



Attention mechanism based adaptive auxiliary task selection for constrained multi-objective optimization and engineering design problems

Qianlong Dang^{1,2} · Xinyu Feng^{1,2} · Linlin Xie^{1,2} · Xianpeng Sun³ · Weijun Wu³

Received: 26 May 2025 / Accepted: 4 October 2025 / Published online: 10 December 2025
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2025

Abstract

Solving constrained multi-objective optimization problems (CMOPs) is a popular research direction in the field of optimization. To address the limitations of using a single strategy in the recent constrained multi-objective evolutionary algorithms (CMOEAs), reinforcement learning (RL) techniques have been applied to solve CMOPs and have achieved remarkable results. However, existing methods use a relatively simplistic state representation to train RL model, which impairs the decision-making capability due to the insufficient training. Moreover, the correlations among tasks are not considered when selecting the strategy, potentially leading to negative knowledge transfer. In this paper, a CMOEA based on attention mechanism assisted auxiliary task selection (AMTCMO) is proposed. Firstly, a population feature fusion method based on self-attention mechanism is used to generate the real-time state of the population for training the RL model. Then, the enhanced state can provide dynamic environment perception ability for the RL model, which can accurately reflect the auxiliary task needed at present. Secondly, a selection strategy based on cross-attention mechanism driven by task association is proposed, which can analyze the benefit of each task. The obtained attention scores and RL collaboratively decide the auxiliary task selection to enhance the reliability. Experimental results show that the proposed AMTCMO demonstrates excellent performance on 4 test suites and 2 real-world engineering problems compared with nine state-of-the-art CMOEAs.

Keywords Constrained multi-objective optimization · Adaptive auxiliary task selection · Reinforcement learning · Self-attention mechanism · Cross-attention mechanism · Engineering design problems

✉ Xianpeng Sun
sunxianpeng@nwsuaf.edu.cn

Qianlong Dang
xidianqldang@163.com

Xinyu Feng
fengxinyu@nwafu.edu.cn

Linlin Xie
xielinlin0213@nwsuaf.edu.cn

Weijun Wu
wuweijun@nwafu.edu.cn

¹ College of Science, Northwest A&F University, Yangling 712100, China

² Joint Laboratory of Algorithm and Simulation for Low Altitude Aircraft, Northwest A&F University, Yangling 712100, China

³ College of Horticulture, Northwest A&F University, Yangling 712100, China

1 Introduction

The study of constrained multi-objective optimization problems (CMOPs) is a significant branch in multi-objective optimization problems (MOPs), and have a wide range of applications in the real world [3, 30], such as speed reducer design problem [1], process synthesis problem [24], power distribution system planning problem [45], and web service location-allocation problems [46]. In general, a simple CMOP can be described as follows [19]:

$$\begin{aligned} &\text{Minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \\ &\text{s.t. } \begin{cases} \mathbf{x} \in \mathcal{D} \\ g_i(\mathbf{x}) \leq 0, i = 1, \dots, s \\ h_j(\mathbf{x}) = 0, j = s + 1, \dots, t \end{cases}, \end{aligned} \quad (1)$$

where $\mathbf{F}(\mathbf{x})$ represents m objectives to be optimized. $\mathbf{x}$ refers to the decision variable with n -dimension, $\mathcal{D}$ is the search

space, $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ represent i -th inequality constraint and j -th equality constraint, respectively.

The k -th constraint violation degree (CV) of $\mathbf{x}$ is calculated as follows [20, 22]:

$$CV_k(\mathbf{x}) = \begin{cases} \max(0, g_k(\mathbf{x})), & k = 1, \dots, s \\ \max(0, |h_k(\mathbf{x})| - \delta), & k = s + 1, \dots, t \end{cases}, \quad (2)$$

where δ is a small positive number used to relax equality constraints. The overall CV of $\mathbf{x}$ is calculated as follows [16]:

$$CV(\mathbf{x}) = \sum_{k=1}^t CV_k(\mathbf{x}), \quad (3)$$

where $\mathbf{x}$ is a feasible solution if and only if $CV(\mathbf{x}) = 0$.

Solving CMOPs involves finding optimal solutions that balance multiple conflicting objective functions while satisfying a set of constraints. Therefore, the key to solving CMOPs is to obtain a set of feasible Pareto optimal solutions, which is mapped to the objective space to be the constrained Pareto front (CPF). Over the past decades, many constraint handling techniques (CHTs) have been proposed and embedded into constrained multi-objective evolutionary algorithms (CMOEAs). They mainly focus on balancing the conflict between constraints and objectives, which is the main difficulty of solving CMOPs. Deb et al. [11] have proposed the constraint dominance principle (CDP), which assigns priority to constraints. The random sorting (SR) [17] method uses a probability function to assign priority to constraints and objectives. The penalty function method [57] converts the CV into the penalty term of the objective function, thereby transforming CMOP into MOP to solve. To combine the advantages of multiple CHTs, many advanced algorithms adopt the multi-stage optimization method [27, 50]. Fan et al. [13] have proposed a two-stage CMOEA, which sequentially prioritizes objectives and constraints in different stages of the algorithm. However, when such algorithms deal with complex problems, the improper balance between constraints and objectives will lead to the loss of convergence and diversity. Specifically, if too much attention is paid to constraint satisfaction may lead the algorithm to fall into the local optima, while excessive emphasis on objective optimization may destroy the constraint dominance relationship and cause the population to diverge in the infeasible regions [66].

To address the limitation of using single CHT and parameters of stage switching, the concept of evolutionary multitasking optimization (EMT) is first introduced to solve CMOPs in [42], and has been widely applied in subsequent researches. Similar to multitasking optimization (MTO), EMT treats multiple related CMOPs as parallel tasks. The main task corresponds to the original CMOP, while auxiliary tasks represent the sub-problems after simplifying the constraints

and employ different strategies. Each task corresponds to a population and the main task achieves optimal performance through knowledge transfer in the process of evolution. Qiao et al. [42] have proposed EMCMO based on EMT. The auxiliary task they constructed ignores the constraints of the original problem and can help the main task converge quickly. In their subsequent work, CMEGL [38], global auxiliary task and local auxiliary task have been constructed to maintain global and local diversity, respectively. The embedding of the EMT framework improves the performance of the algorithm in dealing with complex constraints, but it still exhibits some shortcomings. When the number of auxiliary tasks is insufficient, the algorithm struggles to balance feasibility, convergence, and diversity. However, excessive number of auxiliary tasks may lead to computational resource waste due to parallel multitasking [5, 56].

Reinforcement learning (RL) techniques, which enable agent to interact with environment and learn optimal policies to perform, have recently been applied to address CMOPs to mitigate limitations in the paradigm of EMT [8, 15]. Wang et al. [53] have developed an adaptive CHT selection method based on deep reinforcement learning. The most suitable CHT is selected according to the preference of different problems and the population needs of different evolutionary stages. Ming et al. [33] have integrated RL into the EMT framework, and designed auxiliary task selection methods based on Q-learning (QL) and deep Q-learning (DQL), respectively. QL and DQL are trained to select the most suitable auxiliary tasks to help the main population evolve according to the population state. While the application of RL alleviates the challenge of the limited number of auxiliary tasks in EMT frameworks, they still exhibit critical defects. The state representation of many RLs is relatively simple, making it difficult to capture complex population characteristics. Insufficient state input will affect model training and policy selection. Additionally, similar to the EMT framework, the methods based on RL rarely consider the correlation among tasks. This would undermine the reliability of task selection and the efficiency of knowledge transfer across tasks.

Based on the above considerations, a CMOEA based on attention mechanism assisted auxiliary task selection (AMTCMO) is proposed. The main contributions are as follows:

- (1) A hybrid DQL model enhanced by attention mechanism is proposed. Specifically, the population state fusion method based on self-attention mechanism is used to dynamically fuse multi-dimensional features of convergence, feasibility, and diversity of the main population. The mechanism adaptively integrates the population state information by calculating the correlation weights

among the features to generate a discriminative real-time state. During DQL model training, the enhanced state serves as input, while the selection of auxiliary population is treated as action, and the improvement of the main population state is used as the reward. The enhanced state not only improves the training efficiency of the deep Q-network (DQN), but also precisely reflects the most beneficial auxiliary population required by the main population at different evolution stages.

- (2) A selection strategy based on cross-attention mechanism driven by task association is proposed, which is embedded in the EMT framework to help the algorithm adaptively select auxiliary tasks. The cross-attention mechanism calculates the attention scores between the main task and the auxiliary task to analyze the potential correlation among tasks, and the DQL model uses the DQN to evaluate the benefits of task selection. The selection of auxiliary tasks is collaboratively decided by attention scores and DQL. Attention scores can reflect the attention of the main task to each dimension of the auxiliary task feature, helping the main task filter auxiliary tasks from the task pool that complement the current search blindness. The cooperation of attention scores and DQL utilizes the correlation between tasks, making task selection more reasonable and accurate.

The remainder of this paper is structured as follows. Section 2 introduces the research background and related work. Section 3 describes the motivation of this paper and presents the proposed AMTCMO in detail. Section 4 validates the performance of the algorithm through experiments. Section 5 summarizes this paper and discusses potential future research directions.

2 Related work

2.1 CMOEAs based on multi-stage optimization

Single CHT can not deal with complex problems, and has a risk of local convergence. Many researchers have proposed multi-stage CMOEAs that use different CHTs in different evolutionary stages. Fan et al. [13] proposed the push-pull search framework (PPS). In the push stage, the population is rapidly pushed to the unconstrained Pareto front (UPF) without considering any constraints. In the pull stage, the population is gradually pulled back to the CPF using an improved epsilon constraint handling method. To avoid the difficulty of considering constraints all at once during the switching stage, Ma et al. [31] proposed a UPF-based constraint prioritization strategy. At different stages of evolution, constraints are considered one by one according to their

priority. Liang et al. [27] divided the evolutionary process into learning stage and evolutionary stage. In the learning stage, two auxiliary populations are constructed to approach CPF and UPF respectively and learn the positional relationship between them. In the evolutionary stage, specific strategies are set to guide the populations based on the relationships learned in the previous stage. Bao et al. [2] used a two-population cooperative framework and divided the evolutionary process into two stages: exploration and exploitation. In the exploration stage, the two populations quickly explore the search space. In the exploitation stage, the two populations collaborate to explore the CPF using promising solutions archived from the previous stage. To satisfy the balance between objective optimization and constraint satisfaction, Tian et al. [50] developed a two-stage algorithm based on population state. During the evolutionary process, the priority of constraint and objective is changed by adaptively switching stages according to the state of main population. Yu et al. [60] proposed a purpose-oriented two-stage algorithm. In the first stage, the population ignores constraints to balance convergence and diversity. In the second stage, some promising infeasible solutions are used to enhance the diversity. Wang et al. [54] proposed a three-stage framework for diversity enhancement. At different evolution stages, convergence, diversity, and feasibility are prioritized and specific strategies are designed to additionally enhance diversity.

2.2 CMOEAs based on evolutionary multitasking

Due to the difficulty of controlling the parameters by using different CHTs in stages, most of the popular CMOEAs adopt the EMT paradigm. Qiao et al. [42] first developed a constrained multi-objective optimization framework based on EMT. The optimization of a CMOP is transformed into two related tasks. The main task is used to solve the original CMOP, and the auxiliary task is used to solve the MOP after ignoring the constraints. The useful information provided by the auxiliary task can help the main task across infeasible regions to search for CPF. To prevent the negative impact of knowledge transfer, Tian et al. [49] presented a paradigm of weak co-evolution. The auxiliary task does not consider constraints and the degree of cooperation between the main and auxiliary task is reduced. Liu et al. [29] proposed a bidirectional coevolutionary strategy, where the main and auxiliary task approach the CPF from two directions in the search space, respectively. Qiao et al. [43] constructed a dynamic auxiliary task, whose constraint boundaries are dynamically reduced. Thus, it can continuously provide complementary evolutionary directions for the main task. Since the auxiliary task ignoring all constraints does not provide effective information at most cases in the later evolutionary stage, Li et al. [26] proposed a dynamic population size reduc-

tion mechanism to reduce the consumption of computational resources. Qiao et al. [37] designed a dynamic constraint handling technique for the auxiliary task, which gradually increases the number of constraints to utilize the excellent infeasible solutions during the evolutionary process. In addition, the computational resources are dynamically allocated based on the performance of the tasks.

In order to enhance the performance of the algorithms and to achieve a better balance between feasibility, convergence, and diversity, many algorithms construct multiple auxiliary tasks. Qiao et al. [38] constructed global auxiliary task and local auxiliary task for enhancing global diversity and local diversity, respectively. In addition, a method to adaptively stop updating the global auxiliary task is applied to save computational resources. Ming et al. [34] proposed a three-task framework based on different CHTs. The main task and auxiliary tasks use constraint domination principle (CDP), Pareto domination (PD), and constraint relaxation techniques to balance constraints and objectives, respectively. Liu et al. [28] proposed a collaborative multitasking framework and constructed two adaptive auxiliary tasks. The first auxiliary task uses adaptive constraint relaxation techniques to dynamically shrink constraint boundaries, and the second auxiliary task adaptively selects constraints to focus on single constraint. Yu et al. [61] developed an evolutionary multitasking approach based on constraint subsets. Constraints are divided into multiple constraint subsets and multiple auxiliary tasks are constructed to solve the new CMOP formed by the constraint subsets. Qiao et al. [39] constructed multiple auxiliary tasks to solve the CMOP considering only a single constraint. Furthermore, the method of confirming the priority of auxiliary tasks is applied in order to retain effective auxiliary tasks.

2.3 CMOEAs based on reinforcement learning techniques

RL techniques guide the selection of optimal actions by maximizing cumulative rewards through the interaction of the agent and the environment. RL consists of three core elements: state, action, and reward. During iterative training, the agent takes an action on the current state according to the policy, and the environment generates a reward feedback after receiving the action and enters the next state. Each time the state, action, reward, and state after taking the action are recorded to train the agent. Therefore, designing a reasonable value function to measure the long-term benefits of states and actions is a key point in RL. QL approximates the value function through the Q table, which records the cumulative reward of each action in each state during training. The Q value which is the reward is used to learn the optimal strategy. The update rule is based on the Bellman equation. The Q value corresponding to state c and action a is calculated

as follows [32, 33]:

$$Q(c, a) = Q(c, a) + \alpha[r + \gamma \max_{a'} Q(c', a') - Q(c, a)], \quad (4)$$

where α is the learning rate, and γ is the discount parameter. However, since the Q table can only deal with discrete state space, the generalization ability of QL is insufficient. DQL uses DQN to approximate the value function of continuous state space, which can estimate the expected reward for each action. DQN trains the network through the loss function, and its goal is to minimize the mean square error of the Q value predicted by the network and the target Q value, which is calculated as follows [33, 35]:

$$\mathcal{L} = \frac{1}{|\Gamma|} \sum_{h \in \Gamma} (Q(c_h, a_h) - q_h)^2, \quad (5)$$

where Γ is the training data randomly sampled from the history record. $Q(c_h, a_h)$ is the output of the DQN with the input of the state c_h and action a_h . q_h is the Q value after taking action a_h at state c_h , which is calculated as follows:

$$q_h = r_h + \gamma \max_{a'} Q(c_{h+1}, a'), \quad (6)$$

where r_h is the reward feedback from the environment, and $Q(c_{h+1}, a')$ is the maximum reward for taking the next action under the state where a_h is performed.

Compared with evolutionary algorithm to find the optimal solution in a static environment, RL can interact and learn the optimal strategy in a dynamic environment. In order to deal with the defects of single CHT, Wang et al. [53] developed an adaptive CHT selection method based on DQL. It can dynamically evaluate the influence of various CHTs on the population state to select the best CHT. To solve the resource waste caused by too many tasks paralleling in EMT, Ming et al. [33] applied QL and DQL to the EMT framework, respectively. The Q-table and Q-network are used to select the most appropriate auxiliary task in the evolutionary process. Since the performance of CMOEAs largely depends on the operator used, Ming et al. [35] proposed an online operator selection framework that selects a suitable operator with the help of DQL. Hu et al. [21] proposed a Pareto front search strategy based on QL. The QL adaptively learns the shape and feature of CPF to guide the evolutionary process to better cover the CPF. Yang et al. [58] developed a DQL assisted Voronoi regions selection method. The search space is divided into multiple Voronoi regions by the archive, and DQL is applied to select promising regions to maintain high-quality solutions. Zuo et al. [68] proposed an autonomous evolutionary method based on process knowledge guidance. The embedded RL technique autonomously selects a guidance strategy

for the algorithm based on the state of the population. Zhang et al. [65] designed an optimization model based on DQL for solving the copper composition optimization problem. In their proposed ϵ -constrained approach, the ϵ value is adaptively adjusted by the DQL.

3 Proposed AMTCMO

3.1 Motivation

In the past studies, most of the popular CMOEAs use multiple stages or multiple populations to enhance the performance of the algorithms. Multiple stages based algorithms employ different CHTs or evolutionary strategies at different periods of evolution. Multiple population based algorithms use the framework of EMT. Each population performs a different task to implement a specific strategy, which helps the main population through knowledge transfer. Essentially, these methods are designed to address the limitations of a single strategy for the performance of the algorithm. However, in the multiple stages based algorithms, the parameters of stage switching are set based on subjective experience, which makes it difficult to determine an appropriate parameter. In order to make a balance between feasibility, convergence, and diversity, some EMT-based algorithms employ more than two auxiliary tasks. Due to the purposes of the main task and auxiliary tasks are different, not every task is effective at different stages of evolution. Thus, parallel evolution of these tasks will inevitably result in a waste of computational resources. The transfer of infeasible solutions generated by some auxiliary tasks may even cause negative effects on the main task. For example, as is shown in the Fig. 1, the position relationship between CPF and UPF of LIRCMOP6 and LIRCMOP7 is different. When it comes to the LIRCMOP6, the auxiliary task explores that the solution of UPF is the solution of CPF. On the LIRCMOP7, the solutions generated by the auxiliary task to explore the UPF in the later stage of evolution are not feasible, which is no longer helpful for the main task. Existing algorithms have designed some strategies to solve this problem. Qiao et al. [38] designed an adaptive stop-update strategy for their global auxiliary task, which stops utilizing the UPF by determining the standard deviation of the objective value at the later stage of evolution. Qiao et al. [37] designed a dynamic resource allocation scheme to allocate computational resources to the main task and auxiliary tasks. However, these methods still rely on the parameter settings and the priori knowledge in the population.

However, fixed strategies and auxiliary tasks cannot cope with all problems [55]. In the process of evolution, the state of the population is changing all the time, and the strategies needed in each period are different. It is difficult to implement an appropriate strategy in such a dynamic environment. How-

ever, the RL technology can make the most appropriate action based on the feedback of the reward value through the interaction between the agent and the environment. Compared with traditional CMOEAs, CMOEAs based on RL can adaptively select the best strategy and auxiliary task in a dynamic environment. It has solved the dependence of parameters for some algorithms to control the stage switching and the unnecessary waste of resources caused by the parallelism of multiple tasks in EMT. However, the existing algorithms still have some defects in the use of RL. Firstly, many algorithms have an insufficient representation of the state of the population when using RL. The common practice is to use some index values to reflect the convergence, feasibility, and diversity of the population, respectively. The calculated index is difficult to capture the state characteristics of the population, which may affect the selection of strategy. Secondly, the correlation between tasks is unexploited, which is the same problem that exists in many EMT-based CMOEAs. Knowledge transfer between low correlation tasks may affect the optimization efficiency and the global coordination of the search process. The RL technology can only select the beneficial strategy, thus lacking a basis for rational strategy selection.

Based on the above considerations, this paper proposes a hybrid DQL model enhanced by the attention mechanism for adaptive auxiliary task selection. In order to accurately capture the real-time demand of the main population, a self-attention mechanism is used to fuse the multi-dimensional features of the main population to generate a comprehensive state. Meanwhile, the cross-attention selection mechanism driven by task correlation is applied to improve the reliability of task selection and optimization efficiency.

3.2 General framework of AMTCMO

Algorithm 1 gives the general framework of the proposed AMTCMO. Meantime, Fig. 2 shows the flowchart of the main steps of AMTCMO. Firstly, three populations: main population P_1 , auxiliary population AP_1 , and auxiliary population AP_2 are initialized which consume $3N$ times function evaluation (lines 1-2). A counter is initialized for controlling the update of the DQN model (line 3). After that, the algorithm enters the cycle iteration process until the number of function evaluations (FE) reaches the maximum number of function evaluations ($MaxFE$). The fitness values and state vectors of each population are first calculated at the beginning of each cycle (lines 5-6). The state vector of the main population will be enhanced by Algorithm 2 through the self-attention mechanism (line 7). The attention scores between the main population and auxiliary tasks are calculated by the cross-attention mechanism in Algorithm 3 (line 8). In order to balance the exploration and exploitation, the algorithm is divided into two stages: learning stage and evolutionary stage. When $FE < \eta * MaxFE$, the main population and two

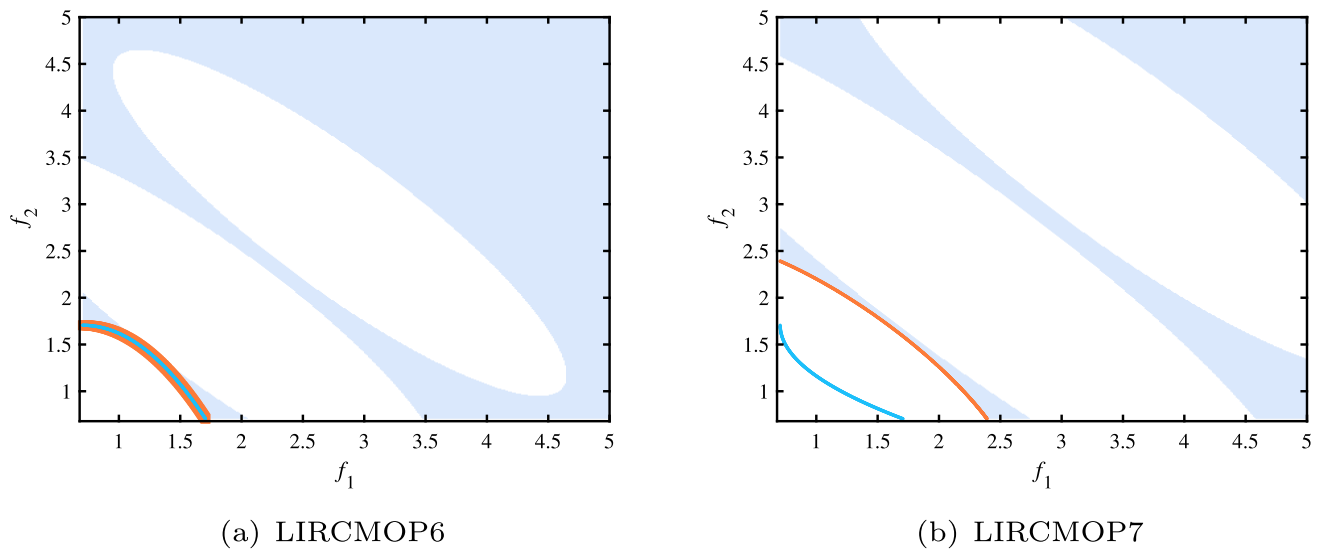


Fig. 1 Two positional relationships between CPF and UPF. Light blue is the feasible region, and orange and cyan lines represent CPF and UPF, respectively

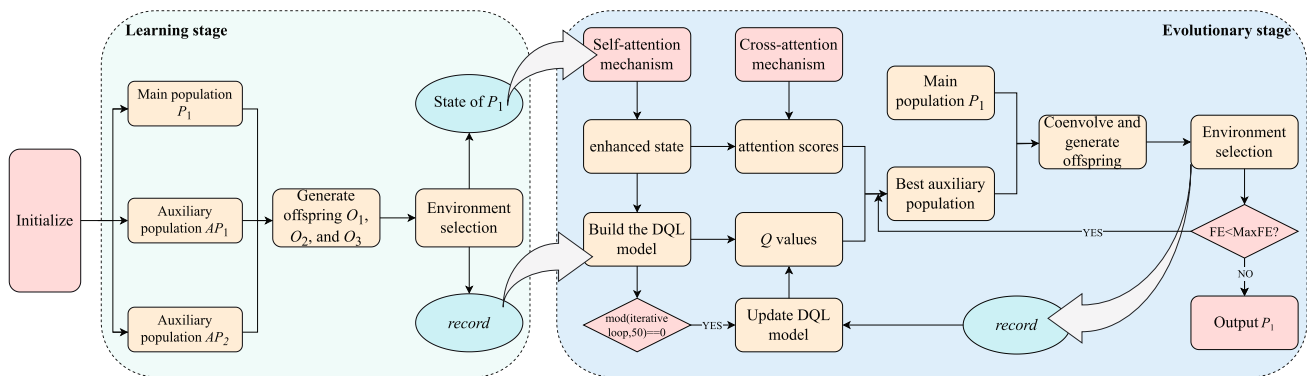


Fig. 2 An overview flowchart of AMTCMO

auxiliary tasks explore the search space together and accumulate data for constructing the DQN model (lines 10–11). When the algorithm enters the evolutionary stage, the stored data is first used to train the DQN. The enhanced state vector of the main population and the selection of the auxiliary task are regarded as the input state and the action, and the state improvement of the main population through a cycle is regarded as the reward (line 13). When making a decision, the trained DQN outputs the Q values of all the actions. The obtained Q values decide together with the attention scores to select the best action, which is the required auxiliary task for this cycle (line 14). The selected auxiliary task and the main task are co-evolved to generate offspring and perform environment selection (lines 15–20). In addition, the DQN is updated every fifty iterations of the cycle to ensure that the DQN is close to the current environment (lines 21–24). Finally, the state of the main population is computed again by Algorithm 2 and the difference between the two states cal-

culated by Eq. (7) is used as the feedback of the interaction between the agent and environment (lines 26–27).

$$reward = newstate_1 - state_1, \quad (7)$$

Both states are processed by the self-attention mechanism, which combines the performance of feasibility, diversity and convergence. The difference between the two states is not limited to the numerical fluctuation of a single dimension, but can show the overall improvement of the population in multi-dimensional performance. Therefore, the difference between the two is used as a reward for training DQN, which promotes the agent to realize the selection of the optimal strategy in subsequent iterations. The two states, the action and the reward are stored for updating the DQN (line 28). After the cycle termination is met, the main population is output as the final solution set.

Algorithm 1 General Framework of AMTCMO

Input: N (population size), $MaxFE$ (maximum number of function evaluations), η (parameter controlling the evolutionary stage)

Output: P_1 (main population)

- 1: Initialize the populations P_1 , AP_1 , and AP_2 ;
- 2: $FE = 3 \cdot N$;
- 3: $count = 0$;
- 4: **while** $FE < MaxFE$ **do**
- 5: Calculate fitness of P_1 , AP_1 , and AP_2 ;
- 6: $state_2 \leftarrow$ Calculate the state of AP_1 and AP_2 ;
- 7: $state_1 \leftarrow$ Enhance the state of P_1 by the self-attention mechanism using **Algorithm 2**;
- 8: $taskscores \leftarrow$ Calculate attention scores between P_1 and AP_1 , AP_2 by the cross-attention mechanism using **Algorithm 3**;
- 9: **if** $FE < \eta * MaxFE$ **then**
- 10: $O_1, O_2, O_3 \leftarrow$ Generate $N/2$ offspring by P_1 , AP_1 , and AP_2 respectively;
- 11: $P_1, AP_1, AP_2 \leftarrow$ Update by environment selection;
- 12: **else**
- 13: Q values $\leftarrow$ Build the DQL model to predict Q value of each action;
- 14: $i \leftarrow$ Select an action according to the Q values and $taskscores$;
- 15: **if** $i == 1$ **then**
- 16: $O_1, O_2 \leftarrow$ Generate offspring by P_1 and AP_1 ;
- 17: **else**
- 18: $O_1, O_3 \leftarrow$ Generate offspring by P_1 and AP_2 ;
- 19: **end if**
- 20: $P_1, AP_1, AP_2 \leftarrow$ Update by environment selection;
- 21: $count = count + 1$;
- 22: **if** $mod(count, 50) == 0$ **then**
- 23: DQL $\leftarrow$ Update the DQL model using $record$;
- 24: **end if**
- 25: **end if**
- 26: $newstate_1 \leftarrow$ Update the state of P_1 by the self-attention mechanism using **Algorithm 2**;
- 27: $reward \leftarrow$ Calculate the improvement in the state of P_1 ;
- 28: $record \leftarrow$ Save the $state_1$, $reward$, $action$, and $newstate_1$;
- 29: $FE = FE + 3 \cdot N/2$;
- 30: **end while**

3.3 Attention mechanism based adaptive task selection

The training of the DQN is realized through the interaction between the agent and the environment. The agent first observes the state of the environment, and after selecting an action to execute, the environment updates the state and gives feedback as the reward. Therefore, the key to training a DQN lies in the determination of states, actions and rewards. In the previous algorithm design, the state is determined by the state of the main population, the state improvement of the main population is regarded as the reward, and the strategy to be selected is viewed as the action. In order to demonstrate the feasibility, convergence, and diversity of the main population, the usual approach uses Eqs (8) to (10) to calculate three indicators as the state vector [33, 35].

$$con = \frac{\sum_{x \in \mathcal{P}} \sum_{j=1}^m f_j(x)}{N}, \quad (8)$$

$$fea = \frac{\sum_{x \in \mathcal{P}} CV(x)}{N}, \quad (9)$$

$$div = \frac{1}{\sum_{j=1}^m (f_j^{\max} - f_j^{\min})}, \quad (10)$$

where $\mathcal{P}$ represents the population to be calculated, N is the size of population, and $f_j(x)$ is the j -th objective function value. However, the calculated state cannot fully characterize the environment and it is difficult to capture key information. As the eyes of RL, the state determines the understanding of the agent for the environment, which can affect the selection of action. The attention mechanism stems from the ability of human eyes to process information from the outside world. When making observations, our attention is focused on what needs to be focused on and filters out unimportant information. The state representation in RL lacks exactly this processing ability. In the proposed AMTCMO, the self-attention mechanism is applied to enhance the representation of the main population state. The self-attention mechanism can capture the internal correlation of the population features, thereby generating a real-time integrated state. The enhanced state can better reflect the current required action. The specific implementation steps are presented in Algorithm 2. Firstly, the calculated indexes of the main population are composed of a state vector (line 1). The three randomly initialized weight matrices and state vectors are linearly transformed to obtain three vectors of query, key, and value, respectively (lines 2-5). The query and the key will obtain the similarity A through the dot product operation (line 6). After that, the softmax function is used to normalize A to obtain the attention weight. Due to the large dot product calculation result, the softmax function gradient may disappear. A is scaled by the dimension of the key vector (line 7). Finally, the weight matrices and the value are weighted and summed to yield the enhanced state vector (line 8). Since the three weight matrices are parameters to be learned, a strategy similar to the gradient descent method is used to adaptively update the weight matrices (lines 10-12). The input state is used as the gradient and the direction of the gradient update is determined by the $reward$. The learning rate is a preset parameter λ , which is discussed in the Sect. 4.5. Specifically, when the $reward$ is greater than zero, the state of the population is improved. The update of the weight matrix strengthens the correlation between the Q , K and V vectors and the original input state, so that a positive weight distribution effect can be generated in the future. On the contrary, the update of the weight matrix suppress the current state. The learning rate regulates the magnitude of the update during the process to prevent the weight matrix from constantly oscillating. This heuristic strategy enables the weight matrices to learn adaptively. In this scenario where the loss function can not be defined, it is beneficial to obtain the attention mode of positive $reward$.

Algorithm 2 Self-attention mechanism

Input: $reward, \lambda$ (learning rate)
Output: $state_1$

- 1: $V \leftarrow [fea, con, div];$
- 2: $W_q, W_k, W_v \leftarrow \text{rand}(3, 3);$
- 3: $Q = W_q * V;$
- 4: $K = W_k * V;$
- 5: $V = W_v * V;$
- 6: $A = Q * K^T;$
- 7: $weight = \text{softmax}(A/\text{sqrt}(3));$
- 8: $state_1 = weight * V;$
- 9: Update attention weights by the gradient descent method according to the feedback:
- 10: $W_q \leftarrow W_q + \lambda * reward * Q * state_1;$
- 11: $W_k \leftarrow W_k + \lambda * reward * K * state_1;$
- 12: $W_v \leftarrow W_v + \lambda * reward * V * state_1;$

In addition, the DQL model not only learns the current Q value for performing each action, but also predicts the maximum cumulative Q value in the future. It is reasonable to select appropriate actions based on the learned Q values, but this approach has limitations when applied to EMT framework to select auxiliary tasks. This is because the Q value only considers the contribution of the auxiliary task, it does not consider the correlation with the main task. The key point of EMT is the knowledge transfer between tasks. The efficiency of knowledge transfer can be improved between tasks with high correlation. Conversely, the transferred knowledge even have a negative impact. Therefore, analyzing the correlation between tasks helps to improve the efficiency of optimization and the reliability of auxiliary task selection. The cross-attention mechanism has a wide range of applications in natural language processing. It is mainly used to model the dynamic association between two different sequences or feature spaces to capture the semantic relevance between them. In the proposed method, the cross-attention mechanism is used to compute the attention scores between the main task and auxiliary tasks to represent the task correlation. The hybrid DQL model uses attention scores and Q values to collaboratively decide the action selection. The pseudo-code for the specific implementation is represented in Algorithm 3. Meanwhile, Fig. 3 shows the flowchart of the hybrid DQL model. Firstly, two weight matrices are randomly initialized (line 1). The enhanced state vector of the main population and the state vectors of the auxiliary population are linearly transformed with the weight matrices to generate the query and key vector, respectively (lines 2-3). Different from the self-attention mechanism, the value vector is the same as the key vector in the cross-attention mechanism (line 4). After that, the similarity A is calculated by the dot product of the query and the key vector (line 5). The attention weight is normalized by the softmax function with the scaling operation (line 6). Finally, the attention scores are obtained by the weighted sum of the attention weight and the value vector (line 7). In order to better adaptively learn the

weight matrix, the contribution rate of the auxiliary tasks to the main task is used to control the direction of the gradient update, which is calculated as follows:

$$succ_j = \frac{|AP_j^t|}{|AP_j|}, \quad (11)$$

where $|AP_j|$ represents the size of the auxiliary population, and $|AP_j^t|$ is the number of solutions transferred to the main population. The state of the auxiliary task is used as the gradient, and the learning rate is consistent with the self-attention mechanism. When the contribution rate of AP_1 is greater than that of AP_2 ($succ_1 - succ_2 > 0$), the weight matrix of AP_1 is enhanced and the weight matrix of AP_2 is suppressed. Conversely, the situation is opposite. Therefore, the weight matrix of the auxiliary population with a high contribution rate can be positively adjusted to obtain higher attention scores (lines 8-14).

Algorithm 3 Cross-attention mechanism

Input: $state_1, state_2, succ, \lambda$ (learning rate)
Output: $taskscores$

- 1: $W_q, W_k \leftarrow \text{rand}(3, 3);$
- 2: $Q = W_q * state_1;$
- 3: $K = W_k * state_2;$
- 4: $V = K;$
- 5: $A = Q * K^T;$
- 6: $weight = \text{softmax}(A/\text{sqrt}(3));$
- 7: $taskscores = weight * V;$
- 8: **if** $succ_1 - succ_2 > 0$ **then**
- 9: $W_q \leftarrow W_q + \lambda * Q * state_2;$
- 10: $W_k \leftarrow W_k + \lambda * K * state_2;$
- 11: **else**
- 12: $W_q \leftarrow W_q - \lambda * Q * state_2;$
- 13: $W_k \leftarrow W_k - \lambda * K * state_2;$
- 14: **end if**

3.4 Auxiliary tasks

The setting of auxiliary tasks in EMT is particularly important. Through knowledge transfer, auxiliary tasks can help the main task quickly achieve convergence and get rid of local optima. However, the requirements of the main task are different in the evolutionary process. Therefore, two auxiliary tasks are set up in the proposed AMTCMO to enhance convergence and diversity respectively for the hybrid DQL model to learn and select. Firstly, to enhance the diversity, the constraint relaxation auxiliary task AP_1 is set as follows:

$$\begin{aligned} \text{Minimize } \mathbf{F}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \\ \text{s.t. } CV(\mathbf{x}) &\leq \varepsilon_G, \end{aligned} \quad (12)$$

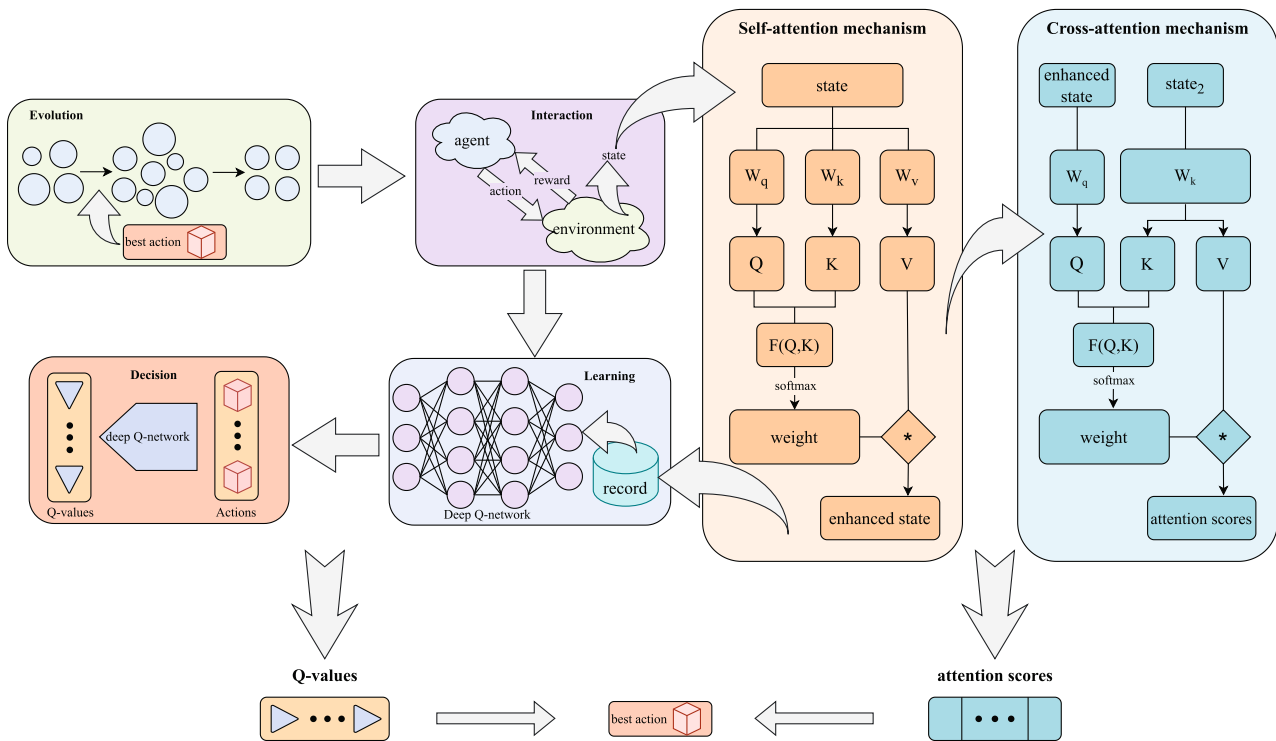


Fig. 3 The illustration of the hybrid DQL model

where ε_G is updated by [52]:

$$\varepsilon_G = \varepsilon_0 * \left(1 - \frac{G}{G_{max}}\right)^{\left(\frac{-\log(\varepsilon_0) - 6}{\log(1-\omega)}\right)}, \quad (13)$$

where G denotes the current generation, G_{max} represents the maximum generations, ε_0 is the initial maximum CV, and ω is used to control the decreasing speed of ε_G . The solution with the CV less than ε_G in the constraint relaxation auxiliary task is regarded as a feasible solution. Therefore, some promising infeasible solutions can be retained in the constraint relaxation auxiliary population to maintain diversity. Since most of the retained infeasible solutions are located around feasible regions, continuing to explore these regions can complement the evolutionary direction of the main population. The degree of constraint relaxation decreases with the number of evolutionary generations, so that the constraint relaxation population and the main population maintain a high degree of correlation in the later stages of evolution. At the same time, to better preserve diversity, the constraint relaxation auxiliary population uses an angle-based environmental selection strategy.

Secondly, in order to help the main task cross the infeasible regions, the unconstrained auxiliary task AP_2 is set as follows:

$$\text{Minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})). \quad (14)$$

Therefore, the optimization task of unconstrained auxiliary population is transformed into solving MOPs. Since the constraints are not considered, the unconstrained auxiliary population can not be blocked by the infeasible regions during the evolution process and can directly converge to the UPF. The fast evolution of the unconstrained auxiliary population prioritizes discovering the potential feasible regions and helps the main task to converge quickly through knowledge transfer. In addition, both the two auxiliary populations adopt the neighbor pairing strategy to generate the offspring [41, 63].

3.5 Computational complexity analysis

The computational complexity of AMTCMO includes the calculation of population state, offspring generation, environment selection, and training of DQN. P_1 , AP_1 and AP_2 calculate the state vector, and the time consumption is $3 * O(mnN)$. In the learning stage, all populations generate offspring, so the time complexity is $3 * O(N/2)$. In the evolutionary stage, only the main population and the selected auxiliary population generate offspring, and the time complexity is $2 * O(N/2)$. Each population performs the environment selection every iteration, so the time complexity is $3 * O(mN^2)$. The complexity of training DQN is $O(4u^2)$, where u is the number of neurons in the neural network [6, 35]. In general, the computational complexity of AMTCMO

is $3 * O(mnN) + 3 * O(N/2) + 3 * O(mN^2) + O(4u^2) = O(mN^2)$, which is consistent with many exiting CMOEAs.

4 Experiment

4.1 Experiment setting

4.1.1 Benchmark problems

Three typical test suites CF [64], LIRCMOP [12], and DASC-MOP [14] are selected to test the performance of the proposed AMTCMO. The CF test suite contains 10 test problems. The constraints of such test problems contain many nonlinear constraints, resulting in very narrow or highly discontinuous feasible regions. The Pareto front (PF) is divided into discrete point sets or segmented continuous surfaces due to the distribution of the feasible regions. At the same time, the constraints and objectives of some problems have a strong coupling relationship making the distribution of feasible solutions closely related to the shape of PF. The LIRC-MOP test suite contains 14 test problems, most of which contain large infeasible regions. The objective function of each test problem is jointly controlled by the shape function and the distance function. The shape function is used to control the concave and convex of PF, and the distance function is used to control the convergence difficulty of the problem. In addition, the CPF of many test problems is located at the constraint boundary due to the control of the shape function, which increases the difficulty of the search. DASC-MOP test suite consists of 9 test problems of adjustable difficulty. The constrained difficulty of each test problem is controlled by a ternary array. Each position of the ternary array represents feasibility difficulty, convergence difficulty, and diversity difficulty, respectively. As a result, the test problems in DASC-MOP are flexible and varied, posing a significant challenge to the performance of algorithms. The three test suites have a total of 33 test problems, which can comprehensively test the performance of the algorithm. Furthermore, we utilize the latest LSCM [40] test suite to demonstrate the advancement of the proposed AMTCMO, which contains 12 test problems. The constraint function of such test problems is directly related to the decision space, including linear and nonlinear relationships. The high-latitude decision space increases the number of constraint functions, resulting in a complex shape of the feasible region and a fragmented local feasible region. In addition, the decision variable and objective function have a non-uniform correlation, so some variables only affect specific objectives, which increases the difficulty of algorithm search.

4.1.2 Comparison algorithms

In order to reflect the performance of the proposed algorithms, nine competitive algorithms of recent years are selected as comparison algorithms. They are DCNSGAI [23], BiCo [29], EMC-MO [42], CMEGL [38], MTCMO [43], DRLOSEMCMO [35], CSEMT [39], DEST [54], and APSEA [48]. DCNSGAI uses a problem transformation technique that transforms the CMOP into a dynamic CMOP. The difficulty of constraints increases gradually, which reduces the difficulty of solving CMOP. To utilize promising infeasible solutions, DCNSGAI uses an improved operator to generate ϵ -feasible solutions. BiCo constructs an auxiliary population to search for the CPF from the two sides of the space with the main population. EMC-MO develops a framework based on EMT to solve CMOP, which constructs an unconstrained auxiliary population to co-evolve with the main population. Meanwhile, a tentative approach is designed to guide knowledge transfer. CMEGL constructs two auxiliary populations, a global auxiliary population and a local auxiliary population for enhancing global diversity and local diversity, respectively. To conserve computational resources, an adaptive approach is used to control the evolution of the global auxiliary population. MTCMO constructs a dynamic auxiliary population in which the constraint boundary is gradually reduced, and designs an improved method to utilize high-quality infeasible solutions. DRLOSEMCMO proposes an online operator selection framework based on DQL technology, which can select the best operator to generate offspring according to the state of the population. CSEMT constructs multiple auxiliary tasks to search for PFs that only consider a single constraint. The auxiliary task priority method is used to confirm the most useful auxiliary task. DEST develops a diversity-enhanced evolutionary framework to assign more priorities to diversity, which improves the exploitation of CPF. In order to prevent the auxiliary population from wasting the function evaluations, APSEA proposes an adaptive population size method, which dynamically adjusts the population size according to the evolutionary state.

4.1.3 Performance indicators

In order to evaluate the superiority between the comparison algorithms, two performance indicators are selected: inverse generation distance plus (IGD+) and hypervolume (HV). IGD+ is calculated as follows [59]:

$$\text{IGD}^+(P^*, S) = \frac{1}{|P^*|} \sum_{y^* \in P^*} \min_{s \in S} d^+(y^*, s), \quad (15)$$

where P^* represents the true PF. S donates the optimal solution set obtained by the algorithm. IGD+ measures the

quality of the output solution set by calculating the distance of the set from the true PF. An improved Euclidean distance is used in IGD+, which is calculated as follows:

$$d^+(y^*, s) = \sqrt{\sum_{i=1}^m (\max\{s_i - y_i^*, 0\})^2}, \quad (16)$$

where m is the dimension of the objective space. Due to the dependence on the true PF, IGD+ cannot measure the performance of the algorithm in real-world problems. At the same time, IGD+ cannot perfectly reflect the diversity of the algorithm. Therefore, HV is introduced to evaluate the performance of the algorithms. HV is calculated as follows [18]:

$$HV(S) = \Lambda(\{q \in \mathbb{R}^m \mid \exists s \in S : s \leq q \text{ and } q \leq r\}), \quad (17)$$

where $\Lambda(\cdot)$ denotes the Lebesgue measure and r is a fixed reference point defined in the objective space $\mathbb{R}^m$. HV reflects the quality of the solution set by calculating the size of the hypervolume surrounded by the output solution set and the reference points. It can be seen that the smaller the IGD+ and the larger the HV represent the better the convergence and diversity of the output solution set. Therefore, the selected two performance indicators can comprehensively evaluate the performance of the comparison algorithms.

4.1.4 Experimental environment

The experiments in this paper are all conducted on the PlatEMO (v4.11) platform built by Tian et al. [47] in MATLAB (2022b). The experimental environment is Intel (R) Core (TM) i7-10700CPU @ 2.90GHz processor, 16GB RAM, and Windows 11 Professional operating system. All the comparison algorithms are from the PlatEMO platform and the parameters are chosen with the same settings as the original literature. During the experiments, the maximum number of evaluations ($MaxFE$) of all the comparison algorithms is set to 100000, and the population size (N) is 100. Each algorithm is run independently 24 times on each test problem.

4.2 Comparison results

Tables 1 and 2 show the IGD+ and HV results for the ten comparison algorithms on the 33 test problems, respectively. Each position of the table shows the mean and standard deviation of each algorithm after 24 runs on the corresponding test problem. The Wilcoxon rank sum test [67] with a significance level of 0.05 is used to analyze the obtained results, and the symbols “+”, “−”, and “=” represent that the algorithms are better than, worse than, and equal to the proposed

algorithm. The best results for each test problem are labelled using bold in Tables 1 and 2. As can be seen from Table 1, AMTCMO outperforms DCNSGAI, BiCo, EMC, CMEGL, MTCMO, DRLOSEMCMO, CSEMT, DEST, and APSEA on 31, 30, 24, 25, 29, 27, 26, 28, and 28 test problems, respectively. In addition, AMTCMO achieves the best results on 24 test problems. The results in Table 2 show that AMTCMO outperforms these comparison algorithms on 31, 29, 24, 24, 29, 25, 25, 28, and 27 test problems, respectively. Meanwhile, AMTCMO achieves the best results on 22 test problems. Combining the results of both IGD+ and HV, AMTCMO has a great advantage on these 33 test problems.

4.2.1 Results for CF

For the CF test suite, AMTCMO achieves the best IGD+ results on 6 of the 10 problems. In terms of HV, AMTCMO achieves the best in 4 test problems and is equal to the best algorithm in 3 test problems. Furthermore, as is shown in Fig. 4, AMTCMO achieves the best distribution on CF4. Due to the high coupling of constraints and objectives and the large scale of infeasible regions, the shape of CPF on many test problems is a discrete point set or surface. This makes it difficult for the algorithm to achieve a balance between convergence and diversity. The auxiliary tasks constructed by EMC and MTCMO only provide one aspect of convergence and diversity for the main population. The parallelism of the two auxiliary tasks in CMEGL cannot achieve balance because of the waste of function evaluation. CSEMT and DCNSGAI transform constraints into single constraints and dynamic constraints respectively, showing advantages in complex problems. In addition, because the feasible regions in the CF test suite are narrow, the unconstrained auxiliary population cannot effectively explore the feasible regions to help the main population. Therefore, the algorithms using an unconstrained auxiliary population have a poor performance. AMTCMO can determine the required auxiliary populations based on the real-time state of the main population during the evolutionary stage. It not only saves the function evaluation, but also co-evolves with the most suitable auxiliary population to maintain a balance between convergence and diversity.

4.2.2 Results for LIRCMOP

On the 14 test problems of LIRCMOP, AMTCMO achieves the best results in both IGD+ and HV for 12 test problems. It can be seen from Fig. 5 that only the final solution set obtained by AMTCMO completely covers the CPF. Due to the long distance between UPF and CPF, the value of exploring UPF is very small. Thus, the algorithm using unconstrained auxiliary population on LIRCMOP1-4 shows a poor result. DEST uses

Table 1 IGD+ results of ten comparison algorithms on CF, LIRC-MOP, and DASC-MOP test suites

Problem	<i>m</i>	<i>n</i>	DCNSGAIII	BiCo	EMCMO	CMEGL	MTCMO
CF1	2	10	5.7774e-3 (1.37e-3) -	1.6675e-2 (1.99e-3) -	6.0069e-3 (6.83e-4) -	4.8131e-3 (5.56e-4) -	1.0462e-2 (1.80e-3) -
CF2	2	10	3.4496e-2 (1.25e-2) -	4.8365e-2 (1.85e-2) -	2.8575e-2 (1.22e-2) -	2.9374e-2 (1.04e-2) -	3.8411e-2 (1.36e-2) -
CF3	2	10	2.3651e-1 (7.44e-2) -	2.3654e-1 (8.13e-2) -	2.2074e-1 (3.44e-2) -	2.2703e-1 (8.68e-2) -	2.3168e-1 (5.66e-2) -
CF4	2	10	8.3125e-2 (2.44e-2) -	8.0065e-2 (2.48e-2) -	6.5485e-2 (2.12e-2) -	7.8331e-2 (2.42e-2) -	7.9705e-2 (2.53e-2) -
CF5	2	10	2.2849e-1 (1.06e-1) -	2.1891e-1 (7.98e-2) -	1.9142e-1 (8.37e-2) -	2.0204e-1 (1.09e-1) -	2.0578e-1 (9.39e-2) -
CF6	2	10	3.3528e-2 (9.13e-3) -	5.3356e-2 (1.71e-2) -	3.5166e-2 (1.26e-2) -	3.8697e-2 (1.23e-2) -	3.6769e-2 (1.36e-2) -
CF7	2	10	2.0505e-1 (1.43e-1) -	1.8357e-1 (1.15e-1) -	1.6311e-1 (9.16e-2) -	1.6695e-1 (8.67e-2) -	1.7673e-1 (1.32e-1) -
CF8	3	10	1.9635e-1 (4.98e-2) -	1.6241e-1 (6.48e-2) =	1.8667e-1 (8.50e-2) =	1.6760e-1 (5.64e-2) =	1.5411e-1 (5.38e-2) -
CF9	3	10	1.2118e-1 (5.95e-2) =	5.7815e-2 (4.37e-3) +	7.3683e-2 (1.78e-2) +	6.4807e-2 (8.44e-3) +	6.0976e-2 (5.78e-3) +
CF10	3	10	2.3486e-1 (1.18e-1) =	4.8133e-1 (1.87e-1) -	2.6622e-1 (8.62e-2) =	2.8705e-1 (9.82e-2) =	3.5723e-1 (9.54e-2) -
LIRC-MOP1	2	30	1.5816e-1 (2.49e-2) -	1.9061e-1 (1.19e-2) -	1.9585e-1 (3.55e-2) -	1.1313e-1 (1.90e-2) -	1.1966e-1 (2.17e-2) -
LIRC-MOP2	2	30	1.0406e-1 (7.05e-3) -	1.1702e-1 (1.14e-2) -	1.3698e-1 (2.71e-2) -	7.2251e-2 (1.15e-2) -	6.9688e-2 (7.95e-3) -
LIRC-MOP3	2	30	1.7277e-1 (2.02e-2) -	1.8713e-1 (2.09e-2) -	2.2909e-1 (3.41e-2) -	1.3844e-1 (2.92e-2) -	1.3123e-1 (2.14e-2) -
LIRC-MOP4	2	30	1.1862e-1 (1.16e-2) -	1.2856e-1 (1.47e-2) -	1.6917e-1 (3.08e-2) -	1.0179e-1 (2.44e-2) -	9.8250e-2 (1.31e-2) -
LIRC-MOP5	2	30	8.0990e-1 (5.04e-1) -	1.2228e+0 (6.04e-3) -	2.1483e-1 (3.23e-2) -	3.0264e-1 (2.83e-1) -	6.0480e-1 (4.86e-1) -
LIRC-MOP6	2	30	7.8258e-1 (5.31e-1) -	1.3453e+0 (1.39e-4) -	2.3650e-1 (3.57e-2) -	4.4816e-1 (4.14e-1) -	7.6006e-1 (5.08e-1) -
LIRC-MOP7	2	30	1.1216e-1 (1.50e-2) -	7.1058e-1 (7.68e-1) -	9.5635e-2 (2.20e-2) -	9.5229e-2 (2.14e-2) -	1.0370e-1 (1.84e-2) -
LIRC-MOP8	2	30	1.7616e-1 (4.14e-2) -	1.0762e+0 (7.32e-1) -	1.4833e-1 (5.34e-2) -	1.5162e-1 (3.77e-2) -	1.5651e-1 (3.75e-2) -
LIRC-MOP9	2	30	7.0733e-1 (2.15e-1) -	9.2167e-1 (1.81e-1) -	3.5113e-1 (1.32e-1) -	2.8546e-1 (1.62e-1) -	7.7219e-1 (1.66e-1) -
LIRC-MOP10	2	30	3.8568e-1 (2.45e-1) -	9.2308e-1 (7.59e-2) -	1.4755e-1 (7.74e-2) -	1.7942e-1 (6.90e-2) -	6.2734e-1 (2.66e-1) -
LIRC-MOP11	2	30	2.2965e-1 (1.34e-1) -	7.1491e-1 (2.30e-1) -	4.4331e-2 (2.86e-2) -	5.8105e-2 (3.99e-2) -	4.2900e-1 (2.14e-1) -
LIRC-MOP12	2	30	3.7179e-1 (1.35e-1) -	5.3857e-1 (2.40e-1) -	1.9463e-1 (5.82e-2) -	1.2977e-1 (6.75e-2) -	3.4398e-1 (1.22e-1) -
LIRC-MOP13	3	30	1.3063e+0 (7.76e-4) -	1.2700e+0 (2.31e-1) -	4.0270e-2 (1.24e-3) +	4.3799e-2 (1.14e-3) +	1.3146e+0 (1.82e-3) -
LIRC-MOP14	3	30	1.2626e+0 (8.73e-4) -	1.2736e+0 (1.76e-3) -	4.6329e-2 (1.19e-3) +	4.5502e-2 (1.33e-3) +	1.2708e+0 (1.57e-3) -
DASC-MOP1	2	30	7.1389e-1 (5.58e-2) -	7.2589e-1 (1.57e-2) -	7.1909e-1 (4.37e-2) -	6.6307e-1 (8.80e-2) -	6.9900e-1 (6.95e-2) -
DASC-MOP2	2	30	1.6000e-1 (1.45e-2) -	1.4726e-1 (6.74e-3) -	1.4212e-1 (6.68e-3) -	1.3071e-1 (1.13e-2) -	1.3362e-1 (7.73e-3) -
DASC-MOP3	2	30	1.6617e-1 (2.56e-2) -	1.8528e-1 (1.74e-2) -	1.8744e-1 (1.87e-2) -	1.6544e-1 (2.69e-2) -	1.7455e-1 (2.12e-2) -
DASC-MOP4	2	30	3.1677e-3 (1.01e-3) -	1.0551e-1 (1.28e-1) -	1.3783e-3 (8.81e-4) =	8.3150e-4 (4.81e-4) -	9.2784e-4 (9.76e-4) -
DASC-MOP5	2	30	4.3547e-3 (1.37e-3) -	6.2875e-2 (1.31e-1) -	1.8997e-3 (1.77e-4) +	1.8873e-3 (9.81e-5) +	1.9762e-3 (9.62e-4) +
DASC-MOP6	2	30	1.6055e-2 (1.97e-2) -	1.7085e-1 (2.45e-1) -	1.8689e-2 (1.94e-2) -	7.3929e-3 (4.66e-3) -	1.4412e-1 (1.92e-1) -
DASC-MOP7	3	30	3.3002e-2 (1.65e-3) -	2.6772e-2 (7.38e-3) -	2.3633e-2 (1.17e-3) +	2.3872e-2 (1.15e-3) +	2.3556e-2 (8.68e-4) +
DASC-MOP8	3	30	2.4813e-2 (1.86e-3) -	1.9237e-2 (1.22e-3) +	1.8674e-2 (8.05e-4) +	1.8557e-2 (7.73e-4) +	1.8259e-2 (8.05e-4) +

Table 1 continued

Problem	<i>m</i>	<i>n</i>	DCNSGAIH	BiCo	EMCMO	CMEGL	MTCMO
DASCMOP9	3	30	2.7105e-1 (6.10e-2) – 0/31/2	2.5047e-1 (3.60e-2) – 2/30/1	2.3928e-1 (4.55e-2) – 6/24/3	1.8194e-1 (2.47e-2) – 6/25/2	2.1851e-1 (3.49e-2) – 4/29/0
+/-/=							
Problem	<i>m</i>	<i>n</i>	DRLOSEMCMO	CSEMT	DEST	APSEA	AMTCMO
CF1	2	10	2.2198e-3 (3.47e-4) +	2.7200e-3 (5.54e-4) +	7.4080e-3 (8.03e-4) –	1.4786e-2 (1.58e-3) –	3.4295e-3 (4.99e-4)
CF2	2	10	7.4041e-3 (2.83e-3) –	5.8516e-3 (8.38e-4) –	4.6851e-2 (2.14e-2) –	2.9206e-2 (1.04e-2) –	3.7424e-3 (6.42e-4)
CF3	2	10	1.6791e-1 (7.15e-2) =	1.8096e-1 (6.25e-2) =	2.4136e-1 (5.06e-2) –	2.1402e-1 (6.97e-2) –	1.6797e-1 (6.72e-2)
CF4	2	10	3.8851e-2 (8.67e-3) –	3.7663e-2 (2.34e-2) =	9.1262e-2 (2.68e-2) –	7.5183e-2 (2.56e-2) –	2.9834e-2 (5.45e-3)
CF5	2	10	1.8804e-1 (9.06e-2) =	1.4420e-1 (7.29e-2) =	2.4802e-1 (1.05e-1) –	2.0288e-1 (9.56e-2) –	1.5599e-1 (8.79e-2)
CF6	2	10	2.7601e-2 (5.65e-3) –	3.1130e-2 (1.22e-2) –	3.6100e-2 (1.39e-2) –	3.6727e-2 (1.22e-2) –	1.8597e-2 (3.37e-3)
CF7	2	10	1.3021e-1 (4.98e-2) –	1.6955e-1 (8.08e-2) –	1.6329e-1 (6.28e-2) –	1.9581e-1 (1.20e-1) –	8.6660e-2 (2.18e-2)
CF8	3	10	1.7215e-1 (2.02e-2) –	2.4539e-1 (1.17e-1) –	1.7348e-1 (8.58e-2) =	2.1693e-1 (1.07e-1) =	1.5263e-1 (1.43e-2)
CF9	3	10	8.8604e-2 (1.17e-2) +	9.4530e-2 (1.80e-2) =	7.4136e-2 (1.76e-2) +	6.3751e-2 (9.72e-3) +	1.0569e-1 (2.08e-2)
CF10	3	10	2.6817e-1 (6.59e-2) –	3.2262e-1 (1.56e-1) –	2.8643e-1 (9.66e-2) =	3.7172e-1 (1.37e-1) –	2.2778e-1 (7.28e-2)
LRCMOP1	2	30	1.9190e-1 (3.01e-2) –	9.2378e-2 (2.68e-2) –	5.1638e-2 (1.90e-2) –	2.5327e-1 (2.15e-2) –	7.8416e-3 (1.20e-3)
LRCMOP2	2	30	1.0598e-1 (3.28e-2) –	8.0108e-2 (3.67e-2) –	4.1843e-2 (1.05e-2) –	1.5085e-1 (2.70e-2) –	8.8076e-3 (1.28e-3)
LRCMOP3	2	30	2.0312e-1 (4.26e-2) –	1.5619e-1 (4.51e-2) –	9.7381e-2 (3.79e-2) –	2.4651e-1 (3.52e-2) –	6.4440e-3 (6.72e-4)
LRCMOP4	2	30	1.4974e-1 (3.07e-2) –	1.0690e-1 (3.67e-2) –	7.3465e-2 (2.24e-2) –	1.6914e-1 (2.85e-2) –	7.4903e-3 (2.24e-3)
LRCMOP5	2	30	1.7483e-2 (7.20e-3) –	2.3211e-1 (4.47e-1) –	2.3211e-1 (4.51e-2) –	1.0872e+0 (3.45e-1) –	7.9249e-3 (8.92e-4)
LRCMOP6	2	30	1.1466e-2 (1.26e-3) –	3.1305e-1 (4.15e-1) –	2.4640e-1 (7.30e-2) –	1.1862e+0 (3.65e-1) –	6.9838e-3 (4.67e-4)
LRCMOP7	2	30	7.1943e-3 (1.21e-3) –	8.7407e-2 (2.59e-2) –	1.0491e-1 (1.97e-2) –	9.1967e-2 (2.72e-2) –	6.4698e-3 (3.31e-4)
LRCMOP8	2	30	6.9544e-3 (3.94e-4) –	9.9200e-2 (5.51e-2) –	1.5252e-1 (3.57e-2) –	1.4487e-1 (4.51e-2) –	6.4471e-3 (2.72e-4)
LRCMOP9	2	30	1.6172e-1 (1.84e-2) –	2.0997e-1 (5.52e-2) –	2.5395e-1 (1.32e-1) –	6.1055e-1 (2.45e-1) –	9.9732e-3 (1.01e-2)
LRCMOP10	2	30	4.5716e-2 (2.61e-2) –	1.1038e-1 (1.07e-1) –	1.4410e-1 (6.04e-2) –	7.0339e-1 (2.24e-1) –	7.5737e-3 (4.59e-4)
LRCMOP11	2	30	1.7935e-2 (1.13e-2) –	6.3743e-2 (5.77e-2) –	5.7583e-2 (2.62e-2) –	5.6959e-1 (1.80e-1) –	2.1203e-3 (1.26e-3)
LRCMOP12	2	30	8.0872e-2 (2.11e-2) –	1.3071e-1 (1.34e-2) –	1.1439e-1 (4.04e-2) –	3.1126e-1 (1.69e-1) –	8.8866e-3 (1.90e-2)
LRCMOP13	3	30	6.5926e-2 (4.27e-3) =	4.2291e-2 (1.12e-3) +	4.2107e-2 (1.04e-3) +	1.3136e+0 (1.36e-3) –	6.5521e-2 (3.45e-3)
LRCMOP14	3	30	5.2100e-2 (1.75e-3) =	4.3717e-2 (1.94e-3) +	4.4522e-2 (9.00e-4) +	1.2703e+0 (1.74e-3) –	5.2918e-2 (2.58e-3)
DASCMOP1	2	30	2.5375e-3 (2.73e-4) –	3.4253e-1 (3.26e-1) –	5.7752e-1 (1.16e-1) –	7.2797e-1 (3.35e-2) –	1.8901e-3 (1.04e-4)
DASCMOP2	2	30	4.4803e-3 (2.15e-4) –	7.9516e-2 (5.76e-2) –	1.1613e-1 (1.04e-2) –	1.5093e-1 (8.33e-3) –	3.4060e-3 (1.14e-4)
DASCMOP3	2	30	7.3272e-3 (1.27e-3) –	1.3484e-1 (1.94e-2) –	1.7730e-1 (2.37e-2) –	1.8923e-1 (1.65e-2) –	5.7692e-3 (2.16e-4)
DASCMOP4	2	30	4.7675e-3 (1.11e-2) –	1.9488e-3 (3.30e-4) –	3.8013e-3 (1.12e-3) –	3.0618e-2 (8.18e-2) –	7.4424e-4 (4.18e-5)
DASCMOP5	2	30	1.6746e-2 (4.57e-2) –	2.8044e-3 (2.40e-4) –	4.3119e-3 (8.23e-4) –	6.2457e-2 (1.37e-1) =	1.9990e-3 (7.36e-5)
DASCMOP6	2	30	1.9266e-2 (4.96e-2) –	6.8234e-3 (2.85e-3) –	9.6226e-3 (4.88e-3) –	3.6455e-1 (1.77e-1) –	5.2899e-3 (6.93e-5)
DASCMOP7	3	30	7.0676e-2 (6.51e-2) –	2.7887e-2 (8.12e-3) –	2.8758e-2 (1.75e-3) –	2.3137e-2 (6.45e-4) +	2.6352e-2 (1.06e-3)
DASCMOP8	3	30	5.7750e-2 (7.81e-2) –	2.0734e-2 (3.71e-3) –	2.0773e-2 (8.55e-4) –	1.8147e-2 (1.02e-3) +	2.0001e-2 (6.58e-4)
DASCMOP9	3	30	2.1535e-2 (1.05e-3) –	5.2674e-2 (6.75e-2) –	2.7754e-2 (1.40e-3) –	2.3073e-1 (2.42e-2) –	1.9959e-2 (1.17e-3)
+/-/=			2/27/4	3/26/4	3/28/2	3/28/2	

Table 2 HV results of ten comparison algorithms on CF, LIRCMOP, and DASCMP test suites

Problem	m	n	DCNSGAlII	BiCo	EMCMO	CMEGL	MTCMO
CF1	2	10	5.5875e-1 (1.56e-3) -	5.4497e-1 (2.59e-3) -	5.5856e-1 (8.03e-4) -	5.6003e-1 (7.07e-4) -	5.5279e-1 (2.41e-3) -
CF2	2	10	6.2889e-1 (1.79e-2) -	5.9934e-1 (2.60e-2) -	6.3148e-1 (2.27e-2) -	6.2615e-1 (2.51e-2) -	6.1375e-1 (3.11e-2) -
CF3	2	10	1.7831e-1 (4.46e-2) -	1.7554e-1 (4.74e-2) -	1.7991e-1 (3.99e-2) -	1.6199e-1 (4.81e-2) -	1.7306e-1 (3.93e-2) -
CF4	2	10	4.0814e-1 (3.35e-2) -	4.1452e-1 (2.92e-2) -	4.3223e-1 (2.97e-2) -	4.1505e-1 (3.35e-2) -	4.1346e-1 (3.38e-2) -
CF5	2	10	2.6931e-1 (8.55e-2) -	2.7629e-1 (5.95e-2) -	3.0331e-1 (6.79e-2) -	2.8917e-1 (8.53e-2) -	2.8781e-1 (7.26e-2) -
CF6	2	10	6.5384e-1 (1.05e-2) -	6.3555e-1 (1.67e-2) -	6.5468e-1 (1.28e-2) -	6.5067e-1 (1.25e-2) -	6.5200e-1 (1.43e-2) -
CF7	2	10	4.2039e-1 (1.25e-1) -	4.5388e-1 (9.53e-2) -	4.6417e-1 (9.97e-2) -	4.5433e-1 (9.57e-2) -	4.6367e-1 (1.14e-1) -
CF8	3	10	2.3627e-1 (7.02e-2) -	3.1381e-1 (4.06e-2) +	2.6173e-1 (9.95e-2) =	2.8173e-1 (8.09e-2) =	2.9775e-1 (5.78e-2) +
CF9	3	10	3.2533e-1 (8.04e-2) =	4.3025e-1 (1.19e-2) +	4.0248e-1 (3.99e-2) +	4.2359e-1 (1.85e-2) +	4.2748e-1 (1.46e-2) +
CF10	3	10	2.0581e-1 (8.83e-2) =	9.8660e-2 (1.17e-2) =	1.6021e-1 (4.20e-2) =	1.5008e-1 (4.30e-2) =	1.3038e-1 (3.13e-2) -
LIRCMOP1	2	30	1.4888e-1 (1.20e-2) -	1.3525e-1 (6.61e-3) -	1.3543e-1 (1.52e-2) -	1.7013e-1 (1.13e-2) -	1.6684e-1 (1.11e-2) -
LIRCMOP2	2	30	2.7293e-1 (5.02e-3) -	2.5724e-1 (9.23e-3) -	2.4636e-1 (1.86e-2) -	2.9041e-1 (1.01e-2) -	2.9208e-1 (7.68e-3) -
LIRCMOP3	2	30	1.3169e-1 (1.01e-2) -	1.2653e-1 (9.78e-3) -	1.0862e-1 (1.42e-2) -	1.4454e-1 (1.20e-2) -	1.4651e-1 (1.01e-2) -
LIRCMOP4	2	30	2.3488e-1 (6.86e-3) -	2.2475e-1 (9.85e-3) -	2.0192e-1 (1.83e-2) -	2.4287e-1 (1.55e-2) -	2.4312e-1 (8.91e-3) -
LIRCMOP5	2	30	6.4156e-2 (7.82e-2) -	0.0000e+0 (0.00e+0) -	1.5904e-1 (1.44e-2) -	1.4478e-1 (4.80e-2) -	9.4380e-2 (7.82e-2) -
LIRCMOP6	2	30	5.2857e-2 (5.00e-2) -	0.0000e+0 (0.00e+0) -	1.1434e-1 (1.30e-2) -	9.1960e-2 (4.46e-2) -	5.4250e-2 (4.80e-2) -
LIRCMOP7	2	30	2.4449e-1 (6.62e-3) -	1.4855e-1 (1.18e-1) -	2.5011e-1 (8.71e-3) -	2.5037e-1 (8.50e-3) -	2.4719e-1 (7.79e-3) -
LIRCMOP8	2	30	2.2984e-1 (8.64e-3) -	8.9778e-2 (1.09e-1) -	2.3879e-1 (1.38e-2) -	2.3662e-1 (9.60e-3) -	2.3382e-1 (1.03e-2) -
LIRCMOP9	2	30	2.3138e-1 (9.36e-2) -	1.3538e-1 (6.69e-2) -	3.7205e-1 (6.48e-2) -	4.0695e-1 (7.93e-2) -	1.8554e-1 (6.95e-2) -
LIRCMOP10	2	30	3.9287e-1 (1.67e-1) -	6.9293e-2 (1.92e-2) -	6.0283e-1 (7.29e-2) -	5.8057e-1 (4.82e-2) -	2.3195e-1 (1.87e-1) -
LIRCMOP11	2	30	5.3181e-1 (9.06e-2) -	2.5253e-1 (1.10e-1) -	6.6566e-1 (2.03e-2) -	6.5937e-1 (2.49e-2) -	4.0965e-1 (1.41e-1) -
LIRCMOP12	2	30	4.6301e-1 (5.15e-2) -	3.6897e-1 (1.14e-1) -	5.0082e-1 (3.65e-2) -	5.4844e-1 (3.60e-2) -	4.5706e-1 (4.09e-2) -
LIRCMOP13	3	30	4.1362e-4 (3.51e-5) -	1.7457e-2 (8.49e-2) -	5.5964e-1 (1.19e-3) +	5.5566e-1 (1.29e-3) +	1.3369e-4 (1.18e-4) -
LIRCMOP14	3	30	9.2851e-4 (7.25e-5) -	4.4437e-4 (3.58e-4) -	5.5431e-1 (1.18e-3) +	5.5499e-1 (1.36e-3) +	5.3109e-4 (3.39e-4) -
DASCMP1	2	30	1.0521e-2 (1.25e-2) -	7.6302e-3 (3.05e-3) -	8.4168e-3 (7.65e-3) -	2.0818e-2 (2.17e-2) -	1.2879e-2 (1.71e-2) -
DASCMP2	2	30	2.5053e-1 (5.90e-3) -	2.4980e-1 (4.39e-3) -	2.5865e-1 (2.58e-3) -	2.6587e-1 (4.45e-3) -	2.6528e-1 (4.69e-3) -
DASCMP3	2	30	2.2331e-1 (1.59e-2) -	2.1226e-1 (9.03e-3) -	2.1100e-1 (9.70e-3) -	2.2385e-1 (1.59e-2) -	2.1897e-1 (8.97e-3) -
DASCMP4	2	30	1.9604e-1 (3.94e-3) -	1.6325e-1 (4.97e-2) -	2.0062e-1 (4.30e-3) =	2.0332e-1 (2.34e-3) =	2.0341e-1 (2.77e-3) -
DASCMP5	2	30	3.4892e-1 (8.95e-4) -	3.1317e-1 (8.11e-2) -	3.5132e-1 (2.77e-4) =	3.5143e-1 (1.24e-4) =	3.5134e-1 (1.14e-3) -
DASCMP6	2	30	3.0374e-1 (1.24e-2) -	2.2916e-1 (1.01e-1) -	3.0337e-1 (1.24e-2) -	3.0932e-1 (6.39e-3) -	2.3946e-1 (9.53e-2) -
DASCMP7	3	30	2.8096e-1 (1.81e-3) -	2.8598e-1 (5.02e-3) -	2.8820e-1 (6.61e-4) +	2.8822e-1 (4.30e-4) +	2.8849e-1 (3.26e-4) +
DASCMP8	3	30	2.0255e-1 (9.36e-4) -	2.0641e-1 (8.28e-4) +	2.0712e-1 (4.80e-4) +	2.0690e-1 (4.45e-4) +	2.0733e-1 (3.64e-4) +

Table 2 continued

Problem	<i>m</i>	<i>n</i>	DCNSGAIH	BiCo	EMCMO	CMEGL	MTCMO
DASCMOP9	3	30	1.1127e-1 (1.94e-2) – 0/31/2	1.2554e-1 (9.83e-3) – 3/29/1	1.2980e-1 (1.36e-2) – 5/24/4	1.5006e-1 (9.03e-3) – 5/24/4	1.3699e-1 (1.14e-2) – 4/29/0
+/-/=							
Problem	<i>m</i>	<i>n</i>	DRLOSEMCMO	CSEMT	DEST	APSEA	AMTCMO
CF1	2	10	5.6340e-1 (4.19e-4) +	5.6279e-1 (6.74e-4) +	5.5690e-1 (9.76e-4) –	5.4753e-1 (1.84e-3) –	5.6193e-1 (6.10e-4)
CF2	2	10	6.7047e-1 (4.47e-3) –	6.7419e-1 (1.16e-3) –	6.1191e-1 (2.85e-2) –	6.3608e-1 (1.61e-2) –	6.7633e-1 (1.32e-3)
CF3	2	10	2.1393e-1 (5.98e-2) =	2.1472e-1 (4.34e-2) =	1.7495e-1 (2.65e-2) –	1.7722e-1 (5.49e-2) –	2.1226e-1 (4.10e-2)
CF4	2	10	4.7792e-1 (1.31e-2) –	4.8010e-1 (2.61e-2) =	3.9593e-1 (3.95e-2) –	4.1975e-1 (3.52e-2) –	4.9016e-1 (8.46e-3)
CF5	2	10	3.0640e-1 (8.90e-2) =	3.5316e-1 (6.68e-2) =	2.5476e-1 (8.25e-2) –	2.9026e-1 (7.71e-2) –	3.4867e-1 (7.29e-2)
CF6	2	10	6.6706e-1 (7.63e-3) –	6.6132e-1 (1.40e-2) –	6.5340e-1 (1.44e-2) –	6.5356e-1 (1.31e-2) –	6.7561e-1 (5.31e-3)
CF7	2	10	5.1586e-1 (5.30e-2) –	4.6661e-1 (8.00e-2) –	4.4926e-1 (8.39e-2) –	4.4591e-1 (9.80e-2) –	5.5350e-1 (3.22e-2)
CF8	3	10	2.9025e-1 (2.81e-2) +	2.0917e-1 (5.23e-2) –	2.8229e-1 (1.18e-1) +	2.6897e-1 (5.67e-2) =	2.7769e-1 (3.22e-2)
CF9	3	10	3.8665e-1 (1.78e-2) +	3.6891e-1 (3.37e-2) =	4.0284e-1 (3.27e-2) +	4.2350e-1 (1.44e-2) +	3.4910e-1 (3.71e-2)
CF10	3	10	1.5362e-1 (4.01e-2) =	1.3251e-1 (5.95e-2) –	1.5551e-1 (3.78e-2) =	1.2300e-1 (4.08e-2) –	1.6845e-1 (4.81e-2)
LRCMOP1	2	30	1.3474e-1 (1.33e-2) –	1.7652e-1 (1.40e-2) –	2.0001e-1 (1.31e-2) –	1.1084e-1 (1.01e-2) –	2.3520e-1 (8.13e-4)
LRCMOP2	2	30	2.7039e-1 (2.82e-2) –	2.8744e-1 (2.97e-2) –	3.2196e-1 (1.29e-2) –	2.3674e-1 (1.82e-2) –	3.5724e-1 (8.19e-4)
LRCMOP3	2	30	1.1966e-1 (1.61e-2) –	1.3624e-1 (1.70e-2) –	1.6174e-1 (1.86e-2) –	1.0484e-1 (1.38e-2) –	2.0386e-1 (6.94e-4)
LRCMOP4	2	30	2.1322e-1 (1.87e-2) –	2.3914e-1 (2.35e-2) –	2.6496e-1 (1.52e-2) –	2.0222e-1 (1.73e-2) –	3.1172e-1 (1.67e-3)
LRCMOP5	2	30	2.8519e-1 (4.11e-3) –	2.2619e-1 (1.10e-1) –	1.5217e-1 (2.00e-2) –	2.1441e-2 (5.82e-2) –	2.9049e-1 (4.04e-4)
LRCMOP6	2	30	1.9385e-1 (5.83e-4) –	1.2779e-1 (6.22e-2) –	1.1346e-1 (1.70e-2) –	1.3262e-2 (3.10e-2) –	1.9598e-1 (2.27e-4)
LRCMOP7	2	30	2.9363e-1 (6.70e-4) –	2.5379e-1 (1.16e-2) –	2.4653e-1 (7.90e-3) –	2.5168e-1 (1.09e-2) –	2.9401e-1 (2.43e-4)
LRCMOP8	2	30	2.9379e-1 (2.13e-4) –	2.5300e-1 (2.11e-2) –	2.3444e-1 (1.05e-2) –	2.3705e-1 (1.33e-2) –	2.9411e-1 (1.17e-4)
LRCMOP9	2	30	4.4590e-1 (1.84e-2) –	4.3110e-1 (5.05e-2) –	4.1872e-1 (7.57e-2) –	2.4464e-1 (1.04e-1) –	5.6143e-1 (4.38e-3)
LRCMOP10	2	30	6.8382e-1 (1.31e-2) –	6.3653e-1 (7.48e-2) –	6.0683e-1 (5.00e-2) –	1.7463e-1 (1.65e-1) –	7.0442e-1 (4.84e-4)
LRCMOP11	2	30	6.8171e-1 (7.61e-3) –	6.4818e-1 (4.62e-2) –	6.5857e-1 (1.74e-1) –	3.2018e-1 (1.08e-1) –	6.9301e-1 (9.71e-4)
LRCMOP12	2	30	5.7497e-1 (1.62e-2) –	5.4195e-1 (9.37e-3) –	5.5417e-1 (2.92e-2) –	4.6693e-1 (7.06e-2) –	6.1639e-1 (9.56e-3)
LRCMOP13	3	30	5.3259e-1 (4.35e-3) =	5.5767e-1 (1.25e-3) +	5.5534e-1 (1.58e-3) +	1.4033e-4 (1.28e-4) –	5.3283e-1 (3.45e-3)
LRCMOP14	3	30	5.4843e-1 (1.79e-3) =	5.5726e-1 (1.91e-3) +	5.5454e-1 (9.67e-4) +	5.9761e-4 (3.02e-4) –	5.4721e-1 (2.36e-3)
DASCMOP1	2	30	2.1118e-1 (7.40e-4) –	1.1161e-1 (9.31e-2) –	4.1197e-2 (3.05e-2) –	7.6847e-3 (5.88e-3) –	2.1212e-1 (4.26e-4)
DASCMOP2	2	30	3.5395e-1 (5.18e-4) –	3.0011e-1 (3.82e-2) –	2.7770e-1 (3.28e-3) –	2.5203e-1 (3.56e-3) –	3.5517e-1 (7.36e-5)
DASCMOP3	2	30	3.1025e-1 (1.35e-3) –	2.4136e-1 (1.38e-2) –	2.1522e-1 (1.14e-2) –	2.1032e-1 (9.12e-3) –	3.1183e-1 (3.44e-4)
DASCMOP4	2	30	1.9785e-1 (1.02e-2) –	2.0263e-1 (4.94e-4) –	1.9978e-1 (3.60e-3) –	1.9134e-1 (3.16e-2) =	2.0394e-1 (2.05e-4)
DASCMOP5	2	30	3.4062e-1 (3.00e-2) –	3.5049e-1 (2.92e-4) –	3.4944e-1 (5.84e-4) –	3.1362e-1 (8.35e-2) =	3.5135e-1 (1.59e-4)
DASCMOP6	2	30	3.0418e-1 (2.49e-2) –	3.1087e-1 (3.88e-3) –	3.0818e-1 (6.21e-3) –	1.2554e-1 (8.84e-2) –	3.1230e-1 (1.49e-4)
DASCMOP7	3	30	2.6409e-1 (3.26e-2) –	2.8546e-1 (3.70e-3) =	2.8484e-1 (9.00e-4) –	2.8862e-1 (2.95e-4) +	2.8699e-1 (3.81e-4)
DASCMOP8	3	30	1.8442e-1 (4.05e-2) –	2.0548e-1 (2.21e-3) –	2.0525e-1 (4.44e-4) –	2.0760e-1 (4.76e-4) +	2.0591e-1 (3.28e-4)
DASCMOP9	3	30	2.0472e-1 (5.12e-4) –	1.9380e-1 (2.27e-2) –	2.0064e-1 (8.71e-4) –	1.3160e-1 (7.78e-3) –	2.0568e-1 (4.61e-4)
+/-/=			3/25/5	3/25/5	4/28/1	3/27/3	

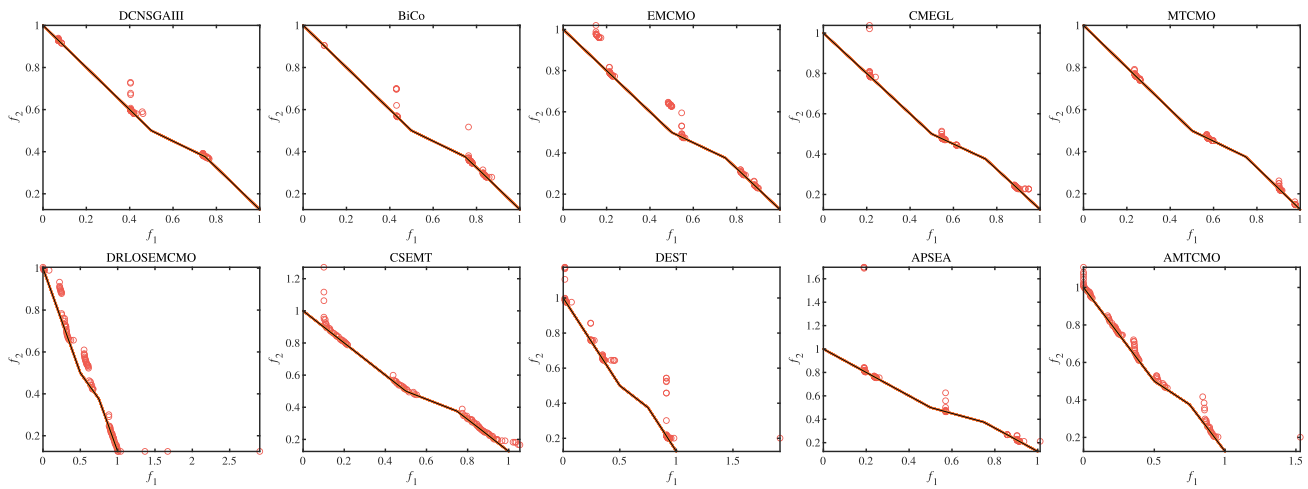


Fig. 4 The distribution of the main population formed by ten comparison algorithms on CF4.

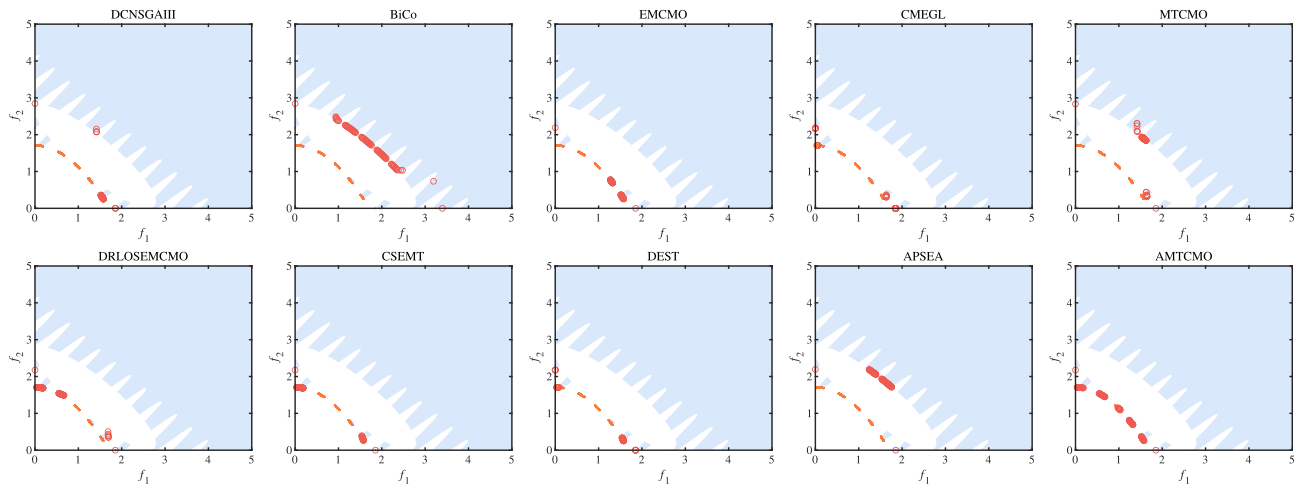


Fig. 5 The distribution of the main population formed by ten comparison algorithms on LIRCMP9

a diversity-enhanced framework to improve the exploration of CPF, performing well in such test problems. The UPF and CPF of LIRCMP5-6 are the same and both are located in the feasible regions, which is beneficial to the algorithm using unconstrained auxiliary population. However, these algorithms lack the use of high-quality infeasible solutions. The loss of diversity leads to an incomplete CPF exploration. The UPF is far from the CPF in LIRCMP7-8, which cannot provide effective information. DRLOSEMCMO uses the preference of the operator for the LIRCMP7-8, which can select the appropriate operator to generate offspring, achieving a good distribution on CPF. LIRCMP9-12 uses global constraints and local constraints to divide CPF into multiple discontinuous segments, which requires the algorithm not only to cross large infeasible regions but also to maintain good diversity. The proposed AMTCMO can select appropriate auxiliary task for the main population at different evolutionary stages with the assistance of the attention

mechanism, which achieves good results. LIRCMP13-14 are all three-objective high-dimensional complex constraint problems, which pose a great challenge to the ability of the algorithm to deal with constraints. The auxiliary population of CSEMT only considers a single constraint, which is easy to search in high-dimensional space and thus makes the CSEMT performs well.

4.2.3 Results for DASCMP

For the DASCMP test suite, AMTCMO achieves the best IGD+ results in 6 of the 9 test problems. In addition, the best HV results are obtained on 6 test problems. Fig. 6 shows the distribution of the solution obtained by the ten comparison algorithms. It is obvious that AMTCMO distributes evenly, which indicates that AMTCMO maintains a good diversity. DASCMP1-3 is mainly a test of the diversity of algorithms. However, in these test problems,

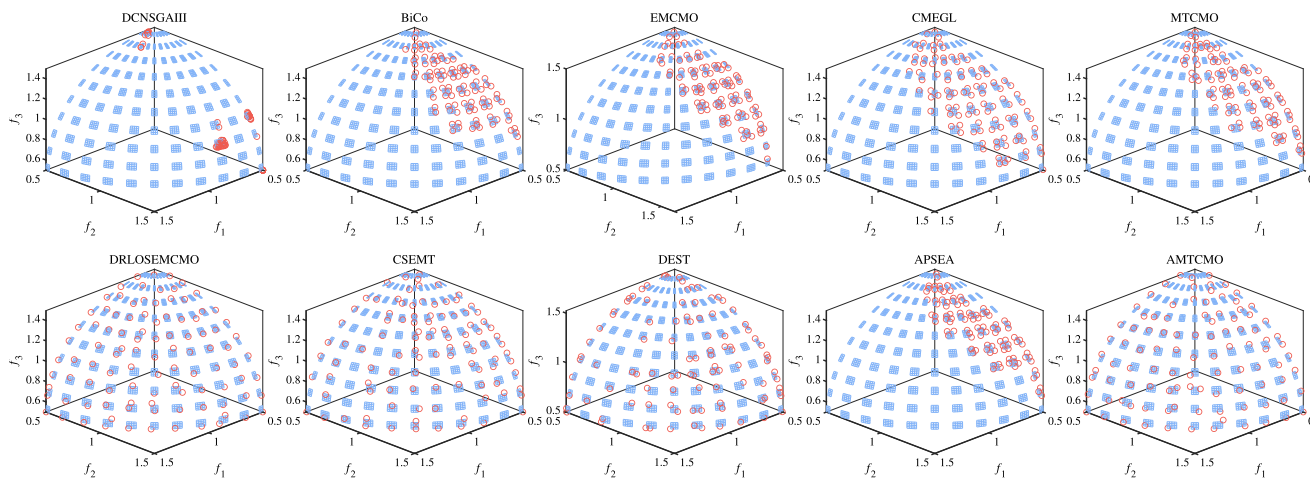


Fig. 6 The distribution of the main population formed by ten comparison algorithms on DASCMP9

the ratio of the feasible region to the search space is very low, and many algorithms fall into local optima. This is because the unconstrained auxiliary population they use quickly approach the UPF during the search process and ignore some potential feasible regions. AMTCMO can give up the selection of unconstrained auxiliary population in this case, and use the constrained relaxation population to explore potential infeasible solutions to enhance diversity. The constraints in DASCMP4-6 form many obstacles to prevent the convergence of the algorithm. CMEGL, EMCMO and these CMOEAs based on unconstrained population assistance quickly achieve convergence with the help of unconstrained auxiliary populations. DASCMP7-9 both contain three objectives, which increase the difficulty in both convergence and diversity. Due to the increased search difficulty, the algorithms consume more function evaluations. APSEA dynamically adjusts the size of the auxiliary population using an adaptive population size method, which saves the number of function evaluations and ensures the convergence of the main population. AMTCMO slightly lags behind APSEA and CMEGL on DASCMP7-8, but achieves the best results on DASCMP9. The proposed AMTCMO uses the hybrid DQL model with the attention mechanism to adaptively select auxiliary task. Compared with other EMT-based algorithms that parallel multiple auxiliary tasks, it has certain potential in high-dimensional problems.

4.2.4 Results for LSCM

Tables 3 and 4 show the IGD+ and HV results of the ten comparison algorithms on the LSCM test suite. The best results for each test problem are labelled using bold in the tables in Tables 3 and 4. Table 3 indicates that AMTCMO is superior to DCNSGAIH, BiCo, EMCMO, CMEGL, MTCMO, DRLOSEMCMO, CSEMT, DEST and APSEA on 9, 8, 8,

9, 9, 8, 10 and 8 test problems, respectively. The results of Table 4 demonstrate that AMTCMO is superior to these comparison algorithms on 4, 3, 3, 4, 4, 3, 6 and 3 test problems, respectively. In addition, AMTCMO achieves the best results on IGD+ of 7 test problems and HV of 3 test problems. Fig. 7 shows the box plot of the rank of comparison algorithms after the Friedman test. It can be seen that AMTCMO ranks best among the comparison algorithms. The high-latitude decision space of LSCM leads to complex constraint functions. Therefore, the test suite poses a great challenge to the convergence of the algorithm. Many algorithms use unconstrained auxiliary population to help the algorithm achieve convergence quickly. However, there is a nonlinear relationship between the objectives and constraints of some test problems in LSCM, which makes the population unable to provide effective information. CSEMT uses a single constraint to construct the auxiliary population. It not only ensures convergence speed but also avoids being plagued by nonlinear relationships, thereby achieving good results. Furthermore, the feasible region of LSCM is complex, and there are many locally infeasible regions. This requires the algorithm to search each region carefully to find a promising search direction while maintaining a fast convergence speed. The proposed AMTCMO can switch the auxiliary population in real time according to the population state. After the unconstrained auxiliary population helps to cross the infeasible regions, the constrained relaxation population is switched to explore the regions. Therefore, AMTCMO is competitive on this test suite.

4.3 Statistics results

In order to analyze the experimental results, two nonparametric statistical analysis models are applied: Wilcoxon signed-ranks test and Friedman aligned test. IGD+ and HV

Table 3 IGD+ results of ten comparison algorithms on LSCM test suite

Problem	M	D	DCNSGAIII	BiCo	EMCMO	CMEGL	MTCMO
LSCM1	2	100	3.9714e-1 (1.93e-2) -	3.9698e-1 (3.21e-2) -	3.6574e-1 (1.91e-2) -	3.6991e-1 (1.93e-2) -	3.6710e-1 (2.15e-2) -
LSCM2	2	100	1.3919e+2 (2.02e+2) -	4.4923e+1 (3.91e+1) -	2.3612e+2 (2.47e+2) -	2.0546e+1 (1.16e+1) -	1.3203e+1 (1.30e+1) -
LSCM3	2	100	1.3290e+0 (8.72e-1) +	6.3203e-1 (1.64e-1) +	1.0942e+0 (1.21e+0) +	1.9700e+0 (1.54e+0) =	1.8052e+0 (9.59e-1) =
LSCM4	2	100	2.5208e+1 (1.33e+0) -	5.7899e+1 (2.91e+1) -	1.6382e+2 (1.27e+2) -	4.1688e+1 (2.51e+1) -	3.6751e+1 (3.04e+1) -
LSCM5	2	100	1.8763e+0 (2.76e-1) -	1.9372e+0 (2.22e-1) -	1.8583e+0 (2.84e-1) -	1.9442e+0 (2.81e-1) -	1.9374e+0 (2.18e-1) -
LSCM6	2	100	3.7492e-1 (6.88e-2) -	3.4020e-1 (9.29e-2) -	3.7104e-1 (1.34e-1) -	3.1030e-1 (4.95e-2) -	3.1225e-1 (4.25e-2) -
LSCM7	2	100	5.8088e-1 (2.31e-1) +	7.9761e-1 (1.70e-1) =	8.3663e-1 (1.56e-1) =	8.3474e-1 (2.14e-1) =	4.6756e-1 (1.05e-1) +
LSCM8	2	100	6.7193e-1 (6.12e-1) =	4.8351e-1 (7.54e-2) +	5.1212e-1 (4.41e-2) =	5.3323e-1 (1.39e-1) =	3.9393e-1 (1.00e-1) +
LSCM9	2	100	5.6745e+2 (6.74e+2) -	1.8482e+2 (2.31e+2) -	2.6890e+2 (2.76e+2) -	4.1152e+1 (3.89e+1) -	1.0513e+2 (1.77e+2) -
LSCM10	3	100	8.6324e-1 (3.90e-1) -	6.2767e-1 (2.53e-1) =	7.0198e-1 (4.06e-1) =	1.2103e+0 (6.78e-1) -	7.3011e-1 (2.55e-1) -
LSCM11	3	100	7.1228e+2 (6.75e+2) -	1.0364e+2 (7.14e+1) -	2.2103e+2 (1.93e+2) -	4.2132e+1 (2.43e+1) -	5.2195e+1 (5.43e+1) -
LSCM12	3	100	6.4020e+2 (7.88e+2) -	4.1410e+1 (3.31e+1) -	1.6220e+2 (1.69e+2) -	1.0046e+1 (7.77e+0) -	1.2058e+1 (1.26e+1) -
+/-/=			2/9/1	2/8/2	1/8/3	0/9/3	2/9/1

Problem	M	D	DRLOSEMCMO	CSEMT	DEST	APSEA	AMTCMO
LSCM1	2	100	3.5125e-1 (1.50e-2) -	3.4595e-1 (1.34e-2) -	3.1384e-1 (1.93e-2) =	3.6828e-1 (1.38e-2) -	3.0907e-1 (1.15e-2)
LSCM2	2	100	1.0259e+1 (7.77e+0) -	2.0989e+1 (9.86e+1) -	5.8901e+2 (8.17e+2) -	5.5474e+1 (8.34e+1) -	4.7249e+0 (2.50e+0)
LSCM3	2	100	4.6469e+0 (2.51e+0) -	6.7426e-1 (9.16e-2) +	NaN (NaN)	6.9486e-1 (2.64e-1) +	1.5795e+0 (4.67e-1)
LSCM4	2	100	2.5224e+1 (5.49e+0) -	1.2861e+1 (1.40e+1) -	6.3346e+2 (6.40e+2) -	6.2542e+1 (4.95e+1) -	1.2767e+1 (3.55e+0)
LSCM5	2	100	1.7607e+0 (3.11e-1) -	1.5100e+0 (3.00e-1) -	1.5590e+0 (1.92e-1) -	1.8806e+0 (2.83e-1) -	1.0764e+0 (3.08e-1)
LSCM6	2	100	3.0872e-1 (6.49e-2) -	3.3357e-1 (1.01e-1) -	5.9618e-1 (2.48e-1) -	3.0112e-1 (6.83e-2) -	2.3520e-1 (2.06e-2)
LSCM7	2	100	1.0007e+0 (7.52e-2) -	8.4047e-1 (2.35e-1) =	1.2882e+0 (3.64e-1) -	8.3967e-1 (2.21e-1) =	8.5764e-1 (1.77e-1)
LSCM8	2	100	5.2368e-1 (5.19e-2) =	4.9287e-1 (4.65e-2) +	6.5910e+0 (1.96e+0) -	5.0866e-1 (6.00e-2) =	5.3926e-1 (6.65e-2)
LSCM9	2	100	1.0582e+0 (1.78e+0) +	1.9508e+1 (8.60e+1) -	2.5004e+2 (2.45e+2) -	2.8022e+2 (4.46e+2) -	3.6384e+0 (6.28e+0)
LSCM10	3	100	1.2751e+0 (3.78e-1) -	1.3274e+0 (9.10e-1) -	6.2450e+0 (4.20e+0) -	6.7402e-1 (4.32e-1) =	5.2725e-1 (1.12e-1)
LSCM11	3	100	1.8364e+0 (1.64e+0) +	3.1717e+0 (3.20e+0) +	2.7566e+2 (1.59e+2) -	1.3030e+2 (1.21e+2) -	1.6186e+1 (9.24e+0)
LSCM12	3	100	1.3775e+0 (5.18e-1) -	9.6096e-1 (1.19e-1) -	7.6764e+2 (6.55e+2) -	3.1268e+1 (2.23e+1) -	7.4313e-1 (2.82e-1)
+/-/=			2/9/1	3/8/1	0/10/1	1/8/3	

Table 4 HV results of ten comparison algorithms on LSCM test suite

Problem	M	D	DCNSGAIH	BiCo	EMCMO	CMEGL	MTCMO
LSCM1	2	100	1.4261e-1 (2.78e-2) -	1.3616e-1 (3.88e-2) -	1.5646e-1 (2.74e-2) -	1.4842e-1 (2.67e-2) -	1.4343e-1 (3.03e-2) -
LSCM2	2	100	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =
LSCM3	2	100	0.0000e+0 (0.00e+0) =	1.3697e-2 (2.80e-2) +	3.6401e-3 (9.20e-3) +	4.3946e-3 (1.45e-2) =	0.0000e+0 (0.00e+0) =
LSCM4	2	100	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =
LSCM5	2	100	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =
LSCM6	2	100	1.6058e-1 (4.74e-2) -	1.8138e-1 (8.52e-2) -	1.5687e-1 (7.67e-2) -	1.6449e-1 (5.78e-2) -	1.6993e-1 (6.36e-2) -
LSCM7	2	100	4.6286e-2 (5.37e-2) +	1.0768e-2 (1.66e-2) =	4.9807e-3 (1.33e-2) =	1.0905e-2 (2.08e-2) =	9.7680e-2 (6.38e-2) +
LSCM8	2	100	3.8171e-2 (4.06e-2) =	3.4376e-2 (4.07e-2) =	2.6235e-2 (2.86e-2) =	2.3667e-2 (3.21e-2) =	4.4061e-2 (4.03e-2) =
LSCM9	2	100	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	1.2283e-2 (3.26e-2) =
LSCM10	3	100	1.1728e-1 (1.56e-1) -	1.6369e-1 (1.14e-1) =	1.7016e-1 (1.14e-1) =	5.3496e-2 (7.87e-2) -	1.0904e-1 (9.93e-2) -
LSCM11	3	100	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =
LSCM12	3	100	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	6.5467e-3 (2.39e-2) -	6.4556e-3 (2.38e-2) -
+/-/=			1/4/7	1/3/8	1/3/8	0/4/8	1/4/7
Problem	M	D	1.9838e-1 (1.14e-2) -	1.9940e-1 (9.99e-3) -	1.8866e-1 (2.26e-2) -	1.4917e-1 (2.01e-2) -	2.2458e-1 (9.15e-3)
LSCM1	2	100	0.0000e+0 (0.00e+0) =	6.5090e-2 (3.42e-2) +	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	4.2349e-4 (2.07e-3)
LSCM2	2	100	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	NaN (NaN)	1.5966e-2 (2.48e-2) +	0.0000e+0 (0.00e+0)
LSCM3	2	100	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
LSCM4	2	100	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
LSCM5	2	100	2.2829e-1 (5.84e-2) -	2.1045e-1 (8.50e-2) -	3.4166e-2 (5.12e-2) -	2.0122e-1 (6.12e-2) -	2.8155e-1 (3.37e-2)
LSCM6	2	100	7.9626e-5 (3.90e-4) -	6.2731e-3 (1.19e-2) =	0.0000e+0 (0.00e+0) -	1.1417e-2 (1.81e-2) =	8.0327e-3 (1.63e-2)
LSCM7	2	100	2.2348e-2 (2.78e-2) =	2.3559e-2 (2.64e-2) =	0.0000e+0 (0.00e+0) -	3.4638e-2 (3.66e-2) =	2.7513e-2 (2.49e-2)
LSCM8	2	100	2.0037e-2 (2.40e-2) +	5.8475e-2 (3.64e-2) +	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	3.6854e-3 (1.81e-2)
LSCM9	2	100	1.3233e-2 (3.35e-2) -	3.0757e-2 (4.96e-2) -	0.0000e+0 (0.00e+0) -	1.6911e-1 (9.81e-2) =	2.0185e-1 (7.46e-2)
LSCM10	3	100	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
LSCM11	3	100	8.3121e-2 (5.29e-2) =	1.4855e-1 (8.14e-3) +	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	1.1295e-1 (3.31e-2)
+/-/=			1/4/7	3/3/6	0/6/5	1/3/8	

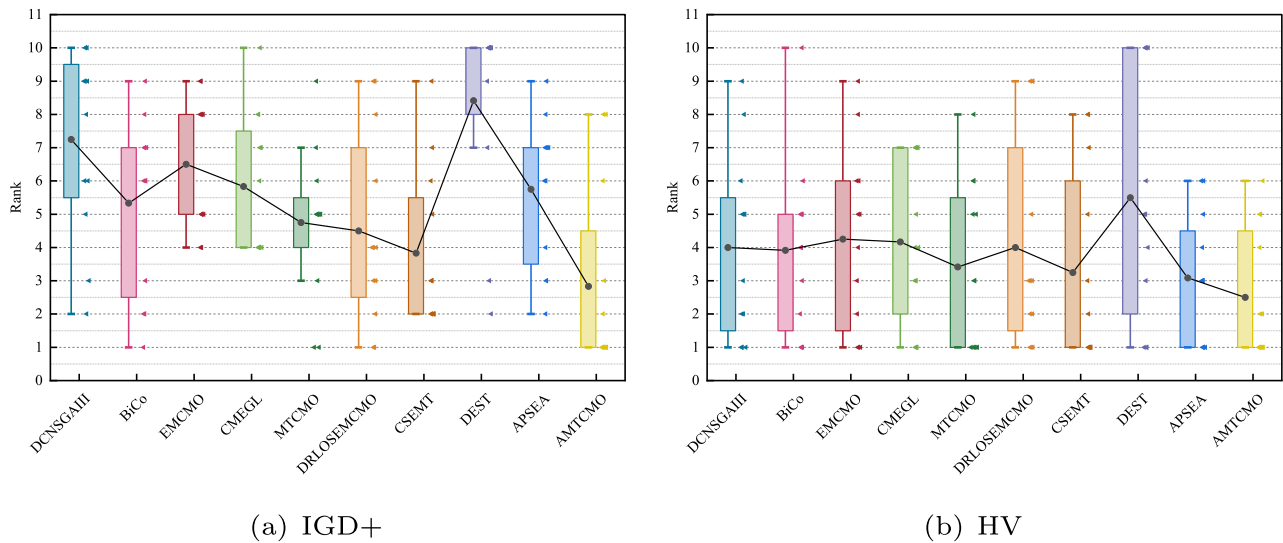


Fig. 7 The box plots of rank obtained by the Friedman test on LSCM test suite

Table 5 IGD+ results obtained by the Wilcoxon test for AMTCMO

AMTCMO VS	R^+	R^-	Exact p value	Asymptotic p value
DCNSGAIH	561.0	0.0	2.328e-10	0.000001
BiCo	552.0	9.0	7.6840e-9	0.000001
EMCMO	525.0	36.0	1.1576e-6	0.000012
CMEGL	520.0	41.0	2.3040e-6	0.000018
MTCMO	542.0	19.0	7.1480e-8	0.000003
DRLOSEMCMO	497.0	31.0	1.1060e-6	0.000013
CSEMT	515.0	46.0	4.3940e-6	0.000027
DEST	530.0	31.0	5.5300e-7	0.000008
APSEA	551.0	10.0	1.0012e-8	0.000001

Table 6 HV results obtained by the Wilcoxon test for AMTCMO

AMTCMO VS	R^+	R^-	Exact p value	Asymptotic p value
DCNSGAIH	551.0	10.0	1.0012e-8	0.000001
BiCo	511.0	17.0	9.6400e-8	0.000004
EMCMO	493.0	35.0	2.0040e-6	0.000017
CMEGL	477.5	50.5	1.5137e-5	0.000058
MTCMO	498.0	30.0	9.4760e-7	0.000011
DRLOSEMCMO	466.5	61.5	5.1620e-5	0.000084
CSEMT	503.5	57.5	1.6991e-5	0.000052
DEST	517.0	44.0	3.4100e-6	0.000019
APSEA	542.0	19.0	7.1480e-8	0.000003

results are imported into the KEEL [51] software for statistical analysis. Tables 5 and 6 demonstrate the results of the Wilcoxon signed-ranks test. A larger R^+ and R^- differences in the tables indicate that AMTCMO outperforms the comparison algorithms on more test problems. Moreover, a p -value less than 0.05 signifies a significant difference in the performance of the two comparison algorithms. It can be

seen that the exact p -values of all the comparison algorithms are significantly less than 0.05, which indicates a significant advantage of the proposed AMTCMO. Figs. 8 and 9 draw the box plot, which present the ranks of all comparison algorithms after the Friedman aligned test. The quartile interval of the rank constitutes the box, and the whisker line is the numerical range of the rank, which can show the overall dis-

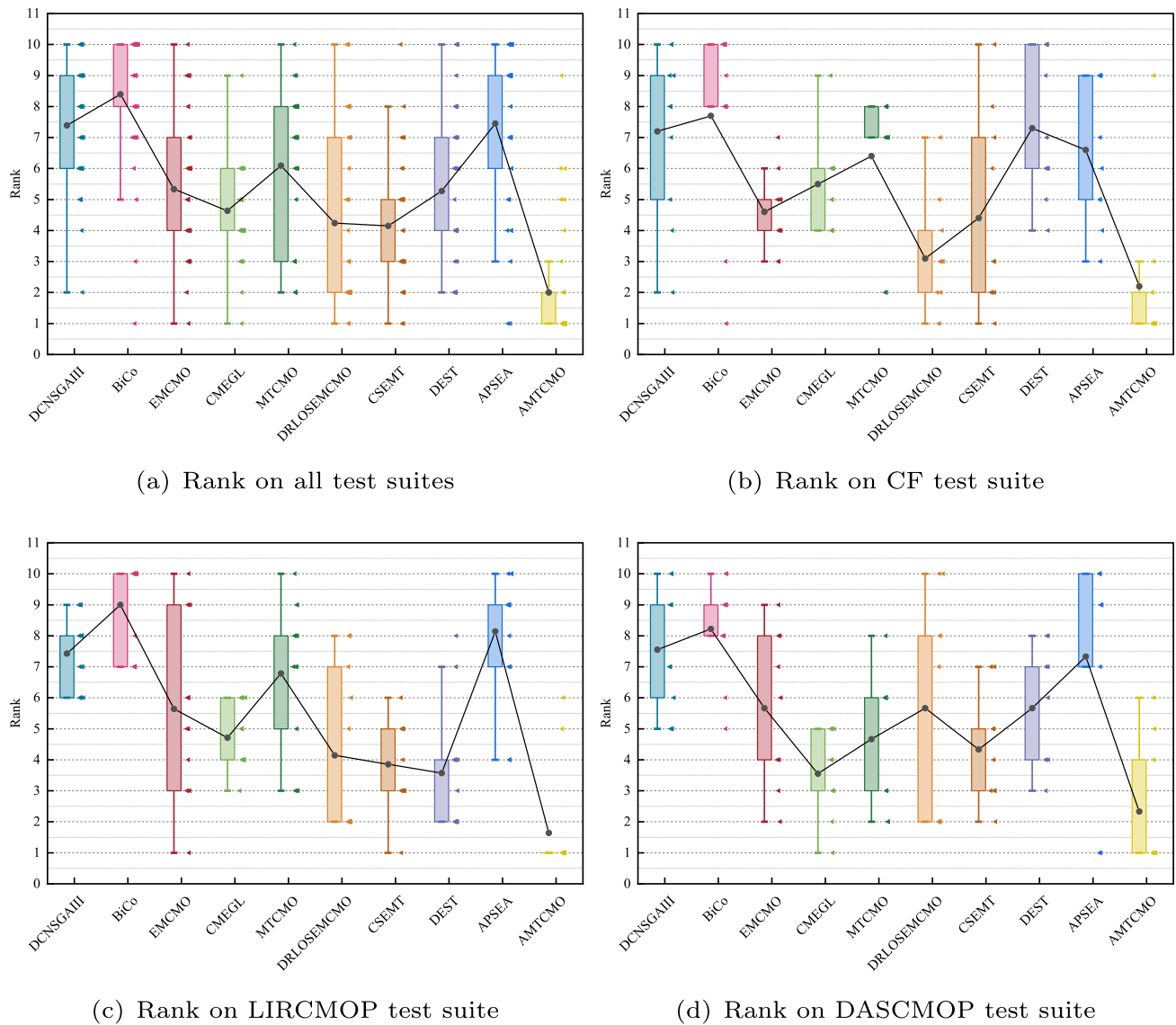


Fig. 8 The box plots of average IGD+ rank obtained by the Friedman test

person of the data. These figures connect the mean of each rank into a line chart. It is obvious that AMTCMO achieves the best rank on all the test suites. In the box chart, the lower the box denotes the better the rank, and the longer the box represents the greater the numerical fluctuation. Therefore, the excellent rank of AMTCMO on each test suite is stable. Through the above statistical tests, there are statistically significant differences between the proposed AMTCMO and the comparison algorithms, which shows the excellent performance of AMTCMO.

4.4 Ablation study

In this section, an ablation study of the proposed AMTCMO is carried out to validate the effectiveness of the pro-

posed strategy. Specifically, three variants are constructed AMTCMO_1, AMTCMO_2, and AMTCMO_3. In AMTCMO_1, the self-attention mechanism is not used for the main population to enhance the state representation. AMTCMO_2 does not use the cross-attention mechanism to calculate the attention scores between the main task and the auxiliary task to decide the selection of the auxiliary task. The self-attention mechanism and cross-attention mechanism in AMTCMO_3 are not used, and the selection of auxiliary tasks is only through the learning of the DQL model. Tables 7 and 8 present the IGD+ and HV results for the three variants. The best results for each test problem are labelled using bold in the tables in Tables 7 and 8. Tables 9 and 10 give a summary of the experimental results and show the results of the Wilcoxon signed-ranks test. It can be seen that these vari-

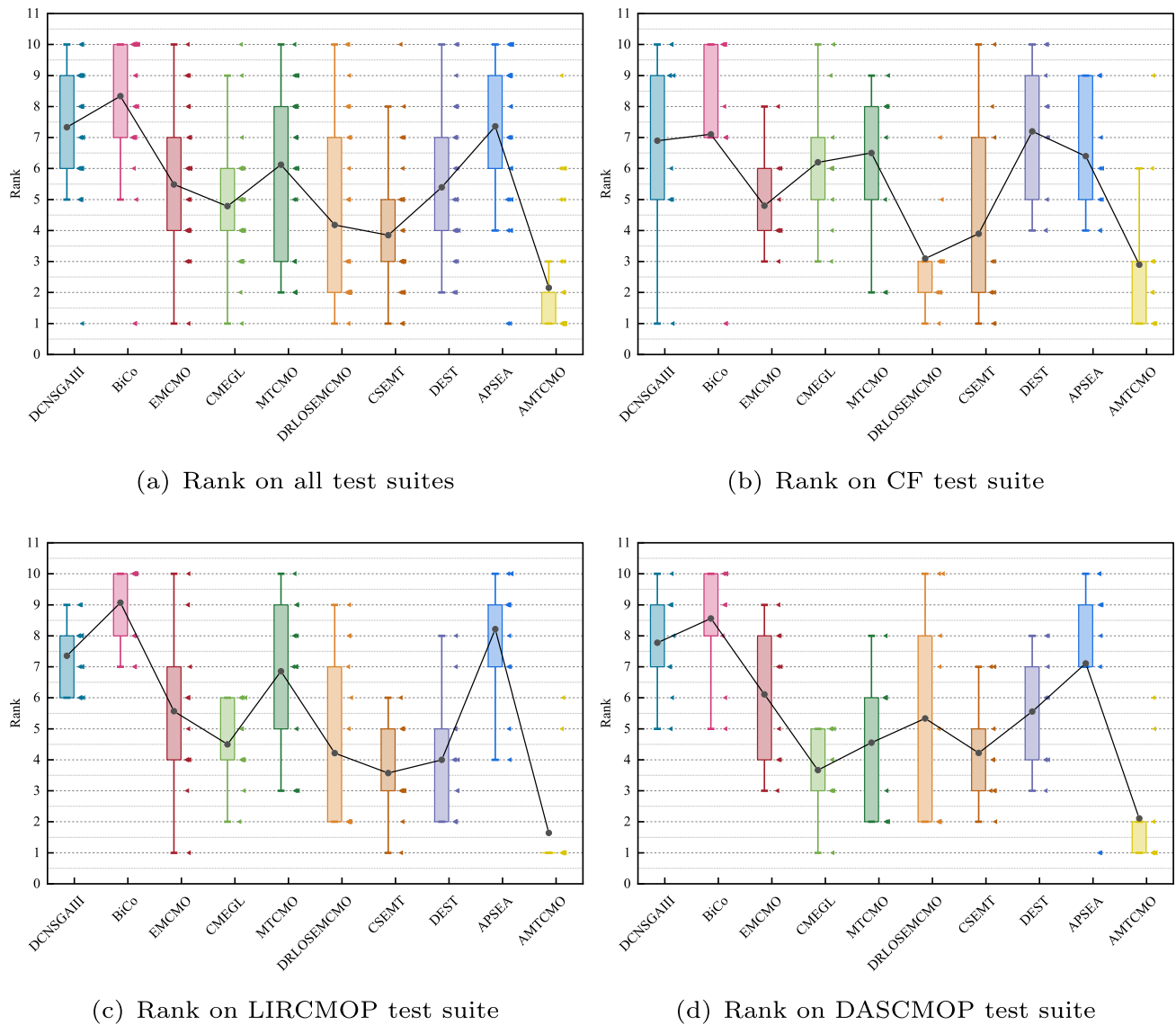


Fig. 9 The box plots of average HV rank obtained by the Friedman test

ants are inferior to the proposed AMTCMO in both IGD+ and HV, and the differences between R^+ and R^- are large. In addition, Fig. 10 illustrates the ranks of the variants and AMTCMO after the Friedman aligned test on each test suite. It is evident that AMTCMO achieves the best rank on each test suite. Thus, AMTCMO shows a significant performance improvement with the help of the self-attention mechanism and the cross-attention mechanism.

The main population state representation enhanced by the self-attention mechanism can not only generate real-time integrated population states to help the DQL better learn the appropriate actions, but also calculate the attention scores between the main task and auxiliary tasks collaborating with the cross-attention mechanism to guide the selection of tasks. Therefore, AMTCMO compared to these variants can select

the most appropriate task to assist the main task during the evolution process. In order to intuitively verify the rationality of the proposed strategy in task selection, we count the number of auxiliary tasks selected by each algorithm in the evolution process on the two test problems DASCOP4 and LIRCMOP5. DASCOP4 needs the help of unconstrained auxiliary tasks to approach CPF in the early stage of evolution due to the existence of infeasible regions. In the later stage of evolution, since CPF is divided into multiple discrete fragments, the algorithm needs to maintain diversity with the help of constraint relaxation auxiliary tasks to explore the complete CPF. The CPF and UPF of LIRCMOP5 are highly overlapping, so choosing an unconstrained auxiliary population leads to better algorithm performance. The statistical results in Figs. 11 and 12 show that AMTCMO selects the

optimal auxiliary task on both test problems, while the variant without the help of the attention mechanism is unable to identify the auxiliary task that is beneficial to the current state, and even makes a wrong decision.

AMTCMO_3 only uses the Q value but does not assist with the attention mechanism to select auxiliary tasks. However, the evaluation based only on the Q value is relatively simple. The infeasible solutions generated by auxiliary tasks may carry key boundary information or maintain diversity, and these potential values are not considered. In addition, DQL compresses multi-dimensional performance into a single Q value, which may lead to conflict of feasibility, convergence and diversity. On diversity-sensitive problems such as DASCOP4, the Q value may be biased towards short-term feasibility gains, resulting in local convergence. The introduction of the attention mechanism solves these limitations, but it brings some trouble at the same time. When the number of function evaluations is limited, the weight matrices of the attention mechanism are difficult to converge due to continuous changes of the environment. Its lag response makes the state representation and task correlation analysis deviate from the real needs and weakens the effectiveness of decision-making. Furthermore, if there is a big conflict between attention score and Q value in collaborative decision-making, the lack of arbitration criteria may reduce the stability of auxiliary task selection.

4.5 Parameter analysis

The proposed AMTCMO contains two parameters: η and λ . η is used to control the transformation of the learning and evolutionary stage, and λ is the learning rate of the weight matrices update in the attention mechanism. For η , four variants $\eta = \{0.2, 0.4, 0.5, 0.6\}$ corresponding to AMTCMO_4, AMTCMO_5, AMTCMO_6 and AMTCMO_7 are constructed. As to λ , four variants $\lambda = \{0.001, 0.1, 1, 10\}$ corresponding to AMTCMO_8, AMTCMO_9, AMTCMO_10 and AMTCMO_11 are constructed. The summary of the experimental results of each variant and the results of the Wilcoxon signed-ranks test are shown in Tables 9 and 10. At the same time, Figs. 13 and 14 show the rank of the variants after the Friedman aligned test on each test suite. It can be seen from the experimental results that the proposed algorithm achieves the best results using $\eta = 0.3$ and $\lambda = 0.01$. AMTCMO uses a hybrid DQL model enhanced by the attention mechanism, and sufficient data needs to be accumulated in the learning stage to construct the DQN model. Therefore, if the parameter set too small, the insufficient amount of data accumulated in the learning stage will lead to inadequate model training. If the parameter set too large, the algorithm will stay in the learning stage for a long time. The appropriate auxiliary tasks cannot be selected in time, which leads to the waste of computing resources. The attention mechanism

uses a strategy similar to the gradient descent method in the weight matrices update process, and λ is used to control the magnitude of the update. The size of η controls the time of using the DQN model. When λ is too small, the attention weight cannot update enough in the dynamical environment, which may make the population state representation lag. If λ is too large, the attention weight continues to oscillate and cannot converge.

4.6 Practical application

Different from the test suite designed specifically for algorithm performance, real-world engineering problems have more objectives and the constraints are more complex [9, 10]. In order to verify the potential value of the proposed AMTCMO in dealing with real-world engineering problems, two real-world engineering problems two bar truss design problem [4] and disc brake design problem [36] are selected. Fig. 15 presents the two bar truss design problem. The goal of this problem is to minimize the structural weight $f_1(\mathbf{x})$ and the position change of joint 3 $f_2(\mathbf{x})$. The cross-sectional area of the members and the distance of x in Fig. 15 are the decision variables to satisfy the truss geometry symmetric which form three constraints. It can be described as follows [44, 62]:

$$\begin{aligned} \text{minimize } f_1(\mathbf{x}) &= x_1 \sqrt{16 + x_3^2} + x_2 \sqrt{1 + x_3^2}, \\ \text{minimize } f_2(\mathbf{x}) &= \frac{20 \sqrt{16 + x_3^2}}{x_1 x_3}, \end{aligned} \quad (18)$$

$$\text{s.t. } \begin{cases} g_1(\mathbf{x}) = f_1(\mathbf{x}) - 0.1 \leq 0, \\ g_2(\mathbf{x}) = f_2(\mathbf{x}) - 10^5 \leq 0, \\ g_3(\mathbf{x}) = \frac{80 \sqrt{1 + x_3^2}}{x_3 x_2} - 10^5 \leq 0, \end{cases}$$

where $10^{-5} \leq x_1 \leq 100$, $10^{-5} \leq x_2 \leq 100$, $1 \leq x_3 \leq 3$.

The disc brake design problem is to minimize the stopping time $f_1(\mathbf{x})$ and the mass of disc brake $f_2(\mathbf{x})$. This problem involves four decision variables including the inner and outer radius (x_1, x_2), the clamping force (x_3), and the number of friction surfaces (x_4). The four decision variables form five constraints, which is defined as follows [7]:

$$\begin{aligned} \text{minimize } f_1(\mathbf{x}) &= 4.9 \times 10^{-5} (x_2^2 - x_1^2) (x_4 - 1), \\ \text{minimize } f_2(\mathbf{x}) &= 9.8 \times 10^6 (x_2^2 - x_1^2) / x_3 x_4 (x_2^3 - x_1^3), \\ \text{s.t. } \begin{cases} g_1(\mathbf{x}) = 20 - (x_2 - x_1) \leq 0, \\ g_2(\mathbf{x}) = 2.5 (x_4 + 1) - 30 \leq 0, \\ g_3(\mathbf{x}) = x_3 / \pi (x_2^2 - x_1^2)^2 - 0.4 \leq 0, \\ g_4(\mathbf{x}) = 2.22 \times 10^{-3} x_3 (x_2^3 - x_1^3) / (x_2^2 - x_1^2)^2 - 1 \leq 0, \\ g_5(\mathbf{x}) = 900 - 2.66 \times 10^{-2} x_3 x_4 (x_2^3 - x_1^3) / (x_2^2 - x_1^2) \leq 0, \end{cases} \end{aligned} \quad (19)$$

Table 7 IGD+ results of ablation study on CF, LIRCMOP, and DASCMP test suites

Problem	<i>m</i>	<i>n</i>	AMTCMO_1	AMTCMO_2	AMTCMO_3	AMTCMO
CF1	2	10	3.4381e-3 (2.89e-4) =	2.8228e-3 (4.38e-4) +	2.6397e-3 (3.55e-4) +	3.4295e-3 (4.99e-4)
CF2	2	10	3.5412e-3 (3.75e-4) =	4.0475e-3 (7.00e-4) -	3.7969e-3 (5.82e-4) =	3.7424e-3 (6.42e-4)
CF3	2	10	1.6148e-1 (6.08e-2) =	1.8207e-1 (7.88e-2) =	1.6388e-1 (6.38e-2) =	1.6797e-1 (6.72e-2)
CF4	2	10	3.3906e-2 (7.79e-3) -	3.4645e-2 (7.37e-3) -	3.6638e-2 (5.56e-3) -	2.9834e-2 (5.45e-3)
CF5	2	10	1.6490e-1 (6.94e-2) =	1.6875e-1 (7.26e-2) =	1.5215e-1 (6.44e-2) =	1.5599e-1 (8.79e-2)
CF6	2	10	1.8882e-2 (4.28e-3) =	1.7616e-2 (2.06e-3) =	1.8242e-2 (3.12e-3) =	1.8597e-2 (3.37e-3)
CF7	2	10	8.0443e-2 (1.46e-2) =	8.7960e-2 (3.48e-2) =	8.3803e-2 (2.61e-2) =	8.6660e-2 (2.18e-2)
CF8	3	10	2.1412e-1 (5.42e-2) -	2.1409e-1 (5.19e-2) -	2.1394e-1 (4.45e-2) -	1.5263e-1 (1.43e-2)
cf9	3	10	1.1495E-1 (2.83E-2) =	1.3258E-1 (3.43E-2) -	1.1494E-1 (2.49E-2) =	1.0569E-1 (2.08E-2)
CF10	3	10	3.3202e-1 (1.08e-1) -	3.0168e-1 (9.50e-2) -	3.6952e-1 (1.52e-1) -	2.2778e-1 (7.28e-2)
LIRCMOP1	2	30	8.2152e-3 (1.43e-3) =	1.2589e-2 (4.04e-3) -	1.2495e-2 (3.83e-3) -	7.8416e-3 (1.20e-3)
LIRCMOP2	2	30	9.2124e-3 (1.23e-3) =	1.4256e-2 (5.64e-3) -	1.3654e-2 (3.06e-3) -	8.8076e-3 (1.28e-3)
LIRCMOP3	2	30	6.8694e-3 (1.24e-3) =	1.2261e-2 (8.03e-3) -	9.5505e-3 (4.40e-3) -	6.4440e-3 (6.72e-4)
LIRCMOP4	2	30	1.0476e-2 (1.34e-2) =	1.2211e-2 (5.86e-3) -	1.7724e-2 (3.41e-2) -	7.4903e-3 (2.24e-3)
LIRCMOP5	2	30	7.8504e-3 (1.03e-3) =	8.8304e-3 (1.14e-3) -	8.2807e-3 (9.94e-4) =	7.9249e-3 (8.92e-4)
LIRCMOP6	2	30	7.3711e-3 (6.69e-4) -	7.0676e-3 (9.75e-4) =	7.0774e-3 (7.56e-4) =	6.9838e-3 (4.67e-4)
LIRCMOP7	2	30	9.6659e-3 (9.10e-3) -	6.8702e-3 (3.60e-4) -	6.8411e-3 (4.92e-4) -	6.4698e-3 (3.31e-4)
LIRCMOP8	2	30	9.5050e-3 (1.01e-2) =	6.6064e-3 (3.35e-4) =	6.7503e-3 (3.03e-4) -	6.4471e-3 (2.72e-4)
LIRCMOP9	2	30	1.5559e-2 (1.58e-2) -	1.5564e-2 (1.89e-2) =	1.8369e-2 (1.73e-2) -	9.9732e-3 (1.01e-2)
LIRCMOP10	2	30	7.7534e-3 (4.15e-4) =	7.8263e-3 (5.66e-4) =	7.9037e-3 (8.66e-4) =	7.5737e-3 (4.59e-4)
LIRCMOP11	2	30	2.0372e-3 (9.35e-4) =	1.8171e-3 (5.29e-4) =	1.7658e-3 (5.63e-4) =	2.1203e-3 (1.26e-3)
LIRCMOP12	2	30	2.5022e-3 (1.92e-3) =	2.3631e-3 (1.75e-3) =	1.6650e-3 (7.95e-4) =	8.8866e-3 (1.90e-2)
LIRCMOP13	3	30	6.4548e-2 (2.28e-3) =	6.6026e-2 (2.55e-3) =	6.4973e-2 (1.89e-3) =	6.5521e-2 (3.45e-3)
LIRCMOP14	3	30	5.2308e-2 (1.56e-3) =	5.1659e-2 (1.76e-3) =	5.2217e-2 (1.72e-3) =	5.2918e-2 (2.58e-3)
DASCMP1	2	30	1.9329e-3 (1.06e-4) =	2.0890e-3 (1.88e-4) -	2.1203e-3 (9.00e-5) -	1.8901e-3 (1.04e-4)
DASCMP2	2	30	3.3807e-3 (1.25e-4) =	3.7128e-3 (2.07e-4) -	3.6598e-3 (1.46e-4) -	3.4060e-3 (1.14e-4)
DASCMP3	2	30	2.0752e-2 (4.94e-2) -	6.1191e-3 (4.45e-4) -	1.3435e-2 (3.57e-2) -	5.7692e-3 (2.16e-4)
DASCMP4	2	30	7.8320e-4 (7.67e-5) =	9.8419e-4 (5.72e-4) =	8.0991e-4 (3.51e-4) =	7.4424e-4 (4.18e-5)
DASCMP5	2	30	2.0179e-3 (7.84e-5) =	2.0007e-3 (1.03e-4) =	2.0010e-3 (8.90e-5) =	1.9990e-3 (7.36e-5)
DASCMP6	2	30	5.3118e-3 (6.38e-5) =	5.3074e-3 (7.68e-5) =	5.3634e-3 (1.88e-4) -	5.2899e-3 (6.93e-5)
DASCMP7	3	30	2.5759e-2 (1.13e-3) +	2.5216e-2 (1.01e-3) +	2.5224e-2 (1.04e-3) +	2.6352e-2 (1.06e-3)
DASCMP8	3	30	1.9826e-2 (9.07e-4) =	1.9090e-2 (8.24e-4) +	1.9183e-2 (7.25e-4) +	2.0001e-2 (6.58e-4)
DASCMP9	3	30	1.9756e-2 (9.71e-4) =	2.0224e-2 (1.02e-3) =	2.0219e-2 (8.97e-4) =	1.9959e-2 (1.17e-3)
+/-/=			1/7/25	3/14/16	3/14/16	

Table 8 HV results of ablation study on CF, LIRCMP, and DASCMP test suites

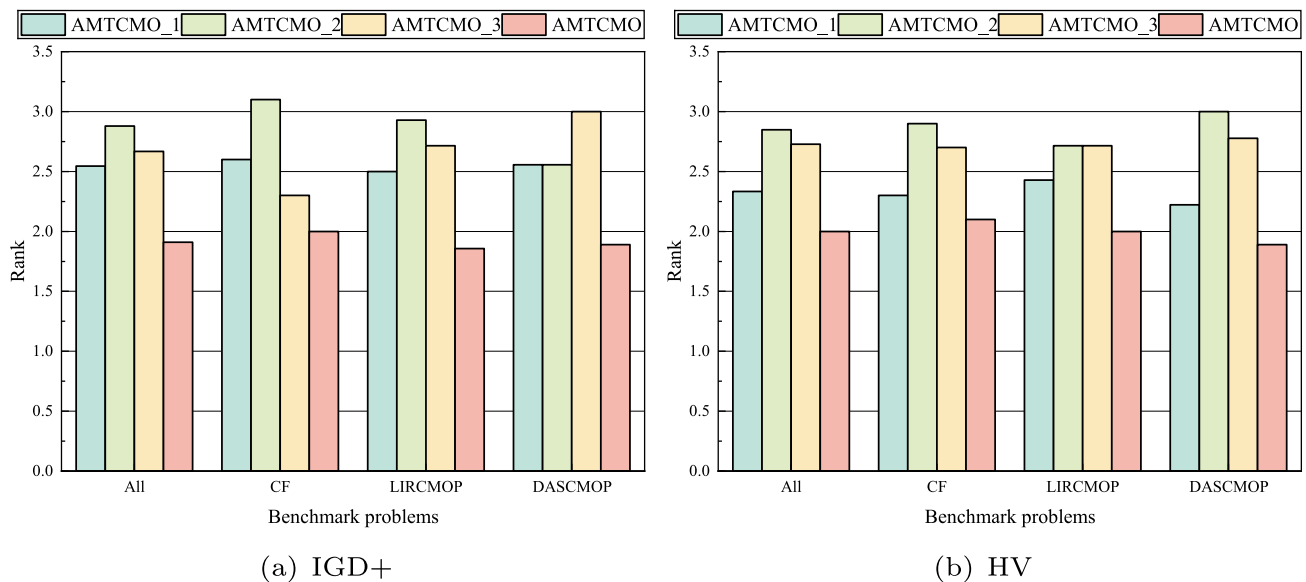
Problem	<i>m</i>	<i>n</i>	AMTCMO_1	AMTCMO_2	AMTCMO_3	AMTCMO
CF1	2	10	5.6192e-1 (3.53e-4) =	5.6267e-1 (5.27e-4) +	5.6289e-1 (4.31e-4) +	5.6193e-1 (6.10e-4)
CF2	2	10	6.7675e-1 (6.60e-4) =	6.7593e-1 (2.14e-3) =	6.7625e-1 (1.09e-3) =	6.7633e-1 (1.32e-3)
CF3	2	10	2.2642e-1 (4.84e-2) =	2.1624e-1 (4.78e-2) =	2.2540e-1 (3.50e-2) =	2.1226e-1 (4.10e-2)
CF4	2	10	4.8339e-1 (1.17e-2) -	4.8265e-1 (1.11e-2) -	4.7915e-1 (9.14e-3) -	4.9016e-1 (8.46e-3)
CF5	2	10	3.3064e-1 (5.78e-2) =	3.3277e-1 (5.92e-2) =	3.4200e-1 (5.93e-2) =	3.4867e-1 (7.29e-2)
CF6	2	10	6.7728e-1 (6.55e-3) =	6.7852e-1 (4.14e-3) =	6.7713e-1 (5.46e-3) =	6.7561e-1 (5.31e-3)
CF7	2	10	5.6100e-1 (2.29e-2) =	5.5140e-1 (5.04e-2) =	5.5992e-1 (3.13e-2) =	5.5350e-1 (3.22e-2)
CF8	3	10	1.9828e-1 (5.16e-2) -	1.9383e-1 (7.00e-2) -	1.9233e-1 (5.17e-2) -	2.7769e-1 (3.22e-2)
CF9	3	10	3.3981e-1 (4.65e-2) =	3.1741e-1 (4.13e-2) -	3.4189e-1 (3.09e-2) =	3.4910e-1 (3.71e-2)
CF10	3	10	1.2346e-1 (4.48e-2) -	1.3708e-1 (4.40e-2) -	1.2106e-1 (5.22e-2) -	1.6845e-1 (4.81e-2)
LIRCMP1	2	30	2.3487e-1 (1.14e-3) =	2.3149e-1 (3.50e-3) -	2.3176e-1 (2.41e-3) -	2.3520e-1 (8.13e-4)
LIRCMP2	2	30	3.5693e-1 (7.63e-4) =	3.5313e-1 (4.25e-3) -	3.5363e-1 (1.97e-3) -	3.5724e-1 (8.19e-4)
LIRCMP3	2	30	2.0369e-1 (8.57e-4) =	1.9953e-1 (5.77e-3) -	2.0136e-1 (2.65e-3) -	2.0386e-1 (6.94e-4)
LIRCMP4	2	30	3.1020e-1 (7.84e-3) =	3.0792e-1 (4.35e-3) -	3.0452e-1 (2.37e-2) -	3.1172e-1 (1.67e-3)
LIRCMP5	2	30	2.9055e-1 (4.71e-4) =	2.9000e-1 (6.85e-4) -	2.9033e-1 (4.74e-4) =	2.9049e-1 (4.04e-4)
LIRCMP6	2	30	1.9580e-1 (2.80e-4) -	1.9589e-1 (3.59e-4) =	1.9590e-1 (3.37e-4) =	1.9598e-1 (2.27e-4)
LIRCMP7	2	30	2.9204e-1 (5.58e-3) =	2.9397e-1 (1.60e-4) =	2.9400e-1 (2.28e-4) =	2.9401e-1 (2.43e-4)
LIRCMP8	2	30	2.9233e-1 (6.01e-3) =	2.9411e-1 (1.42e-4) =	2.9408e-1 (1.41e-4) =	2.9411e-1 (1.17e-4)
LIRCMP9	2	30	5.5884e-1 (6.58e-3) -	5.5879e-1 (8.83e-3) =	5.5774e-1 (6.95e-3) -	5.6143e-1 (4.38e-3)
LIRCMP10	2	30	7.0448e-1 (3.62e-4) =	7.0451e-1 (4.01e-4) =	7.0440e-1 (5.75e-4) =	7.0442e-1 (4.84e-4)
LIRCMP11	2	30	6.9310e-1 (6.34e-4) =	6.9334e-1 (3.27e-4) =	6.9334e-1 (3.73e-4) =	6.9301e-1 (9.71e-4)
LIRCMP12	2	30	6.1955e-1 (8.15e-4) =	6.1958e-1 (7.46e-4) =	6.1982e-1 (3.40e-4) =	6.1639e-1 (9.56e-3)
LIRCMP13	3	30	5.3403e-1 (1.99e-3) =	5.3236e-1 (2.48e-3) =	5.3326e-1 (2.05e-3) =	5.3283e-1 (3.45e-3)
LIRCMP14	3	30	5.4791e-1 (1.65e-3) =	5.4844e-1 (1.76e-3) =	5.4787e-1 (1.43e-3) =	5.4721e-1 (2.36e-3)
DASCMP1	2	30	2.1219e-1 (3.62e-4) =	2.1174e-1 (8.01e-4) =	2.1150e-1 (6.91e-4) -	2.1212e-1 (4.26e-4)
DASCMP2	2	30	3.5518e-1 (8.02e-5) =	3.5491e-1 (1.70e-4) -	3.5498e-1 (1.08e-4) -	3.5517e-1 (7.36e-5)
DASCMP3	2	30	3.0402e-1 (2.54e-2) -	3.1140e-1 (6.05e-4) -	3.0783e-1 (1.83e-2) -	3.1183e-1 (3.44e-4)
DASCMP4	2	30	2.0376e-1 (4.47e-4) =	2.0285e-1 (2.79e-3) =	2.0351e-1 (1.68e-3) =	2.0394e-1 (2.05e-4)
DASCMP5	2	30	3.5135e-1 (1.44e-4) =	3.5131e-1 (1.82e-4) =	3.5129e-1 (1.55e-4) =	3.5135e-1 (1.59e-4)
DASCMP6	2	30	3.1222e-1 (1.86e-4) =	3.1219e-1 (1.35e-4) -	3.1225e-1 (1.57e-4) =	3.1230e-1 (1.49e-4)
DASCMP7	3	30	2.8713e-1 (3.94e-4) =	2.8745e-1 (4.93e-4) +	2.8747e-1 (3.19e-4) +	2.8699e-1 (3.81e-4)
DASCMP8	3	30	2.0587e-1 (4.04e-4) =	2.0627e-1 (3.88e-4) +	2.0621e-1 (4.35e-4) +	2.0591e-1 (3.28e-4)
DASCMP9	3	30	2.0581e-1 (4.17e-4) =	2.0557e-1 (4.30e-4) =	2.0558e-1 (5.45e-4) =	2.0568e-1 (4.61e-4)
+/-/=			0/6/27	3/12/18	3/11/19	

Table 9 The summary of IGD+ results and the Wilcoxon test for ablation study and parameter analysis

AMTCMO VS	+/-/=	R^+	R^-	Exact p value	Asymptotic p value
AMTCMO_1	1/7/25	372.0	189.0	0.10454	0.100209
AMTCMO_2	3/14/16	435.0	126.0	0.00485	0.005614
AMTCMO_3	3/14/16	361.0	200.0	0.15472	0.147815
AMTCMO_4	2/12/19	406.0	155.0	0.02412	0.024364
AMTCMO_5	0/10/23	455.0	106.0	1.276e-3	0.001725
AMTCMO_6	0/17/16	496.0	65.0	3.766e-5	0.000114
AMTCMO_7	1/24/8	516.0	45.0	3.874e-6	0.000025
AMTCMO_8	2/12/19	449.5	111.5	0.001883	0.002349
AMTCMO_9	0/8/25	424.0	137.0	0.009304	0.009917
AMTCMO_10	1/8/24	408.5	152.5	0.0213	0.021383
AMTCMO_11	1/9/23	462.5	98.5	7.314e-4	0.001082

Table 10 The summary of HV results and the Wilcoxon test for ablation study and parameter analysis

AMTCMO VS	+/-/=	R^+	R^-	Exact p value	Asymptotic p value
AMTCMO_1	0/6/27	335.0	193.0	0.1901	0.180211
AMTCMO_2	3/12/18	393.0	135.0	0.0148	0.014979
AMTCMO_3	3/11/19	378.0	183.0	0.0831	0.079938
AMTCMO_4	3/8/22	364.5	163.5	0.0609	0.057767
AMTCMO_5	1/9/23	431.0	130.0	6.188e-3	0.006849
AMTCMO_6	0/17/16	441.5	86.5	5.355e-4	0.000826
AMTCMO_7	1/23/9	503.0	25.0	4.210e-7	0.000007
AMTCMO_8	2/7/24	410.0	151.0	0.01974	0.019362
AMTCMO_9	0/7/26	383.0	178.0	0.06796	0.044838
AMTCMO_10	0/7/26	368.0	160.0	0.05208	0.039402
AMTCMO_11	1/7/25	390.0	138.0	0.01746	0.002739

**Fig. 10** IGD+ and HV rank of ablation study obtained by Friedman test

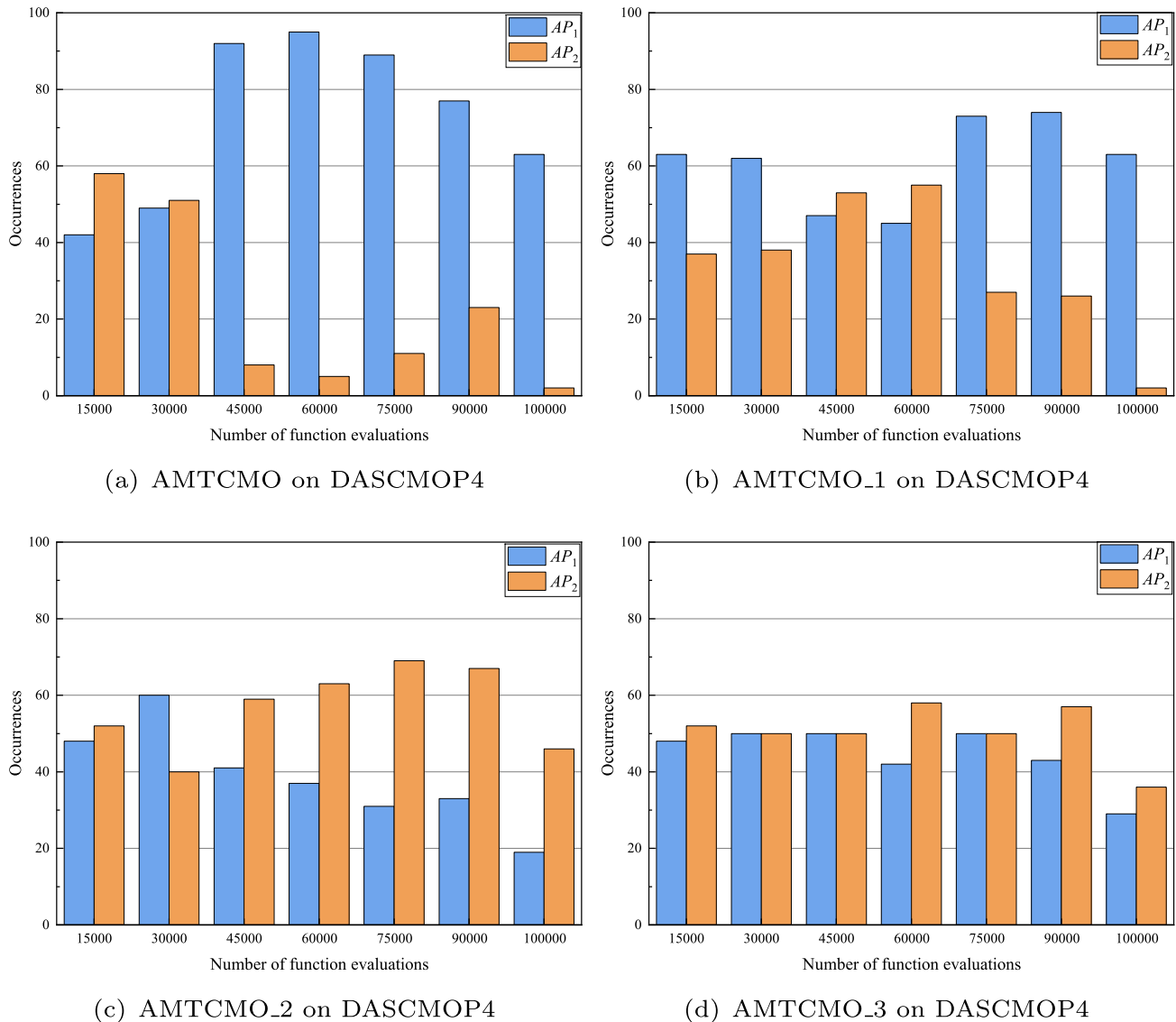


Fig. 11 The number of auxiliary tasks selected by each algorithm in the evolution process

where $55 \leq x_1 \leq 80$, $75 \leq x_2 \leq 110$, $1000 \leq x_3 \leq 3000$, $2 \leq x_4 \leq 20$.

All comparison algorithms run independently 24 times, and the maximum number of function evaluations for each problem is set according to [25]. Since the true PF of real-world engineering problems is unknown, the IGD+ cannot be calculated. Therefore, only the HV is selected as the performance indicator. The final results are shown in Table 11. It can be seen that the proposed AMTCMO achieves the best results on the two problems. Therefore, AMTCMO shows the potential value to solve real-world engineering problems.

5 Conclusion

In this paper, in order to address the defects of insufficient population state representation and underdevelopment of task correlation when applying RLs for solving CMOPs, a CMOEA based on attention mechanism assisted auxiliary task selection (AMTCMO) is proposed. Firstly, when training the RL model, the self-attention mechanism is used to generate the enhanced population state as input. The enhanced state integrates the multi-dimensional feature of the population, which can better reflect the current needs and make the model prediction more accurate. Secondly, the cross-attention mechanism is used to calculate the attention scores between tasks to exploit the correlation among tasks. The attention score can reflect the attention degree of the

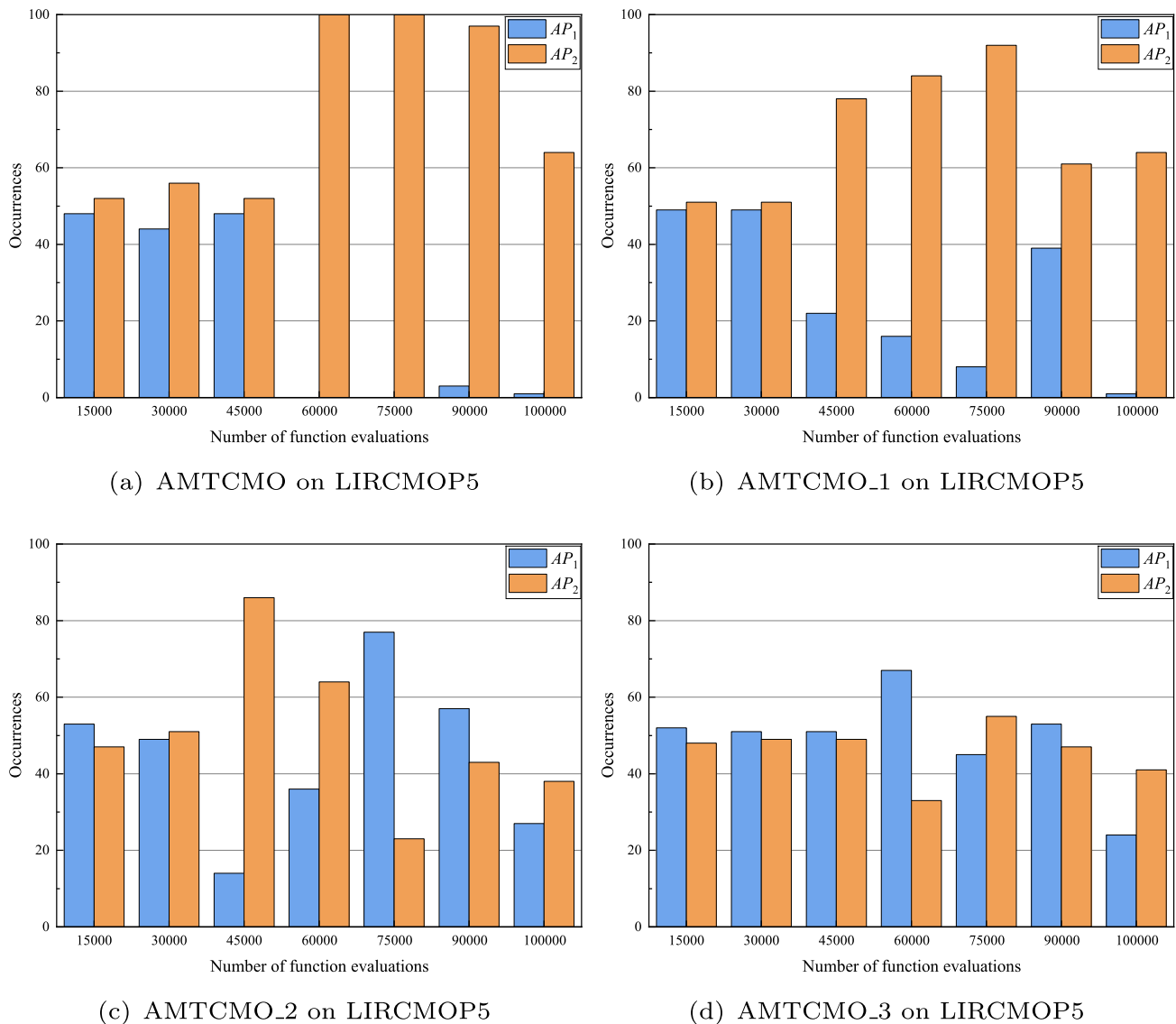
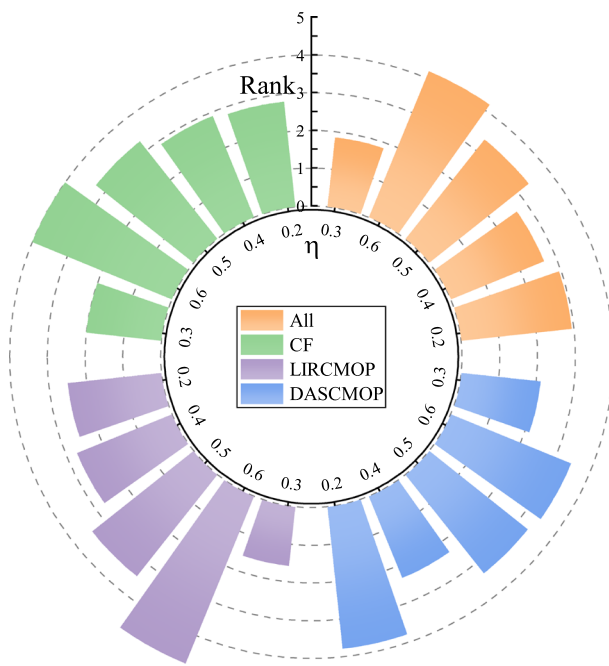


Fig. 12 The number of auxiliary tasks selected by each algorithm in the evolution process

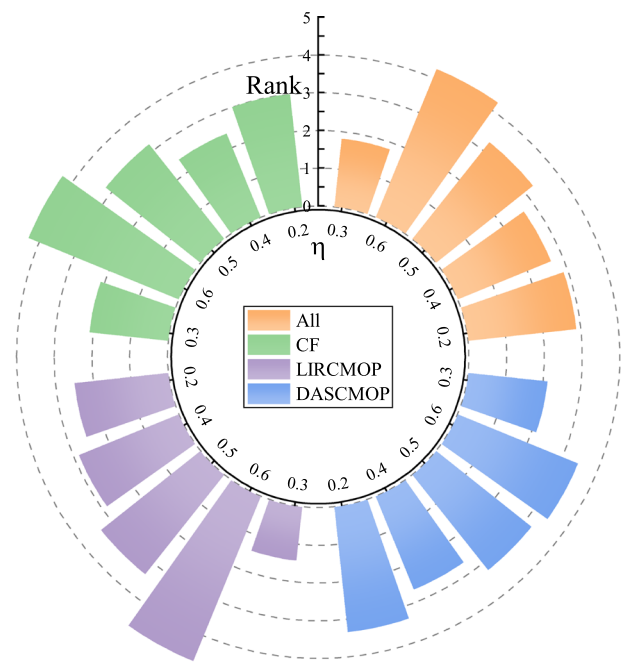
main task to each auxiliary task, which can complement the decision-making with RL and increase the reliability of task selection. Compared with nine state-of-the-art algorithms on 4 test suites and 2 real-world problems, the proposed AMTCMO shows excellent performance. At the same time, a large number of statistical results and experimental studies show the effectiveness of the proposed strategy. In particular, it can be seen from the results of ablation experiments that the proposed strategy can better select effective strategies than traditional RLs.

In the future, there are still some noteworthy research directions. Firstly, the DQL model used in AMTCMO requires a large amount of training data. When the number of function evaluations is limited, the training of the model cannot be guaranteed, which may affect the selection of strategy.

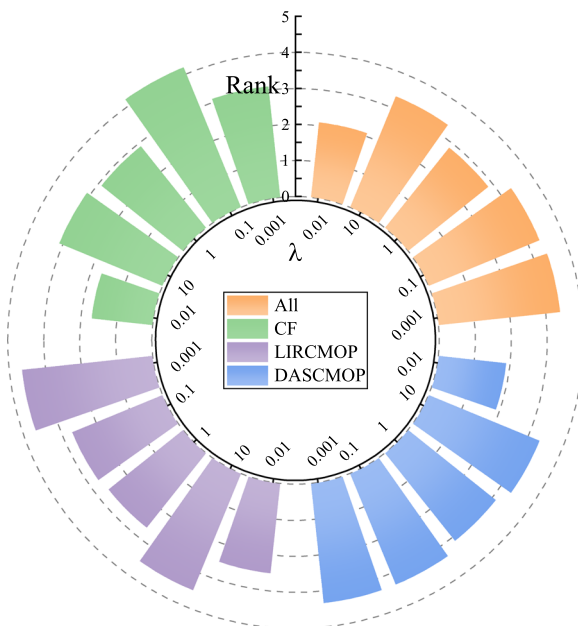
Therefore, there is potential to nest some advanced unsupervised learning techniques in the algorithm. Furthermore, AMTCMO does not perform well on some high-dimensional complex problems, which requires specific auxiliary tasks to provide assistance. Therefore, more effective auxiliary tasks are expected to be designed for selection to enhance the ability to cope with complex and changing problems.



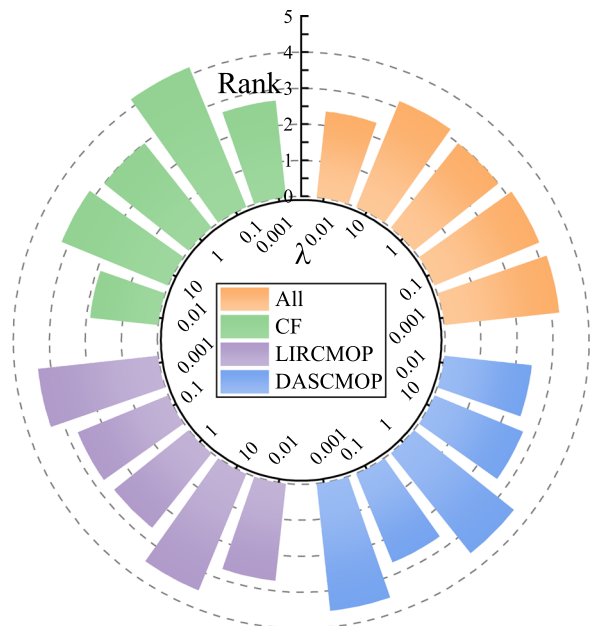
(a) IGD+



(b) HV

Fig. 13 IGD+ and HV rank of η under different values obtained by Firedman test.


(a) IGD+

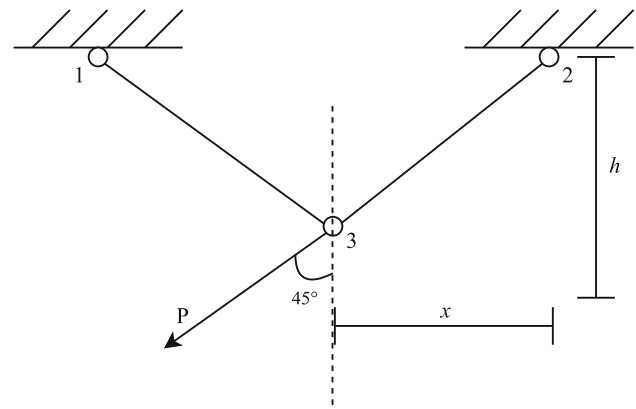


(b) HV

Fig. 14 IGD+ and HV rank of λ under different values obtained by Firedman test

Table 11 HV results of ten comparison algorithms on two real-world engineering problems

Problem	m	n	DCNSGAIII	BiCo	EMCMO	CMEGL	MTCMO
Two bar truss design	2	3	8.9434e-1 (1.84e-3) –	8.9952e-1 (6.59e-4) –	8.9919e-1 (6.52e-4) –	8.9894e-1 (8.77e-4) –	8.9918e-1 (5.38e-4) –
Disc brake design	2	4	3.6168e-1 (4.13e-2) –	4.3467e-1 (2.71e-4) –	4.3351e-1 (1.31e-3) –	4.3417e-1 (3.39e-4) –	4.3453e-1 (5.31e-4) –
+/-/=			0/2/0	0/0/2	0/2/0	0/2/0	0/1/1
Problem	m	n	DRLOSEMCMO	CSEMT	DEST	APSEA	AMTCMO
Two bar truss design	2	3	8.9969e-1 (4.82e-4) –	8.9943e-1 (7.11e-4) –	8.1809e-1 (6.31e-2) –	8.9905e-1 (4.41e-4) –	8.9972e-1 (4.01e-4) –
Disc brake design	2	4	4.3433e-1 (2.28e-4) –	4.3353e-1 (8.03e-4) –	3.9704e-1 (1.06e-2) –	4.3404e-1 (1.02e-3) –	4.3467e-1 (1.37e-4) –
+/-/=			0/1/1	0/1/1	0/2/0	0/2/0	

**Fig. 15** The illustration of the two bar truss design problem

Acknowledgements This work was supported in part by the National Nature Science Foundation of China under Grant 12501711, in part by the China Postdoctoral Science Foundation under Grant 2024M762632, in part by the Natural Science Foundation of Shaanxi under Grant 2024JC-YBQN-0687, in part by the Social Science Foundation of Shaanxi under Grant 2024R027, and in part by the Fundamental Research Funds of the Central Universities of China under Grant 2452023013.

Author Contributions Qianlong Dang: Conceptualization, methodology, software, writing—original draft, visualization. Xinyu Feng: Method-ology. Linlin Xie: Supervision, Funding acquisition. Xianpeng Sun: Validation. Weijun Wu: Writing—review & editing, Software.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Azarm S, Tits A, Fan M (1992) Tradeoff-driven optimization-based design of mechanical systems. In: 4th symposium on multidisciplinary analysis and optimization
2. Bao Q, Wang M, Dai G, Chen X, Song Z, Li S (2022) An archive-based two-stage evolutionary algorithm for constrained multi-objective optimization problems. *Swarm Evol Comput* 75:101161. <https://doi.org/10.1016/j.swevo.2022.101161> (<https://www.sciencedirect.com/science/article/pii/S2210650222001298>)
3. Bao Q, Wang M, Dai G, Chen X, Song Z, Li S (2023) A dual-population based bidirectional coevolution algorithm for constrained multi-objective optimization problems. *Expert Syst Appl* 215:119258. <https://doi.org/10.1016/j.eswa.2022.119258> (<https://www.sciencedirect.com/science/article/pii/S095741742202276X>)
4. Chiandussi G, Codegone M, Ferrero S, Varesio F (2012) Comparison of multi-objective optimization methodologies for engineering applications. *Comput Math Appl* 63(5):912–942. <https://doi.org/10.1016/j.camwa.2011.11.057> (<https://www.sciencedirect.com/science/article/pii/S0898122111010406>)
5. Chu X, Ming F, Gong W (2024) Competitive multitasking for computational resource allocation in evolutionary constrained multi-

- objective optimization. *IEEE Trans Evol Comput* 1–1. <https://doi.org/10.1109/TEVC.2024.3376729>
6. Dang Q, Liu Q, Yang S, He X (2025) Data-driven evolutionary algorithm based on inductive graph neural networks for multimodal multi-objective optimization. *IEEE Trans Evol Comput* 1–1. <https://doi.org/10.1109/TEVC.2025.3541046>
 7. Dang Q, Shang W, Huang Z, Yang S (2025) Constrained multi-objective optimization assisted by convergence and diversity auxiliary tasks. *Eng Appl Artif Intell* 139:109546. <https://doi.org/10.1016/j.engappai.2024.109546> (<https://www.sciencedirect.com/science/article/pii/S0952197624017044>)
 8. Dang Q, Zhang G, Wang L, Yang S, Zhan T (2024) A generative adversarial networks model based evolutionary algorithm for multimodal multi-objective optimization. *IEEE Trans Emerg Top Comput Intell* 1–10. <https://doi.org/10.1109/TETCI.2024.3397996>
 9. Dang Q, Zhang G, Wang L, Yang S, Zhan T (2024) Hybrid IoT device selection with knowledge transfer for federated learning. *IEEE Internet Things J* 11(7):12216–12227. <https://doi.org/10.1109/JIOT.2023.3334018>
 10. Dang Q, Zhang G, Wang L, Yu Y, Yang S, He X (2025) Transformer-based intelligent prediction model for multimodal multi-objective optimization. *IEEE Comput Intell Mag* 20(1):34–49. <https://doi.org/10.1109/MCI.2024.3486284>
 11. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
 12. Fan Z, Li W, Cai X, Huang H, Fang Y, You Y, Goodman ED (2019) An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. *Soft Comput*
 13. Fan Z, Li W, Cai X, Li H, Wei C, Zhang Q, Goodman E (2019) Push and pull search for solving constrained multi-objective optimization problems. *Swarm Evol Comput* 44:665–679. <https://doi.org/10.1016/j.swevo.2018.08.017> (<https://www.sciencedirect.com/science/article/pii/S2210650218300233>)
 14. Fan Z, Li W, Cai X, Li H, Wei C, Zhang Q, Goodman E (2020) Difficulty adjustable and scalable constrained multiobjective test problem toolkit. *Evol Comput* 28(3):339–378
 15. Gao X, Bai W, Dang Q, Yang S, Zhang G (2025) Learnable self-supervised support vector machine based individual selection strategy for multimodal multi-objective optimization. *Inf Sci* 690:121553. <https://doi.org/10.1016/j.ins.2024.121553> (<https://www.sciencedirect.com/science/article/pii/S0020025524014671>)
 16. Gu Q, Liu R, Hui Z, Wang D (2024) A constrained multi-objective optimization algorithm based on coordinated strategy of archive and weight vectors. *Expert Syst Appl* 244:122961. <https://doi.org/10.1016/j.eswa.2023.122961> (<https://www.sciencedirect.com/science/article/pii/S0957417423034632>)
 17. Gu Q, Wang Q, Xiong NN, Jiang S, Chen L (2022) Surrogate-assisted evolutionary algorithm for expensive constrained multi-objective discrete optimization problems. *Complex Intell Syst* 8(4):2699–2718
 18. Guerreiro AP, Fonseca CM, Paquete L (2021) The hypervolume indicator: computational problems and algorithms. *ACM Comput Surv (CSUR)* 54(6) <https://doi.org/10.1145/3453474>
 19. Hao Y, Zhao C, Zhang Y, Cao Y, Li Z (2024) Constrained multi-objective optimization problems: methodologies, algorithms and applications. *Knowl-Based Syst* 299:111998. <https://doi.org/10.1016/j.knosys.2024.111998> (<https://www.sciencedirect.com/science/article/pii/S0950705124006324>)
 20. He Z, Liu H (2025) Dual-stage dual-population diversity maintenance for global and local exploration of constrained multi-objective optimization. *Expert Syst Appl* 268:126229. <https://doi.org/10.1016/j.eswa.2024.126229> (<https://www.sciencedirect.com/science/article/pii/S0957417424030963>)
 21. Hu Y, Qu Y, Li W, Huang Y (2025) A pareto front searching algorithm based on reinforcement learning for constrained multi-objective optimization. *Inf Sci* 705:121985. <https://doi.org/10.1016/j.ins.2025.121985> (<https://www.sciencedirect.com/science/article/pii/S0020025525001173>)
 22. Jiang M, Gao X, Dang Q, Ruan J (2025) An evolutionary multitasking algorithm based on k-nearest neighbors pre-selection strategy for constrained multi-objective optimization. *Expert Syst Appl* 283:127768. <https://doi.org/10.1016/j.eswa.2025.127768> (<https://www.sciencedirect.com/science/article/pii/S0957417425013909>)
 23. Jiao R, Zeng S, Li C, Yang S, Ong YS (2021) Handling constrained many-objective optimization problems via problem transformation. *IEEE Trans Cybern* 51(10):4834–4847. <https://doi.org/10.1109/TCYB.2020.3031642>
 24. Kocis GR, Grossmann IE (1989) A modelling and decomposition strategy for the MINLP optimization of process flowsheets. *Comput Chem Eng* 13(7):797–819
 25. Kumar A, Wu G, Ali MZ, Luo Q, Mallipeddi R, Suganthan PN, Das S (2021) A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results. *Swarm Evol Comput* 67:100961. <https://doi.org/10.1016/j.swevo.2021.100961> (<https://www.sciencedirect.com/science/article/pii/S2210650221001231>)
 26. Liang J, Chen Z, Wang Y, Ban X, Qiao K, Yu K (2023) A dual-population constrained multi-objective evolutionary algorithm with variable auxiliary population size. *Complex Intell Syst* 9(5):5907–5922
 27. Liang J, Qiao K, Yu K, Qu B, Yue C, Guo W, Wang L (2023) Utilizing the relationship between unconstrained and constrained pareto fronts for constrained multiobjective optimization. *IEEE Trans Cybern* 53(6):3873–3886. <https://doi.org/10.1109/TCYB.2022.3163759>
 28. Liu S, Wang Z, Lin Q, Chen J (2024) Coevolutionary multitasking for constrained multiobjective optimization. *Swarm Evol Comput* 91:101727. <https://doi.org/10.1016/j.swevo.2024.101727> (<https://www.sciencedirect.com/science/article/pii/S2210650224002657>)
 29. Liu Z, Wang B, Tang K (2021) Handling constrained multiobjective optimization problems via bidirectional coevolution. *IEEE Trans Cybern* 52(10):10163–10176
 30. Liu Z, Zhao P, Cao J, Zhang J, Chen Z (2024) A constrained multi-objective evolutionary algorithm with pareto estimation via neural network. *Expert Syst Appl* 237:121718. <https://doi.org/10.1016/j.eswa.2023.121718> (<https://www.sciencedirect.com/science/article/pii/S0957417423022200>)
 31. Ma H, Wei H, Tian Y, Cheng R, Zhang X (2021) A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints. *Inf Sci* 560:68–91. <https://doi.org/10.1016/j.ins.2021.01.029> (<https://www.sciencedirect.com/science/article/pii/S0020025521000566>)
 32. Meyn S (2024) The projected bellman equation in reinforcement learning. *IEEE Trans Autom Control* 69(12):8323–8337. <https://doi.org/10.1109/TAC.2024.3409647>
 33. Ming F, Gong W, Gao L (2023) Adaptive auxiliary task selection for multitasking-assisted constrained multi-objective optimization [feature]. *IEEE Comput Intell Mag* 18(2):18–30. <https://doi.org/10.1109/MCI.2023.3245719>
 34. Ming F, Gong W, Wang L, Gao L (2024) Constrained multiobjective optimization via multitasking and knowledge transfer. *IEEE Trans Evol Comput* 28(1):77–89. <https://doi.org/10.1109/TEVC.2022.3230822>
 35. Ming F, Gong W, Wang L, Jin Y (2024) Constrained multi-objective optimization with deep reinforcement learning assisted operator selection. *IEEE/CAA J Autom Sin* 11(4):919–931. <https://doi.org/10.1109/JAS.2023.123687>

36. Osyczka A, Kundu S (1995) A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Struct Optim* 10(2):94–99
37. Qiao K, Chen Z, Qu B, Yu K, Yue C, Chen K, Liang J (2024) A dual-population evolutionary algorithm based on dynamic constraint processing and resources allocation for constrained multi-objective optimization problems. *Expert Syst Appl* 238:121707
38. Qiao K, Liang J, Liu Z, Yu K, Yue C, Qu B (2023) Evolutionary multitasking with global and local auxiliary tasks for constrained multi-objective optimization. *IEEE/CAA J Autom Sin* 10(10):1951–1964. <https://doi.org/10.1109/JAS.2023.123336>
39. Qiao K, Liang J, Yu K, Ban X, Yue C, Qu B, Suganthan PN (2024) Constraints separation based evolutionary multitasking for constrained multi-objective optimization problems. *IEEE/CAA J Autom Sin* 11(8):1819–1835. <https://doi.org/10.1109/JAS.2024.124545>
40. Qiao K, Liang J, Yu K, Guo W, Yue C, Qu B, Suganthan P (2024) Benchmark problems for large-scale constrained multi-objective optimization with baseline results. *Swarm Evol Comput* 86:101504. <https://doi.org/10.1016/j.swevo.2024.101504> (<https://www.sciencedirect.com/science/article/pii/S2210650224000373>)
41. Qiao K, Liang J, Yu K, Yue C, Lin H, Zhang D, Qu B (2024) Evolutionary constrained multiobjective optimization: scalable high-dimensional constraint benchmarks and algorithm. *IEEE Trans Evol Comput* 28(4):965–979. <https://doi.org/10.1109/TEVC.2023.3281666>
42. Qiao K, Yu K, Qu B, Liang J, Song H, Yue C (2022) An evolutionary multitasking optimization framework for constrained multiobjective optimization problems. *IEEE Trans Evol Comput* 26(2):263–277
43. Qiao K, Yu K, Qu B, Liang J, Song H, Yue C, Tan KC (2022) Dynamic auxiliary task-based evolutionary multitasking for constrained multiobjective optimization. *IEEE Trans Evol Comput* 27(3):642–656
44. Ray T, Tai K, Seow KC (2001) Multiobjective design optimization by an evolutionary algorithm. *Eng Optim* 33(4):399–424
45. RivasDávalos F, Irving MR (2005) An approach based on the strength pareto evolutionary algorithm 2 for power distribution system planning. *Evolut Multi Criterion Optim* 3410:707–720
46. Tan B, Ma H, Mei Y, Zhang M (2021) Evolutionary multi-objective optimization for web service location allocation problem. *IEEE Trans Serv Comput* 14(2):458–471. <https://doi.org/10.1109/TSC.2018.2793266>
47. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput Intell Mag* 12(4):73–87. <https://doi.org/10.1109/MCI.2017.2742868>
48. Tian Y, Wang R, Zhang Y, Zhang X (2025) Adaptive population sizing for multi-population based constrained multi-objective optimization. *Neurocomputing* 621:129296. <https://doi.org/10.1016/j.neucom.2024.129296> (<https://www.sciencedirect.com/science/article/pii/S0925231224020678>)
49. Tian Y, Zhang T, Xiao J, Zhang X, Jin Y (2020) A coevolutionary framework for constrained multiobjective optimization problems. *IEEE Trans Evol Comput* 25(1):102–116
50. Tian Y, Zhang Y, Su Y, Zhang X, Tan KC, Jin Y (2022) Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization. *IEEE Trans Cybern* 52(9):9559–9572. <https://doi.org/10.1109/TCYB.2020.3021138>
51. Triguero I, González S, Moyano JM, García S, AlcaláFdez J, Luengo J, Herrera F (2017) Keel 3.0: an open source software for multi-stage analysis in data mining. *Int J Comput Intell Syst* 10(1):1238–1249
52. Wang B, Li H, Li J, Wang Y (2019) Composite differential evolution for constrained evolutionary optimization. *IEEE Trans Syst Man Cybern Syst* 49(7):1482–1495. <https://doi.org/10.1109/TSMC.2018.2807785>
53. Wang C, Liu Z, Qiu J, Zhang L (2024) Adaptive constraint handling technique selection for constrained multi-objective optimization. *Swarm Evol Comput* 86:101488. <https://doi.org/10.1016/j.swevo.2024.101488> (<https://www.sciencedirect.com/science/article/pii/S221065022400021X>)
54. Wang Y, Hu C, Ming F, Li Y, Gong W, Gao L (2024) A diversity-enhanced tri-stage framework for constrained multi-objective optimization. *IEEE Trans Evol Comput* 1–1. <https://doi.org/10.1109/TEVC.2024.3458079>
55. Wolpert D, Macready W (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82. <https://doi.org/10.1109/4235.585893>
56. Xia M, Chong Q, Dong M (2023) A constrained multi-objective evolutionary algorithm with two-stage resources allocation. *Swarm Evol Comput* 79:101313. <https://doi.org/10.1016/j.swevo.2023.101313> (<https://www.sciencedirect.com/science/article/pii/S221065022300086X>)
57. Xia Z, Liu Y, Lu J, Cao J, Rutkowski L (2020) Penalty method for constrained distributed quaternion-variable optimization. *IEEE Trans Cybern* 51(11):5631–5636
58. Yang Y, Zhang C, Liu Y, Ning J, Guo Y (2024) Deep reinforcement learning assisted novelty search in Voronoi regions for constrained multi-objective optimization. *Swarm Evol Comput* 91:101732. <https://doi.org/10.1016/j.swevo.2024.101732> (<https://www.sciencedirect.com/science/article/pii/S2210650224002700>)
59. Yang Z, Yao T, Jiang Y, Zhang J, Zhang X (2024) Indexes-based and partial restart-based constrained multiobjective optimization. *IEEE Trans Evol Comput* 1–1. <https://doi.org/10.1109/TEVC.2024.3400610>
60. Yu K, Liang J, Qu B, Yue C (2021) Purpose-directed two-phase multiobjective differential evolution for constrained multiobjective optimization. *Swarm Evol Comput* 60:100799. <https://doi.org/10.1016/j.swevo.2020.100799> (<https://www.sciencedirect.com/science/article/pii/S2210650220304521>)
61. Yu K, Wang L, Liang J, Wang H, Qiao K, Liang T (2024) Constraint subsets-based evolutionary multitasking for constrained multiobjective optimization. *Swarm Evol Comput* 86:101531. <https://doi.org/10.1016/j.swevo.2024.101531> (<https://www.sciencedirect.com/science/article/pii/S2210650224000695>)
62. Yuan J, Liu HL, He Z (2022) A constrained multi-objective evolutionary algorithm using valuable infeasible solutions. *Swarm Evol Comput* 68:101020. <https://doi.org/10.1016/j.swevo.2021.101020> (<https://www.sciencedirect.com/science/article/pii/S2210650221001826>)
63. Yuan J, Liu HL, Ong YS, He Z (2022) Indicator-based evolutionary algorithm for solving constrained multiobjective optimization problems. *IEEE Trans Evol Comput* 26(2):379–391. <https://doi.org/10.1109/TEVC.2021.3089155>
64. Zhang Q, Zhou A, Zhao S, Suganthan PN, Tiwari S (2008) Multiobjective optimization test instances for the CEC 2009 special session and competition. *Mech Eng (New York, N.Y.)* 1919
65. Zhang X, Han Z, Wang Z, Zhao J, Wang W (2024) Ingredient planning for copper industry: A deep reinforcement learning-based ϵ -constrained multi-objective optimization framework. In: 2024 IEEE congress on evolutionary computation (CEC), pp 1–8
66. Zhou J, Zhang Y, Wang J, Suganthan PN (2024) Localized constrained-domination principle for constrained multiobjective optimization. *IEEE Trans Syst Man Cybern Syst* 54(3):1376–1387. <https://doi.org/10.1109/TSMC.2023.3324797>
67. Zitzler E, Knowles J, Thiele L (2008) Quality assessment of pareto set approximations. *Interactive and Evolutionary Approaches, Multiobjective Optimization*, pp 373–404

68. Zuo M, Gong D, Wang Y, Ye X, Zeng B, Meng F (2024) Process knowledge-guided autonomous evolutionary optimization for constrained multiobjective problems. *IEEE Trans Evol Comput* 28(1):193–207. <https://doi.org/10.1109/TEVC.2023.3243109>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.