✓	✓		✓
$f_{Emission}$		✓	✓		✓	✓
system	IEEE 30	IEEE 30	IEEE 30	IEEE 57	IEEE 57	IEEE 57

Table 12 Setting of algorithm parameters

Algorithm	Parameters
MOCOA	Number of pack (n_packs)=10; Number of coyotes in each pack(n_coy)=10; Number of Grid=10; Archive size=100
MOCOA-ML	Number of pack (n_packs)=10; Number of coyotes in each pack(n_coy)=10; Number of Grid=10; Archive size=100
MOPSO Hazra and Sinha (2011)	Number of Grid=10; Inertia weight(w)=0.5; coefficient (c_1)=1; Social coefficient (c_2)=2; Archive size=100
MOGWO Mirjalili et al. (2016)	Number of Grid=10; Archive size=100
MSSA Mirjalili et al. (2017)	Archive size=100

was rejected. If the p-value is greater than the significance level, we fail to reject the null hypothesis.

Wilcoxon signed-rank tests are conducted to compare the IGD and HV values of MOCOA-ML on different test functions with eight other algorithms. The results are shown in Tables 6 and 7. The “+” sign indicates that the algorithm is significantly better than MOCOA-ML, the “−” sign indicates that the algorithm is significantly worse than MOCOA-ML, and the “=” sign indicates that there is no significant difference between the algorithm and MOCOA-ML. From the results in the tables, it can be concluded that in most cases, MOCOA-ML performs better than the eight compared algorithms.

4.6 Mean absolute error

The mean absolute error (MAE) is used to analyze the IGD indicators obtained by all algorithms to determine the difference between the obtained Pareto frontier and the real Pareto frontier. The calculation formula of MAE is shown in Eq. (42) (Zamani et al. 2021). The obtained results are shown in Table 8. MOCOA-ML ranked second in the ZDT test functions and first in the DTLZ and UF test functions. MOCOA-ML also performs well in MAE analysis.

$$MAE = \frac{\sum_{j=1}^{Nf} |C_j - D_j|}{Nf} \quad (42)$$

where, Nf is the number of functions, C_j is the optimal IGD value of the j -th function, D_j is the optimal IGD value obtained from the j -th function.

Table 13 Solution set obtained in Case 1

Algorithm	MOCOA-ML			MOCOA		
	MIN C	MIN P	COMP	MIN C	MIN P	COMP
P_{G_2} (MW)	47.6877	80	53.2904	47.5353	80	51.5488
P_{G_5} (MW)	21.5551	50	32.3423	20.3466	50	32.9151
P_{G_8} (MW)	23.0780	35	35	19.7809	35	35
$P_{G_{11}}$ (MW)	13.0492	30	27.9989	13.8224	30	30
$P_{G_{13}}$ (MW)	12	40	21.8586	12.3047	40	20.9186
V_{G_1} (p.u.)	1.0821	1.0574	1.0714	1.0859	1.0597	1.0701
V_{G_2} (p.u.)	1.0642	1.0539	1.0588	1.0664	1.0589	1.0593
V_{G_5} (p.u.)	1.0326	1.0394	1.0280	1.0349	1.0432	1.0339
V_{G_8} (p.u.)	1.0359	1.0442	1.0410	1.0408	1.0471	1.0480
$V_{G_{11}}$ (p.u.)	1.0284	0.9874	1.0161	1.0364	0.9888	1.0027
$V_{G_{13}}$ (p.u.)	1.0313	1.0586	1.0568	1.0241	1.0551	1.0515
T_{11} (p.u.)	1.0948	1.0676	1.0373	1.0533	1.1000	1.1000
T_{12} (p.u.)	0.9527	0.9303	0.9608	0.9878	0.9000	0.9000
T_{15} (p.u.)	0.9694	1.0059	0.9860	0.9584	0.9862	0.9714
T_{36} (p.u.)	0.9760	0.9712	0.9851	0.9976	0.9836	0.9788
$Q_{C_{10}}$ (MVar)	3.8768	3.2591	2.8268	4.4249	0.1192	0
$Q_{C_{12}}$ (MVar)	3.5960	4.9238	1.7195	0.6618	0	0.0013
$Q_{C_{15}}$ (MVar)	2.7789	4.6953	2.9395	3.0137	5	0
$Q_{C_{17}}$ (MVar)	3.4381	5	3.2662	2.5474	4.9838	5
$Q_{C_{20}}$ (MVar)	4.6003	4.8923	3.5359	1.7598	4.9574	4.1573
$Q_{C_{21}}$ (MVar)	4.9832	4.9988	4.9652	0.2547	5	5
$Q_{C_{23}}$ (MVar)	1.3731	1.4648	3.6794	4.2175	5	4.5847
$Q_{C_{24}}$ (MVar)	4.1488	4.8924	2.4257	5	0	5
$Q_{C_{29}}$ (MVar)	2.0625	1.7212	2.7909	1.2101	2.1292	0.3694
P_{G_1} (MW)	174.8879	51.5146	118.2429	178.8126	51.5311	118.3008
Fuel cost(\$/h)	800.7669	967.6973	834.6730	800.9718	967.7366	836.5858
P_{loss} (MW)	8.8580	3.1147	5.3332	9.2025	3.1312	5.2834

Optimal values of objective functions are bolded

5 Case study of optimal power flow

In order to verify the performance of the proposed MOCOA-ML to solve the MOOPF problem, simulation studies are carried out in IEEE 30-bus system and IEEE 57-bus system respectively in this section.

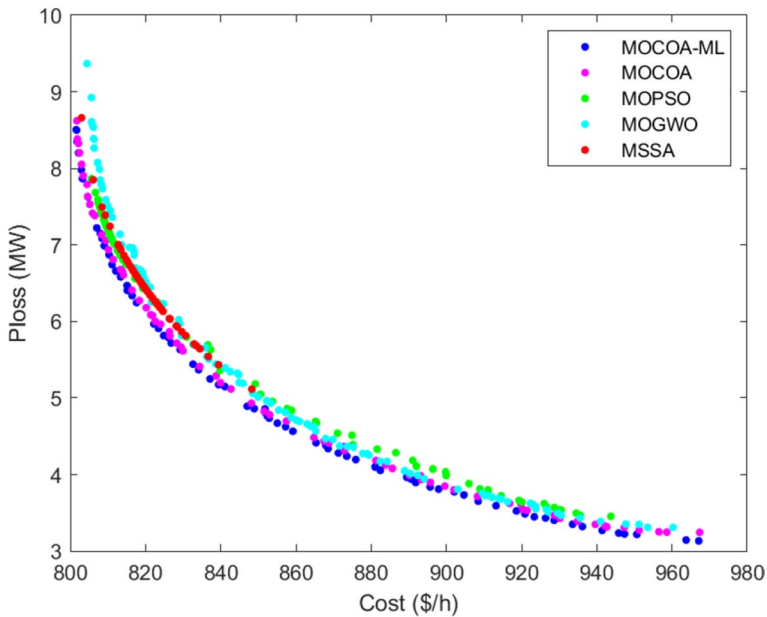


Fig. 13 Pareto frontier obtained by each algorithm in Case 1

Table 14 Compromise solution obtained by each algorithm

Algorithm	Cost(\$/h)	Ploss(MW)
MOCOA-ML	834.6730	5.3332
MOCOA	836.5858	5.2834
MOPSO	836.7859	5.4029
MOGWO	836.7707	5.5058
MSSA	834.8014	5.6435
NSGA-II Sivasubramani and Swarup (2011)	837.416	5.2397
NSGA-III Chen et al. (2019)	835.0259	5.9213
ICA Ghasemi et al. (2014)	848.0544	4.5603
PSO-Fuzzy Liang et al. (2011)	847.01	5.67
EGA Herbadji et al. (2019)	847.011	5.6658

5.1 Test system and parameter setting

IEEE 30-node system is shown in Fig. 11. The system consists of 6 generators, 4 adjustable transformers and 9 capacitors. On the basis of 100 MVA, the active power demand at the load side is 283.4 MW and the reactive power demand is 126.2 MVar. The voltage range

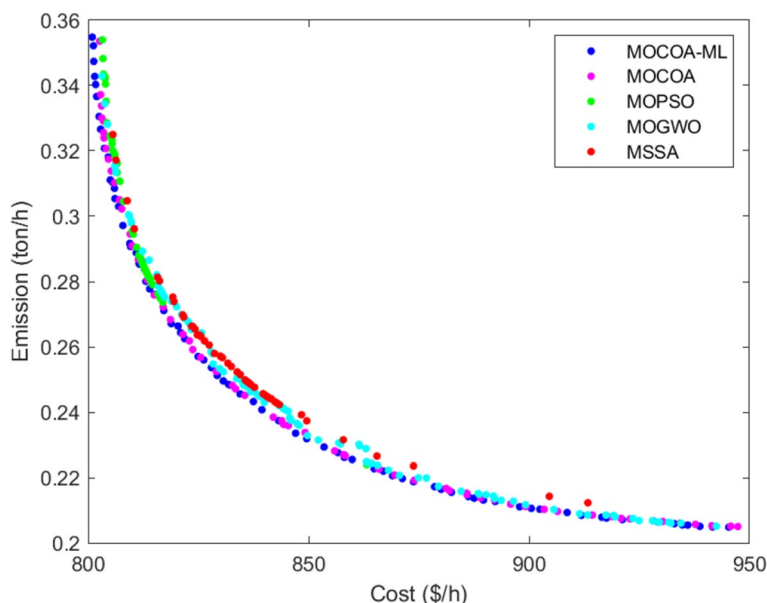


Fig. 14 Pareto frontier obtained by each algorithm in Case 2

of generator bus is 0.95–1.1 p.u.. The normal range of load bus voltage is 0.95–1.05 p.u. Bus 1 is the balance bus. The parameters of the generators are listed in Table 9, including cost coefficients and emission coefficients.

The structure of the IEEE 57 node system is shown in Fig. 12. The system consists of 7 generators, 50 load buses, 80 transmission lines, 17 adjustable transformers and 3 capacitors. The total active power demand at the load side is 1250.8 MW and the total reactive power demand is 336.4MVar. The load bus voltage range is 0.94–1.06 p.u., and the transformer tap range is 0.9–1.1 p.u.. Shunt capacitor maximum reactive power is 30 MVar. The cost and emission coefficients of generators are shown in Table 10. In this experiment, fuel cost, active power loss and total emissions were selected as objective functions and tested in two systems, with a total of 6 cases. The specific case combinations are shown in Table 11.

5.2 Simulation results and analysis

In order to verify the performance of MOCOA-ML, the results were compared with MOCOA, MOPSO, MOGWO and MSSA. For the IEEE 30-node test system, the population of each algorithm is set to 100, and the maximum number of iterations is 300. For the IEEE 57 node test system, the population of each algorithm is set to 100, and the maximum number of iterations is 700. Detailed parameter settings of each algorithm are shown in Table 12. They are independently run 30 times in each case.

Table 15 Solution set obtained in Case 2

Algorithm	MOCOA-ML			MOCOA		
	MIN C	MIN E	COMP	MIN C	MIN E	COMP
P_{G_2} (MW)	48.7222	68.1878	55.9650	48.0114	67.7323	60.1727
P_{G_5} (MW)	21.8651	50	28.3170	21.5952	50	27.8339
P_{G_8} (MW)	22.2842	35	34.4453	21.3186	35	34.9741
$P_{G_{11}}$ (MW)	11.8491	30	27.4468	11.6491	30	26.3913
$P_{G_{13}}$ (MW)	12	40.0000	26.3893	12	40	24.5596
V_{G_1} (p.u.)	1.0796	1.0592	1.0684	1.0868	1.0377	1.0646
V_{G_2} (p.u.)	1.0638	1.0513	1.0570	1.0654	1.0292	1.0466
V_{G_5} (p.u.)	1.0345	1.0314	1.0357	1.0359	1.0021	1.0175
V_{G_8} (p.u.)	1.0404	1.0417	1.0380	1.0413	1.0037	1.0269
$V_{G_{11}}$ (p.u.)	1.0128	1.0687	1.0351	1.0263	1.0604	0.9826
$V_{G_{13}}$ (p.u.)	1.0274	1.0602	1.0506	1.0233	1.0215	1.0059
T_{11} (p.u.)	1.0439	1.1000	1.0733	0.9605	1.0771	1.0532
T_{12} (p.u.)	0.9829	0.9777	0.9381	1.1000	0.9073	1.0312
T_{15} (p.u.)	0.9817	0.9950	0.9907	0.9683	0.9396	1.0174
T_{36} (p.u.)	1.0086	0.9901	0.9903	1.0040	0.9582	1.0003
$Q_{C_{10}}$ (MVar)	5	3.1589	5	2.6280	5	3.3710
$Q_{C_{12}}$ (MVar)	3.5874	0	5	4.4242	0.6941	2.1905
$Q_{C_{15}}$ (MVar)	4.7660	4.7160	1.1163	3.1855	5	4.1140
$Q_{C_{17}}$ (MVar)	5	2.6086	4.8809	4.8762	3.0405	4.3360
$Q_{C_{20}}$ (MVar)	5	1.8158	3.4031	0.8327	5	4.2750
$Q_{C_{21}}$ (MVar)	5	5	0	4.2959	2.0534	2.5529
$Q_{C_{23}}$ (MVar)	3.2267	3.2192	2.1246	2.5456	3.3298	4.4370
$Q_{C_{24}}$ (MVar)	3.1175	3.5379	3.2664	0	4.8975	4.7605
$Q_{C_{29}}$ (MVar)	4.2335	0	3.2616	0.9919	1.9627	0.4602
P_{G_1} (MW)	175.6405	63.5148	116.3380	177.9974	64.0755	115.1995
Fuel cost(\$/h)	800.7411	945.5703	834.2074	800.9233	945.0795	834.2965
Emission(ton/h)	0.3621	0.20485	0.2454	0.3686	0.20488	0.2457

Optimal values of objective functions are bolded

Table 16 Compromise solution obtained by each algorithm

Algorithm	Cost(\$/h)	Emission(ton/h)
MOCOA-ML	834.2074	0.2454
MOCOA	834.2965	0.2457
MOPSO	839.1628	0.2431
MOGWO	835.7207	0.2457
MSSA	836.8744	0.2493
AGSO Daryani et al. (2016)	843.5473	0.2539
ESDE Pulluri et al. (2017)	833.4743	0.2540
BSA Chaib et al. (2016)	835.0199	0.2425
NSGA-II Zhang et al. (2016)	835.59	0.2449

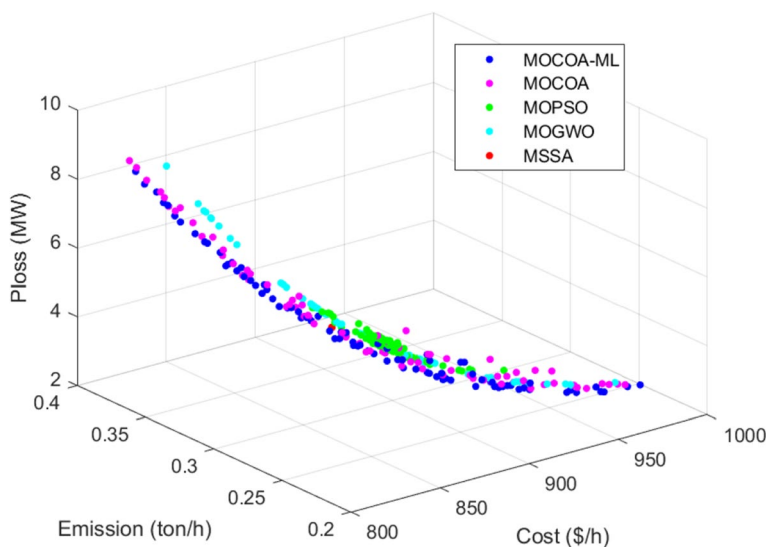


Fig. 15 Pareto frontier obtained by each algorithm in Case 3

5.2.1 Case 1

Fuel cost and active power loss are considered in this case. The optimal solutions and compromise solutions of MOCOA-ML and MOCOA on each objective are shown in Table 13. MIN C represents the solution corresponding to the minimum fuel cost, MIN P represents the solution corresponding to the minimum active power loss, and COMP represents the best compromise solution. The minimum fuel cost and the minimum power loss obtained by MOCOA-ML are 800.7669 \$/h and 3.1147 MW, respectively, and the compromise solutions are 834.6730 \$/h and 5.3332 MW, which are not dominated by the compromise solutions of 836.5858\$/h and 5.2834 MW obtained by MOCOA. The simulation result is shown in Fig. 13, showing the Pareto frontier found by each algorithm. It can be seen that MOCOA-ML has a Pareto solution set with more advanced position, and the result is better than MOCOA. Table 14 compares the compromise solutions of each algorithm. The compromise of MOCOA-ML is superior to MOPSO, MOGWO, MSSA, NSGA-III (Chen et al. 2019), PSO-Fuzzy (Liang et al. 2011) and EGA (Herbadji et al. 2019).

5.2.2 Case 2

Fuel cost and emissions are considered in this case. The simulation result is shown in Fig. 14, showing the Pareto frontier found by each algorithm. It can be seen that MOCOA-ML has a more advanced Pareto solution set, with slightly better results than MOCOA. The optimal solutions and compromise solutions of MOCOA-ML and MOCOA on each objective are shown in Table 15. The minimum fuel cost and minimum emissions obtained by MOCOA-ML are 800.7411 \$/h and 0.20485 ton/h respectively, and the compromise solutions are 834.2074 \$/h and 0.2454 ton/h. Table 16 compares the compromise solutions of each algorithm. The compromise scheme of MOCOA-ML is superior to that of MOCOA, MOGWO, MSSA and AGSO (Daryani et al. 2016), and it has the same dominant level as the compromise scheme of other algorithms.

Table 17 Solution set obtained in Case 3

Algorithm	MOCOA-ML				MOCOA			
	MIN C	MIN E	MIN P	COMP	MIN C	MIN E	MIN P	COMP
P_{G_2} (MW)	49.0086	67.5909	80	60.3497	47.7642	67.7715	80	63.3827
P_{G_5} (MW)	21.4140	50.0000	50	37.0525	20.6283	50.0000	50	36.8097
P_{G_8} (MW)	21.4771	35.0000	35	35	21.9645	35.0000	35	35
$P_{G_{11}}$ (MW)	12.1804	30	30	29.7698	12.9960	30	30	29.7536
$P_{G_{13}}$ (MW)	12	40.0000	40	34.3586	12	40.0000	40	33.1526
V_{G_1} (p.u.)	1.0809	1.0600	1.0593	1.0592	1.0811	1.0612	1.0505	1.0674
V_{G_2} (p.u.)	1.0567	1.0551	1.0541	1.0490	1.0621	1.0541	1.0493	1.0547
V_{G_5} (p.u.)	1.0265	1.0353	1.0361	1.0206	1.0299	1.0323	1.0277	1.0403
V_{G_8} (p.u.)	1.0417	1.0475	1.0394	1.0343	1.0393	1.0332	1.0386	1.0469
$V_{G_{11}}$ (p.u.)	1.0096	1.0164	1.0528	1.0640	1.0887	1.0346	1.0138	0.9699
$V_{G_{13}}$ (p.u.)	1.0484	1.0335	1.0541	1.0333	1.0264	1.0571	1.0626	0.9891
T_{11} (p.u.)	1.1000	0.9831	0.9927	0.9949	1.0548	1.0232	1.0886	1.0886
T_{12} (p.u.)	0.9214	1.0715	1.0322	1.0397	1.0878	0.9764	0.9026	0.9173
T_{15} (p.u.)	0.9792	0.9895	0.9844	0.9950	0.9485	1.0288	0.9781	0.9944
T_{36} (p.u.)	1.0003	0.9721	0.9609	0.9757	1.0191	0.9901	0.9784	0.9872
$Q_{C_{10}}$ (MVar)	4.2247	4.4537	2.1961	0	1.6611	2.2210	0	0.1675
$Q_{C_{12}}$ (MVar)	3.2150	3.1316	1.5841	4.6978	3.3284	3.5205	0.6466	5
$Q_{C_{15}}$ (MVar)	1.9102	4.7593	2.1551	4.3384	0.0538	0.0096	0	3.7618
$Q_{C_{17}}$ (MVar)	2.2557	5	2.1441	0.8411	4.8803	2.0941	0.8304	4.9962
$Q_{C_{20}}$ (MVar)	2.6381	4.0734	4.0030	1.8353	4.7869	2.1753	1.7956	4.9890
$Q_{C_{21}}$ (MVar)	4.5050	3.8487	5	5	2.3472	1.4363	3.0354	4.7117
$Q_{C_{23}}$ (MVar)	3.3162	4.5087	0	3.7527	4.4179	1.5020	4.9957	0.3167
$Q_{C_{24}}$ (MVar)	4.8703	0.9969	2.2774	4.0734	2.6045	4.9884	4.1756	0.9253
$Q_{C_{29}}$ (MVar)	2.5213	3.0827	0.8025	1.8427	5	4.9057	0	0
P_{G_1} (MW)	176.4041	64.1090	51.5407	91.2504	177.1849	63.9732	51.6025	89.8133
Fuel cost(\$/h)	800.8717	944.5804	967.7595	873.9523	801.0532	944.9874	967.9071	874.8882
Emission(ton/h)	0.3642	0.20484	0.2073	0.2191	0.3661	0.20486	0.2073	0.2191
P-loss(MW)	9.0844	3.3000	3.1408	4.3810	9.1379	3.3448	3.2026	4.5120

Optimal values of objective functions are bolded

Table 18 Compromise solution obtained by each algorithm

Algorithm	Cost(\$/h)	Ploss(MW)	Emission(ton/h)
MOCOA-ML	873.9523	4.3810	0.2191
MOCOA	874.8882	4.5120	0.2191
MOPSO	875.7404	4.6815	0.2198
MOGWO	880.8689	4.5705	0.2209
MSSA	856.0247	5.3074	0.2369
WA Anantasate and Bhasaputra (2011)	897.2797	4.6211	0.2175
NSGA-III Chen et al. (2019)	898.5219	4.1419	0.2115
MOEA/D Zhang et al. (2016)	902.54	3.4594	0.2107
NSGA-II Sivasubramani and Swarup (2011)	866.0169	4.5583	0.2165

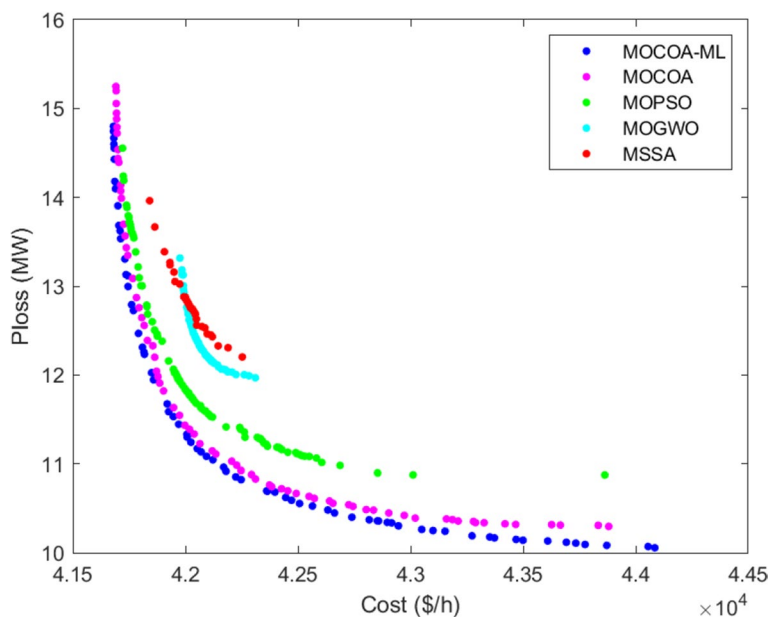


Fig. 16 Pareto frontier obtained by each algorithm in Case 4

5.2.3 Case 3

Fuel cost, emissions and active power loss are considered in this case. The simulation result is shown in Fig. 15, which shows the Pareto frontier found by each algorithm. It can be seen that MOCOA-ML has a more advanced Pareto solution set, and the result is better than MOCOA. The optimal solutions and compromise solutions of MOCOA-ML and MOCOA in each objective are shown in Table 17. The minimum fuel cost, minimum emission and minimum active power loss obtained by MOCOA-ML are 800.8717 \$/h, 0.20484 ton/h and 3.1408 MW respectively, and the compromise solutions are 873.9523 \$/h, 0.2191 ton/h and 4.3810 MW. Table 18 compares the compromise solutions of each algorithm. The compromise of MOCOA-ML is superior to that of MOCOA, MOPSO and MOGWO, and is at the same dominant level as that of other algorithms.

5.2.4 Case 4

In this case, fuel cost and active power loss are considered and simulated in IEEE 57 node system. The simulation result is shown in Fig. 16, showing the Pareto frontier found by each algorithm. It can be seen that MOCOA-ML has a more advanced and more malleable Pareto solution set, and its results are superior to MOCOA. The optimal solutions and compromise solutions of MOCOA-ML and MOCOA in each objective are shown in Table 19. The minimum fuel cost and minimum active power loss obtained by MOCOA-ML are 41,675.44 \$/h and 10.0428 MW respectively, and the compromise solution is 42,146.23 \$/h and 11.0192 MW. Table 20 compares the compromise solutions of each algorithm. The compromise of MOCOA-ML is superior to that of MOCOA, MOPSO, MOGWO, MSSA, BMPSO (Qian and Chen 2022) and MOJFS (Shaheen et al. 2021).

Table 19 Solution set obtained in Case 4

Algorithm	MOCO-ML			MOCO		
	MIN C	MIN P	COMP	MIN C	MIN P	COMP
P_{G_2} (MW)	91.0352	30	71.5206	86.2769	30	72.4361
P_{G_3} (MW)	44.9291	139.8292	63.4934	44.5440	115.8068	58.7525
P_{G_6} (MW)	64.9920	100	99.9807	64.5510	99.5688	100
P_{G_8} (MW)	458.3547	312.1038	358.9955	462.9583	303.3304	352.3207
P_{G_9} (MW)	99.8533	100	99.5027	99.8074	99.9601	100
$P_{G_{12}}$ (MW)	365.9525	410	410	363.9872	409.9514	409.9998
V_{G_1} (p.u.)	1.0715	1.0689	1.0577	1.0571	1.0592	1.0560
V_{G_2} (p.u.)	1.0683	1.0625	1.0542	1.0549	1.0541	1.0526
V_{G_3} (p.u.)	1.0595	1.0602	1.0500	1.0506	1.0558	1.0489
V_{G_6} (p.u.)	1.0617	1.0633	1.0580	1.0592	1.0538	1.0588
V_{G_8} (p.u.)	1.0780	1.0702	1.0679	1.0719	1.0600	1.0689
V_{G_9} (p.u.)	1.0538	1.0505	1.0445	1.0459	1.0391	1.0426
$V_{G_{12}}$ (p.u.)	1.0594	1.0592	1.0475	1.0475	1.0428	1.0417
T_{19} (p.u.)	1.0937	1.0808	0.9841	1.0565	0.9089	1.0960
T_{20} (p.u.)	0.9437	1.0218	1.0730	0.9434	1.0989	0.9848
T_{31} (p.u.)	1.0347	1.0423	1.0319	0.9887	1.0213	1.1000
T_{35} (p.u.)	0.9141	0.9479	1.1000	0.9848	1.0900	0.9763
T_{36} (p.u.)	1.0716	1.0711	1.0238	1.1000	0.9543	1.0991
T_{37} (p.u.)	1.0523	1.0098	1.0157	1.0193	1.0110	1.0202
T_{41} (p.u.)	0.9963	0.9948	0.9962	1.0061	0.9948	0.9942
T_{46} (p.u.)	0.9657	0.9906	0.9570	0.9743	0.9844	1.0150
T_{54} (p.u.)	0.9027	0.9392	0.9000	0.9000	0.9034	0.9521
T_{58} (p.u.)	0.9857	0.9853	0.9762	0.9820	0.9759	0.9778
T_{59} (p.u.)	0.9748	0.9717	0.9661	0.9800	0.9582	0.9593
T_{65} (p.u.)	0.9854	0.9805	0.9727	0.9758	0.9755	0.9695
T_{66} (p.u.)	0.9352	0.9476	0.9366	0.9302	0.9441	0.9194
T_{71} (p.u.)	0.9834	0.9728	0.9819	0.9992	0.9769	0.9640
T_{73} (p.u.)	1.0189	1.0003	1.0101	0.9681	0.9678	0.9553
T_{76} (p.u.)	0.9806	0.9825	0.9844	0.9352	0.9950	0.9997
T_{80} (p.u.)	1.0055	0.9968	1.0020	1.0235	0.9804	0.9824
$Q_{C_{18}}$ (MVar)	0.6017	5.5454	6.1419	0	0.0004	0.4690
$Q_{C_{25}}$ (MVar)	2.6376	8.6312	13.0960	11.2875	11.3191	10.0866
$Q_{C_{53}}$ (MVar)	3.4088	5.4130	7.6298	8.3036	9.8390	8.0241
P_{G_1} (MW)	140.4018	168.9085	158.3252	143.7006	202.3575	168.3859
Fuel cost(\$/h)	41,675.44	44,714.45	42,146.23	41,679.64	44,013.66	42,183.76
P_{loss} (MW)	14.7208	10.0428	11.0192	15.0265	10.1759	11.0961

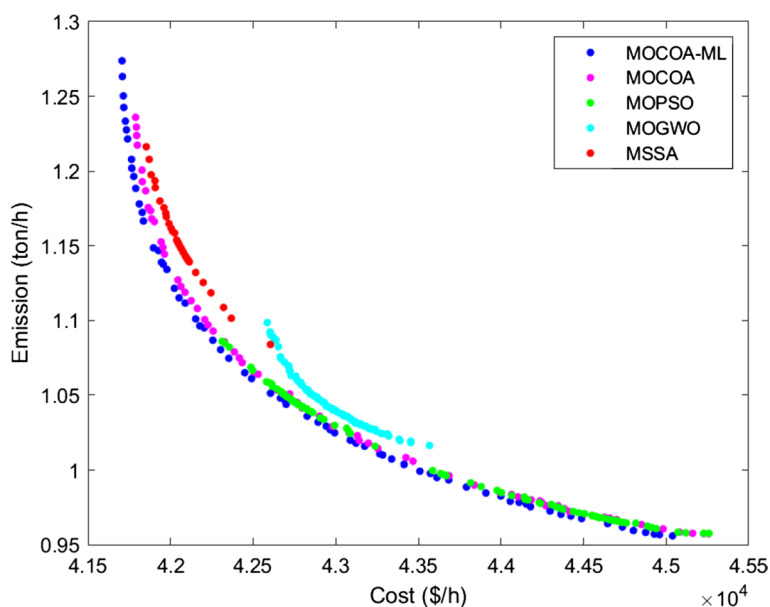
Optimal values of objective functions are bolded

5.2.5 Case 5

Fuel cost and emissions are considered in this case. The simulation result is shown in Fig. 17, showing the Pareto frontier found by each algorithm. It can be seen that the Pareto

Table 20 Compromise solution obtained by each algorithm

Algorithm	Cost(\$/h)	Ploss(MW)
MOCOA-ML	42,146.23	11.0192
MOCOA	42,183.76	11.0961
MOPSO	42,181.35	11.4155
MOGWO	42,070.82	12.2947
MSSA	42,155.31	13.1737
ICA Ghasemi et al. (2014)	42,850.6407	10.289
MICA Ghasemi et al. (2014)	42,834.7597	10.2511
BMPSO Qian and Chen (2022)	42,150.7986	11.1328
MOJFS Shaheen et al. (2021)	42,591.8712	15.1461

**Fig. 17** Pareto frontier obtained by each algorithm in Case 5

solution set of MOCOA-ML has obvious advantages, and the result is better than that of MOCOA. The optimal solutions and compromise solutions of MOCOA-ML and MOCOA in each objective are shown in Table 21. The minimum fuel cost and minimum emissions obtained by MOCOA-ML are 41,698.88 \$/h and 0.9546 ton/h respectively, and the compromise solutions are 42,474.51 \$/h and 1.0632 ton/h. Table 22 compares the compromise solutions of each algorithm. The compromise of MOCOA-ML is superior to that of MOCOA, MOPSO, MSSA, MPIO-PFM (Chen et al. 2020), NSGA-III (Chen et al. 2019) and MOJFS (Shaheen et al. 2021).

Table 21 Solution set obtained in Case 5

Algorithm	MOCOA-ML			MOCOA		
	MIN C	MIN E	COMP	MIN C	MIN E	COMP
P_{G_2} (MW)	99.9373	99.9755	100	99.9834	99.9988	99.9780
P_{G_3} (MW)	41.4842	140	81.4389	40.8131	140	81.2529
P_{G_6} (MW)	91.4852	100	99.9977	98.1770	99.9990	100
P_{G_8} (MW)	451.7860	272.0993	343.8289	442.7961	278.0674	340.7195
P_{G_9} (MW)	90.9582	100	100	96.2657	99.9432	99.9851
$P_{G_{12}}$ (MW)	351.0554	356.1969	365.2878	347.6511	355.7908	369.1975
V_{G_1} (p.u.)	1.0510	1.0722	1.0476	1.0469	1.0641	1.0007
V_{G_2} (p.u.)	1.0464	1.0676	1.0466	1.0437	1.0630	0.9968
V_{G_3} (p.u.)	1.0478	1.0583	1.0410	1.0368	1.0605	0.9892
V_{G_6} (p.u.)	1.0589	1.0553	1.0474	1.0534	1.0508	1.0035
V_{G_8} (p.u.)	1.0711	1.0618	1.0468	1.0655	1.0535	1.0061
V_{G_9} (p.u.)	1.0400	1.0403	1.0222	1.0333	1.0314	0.9774
$V_{G_{12}}$ (p.u.)	1.0414	1.0468	1.0223	1.0290	1.0381	0.9748
T_{19} (p.u.)	0.9672	1.0795	1.0007	1.0469	0.9911	1.0675
T_{20} (p.u.)	1.0357	0.9659	1.0510	1.0999	1.0468	1.1000
T_{31} (p.u.)	0.9988	1.0825	1.0194	1.0719	1.0724	1.0037
T_{35} (p.u.)	1.0260	1.0614	1.0069	1.0118	1.0222	0.9015
T_{36} (p.u.)	1.0114	1.1000	1.0315	1.0989	0.9492	1.0238
T_{37} (p.u.)	1.0312	1.0168	1.0190	1.0688	1.0784	1.0145
T_{41} (p.u.)	1.0083	1.0049	0.9801	1.0392	1.0419	0.9465
T_{46} (p.u.)	0.9713	0.9700	0.9897	0.9877	0.9010	0.9210
T_{54} (p.u.)	0.9233	0.9364	0.9006	0.9736	0.9147	0.9368
T_{58} (p.u.)	0.9667	0.9949	0.9646	0.9650	0.9842	0.9047
T_{59} (p.u.)	0.9681	0.9674	0.9527	0.9514	0.9715	0.9189
T_{65} (p.u.)	0.9755	0.9827	0.9483	0.9689	0.9626	0.9237
T_{66} (p.u.)	0.9380	0.9386	0.9101	0.9002	0.9009	0.9000
T_{71} (p.u.)	0.9616	0.9741	0.9885	0.9491	0.9795	0.9002
T_{73} (p.u.)	1.0041	1.0123	1.0293	0.9515	0.9887	1.0412
T_{76} (p.u.)	0.9658	0.9745	0.9808	1.0940	1.0294	0.9540
T_{80} (p.u.)	0.9879	0.9913	0.9638	1.0274	1.0168	0.9264
$Q_{C_{18}}$ (MVar)	3.4666	5.2396	6.7966	15.9064	2.9703	20
$Q_{C_{25}}$ (MVar)	8.1369	11.0002	10.8232	13.1724	0	0.1437
$Q_{C_{53}}$ (MVar)	7.6256	5.5510	7.9909	5.0102	11.9368	19.6699
P_{G_1} (MW)	139.7307	195.7842	172.8223	141.0471	190.5239	173.6028
Fuel cost(\$/h)	41,698.88	45,192.39	42,474.51	41,727.86	45,102.96	42,559.47
Emission(ton/h)	1.3114	0.9546	1.0632	1.2804	0.9555	1.0646

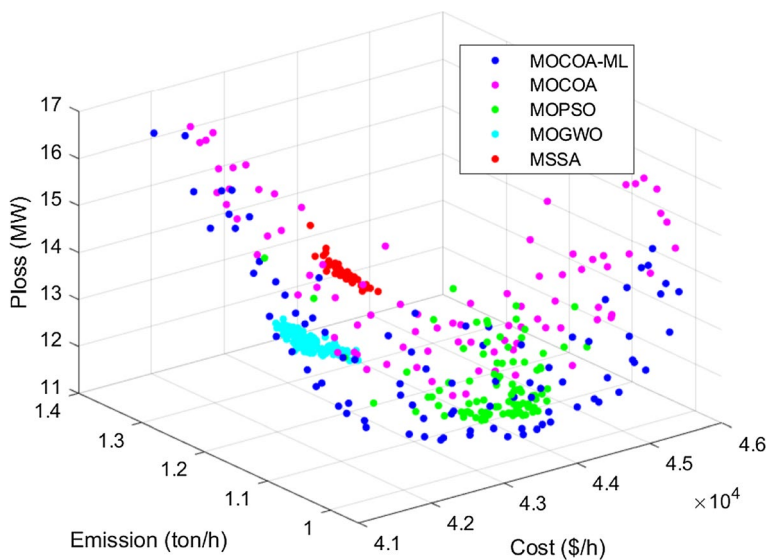
Optimal values of objective functions are bolded

5.2.6 Case 6

Fuel cost, emissions and active power loss are considered in this case. The simulation result is shown in Fig. 18, showing the Pareto frontier found by each algorithm. It can be seen

Table 22 Compromise solution obtained by each algorithm

Algorithm	Cost(\$/h)	Emission(ton/h)
MOCOA-ML	42,474.51	1.0632
MOCOA	42,559.47	1.0646
MOPSO	42,625.42	1.0658
MOGWO	42,118.55	1.1624
MSSA	42,665.47	1.1559
MPIO-PFM Chen et al. (2020)	43,205.8477	1.2386
NSGA-III Chen et al. (2019)	43,323.7670	1.2592
MOJFS Shaheen et al. (2021)	43,888.232	1.2384

**Fig. 18** Pareto frontier obtained by each algorithm in Case 6

that MOCOA-ML has a more advanced Pareto solution set, and the result is better than MOCOA. The optimal solutions and compromise solutions of MOCOA-ML and MOCOA on each objective are shown in Table 23. The minimum fuel cost, minimum emission and minimum active power loss obtained by MOCOA-ML are 41,695.75 \$/h, 0.9556 ton/h and 10.3392 MW respectively, and the compromise solutions are 42,669.53 \$/h, 1.0682 ton/h and 11.0802 MW. Table 24 compares the compromise solutions of each algorithm. The compromise of MOCOA-ML is superior to that of MOCOA, MOGWO, MSSA, MPIO-PFM (Chen et al. 2020) and MOALO (Herbadji et al. 2019), and is at the same dominant level as the compromise of other algorithms.

Table 23 Solution set obtained in Case 6

Algorithm	MOCOA-ML				MOCOA			
	MIN C	MIN E	MIN P	COMP	MIN C	MIN E	MIN P	COMP
P_{G_2} (MW)	93.8442	100	30	99.6217	91.1750	100	33.6875	97.5136
P_{G_3} (MW)	40	140	120.2669	80.8993	46.7192	140	119.2988	82.2691
P_{G_6} (MW)	78.5192	99.9338	99.9671	100	81.8374	100	98.0136	98.6241
P_{G_8} (MW)	459.7178	274.7273	307.6299	317.1069	449.3197	273.6402	324.2483	317.8328
P_{G_9} (MW)	94.7586	99.9998	99.9835	99.7651	98.5632	100	100	99.6736
$P_{G_{12}}$ (MW)	359.3066	350.2921	410	410	354.7121	357.4455	410	410
V_{G_1} (p.u.)	1.0642	1.0546	1.0602	1.0627	1.0604	1.0511	1.0566	1.0471
V_{G_2} (p.u.)	1.0589	1.0539	1.0536	1.0599	1.0545	1.0456	1.0529	1.0449
V_{G_3} (p.u.)	1.0452	1.0550	1.0522	1.0494	1.0490	1.0404	1.0571	1.0449
V_{G_6} (p.u.)	1.0587	1.0527	1.0632	1.0611	1.0641	1.0539	1.0634	1.0545
V_{G_8} (p.u.)	1.0729	1.0536	1.0609	1.0625	1.0676	1.0563	1.0597	1.0666
V_{G_9} (p.u.)	1.0414	1.0250	1.0399	1.0394	1.0430	1.0289	1.0367	1.0320
$V_{G_{12}}$ (p.u.)	1.0353	1.0209	1.0474	1.0445	1.0433	1.0235	1.0369	1.0221
T_{19} (p.u.)	0.9893	0.9561	0.9127	1.1000	0.9667	0.9993	0.9358	1.0960
T_{20} (p.u.)	1.0017	1.0999	1.1000	1.0340	1.0997	1.0941	1.0933	0.9071
T_{31} (p.u.)	1.0182	1.0194	1.0156	1.0443	1.0687	1.0652	0.9815	1.0252
T_{35} (p.u.)	0.9877	0.9517	0.9883	1.0608	1.0972	1.0790	1.1000	0.9000
T_{36} (p.u.)	1.0037	0.9266	0.9715	0.9915	1.0828	1.0809	1.0978	1.0042
T_{37} (p.u.)	1.0130	1.0072	1.0175	1.0430	1.0901	1.0369	1.1000	0.9965
T_{41} (p.u.)	1.0036	1.0015	0.9925	1.0132	1.0527	1.0556	1.0633	0.9978
T_{46} (p.u.)	0.9832	1.00427	0.9826	0.9652	0.9860	0.9833	0.9610	0.9658
T_{54} (p.u.)	0.9152	0.9334	0.9702	0.9114	0.9570	0.9248	0.9267	1.0965
T_{58} (p.u.)	0.9931	0.9700	0.9751	0.9722	0.9763	0.9729	0.9754	0.9677
T_{59} (p.u.)	0.9930	0.9602	0.9828	0.9919	0.9784	0.9847	0.9822	0.9806
T_{65} (p.u.)	0.9606	0.9543	0.9838	0.9958	0.9870	0.9600	0.9689	0.9697
T_{66} (p.u.)	0.9671	0.9151	0.9214	0.9489	0.9163	0.9000	0.9300	0.9025
T_{71} (p.u.)	0.9692	0.9548	0.9685	0.9659	0.9743	0.9608	0.9950	0.9442
T_{73} (p.u.)	0.9828	1.0258	0.9831	1.0215	0.9078	0.9192	1.0412	1.0424
T_{76} (p.u.)	0.9614	0.9870	0.9787	0.9240	1.0749	1.0085	0.9397	0.9020
T_{80} (p.u.)	0.9904	0.9924	0.9924	1.0054	1.0805	1.0591	1.0616	0.9799
$Q_{C_{18}}$ (MVar)	0	9.9830	6.3163	7.6720	0.2296	11.6840	9.1728	0.9397
$Q_{C_{25}}$ (MVar)	7.3369	1.1706	4.2494	10.0313	17.7532	16.5342	15.6382	0
$Q_{C_{53}}$ (MVar)	6.4393	8.2051	9.3938	4.7346	7.4268	7.9233	20	14.9821
P_{G_1} (MW)	140.2227	199.6906	193.2909	154.4862	144.0915	193.8403	176.4072	156.7346
Fuel cost(\$/h)	41,695.75	45,220.43	44,070.80	42,669.53	41,708.66	45,199.00	43,805.79	42,726.38
Emission(ton/h)	1.3501	0.9556	1.1220	1.0682	1.3088	0.9564	1.1277	1.0716
P_{loss} (MW)	15.5701	13.8441	10.3392	11.0802	15.6242	14.1267	10.8631	11.8483

Optimal values of objective functions are bolded

5.3 Evaluation on performance metrics

In this section, two performance indicators, IGD and HV, are selected to evaluate the algorithm. The former is an inverse index, reflecting the difference between the Pareto solution set found by the algorithm and the real Pareto solution set. The latter is a positive

Table 24 Compromise solution obtained by each algorithm

Algorithm	Cost(\$/h)	Ploss(MW)	Emission(ton/h)
MOCOA-ML	42,669.53	11.0802	1.0682
MOCOA	42,726.38	11.8483	1.0716
MOPSO	42,723.47	11.9625	1.0645
MOGWO	42,922.23	13.4192	1.3971
MSSA	42,865.42	12.5382	1.1415
MOFA-CPA Chen et al. (2018)	42,665.51	11.7785	1.5234
MPIO-PFM Chen et al. (2020)	43,133.9896	11.7899	1.5027
MOEA/D-SF Biswas et al. (2020)	42,648.69	11.8862	1.3437
MOALO Herbadji et al. (2019)	42,931.4007	15.0270	1.6349

Table 25 Statistical results of IGD in different cases

Case #		MOCOA-ML	MOCOA	MOPSO	MOGWO	MSSA
Case 1	Mean	4.03e−04	5.89e−04 (3.34e−11)	0.0034 (3.02e−11)	0.0041 (3.34e−11)	0.0057 (3.02e−11)
	Std	2.61e−05	1.44e−04	0.0028	0.0027	0.0028
Case 2	Mean	3.37e−04	3.88e−04 (4.94e−05)	0.0074 (3.02e−11)	0.0016 (3.02e−11)	0.0029 (3.02e−11)
	Std	3.22e−05	4.77e−05	0.0038	0.0011	0.0014
Case 3	Mean	6.10e−04	7.43e−04 (2.03e−09)	0.0071 (3.02e−11)	0.0018 (3.02e−11)	0.0108 (3.02e−11)
	Std	4.68e−05	1.11e−04	0.0024	4.77e−04	0.0020
Case 4	Mean	0.0045	0.0120 (4.44e−07)	0.0262 (9.92e−11)	0.0545 (3.02e−11)	0.0375 (3.02e−11)
	Std	0.0028	0.0086	0.0241	0.0250	0.0075
Case 5	Mean	8.22e−04	0.0017 (9.53e−07)	0.0081 (3.02e−11)	0.0283 (3.02e−11)	0.0183 (3.02e−11)
	Std	2.34e−04	0.0016	0.0050	0.0142	0.0049
Case 6	Mean	0.0031	0.0073 (3.81e−07)	0.0154 (3.69e−11)	0.0304 (3.02e−11)	0.0224 (3.02e−11)
	Std	0.0010	0.0059	0.0056	0.0138	0.0050
Avg		1	2	3.5	4.1667	4.333
Rank		1	2	3	4	5

indicator, reflecting convergence and coverage. This is described in details in the previous section. Since the real Pareto frontier of OPF problem cannot be obtained, all the solution sets obtained by running each algorithm for 30 times are taken as a whole, and the non-dominant solution is found to replace the real Pareto frontier. The experimental results of IGD and the average rankings obtained from the Friedman test are shown in Table 25, and the box diagram is shown in Fig. 19. The results of HV and the average rankings obtained from the Friedman test are shown in Table 26, and the box diagram is shown in Fig. 20. The values within parentheses in Tables 25, 26 are the p-values obtained from the Wilcoxon signed-rank test (at a significance level of 95%). This test compares the results of

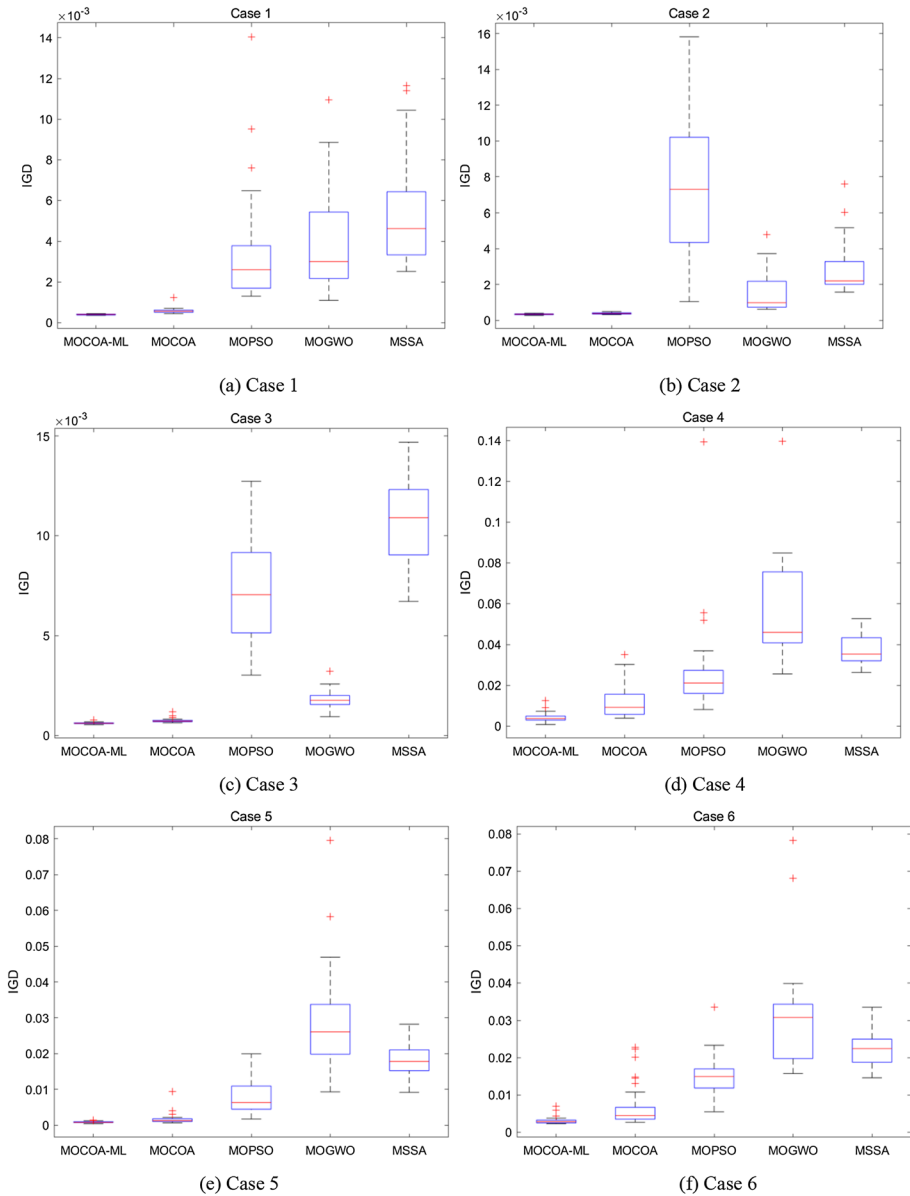


Fig. 19 Boxplots of the IGD

MOCOA-ML with those of other algorithms. The IGD and HV values of the proposed algorithm are better than those of other algorithms. And MOCOA-ML obtained the optimal index in all cases. At the same time, its standard deviation is also smaller, which indicates the stability of the optimization algorithm to a certain extent.

Table 26 Statistical results of HV in different cases

Case #		MOCOA-ML	MOCOA	MOPSO	MOGWO	MSSA
Case 1	Mean	0.1465	0.1449 (3.02e-11)	0.1351 (3.02e-11)	0.1312 (3.02e-11)	0.1252 (3.02e-11)
	Std	1.79e-04	6.61e-04	0.0080	0.0093	0.0071
Case 2	Mean	0.1022	0.1019 (4.08e-11)	0.0901 (3.02e-11)	0.0992 (3.02e-11)	0.0969 (3.02e-11)
	Std	5.6e-05	1.36e-04	0.0080	0.0024	0.0020
Case 3	Mean	0.0718	0.0706 (1.21e-10)	0.0544 (3.02e-11)	0.0647 (3.02e-11)	0.0369 (3.02e-11)
	Std	3.04e-04	7.45e-04	0.0079	0.0026	0.0061
Case 4	Mean	0.0538	0.0485 (6.74e-06)	0.0387 (1.09e-10)	0.0194 (3.02e-11)	0.0283 (3.02e-11)
	Std	0.0024	0.0072	0.0106	0.0128	0.0051
Case 5	Mean	0.0508	0.0499 (7.04e-07)	0.0457 (3.02e-11)	0.0245 (3.02e-11)	0.0315 (3.02e-11)
	Std	2.40e-04	0.0018	0.0031	0.0107	0.0054
Case 6	Mean	0.0186	0.0149 (2.03e-07)	0.0098 (3.34e-11)	0.0046 (3.02e-11)	0.0062 (3.02e-11)
	Std	0.0013	0.0041	0.0031	0.0034	0.0020
Avg		5	4	2.5	1.8333	1.6667
Rank		1	2	3	4	5

6 Conclusions and future works

In this paper, a MOCOA based on hybrid elite framework and Meta-Lamarckian learning strategy is proposed to deal with multi-objective OPF problems with complex constraints. MOCOA-ML retains the main position updating formula of COA, and selects better populations by non-dominated sorting. Additional external archive can store uniform and diverse Pareto solution sets. Combined with the Meta-Lamarckian learning strategy, a local optimizer is established to further improve the performance of the algorithm. The experimental contents and results are as follows. (1) The proposed method was tested on 20 test functions, including ZDT series, DTLZ series and UF series, which were run independently 10 times in each case to record the performance metrics of the algorithm. The results of the test functions show that MOCOA-ML can find true Pareto frontiers on most functions, and the diversity of solution sets is best. (2) Simulation experiments were conducted on 6 OPF cases in IEEE 30-bus system and IEEE 57-bus system with fuel cost, active power loss and emissions as objective functions. The experimental results of OPF demonstrate that MOCOA-ML outperforms other advanced multi-objective optimization algorithms, such as MOPSO, MSSA and MOGWO. It effectively balances convergence performance with ductility, resulting in a superior and more uniformly distributed Pareto solution set.

The OPF problem has complex constraints, so proposing and selecting different constraint processing methods will directly affect the quality of the solution set. This paper only uses the basic penalty function method. The future work will focus on the constraint processing technology. In addition, the integration of new energy into the power

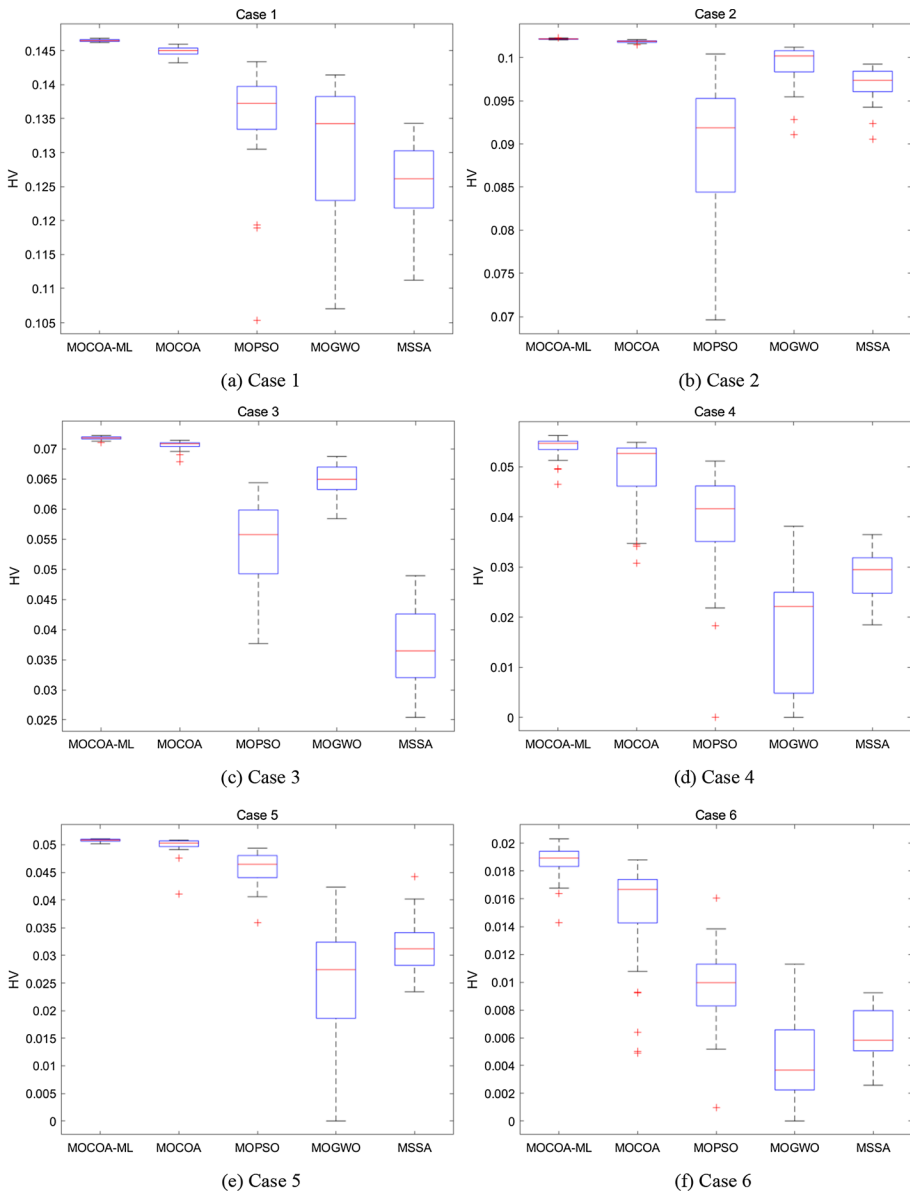


Fig. 20 Boxplots of the HV

grid is the future research trend. In the future work, the modeling of wind turbines and photovoltaic power stations will be carried out, and new energy will be added into the OPF problem and simulation will be carried out so as to deal with the challenges brought by the uncertainty of power and load demand of distributed generation.

Acknowledgements This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJKZ0293), and the Postgraduate Education Reform Project of Liaoning Province (Grant No. LNYJG2022137).

Author contributions J-HZ participated in the data collection, analysis, algorithm simulation, and draft writing. J-SW participated in the concept, design, interpretation and commented on the manuscript. X-YZ, Y-CW, H-MS, YZ and XL participated in the critical revision of this paper.

Funding This study was funded by Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province, LJKZ0293, Postgraduate Education Reform Project of Liaoning Province, LNYJG2022137.

Data availability There are no data available for this paper.

Declarations

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abd El-sattar S, Kamel S, Ebeed M et al (2021) An improved version of salp swarm algorithm for solving optimal power flow problem. *Soft Comput* 25:4027–4052
- Abdelaziz AY, Ali ES, Abdelazim SM (2016) Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems. *Energy* 101:506–518
- Ali ES, Abdelazim SM (2018) Mine blast algorithm for environmental economic load dispatch with valve loading effect. *Neural Comput Appl* 30:261–270
- Ali ES, Abdelazim SM, Abdelaziz AY (2016) Ant lion optimization algorithm for renewable distributed generations. *Energy* 116:445–458
- Ali ES, Abdelazim SM, Balobaid AS (2023) Implementation of coyote optimization algorithm for solving unit commitment problem in power systems. *Energy* 263:125697
- Anantasate S, Bhasaputra P (2011) A multi-objective bees algorithm for multi-objective optimal power flow problem. The 8th electrical engineering/electronics, computer, telecommunications and information technology (ECTI) Association of Thailand-Conference 2011. IEEE, pp 852–856
- Awad NH, Ali MZ, Mallipeddi R et al (2019) An efficient differential evolution algorithm for stochastic OPF based active–reactive power dispatch problem considering renewable generators. *Appl Soft Comput* 76:445–458
- Azizi M, Talatahari S, Khodadadi N et al (2022) Multiobjective atomic orbital search (MOAOS) for global and engineering design optimization. *IEEE Access* 10:67727–67746
- Bentouati B, Khelifi A, Shaheen AM et al (2021) An enhanced moth-swarm algorithm for efficient energy management based multi dimensions OPF problem. *J Ambient Intell Human Comput* 12(10):9499–9519
- Berrouk F, Bouchekara H, Chaib AE et al (2018) A new multi-objective Jaya algorithm for solving the optimal power flow problem. *J Electr Syst* 14(3):165–181
- Biswas PP, Suganthan PN, Mallipeddi R et al (2018) Optimal power flow solutions using differential evolution algorithm integrated with effective constraint handling techniques. *Eng Appl Artif Intell* 68:81–100

- Biswas PP, Suganthan PN, Mallipeddi R et al (2020) Multi-objective optimal power flow solutions using a constraint handling technique of evolutionary algorithms. *Soft Comput* 24(4):2999–3023
- Carpentier J (1962) Contribution to the economic dispatch problem. *Bullet Soc Francoise Electric* 3(8):431–447
- Chaib AE, Boucekara H, Mehasni R et al (2016) Optimal power flow with emission and non-smooth cost functions using backtracking search optimization algorithm. *Int J Electr Power Energy Syst* 81:64–77
- Chen G, Yi X, Zhang Z et al (2018) Solving the multi-objective optimal power flow problem using the multi-objective firefly algorithm with a constraints-prior pareto-domination approach. *Energies* 11(12):3438
- Chen G, Qian J, Zhang Z et al (2019) Applications of novel hybrid bat algorithm with constrained Pareto fuzzy dominant rule on multi-objective optimal power flow problems. *IEEE Access* 7:52060–52084
- Chen G, Qian J, Zhang Z et al (2020) Application of modified pigeon-inspired optimization algorithm and constraint-objective sorting rule on multi-objective optimal power flow problem. *Appl Soft Comput* 92:106321
- Coello CAC, Cortés NC (2005) Solving multiobjective optimization problems using an artificial immune system. *Genet Program Evol Mach* 6(2):163–190
- Coello CAC, Lechuga MS (2002) MOPSO: a proposal for multiple objective particle swarm optimization. *Proc Congress Evol Comput* 2:1051–1056
- Daryani N, Hagh MT, Teimourzadeh S (2016) Adaptive group search optimization algorithm for multi-objective optimal power flow problem. *Appl Soft Comput* 38:1012–1024
- Davoodi E, Babaei E, Mohammadi-ivatloo B (2018) An efficient convexified SDP model for multi-objective optimal power flow. *Int J Electr Power Energy Syst* 102:254–264
- de Souza RCT, de Macedo CA, dos Santos CL et al (2020) Binary coyote optimization algorithm for feature selection. *Pattern Recogn* 107:107470
- Deep K, Thakur M (2007) A new mutation operator for real coded genetic algorithms. *Appl Math Comput* 193(1):211–230
- Dommel HW, Tinney WF (1968) Optimal power flow solutions. *IEEE Trans Power Appar Syst* 10:1866–1876
- El-Sattar SA, Kamel S, El Sehiemy RA et al (2019) Single-and multi-objective optimal power flow frameworks using Jaya optimization technique. *Neural Comput Appl* 31(12):8787–8806
- Farhat M, Kamel S, Atallah AM et al (2022) ESMA-OPF: Enhanced slime mould algorithm for solving optimal power flow problem. *Sustainability* 14(4):2305
- Fonseca CM, Fleming PJ (1993) Genetic algorithms for multiobjective optimization: formulation discussion and generalization. *Icga* 93:416–423
- Ghasemi M, Ghavidel S, Ghanbarian MM et al (2014) Multi-objective optimal power flow considering the cost, emission, voltage deviation and power losses using multi-objective modified imperialist competitive algorithm. *Energy* 78:276–289
- Gomez-Gonzalez M, López A, Jurado F (2012) Optimization of distributed generation systems using a new discrete PSO and OPF. *Electric Power Syst Res* 84(1):174–180
- Hazra J, Sinha AK (2011) A multi-objective optimal power flow using particle swarm optimization. *Eur Trans Electr Power* 21(1):1028–1045
- Herbadji O, Slimani L, Bouktir T (2019) Optimal power flow with four conflicting objective functions using multiobjective ant lion algorithm: a case study of the Algerian electrical network. *Iran J Electr Electron Eng* 15(1):94–113
- Jeyadevi S, Baskar S, Babulal CK et al (2011) Solving multiobjective optimal reactive power dispatch using modified NSGA-II. *Int J Electr Power Energy Syst* 33(2):219–228
- Khan A, Hizam H, Abdul-Wahab NI et al (2020) Solution of optimal power flow using non-dominated sorting multi objective based hybrid firefly and particle swarm optimization algorithm. *Energies* 13(16):4265
- Khunkitti S, Siritratiwat A, Premrudeepreechacharn S (2021) Multi-objective optimal power flow problems based on slime mould algorithm. *Sustainability* 13(13):7448
- Konstantinidis A, Pericleous S, Charalambous C (2018) Meta-Lamarckian learning in multi-objective optimization for mobile social network search. *Appl Soft Comput* 67:70–93
- Kumar S, Jangir P, Tejani GG et al (2022) MOTEQ: a novel physics-based multiobjective thermal exchange optimization algorithm to design truss structures. *Knowl-Based Syst* 242:108422
- Lavaei J, Low SH (2011) Zero duality gap in optimal power flow problem. *IEEE Trans Power Syst* 27(1):92–107
- Li L, Sun L, Xue Y et al (2021) Fuzzy multilevel image thresholding based on improved coyote optimization algorithm. *IEEE Access* 9:33595–33607

- Liang RH, Tsai SR, Chen YT et al (2011) Optimal power flow by a fuzzy based hybrid particle swarm optimization approach. *Electric Power Syst Res* 81(7):1466–1474
- Medina MA, Das S, Coello CAC et al (2014) Decomposition-based modern metaheuristic algorithms for multi-objective optimal power flow—a comparative study. *Eng Appl Artif Intell* 32:10–20
- Meng A, Chen Y, Yin H et al (2014) Crisscross optimization algorithm and its application. *Knowl-Based Syst* 67:218–229
- Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 27:1053–1073
- Mirjalili S, Saremi S, Mirjalili SM et al (2016) Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert Syst Appl* 47:106–119
- Mirjalili S, Gandomi AH, Mirjalili SZ et al (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
- Ong YS, Keane AJ (2004) Meta-Lamarckian learning in memetic algorithms. *IEEE Trans Evol Comput* 8(2):99–110
- Pierezan J, Coelho LDS (2018) Coyote optimization algorithm: a new metaheuristic for global optimization problems. In: 2018 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8
- Ponnusamy A, Rengarajan N (2014) Optimal power flow solution using cuckoo search algorithm. *ARPN J Eng Appl Sci* 9(12):2687–2691
- Pulluri H, Naresh R, Sharma V (2017) An enhanced self-adaptive differential evolution based solution methodology for multiobjective optimal power flow. *Appl Soft Comput* 54:229–245
- Qian J, Chen G (2022) Improved multi-goal particle swarm optimization algorithm and multi-output BP network for optimal operation of power system. *IAENG Int J Appl Math* 52(3):1
- Rizk-Allah RM, Hassanien AE, Slowik A (2020) Multi-objective orthogonal opposition-based crow search algorithm for large-scale multi-objective optimization. *Neural Comput Appl* 32:13715–13746
- Santos AJ, Da Costa GRM (1995) Optimal-power-flow solution by Newton's method applied to an augmented Lagrangian function. *IEE Proc-Gener, Transmission Distrib* 142(1):33–36
- Sayah S, Zehar K (2008) Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. *Energy Convers Manag* 49(11):3036–3042
- Shabanpour-Haghighi A, Seifi AR, Niknam T (2014) A modified teaching–learning based optimization for multi-objective optimal power flow problem. *Energy Convers Manag* 77:597–607
- Shaheen AM, El-Sehiemy RA, Alharthi MM et al (2021) Multi-objective jellyfish search optimizer for efficient power system operation based on multi-dimensional OPF framework. *Energy* 237:121478
- Singh MK, Kekatos V, Giannakis GB (2021) Learning to solve the AC-OPF using sensitivity-informed deep neural networks. *IEEE Trans Power Syst* 37(4):2833–2846
- Sivasubramani S, Swarup KS (2011) Multi-objective harmony search algorithm for optimal power flow problem. *Int J Electr Power Energy Syst* 33:745–752
- Snášel V, Rizk-Allah RM, Hassanien AE (2023) Guided golden jackal optimization using elite-opposition strategy for efficient design of multi-objective engineering problems. *Neural Comput Appl* 35(28):20771–20802
- Taher MA, Kamel S, Jurado F et al (2019) An improved moth-flame optimization algorithm for solving optimal power flow problem. *Int Trans Electr Energy Syst* 29(3):e2743
- Zamani H, Nadimi-Shahraki MH, Gandomi AH (2021) QANA: quantum-based avian navigation optimizer algorithm. *Eng Appl Artif Intell* 104:104314
- Zamani H, Nadimi-Shahraki MH, Gandomi AH (2022) Starling murmuration optimizer: a novel bio-inspired algorithm for global and engineering optimization. *Comput Methods Appl Mech Eng* 392:114616
- Zhang J, Tang Q, Li P et al (2016) A modified MOEA/D approach to the solution of multi-objective optimal power flow problem. *Appl Soft Comput* 47:494–514
- Zhang J, Wang S, Tang Q et al (2019) An improved NSGA-III integrating adaptive elimination strategy to solution of many-objective optimal power flow problems. *Energy* 172:945–957
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 3(4):257–271