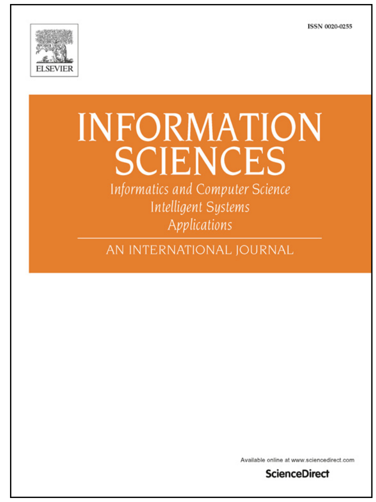
Journal Pre-proof



A Novel Multi-State Reinforcement Learning-Based Multi-Objective Evolutionary Algorithm

Jing Wang, Yuxin Zheng, Ziyun Zhang, Hu Peng and Hui Wang

PII: S0020-0255(24)01311-2
DOI: https://doi.org/10.1016/j.ins.2024.121397
Reference: INS 121397

To appear in: *Information Sciences*

Received date: 16 April 2024
Revised date: 25 July 2024
Accepted date: 24 August 2024

A Novel Multi-State Reinforcement Learning-Based Multi-Objective Evolutionary Algorithm

Jing Wang[a] (wangjing@jxufe.edu.cn), Yuxin Zheng[a,*] (17854338276@163.com), Ziyun Zhang[a] (2202220775@stu.jxufe.edu.cn), Hu Peng[b] (hu_peng@whu.edu.cn), Hui Wang[c] (huiwang@whu.edu.cn)

[a] School of Software and IoT Engineering, Jiangxi University of Finance and Economics, Nanchang 330032, PR China
[b] School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, PR China
[c] School of Information Engineering, Nanchang Institute of Technology, Nanchang, China

Corresponding Author:
Yuxin Zheng
School of Software and IoT Engineering, Jiangxi University of Finance and Economics, Nanchang 330032, PR China
Tel: 17854338276
Email: 17854338276@163.com

A Novel Multi-State Reinforcement Learning-Based Multi-Objective Evolutionary Algorithm

Jing Wang[a], Yuxin Zheng[a,*], Ziyun Zhang[a], Hu Peng[b], Hui Wang[c]

[a]School of Software and IoT Engineering, Jiangxi University of Finance and Economics, Nanchang 330032, PR China
[b]School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, PR China
[c]School of Information Engineering, Nanchang Institute of Technology, Nanchang, China

Abstract

Multi-objective evolutionary algorithms (MOEAs) are widely employed to tackle multi-objective optimization problems (MOPs). However, the choice of different crossover operators significantly impacts the algorithm's ability to balance population diversity and convergence effectively. To enhance algorithm performance, this paper introduces a novel multi-state reinforcement learning-based multi-objective evolutionary algorithm, MRL-MOEA, which utilizes reinforcement learning (RL) to select crossover operators. In MRL-MOEA, a state model is established according to the distribution of individuals in the objective space, and different crossover operators are designed for the transition between different states. Additionally, in the process of evolution, the population still exhibits inadequate convergence in certain regions, leading to sparse areas within the regular Pareto Front (PF). To address this issue, a strategy for adjusting weight vectors has been devised to achieve uniform distribution of the PF. The experimental results of MRL-MOEA on several benchmark suites with a varying number of objectives ranging from 3 to 10, including WFG and DTLZ, demonstrate MRL-MOEA's competitiveness compared to other algorithms.

Keywords: Multi-objective evolutionary algorithm, Decomposition, Reinforcement learning, Matching mechanism, Multiple strategies

1. Introduction

In recent years, researchers have introduced a variety of enhanced multi-objective optimization evolutionary algorithms (MOEAs), developed to address real-world engineering problems [1], including logistics scheduling [2], machine scheduling [3], power grid optimization [4], and parameter optimization in machine learning [5]. Within these MOEAs, numerous strategies aimed at enhancing environmental selection pressure have been introduced to boost algorithms' performance. Furthermore, the choice of parent individuals and crossover operators directly influence algorithm performance [6]. Two main approaches are utilized to determine the optimal evolutionary operator for a given problem: offline and online operator selection. Offline selection involves training models to understand the relationship between problem features and operator performance, enabling the selection of the most suitable operator for a new problem. For instance, Kerschke et al. [7] and Tian et al. [8] suggested using training models to learn how problem characteristics relate to operator performance.

However, accurately capturing the characteristics of a specific problem can be challenging. On the other hand, online operator selection, also known as the hyperheuristic method, dynamically selects operators based on their performance in past iterations. These methods propose various credit allocation strategies to assess each operator's contribution to solution enhancement and suggest selection strategies to identify the next operator for generating offspring. Noteworthy algorithms like MCHH [9], and FRRMAB [10] are commonly employed in this context. Nevertheless, this approach may introduce a delicate balance between exploration and exploitation in MOEAs. Operators with a strong historical performance may receive higher priority, potentially limiting population development and preventing offspring from escaping local optima. Conversely, prioritizing operators with weaker historical performance could diminish population exploration, making it challenging for future generations to uncover the global optimal solution.

*Corresponding author
Email addresses: wangjing@jxufe.edu.cn (Jing Wang), 17854338276@163.com (Yuxin Zheng), 2202220775@stu.jxufe.edu.cn (Ziyun Zhang), hu_peng@whu.edu.cn (Hu Peng), huiwang@whu.edu.cn (Hui Wang)

Reinforcement learning (RL) has been extensively utilized in evolutionary algorithms to guide environment selection [11], parameter adaptation [12], and other domains [13]. One of the fundamental RL models employed is the Q-learning model, which can learn and update itself through continuous interaction with the environment. This confers Q-learning an advantage in real-time decision-making and solving problems, enabling it to adapt to environmental changes swiftly. However, simply applying the Q-learning model to MOEAs does not effectively enhance the algorithm's performance. This outcome depends not only on the strategies employed by the algorithm, the characteristics of the crossover operators, and the information generated during the population evolution process, but more importantly, it requires designing a new interaction method tailored to these factors as inputs and outputs understandable by the Q-learning model.

According to the above description, this paper integrates RL into the adaptive selection of operators in MOEAs. This integration enables operators to collaboratively make decisions based on individual states and historical experience, with the results influencing subsequent operator selection processes. Additionally, a reverse learning mechanism is implemented to ensure that operators with poor historical performance have an appropriate probability of being selected. This approach aims to address the limitations observed in traditional hyperheuristic methods. The following outlines the primary contributions of this paper.

- A new matching strategy is proposed, and each weight vector is allowed to match multiple individuals, which reduces the waste of computing resources and improves the diversity of the population.
- Based on different crossover operators and the characteristics of parent individuals, a generation method of offspring is proposed to improve the robustness of the algorithm.
- A state model is proposed based on the geometric distribution of individuals corresponding to each subproblem in the matching strategy. This model provides state information for the Q-learning algorithm, with different crossover operators used as actions to guide the evolution of the population.

The subsequent content is as follows: Section 2 illustrates the basic concepts of the Q-learning model and related works; Section 3 introduces the motivation of this paper; Section 4 provides a detailed description of the algorithm proposed in this paper; Section 5 introduces the test problems and comparative algorithms, followed by an analysis of the experimental results; finally, Section 6 summarizes the work of this paper and outlines directions for future work.

2. Preliminaries and related works

2.1. Multi-objective optimization problems

Without sacrificing generality, the minimizing of multi-objective optimization problems (MOPs) can be delineated as follows:

$$\mathbf{M}inmize : F(x) = (f_1(x), f_2(x), ..., f_m(x))^T \qquad x = (x_1, x_2, ..., x_n) \in \Omega \tag{1}$$

where Ω is the decision (variable) space, x is a candidate with n-dimensional variables, $f_1(x)$, $f_2(x)$, ... , $f_m(x)$ represent m objective functions. Generally, in MOPs, the objective functions exhibit conflicts, where improving one objective function often leads to a degradation of the others. Several relevant definitions are presented.

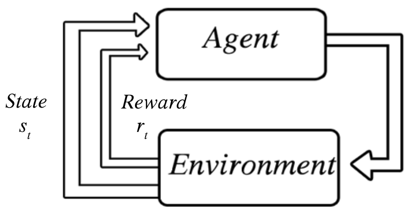
- Pareto Dominance: Two solutions x_1 and x_2, if and only if for all dimension i, $f_i(x_1) \leq f_i(x_2)$ and there exists at least one dimension j, $f_j(x_1) < f_j(x_2)$, then x_1 is said to dominate x_2, denoted as $x_1 \prec x_2$.
- Pareto optimal solution: There exists a solution $x_1 \in \Omega$, such that there is no solution dominating x_1: x_1 is called a Pareto optimal solution.
- Pareto Set (PS): The set is composed of multiple Pareto optimal individuals.
- Pareto Front (PF): The mapping of the PS on the objective space is called the PF.

2.2. Q-learning

The Q-learning is a model of RL, mainly composed of an agent, a state set S $=\{s_1, s_2, s_3, ..., s_N\}$, and an action set A $= \{a_1, a_2, ..., a_N\}$, as illustrated in Fig. 1(a). In this context, each state in the state set can correspond to any elements in the action set. The effectiveness of an action corresponding to a particular state at a given moment is learned by the agent and recorded as $Q(s_t, a_t)$. The value of $Q(s_t, a_t)$ is continuously updated by the agent.

Typically, Eq. 2 is used to update the Q-table shown in Fig. 1(b), where α represents the learning rate, and γ represents the reward discount factor, which is defined to control the degree of discounting future rewards. A higher value of γ suggests a greater emphasis on future rewards, encouraging the agent to prioritize actions that yield higher long-term rewards, and vice versa. The values of these parameters fall within the range (0,1), and r_t represents the penalty or reward obtained after taking an action. During the execution of the algorithm, the agent is presented with a state and is required to choose an action with the maximum Q-value. Subsequently, the Bellman equation, i.e., Eq. 2, is used to update the Q-value $Q(s_t, a_t)$ for the chosen action under that state.

$$\mathrm{Q}_{\mathrm{t+1}}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[r_t + \gamma * Max_a Q_t(s_{t+1}, a_t) - Q_t(s_t, a_t)] \tag{2}$$



(a) The Q-learning model.

	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$	$\mathbf{a_5}$	$\mathbf{a_6}$
$\mathbf{s_1}$	$Q(\mathbf{s_1}, \mathbf{a_1})$	$Q(\mathbf{s_1}, \mathbf{a_2})$	$Q(\mathbf{s_1}, \mathbf{a_3})$	$Q(\mathbf{s_1}, \mathbf{a_4})$	$Q(\mathbf{s_1}, \mathbf{a_5})$	$Q(\mathbf{s_1}, \mathbf{a_6})$
$\mathbf{s_2}$	$Q(\mathbf{s_2}, \mathbf{a_1})$	$Q(\mathbf{s_2}, \mathbf{a_2})$	$Q(\mathbf{s_2}, \mathbf{a_3})$	$Q(\mathbf{s_2}, \mathbf{a_4})$	$Q(\mathbf{s_2}, \mathbf{a_5})$	$Q(\mathbf{s_2}, \mathbf{a_6})$
$\mathbf{s_3}$	$Q(\mathbf{s_3}, \mathbf{a_1})$	$Q(\mathbf{s_3}, \mathbf{a_2})$	$Q(\mathbf{s_3}, \mathbf{a_3})$	$Q(\mathbf{s_3}, \mathbf{a_4})$	$Q(\mathbf{s_3}, \mathbf{a_5})$	$Q(\mathbf{s_3}, \mathbf{a_6})$

(b) The Q-table.

Fig. 1. An exposition on the Q-learning model structure and the contents of the Q-table. (a) represents the model for Q-learning. (b) is the Q-table containing Q-values for states and actions.

2.3. Related work

MOEAs have gained significant attention in the research field in recent decades. The primary objective is to find a set of solutions that can achieve optimal or near-optimal states for multiple conflicting objective functions without sacrificing each other. Balancing exploration and exploitation strategies is crucial during the evolutionary process of multi-objective optimization to obtain better solutions. To achieve this objective, numerous scholars have conducted in-depth research on MOEAs and published a substantial number of related research papers. Among these studies, two main research directions deserve particular attention.

2.3.1. Multicrossover operator improvement algorithms

The selection and adjustment of operator parameters have a significant impact on the convergence speed, solution quality, and exploration capability of the algorithm. Improving operator parameters is a common enhancement method in optimization algorithms. One of the earliest works in this field is the Fuzzy Adaptive Differential Evolution algorithm proposed by Liu et al. [14]. This algorithm utilizes relative function values and individual values from successive generations as inputs to design a fuzzy logic controller, which dynamically adapts the scaling factor and crossover probability. Qin et al. introduced the well-known SaDE algorithm [15], which incorporates two learning strategies and associated parameter controls for the crossover operator. These parameters are adaptively adjusted based on previous learning experiences. Another notable adaptive scheme is proposed by Brest et al. [16] in their jDE algorithm. They encode F and Cr into each individual, allowing for the acquisition of better individuals and propagation of parameter values during selection and offspring generation. The ACGDE [17] algorithm proposed by Qiu et al. follows a similar approach but incorporates Gaussian variation to control the parameter distribution.

In addition to the modification of operator parameters, the improvement of operators also includes the selection of combination operators. Similarly, Xie et al. proposed the MOEA/D-IMA algorithm [18], which assigns the selection probability of an operator based on the sum of the number of offspring that outperform their parents and the number of evolutionary iterations. Yan et al. proposed the MOEA/D-DOS algorithm [19], which introduces a two-operator co-evolutionary strategy based on operators' historical performance and the offspring's fitness improvement rate. Additionally, they proposed the MOEA/D-OPS algorithm [20], which utilizes an operator pre-selection mechanism to directly identify the corresponding crossover operator for each offspring, thereby avoiding the influence of historical information. Li et al. developed a variant of the Artificial Bee Colony algorithm [21], which proposes five local search operators, each of which is used through different stages of the scheduling process. This rational arrangement of heuristic operators aims to improve the overall efficiency of the algorithm and thus more skillfully address the inherent challenges posed by parallel batch processing in distributed pipelined systems. The algorithm is further improved in this paper through combinatorial crossover operator selection, which will be elaborated in the following sections.

2.3.2. RL-based evolutionary algorithms

In the context of integrating RL into MOEAs, one of the earliest works was proposed by Liao et al. [22]. Their approach involved discretizing the search space into a set of cells and identifying all possible search paths for each cell. By combining individual search paths with RL, they aimed to determine the optimal probability distribution for individual search paths. To assign different immediate rewards, they compared the objective vector of the current state (function values on each dimension) with non-dominated solutions. In the same year, Liao et al. [23] applied this approach to the practical problem of power system scheduling and voltage stabilization (MORL). Building upon this work, Kristof et al. proposed HB-MORL [11], which utilizes the HyperVolume(HV) [24] indicator as an evaluation criterion for selecting better individuals in the environmental selection process. Subsequently, several other papers have been published on MOEAs incorporating RL.

In 2014, Giorgos Karafotias et al. [12] proposed a generalized RL controller for controlling the parameters of evolutionary algorithms. This method represents the state using the set of all gene combinations and current parameters in the algorithm population. The action space includes optional parameter values, and the reward function evaluates the selection of individuals based on the improvement of the algorithm's fitness value. This reward signal guides the learning process of intelligence. Ning et al. [25] aimed to improve the performance of the multi-objective optimization algorithm based on decomposition(MOEA/D) [26] by adaptively controlling the neighborhood size and the differential evolution operator used in MOEA/D. Their algorithm utilizes a set of five observations to describe the population state. The reward is computed as the sum of improvements on each dimension of each subproblem, which is then used to update the reward.

Cheng et al. [27] presented a paper on applying a decomposition-based multi-objective evolutionary algorithm for multi-objective RL in an Extended Classifier System (XCS). The approach decomposes the MOPs into a set of subproblems, and an XCS classifier system is designed for each subproblem. Each subproblem has an objective function for evaluating the solution quality. Through the iteration of the evolutionary algorithm, the solutions of each subproblem are gradually improved, resulting in a set of approximate PS. Liu et al. [28] proposed an adaptive parameter adjustment approach for the Particle Swarm Optimization algorithm using RL. The algorithm defines four states based on the distance to the global optimal position: nearest, closer, farther, and farthest. Actions are divided into two groups: local search and global search. The performance improvement of an individual, measured by its single-objective fitness value or multi-objective dominance relation, is used as a reward signal to guide the individual's evolution.

In the most recent paper by Han et al. [29], RL is utilized to adaptively adjust the weight vectors and improve their utilization. The selection of states and actions for these strategies varies, but most of them employ the improvement of individual fitness values as a criterion to update the rewards and enhance the algorithm's performance. In addition to the applications mentioned earlier, RL can also be used to select operators adaptively. Sallam et al. [30] proposed a method that utilizes the entire algorithm to calculate the improved sum of fitness values as an RL's action for rewarding and punishing. This approach enables the adaptive adjustment of variance and crossover probability. Karam et al. [31] introduced RL to adaptively select operators for the population. The state of the population is determined by its diversity and quality, and different operators are selected based on different actions. The number of improved individuals in the population serves as the criterion for rewarding and punishing. Huang et al. [32] proposed a mechanism to update the optimal probability distribution of the search variation strategy in the combined fitness landscape using RL's reward and punishment mechanism.

In 2023, Jiao et al. [33] combined operator selection with RL to control the adaptive selection of operators. Different populations generated by different operators are defined as different states, and the survival rate of populations with different operators is used to update the Q-value table with rewards and penalties. Tian et al. [34] proposed an approach that utilizes deep RL to adaptively select optimal operators. RL is also incorporated in this paper for operator selection, and the specific algorithm will be described in the subsequent sections.

3. Motivation

After understanding the MOEAs mentioned above, it can be observed that the existing dominance-based and decomposition-based algorithms have to some extent increased the environmental selection pressure. However, in order to adapt the population to different stages during the evolutionary process, it is necessary to utilize search strategies with different characteristics to achieve the exploratory and exploitative nature of the algorithm. Additionally, in decomposition-based algorithms, each weight vector corresponds to a subproblem of varying difficulty, which may result in the algorithm failing to find the optimal solutions in certain regions of the decision space, leading to sparse areas in the objective space. Ideally, the population should tend towards convergence in the early stages and towards diversity in the later stages. Although the combination of multiple strategies has mitigated this issue, it is only for certain specific problems. To address this issue, it is necessary to establish the relationship between weight vectors and individuals, as well as the adaptive allocation of computational resources. Finally, the strategies proposed for different problems need to establish connections with each other rather than simply being a superposition of multiple strategies.

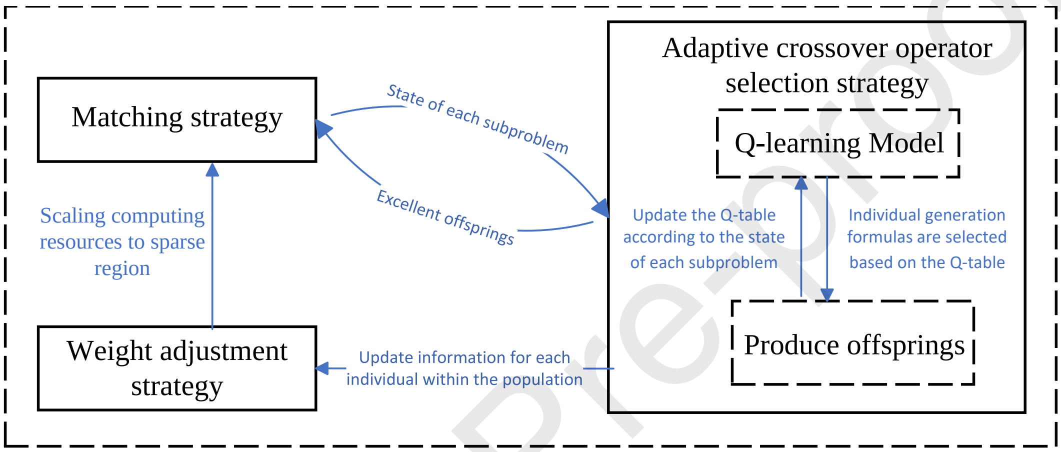


Fig. 2. The design structure of the algorithm presented in this paper.

In this paper, two strategies to improve the existing MOEAs are proposed, the matching strategy proposed matches multiple individuals to each weight vector to store historical population information, while also finding suitable weight vectors for offspring, avoiding the waste of computational resources. In the adaptive crossover operator strategy, actions for generating offspring with different exploration and exploitation characteristics are designed using the individual information from the matching strategy. This paper integrates the Q-learning framework into the proposed adaptive crossover operator selection strategy by designing three different states and enhances the Q-learning framework with the concept of reverse learning to refine the execution of the adaptive crossover operator strategy. The design structure is shown in Fig. 2. Initially, the weight vector adjustment strategy is employed to overcome the shortcomings within the matching strategy and reallocate computational resources. The matching strategy is interconnected with the adaptive crossover operator selection strategy; the former provides population information to the latter, which uses this information to generate superior offspring as effectively as possible. Similarly, the adaptive crossover operator selection strategy controls the timing of weight vector adjustments. These strategies are interdependent and complementary, continuously evolving the population towards the true PF at different stages.

4. The proposed MRL-MOEA

In this section, a novel multi-state reinforcement learning-based multi-objective evolutionary algorithm, MRL-MOEA, is proposed. MRL-MOEA employs a matching strategy to preserve historical information about the population by recording subproblem-matching individuals in an online form. Additionally, an adaptive selection strategy for crossover operators is introduced, which controls the operator selection for each individual in the population. This strategy utilizes Q-learning to learn from feedback received during the optimization process and to adjust the crossover operator selection accordingly, effectively responding to changes in the search environment.

4.1. Matching strategy

In the framework of MOEA/D, MOPs are decomposed into single-objective subproblems using a set of weight vectors to guide the population towards optimal solutions. Typically, this process involves a one-to-one matching mechanism between weights and individuals. However, this strict correspondence can lead to inefficiencies where an individual might be discarded due to lower fitness under one weight vector, despite performing well under others, wasting computational resources. Relaxing the one-to-one rule allows for a more flexible matching process where a single weight vector can correspond to multiple individuals, or vice versa.

Based on the analysis, a new matching strategy is proposed to preserve population diversity, convergence, and historical information. An online table records historical matches between weight vectors and individuals, with different rules for each table position. The process is illustrated in Algorithm 1. Initially, individual objective function values are normalized. Then, cosine similarities between each weight vector and normalized objective vectors are computed. The individual with the highest cosine similarity is matched with the corresponding weight vector. If a weight vector matches multiple individuals, they are positioned in the online table based on their aggregated function values and criteria like non-dominated ranking.

Algorithm 1 Matching strategy

Input: the population P, population size N, the weight vector λ, the ideal point Z^*;
Output: the online table $Vec\_MatchPop$;
for $i = 1 : N$ **do**
  $f_i^{'}(x) = f_i(x) - Z^*$;
end for
Population normalization by Eq. 4;
for $i = 1 : N$ **do**
  for $j = 1 : |\lambda|$ **do**
    Cosine_Similarity$_{i,j} = \frac{P_i \cdot \lambda_j}{\|P_i\| \cdot \|\lambda_j\|}$;
  end for
  if Multiple individuals matched the same weight vector **then**
    Update $Vec\_MatchPop$ according to sorting rules.
  end if
end for

Each weight vector is matched to an individual by calculating the cosine similarity value between the objective vector and the weight vector. The weight vectors are typically uniformly distributed in the first quadrant and on the axes, while the populations are initially random and may have different value ranges for each objective. To maximize the coverage of the weight vector, it is necessary to normalize the population values. In this paper, a similar approach to NSGA-III [35] is adopted, where the minimum value of the function representing the ideal point Z^* is determined for each objective prior to normalization, $Z_i^* = min\{f_i(x) \,|\, x \in \Omega, i = 1, ..., m\}$. The ideal point Z^* is subtracted from each objective value to obtain the minimum function value. Subsequently, the objective value minus the ideal point is translated, resulting in an objective vector with the ideal point as the origin. This translated objective vector can be expressed as:

$$f_i^{'}(x) = f_i(x) - Z^* \tag{3}$$

The extreme point $Z^{i,max}$ is determined by minimizing the corresponding achievement scalar function (ASF) for each axis in $f_i^{'}$. Once the extreme points are identified, these M extreme points are used to construct an M-dimensional hyperplane. The intercept a_j is then calculated for each objective axis and hyperplane. Finally, the objective function values can be normalized as:

$$f_j^{''}(x) = \frac{f_j^{'}(x)}{a_j}, j = 1, 2, ..., M \tag{4}$$

As shown in Eq. 4, the variable j refers to the number of objective functions, and a_j represents the intercept on the j-th objective axis that has been previously determined. Note that the intercept on each normalized objective axis is now at $f_j^{''} = 1$, and a hyper-plane constructed with these intercept points will satisfy $\sum_{j=1}^{M} f_j^{''} = 1$.

The next step is to calculate the cosine similarity of each objective function value vector to the weight vectors after normalization, which is calculated as follows:

$$\text{Cosine\_Similarity}_{i,j} = \frac{P_i \cdot \lambda_j}{\|P_i\| \cdot \|\lambda_j\|} \tag{5}$$

where P_i represents the i-th individual in the population and λ_j represent the j-th weight vector. $Cosine\_Similarity_{i,j}$ denotes the cosine value of the angle between the two. A higher value indicates a smaller angle. Therefore, the objective is to identify the individual that corresponds to the largest cosine value with the weight vector, and then add this individual to the set of matched weight vectors.

In addition, this paper sets the size of the online table to K, indicating that a weight vector can be matched with up to K individuals. Furthermore, a sorting mechanism is employed to arrange the multiple individuals that satisfy the matching criteria according to predefined rules.

$$Vec\_MatchPop = \{M_1, M_2, ..., M_K\} \tag{6}$$

where $\{M_1, M_2, ..., M_K\}$ initially stores a null value and subsequently stores the different individuals corresponding to the weight vectors.

Regarding the sorting rules of the online table, named Vec_MatchPop, M_1 is intended to maintain population diversity, M_2 is aimed at algorithm convergence, and the remaining K-2 individuals are designated to preserve the information of individuals with a favorable history, as shown in Fig. 3. The specific sorting rule is as follows: Firstly, the weight vector with the highest cosine similarity value to the objective vector P_i is identified. Next, it is checked whether there exists an individual in M_1 corresponding to this weight vector. If not, the individual is directly placed into M_1; otherwise, the PBI value of P_i is compared with that of the individual in M_1. The individual with the smaller PBI value is placed into M_1. If the individual has a larger PBI value, it is compared with the individuals in M_2. If M_2 is empty, the individual is directly placed into M_2. Otherwise, it is compared for domination with the individuals in M_2. The non-dominated individual is placed into M_2, and the dominated individual is placed into one of the remaining K-2 positions that are unoccupied. Ultimately, if all K-2 positions are filled, the individual is subjected to non-dominated sorting and crowding distance assessment with the individuals in the remaining K-2 positions, eliminating the dominated individuals or those with lower crowding distance.

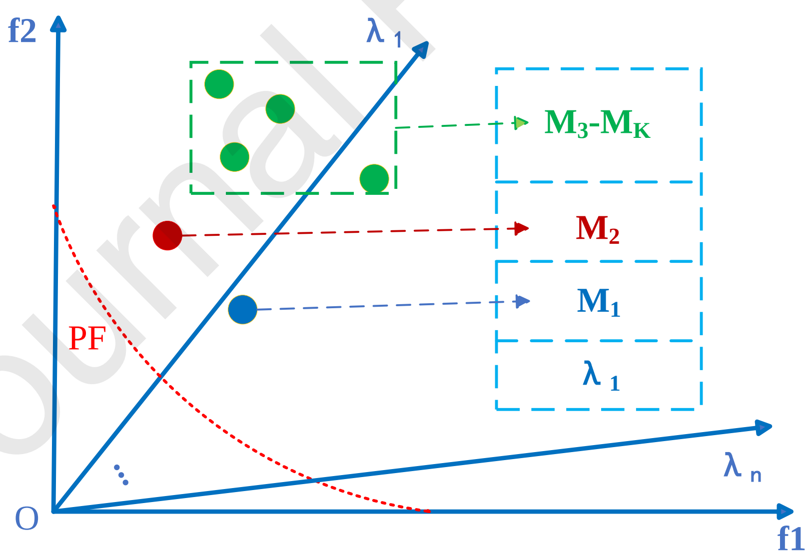


Fig. 3. The index of the storage location of individuals in the online table.

4.2. Adaptive crossover operator selection strategy

Evolutionary algorithms employ survival of the fittest to guide population evolution by generating high-quality offspring and eliminating inferior individuals for continuous improvement. Choosing suitable crossover operators enhances the likelihood of favorable offspring, expediting evolution. In MOEAs, using a single crossover operator can disrupt the balance between diversity and convergence. Scholars emphasize the critical role of crossover operator types and design strategies to balance exploration and exploitation with multiple operators. However, these strategies often depend on specific indicators, risking local optima and increased computational resources in complex optimization problems.

To address this issue, this paper proposes an adaptive selection strategy for SBX and DE crossover operators combined with Q-learning. The authors of the MOEA/D-DAS algorithm [36] have shown the exploration range of these operators using illustrative figures. The SBX operator excels in search speed and stability but may cause premature convergence, while the DE operator offers better global search capability and maintains high diversity. As shown in Fig. 4, the proposed strategy uses Q-learning to select the optimal crossover operator based on historical experience and the current environment. Although two different crossover operators are employed, parent selection impacts their effectiveness. Therefore, the strategy uses an online table of different individual types and search operators to maximize the probability of generating superior offspring. The implementation process is illustrated in Algorithm 2.

Algorithm 2 Adaptive crossover operator selection strategy

```
Input: the population P , the online table Vec_MatchPop, the number of updates for individuals corresponding to
  M_1 and M_2 for all weight vectors Global_flag , the Q-table.
Output: Q-table
for each i from 1 to N do
  if M_1 ≺ M_2 then
    The current state of operator s ← s_1;
  else if M_2 ≺ M_1 then
    The current state of operator s ← s_2;
  else
    The current state of operator s ← s_3;
  end if
  Choose an action a in the current state s that corresponds to the maximum value in the Q-table.
  Generation of new offspring O based on the action a selection crossover operator.
  Vec_MatchPop ← Matching individuals to Weight Vectors (λ_i, P, O) by Algorithm 1
  Update Global_flag(i, M_1, M_2);
  Update the current state s'.
  if the current state s' unlike previous state s then
    Update the learning rate a and the Q-value Q(s_t, a_t) in the Q-table according to Eq. 8, Table 1 and Eq. 2.
  end if
end for
```

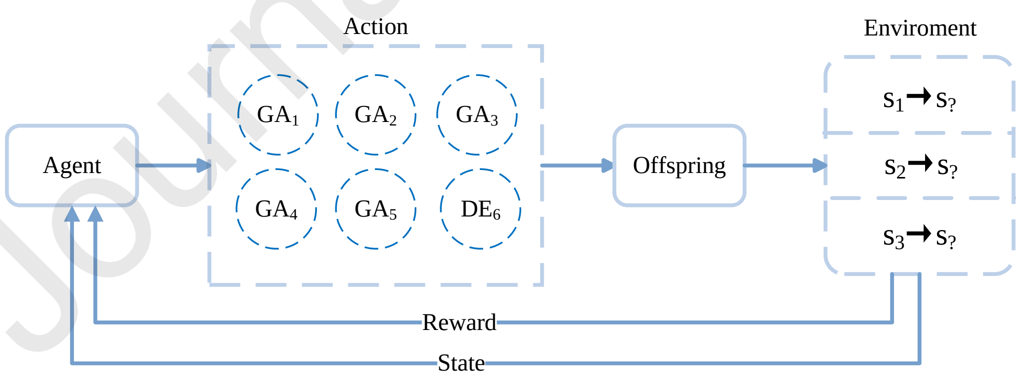


Fig. 4. The structure of the adaptive crossover operator selection strategy.

Initially, the state set S and action set A are defined, with three states and six actions, respectively. These six actions represent the use of six different methods to generate offspring, and the parent individuals used in these actions are based on those in the online table from the previously proposed matching strategy. For each subproblem, the individual matched to its corresponding weight vector can not only assess the current evolutionary state but also add more effective information about the search area.

When the state of a subproblem is judged to be unchanged or transitions to another state, it can provide a basis for rewards or penalties after the agent in Q-learning makes an action selection. The first state, s_1, is defined as the weight-associated individual M_1 dominating individual M_2. The second state, s_2, is defined as individual M_2 dominating individual M_1. The third state, s_3, is defined as individuals M_1 and M_2 not dominating each other. Additionally, by combining these weight-matched individuals with the two basic operators, six actions are generated, namely:

$$\begin{aligned}
&GA_1 : V_i = 0.5*((1+\gamma)*x_{r1}) + 0.5*((1-\gamma)*x_{r2})\\
&GA_2 : V_i = 0.5*((1+\gamma)*x_{r1}) + 0.5*((1-\gamma)*x_{rand})\\
&GA_3 : V_i = 0.5*((1+\gamma)*x_{r2}) + 0.5*((1-\gamma)*x_{rand})\\
&GA_4 : V_i = 0.5*((1+\gamma)*x_{rand}) + 0.5*((1-\gamma)*x_{rand})\\
&GA_5 : V_i = 0.5*((1+\gamma)*x_{r1}) + 0.5*((1-\gamma)*x_{T})\\
&DE_6 : V_i = x_{r1} + F*(x_{T1} - x_{T2})
\end{aligned} \tag{7}$$

Where x_{r1} and x_{r2} represent the individuals M_1 and M_2, respectively, in the online table of weight-matched individuals. x_{rand} denotes the remaining individuals in the online table $Vec\_MatchPop$, while x_T, x_{T1}, and x_{T2} represent individuals randomly selected from the neighborhood of the weight vector corresponding to the subproblem. By combining the distribution information of individuals in Fig. 3 and 5, it can be observed that the individuals at positions M_1 and M_2 are the optimal solutions for the current subproblems. Thus, there are two exploration directions: one is to explore the objective space around the current individuals, and the other is to explore other locations in the objective space, as the current individuals might be local optima.

In Eq.7, GA_1,GA_2, and GA_3 represent the exploitation of the decision space based on the characteristics of GA crossover operators. These operators are used to exploit the decision space of individuals at positions M_1 or M_2 to seek individuals with higher fitness. Similarly, the GA_4 exploits the decision space of individuals at positions M_3 to M_K. Although these individuals have lower fitness values, they are closer to the corresponding weight vectors in the objective space. Therefore, it is necessary to exploit these decision spaces, which may contain optimal individuals. Unlike GA_1 to GA_4, the GA_5 and DE_6 represent two different types of crossover operators that explore the decision space of individuals at position M_1 using information from neighboring individuals. This involves exploring the decision space between these optimal individuals at different scales.

During the evolutionary process of the population, the dominance ranking of individuals matched with the weight vectors corresponding to subproblems is constantly changing. Positions M_1 and M_2 hold two optimal individuals, and their dominance relationship corresponds to three different states, as shown in Fig. 5. By continuously generating offspring to replace these two individuals, the state also changes accordingly. When the state of a subproblem transitions to another state, it indicates that the current crossover operator has produced offspring with higher fitness, and this action is effective; it can be rewarded based on the data in Table 1. Conversely, when there is no state transition, it generally falls into two scenarios: the subproblem has only achieved a local optimum, or the subproblem has already approached the true PF.

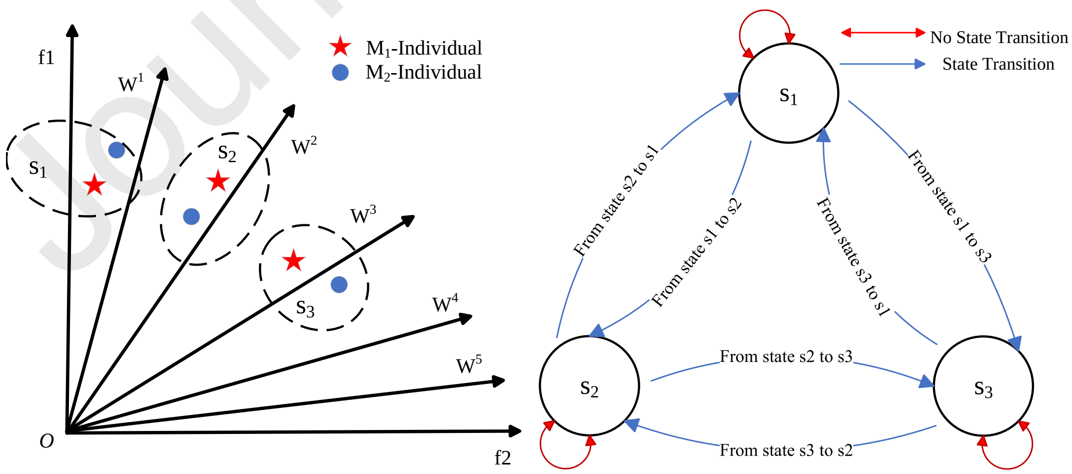


Fig. 5. Representation of the three states s_1, s_2, and s_3 of the subproblems in the objective space and their relationships.

Table. 1. The setting of reward and penalty values after transition between different states.

state	s_1	s_2	s_3
s_1	0	0.5	0.5
s_2	0.5	0	0.5
s_3	0.5	0.5	0

Additionally, the selection of operators based on the "state-action" values in the Q-table may sometimes lead to "local optima." To address this issue, the concept of reverse learning is introduced. When the state of a subproblem remains at s_3 multiple times, the operator selected according to the Q-table is considered ineffective in inducing a state change. Therefore, the maximum value in the Q-table is identified, and this value is subtracted from all the data in the table to form the updated Q-table. In practice, this manifests as repeatedly determining, based on the Q-table data, that the DE operator is more effective for a certain subproblem and using it to generate offspring, yet still being unable to change the current state. Consequently, the "least effective action" from the Q-table is chosen for a trial. In such instances, a reverse learning mechanism is employed, where the maximum and minimum operator selection probabilities for each state in the Q-table are interchanged. This strategy effectively enables the population to break free from the current state of stagnation.

Furthermore, it should be emphasized that in RL the value in the reward table represents the immediate reward obtained after taking an action in the current state, denoted as r_t in Eq. 2. The learning rate, denoted as α in Eq. 2, determines the trade-off between the new Q-value and the previous Q-value during each update. A higher learning rate allows the new Q-value to approach the target value more quickly, but it may lead to instability. On the other hand, a smaller learning rate provides more stability but requires more time for convergence. Therefore, during the initial stages of the algorithm, a higher value of α is established. As the algorithm progresses, α should decrease to maintain a balance between convergence and exploration. The value of α is set as Eq. 8, where FE represents the current number of iterations, and $MaxFE$ is the maximum number of iterations. This learning rate adjustment methodology has garnered widespread acceptance in practical applications.

$$\alpha = 1 - 0.9 * \left(\frac{\text{FE}}{\text{MaxFE}}\right) \tag{8}$$

4.3. The framework of MRL-MOEA

The proposed algorithmic framework in this paper comprises several main components, as illustrated in Algorithm 3. Firstly, the Q-table, which records the "state-action" key-value pairs $Q(s_t, a_t)$, is also initialized. Additionally, Algorithm 1 can then be used to update this table regarding the initial population and their corresponding weight vectors. Lines 6 to 10 of the pseudocode count the number of updates for individuals corresponding to positions M_1 and M_2 in the current *Vec_MatchPop* table for N rows, primarily to monitor the population changes following the execution of Algorithm 1 and Algorithm 2. The variables $h1$ and $h2$ represent the updated status of the previous and current generations, respectively. In lines 12 to 15 of the pseudocode, if the population is determined to be unchanged, the switch $FLAG$ for the weight vector adjustment strategy is activated at line 14, and the Q-table is updated based on the concept of reverse learning.

Additionally, the execution of the weight vector adjustment strategy not only requires the activation of the $FLAG$ variable but also depends on certain stages of the population's evolution. This is because, although Algorithm 1 reestablishes the correspondence between weight vectors and individuals and effectively preserves valuable information during the population's evolutionary process, there are still instances where weight vectors do not correspond to any individual. The inability of a weight vector to match an individual may indicate that the subproblem is challenging and no effective solution has been found, leading to sparse areas in the PF at that stage. As the population evolves into the later stages, most subproblems have found their best solutions, and the algorithm has approached the true PF infinitely closely. At this point, by using weight vectors to eliminate individuals in dense areas of the objective space and constructing new weight vectors in sparse areas (lines 16 to 18), computational resources can be more rationally allocated to these regions, allowing the algorithm to achieve a uniformly distributed approximation of the PF.

Algorithm 3 The framework of MRL-MOEA

Initialize the population P, Population size N, the weight vector λ, reference point Z^*, the counter $FLAG$, the maximum number of iterations $MaxFE$, the number of updates for individuals corresponding to M_1 and M_2 for all weight vectors $Global\_flag$, the online table $Vec\_MatchPop$, and the Q-table.
$FE = N$, $FLAG = 0$;
$Vec\_MatchPop \leftarrow$ Matching individuals to Weight Vectors (λ, P) by Algorithm 1
/* Main loop */;
while $FE < MaxFE$ **do**
    $h1 \leftarrow sum(Global\_flag(1:N,1:2))$;
    for each i from 1 to N **do**
        Execute Algorithm 2 to update the Q-table based on the current state s and the subsequent state s';
    end for
    $h2 \leftarrow sum(Global\_flag(1:N,1:2))$;
    $FE \leftarrow FE + N$;
    if $h1/h2 == 1$ **then**
        Q-table $\leftarrow max($Q-table$,2)-$Q-table; /* The reverse learning mechanism is executed.*/
        $FLAG \leftarrow 1$
    end if
    if $FLAG == 1 \&\& FE \leq 0.8 * MaxFE$ **then**
        $\lambda \leftarrow$ weight adjustment strategy of the MOEA/D-AWA;
    end if
end while
for each i from 1 to N **do**
    $P(i) \leftarrow Vec\_MatchPop(i,1)$; /* Store the individuals from the online table sequentially into the population. */
end for
return P;

4.4. Computational complexity of MRL-MOEA

To analyze the computational complexity of the proposed MRL-MOEA algorithm, the main computations are concentrated at each step in Algorithm3. The primary computational costs stem from the strategies involved. We denote the population size, number of decision variables, neighborhood size, matching size of weighted individuals, and number of weight adjustments as N, m, T, k, and Q, respectively. In Strategy 1, the matching strategy, the computational complexity is $O(k*T*N)$, for Strategy 2, it is $O(N^2)$, and for the weight adjustment strategy, it is $O(Q*m*N^2)$. Considering all strategies, the overall iterative complexity of the algorithm is expressed as $O(Q*m*N^2)$. Taking into account the initialization time complexity of $O(m*T*N)$, and considering $N >> T > k$, the total time complexity of MRL-MOEA is $O(Q*m*N^2)$.

5. Analysis of experimental results

This section evaluates and compares the MRL-MOEA algorithm with other algorithms using commonly used benchmark functions. The aim is to assess the effectiveness of these algorithms based on the obtained indicator results. The comparison algorithms and test problems used in the experiments were executed on Platform 4.1 [37] using Matlab version R2021a, operating system version Windows 10, and 12 GB of RAM.

5.1. Experimental design

5.1.1. Benchmark test problems

Benchmark test problems from the DTLZ [38] and WFG [39] test suites are utilized in this paper. These test suites consist of 13 test problems with distinct features of PF. Specifically, the selected problems include DTLZ1-DTLZ4 and WFG1-WFG9. The DTLZ1-DTLZ4 test problems exhibit characteristics such as nonlinearity, nonconvexity, and discrete decision variables. Similarly, the WFG1-WFG9 test problems also possess these features. These inherent characteristics make these problems particularly challenging for evaluating and comparing the performance and robustness of MOEAs. Additionally, the DTLZ1-DTLZ4 and WFG1-WFG9 test suites offer the flexibility to extend the number of objectives and decision variables, making them suitable for evaluating algorithms designed for MOPs and high-dimensional decision spaces.

5.1.2. Performance indicators

In order to assess the algorithm's performance, this paper employs two evaluation indicators: the Inverted Generational Distance (IGD) [40] and the HV indicators. The IGD indicator measures the average distance between the set of optimal points in the true PF, which is uniformly distributed, and the set of solutions obtained by the algorithm to approximate the PF.

A smaller IGD value indicates better convergence and diversity of the algorithm, as well as a closer proximity of the generated solution set to the true PF. To compare the IGD indicator between algorithms, the Wilcoxon rank sum test [41] was conducted with a confidence level of 95%.

$$\mathrm{IGD(PF,P)} = \frac{1}{|\mathrm{PF}|}\sum_{\mathrm{i=1}}^{|\mathrm{PF}|}\min_{\mathrm{j=1...|P|}}\mathrm{d(PF_i,P_j)} \tag{9}$$

The expression describes the notation used in the context of multi-objective optimization. Here, PF represents the solution set of the true PF [42], P represents the approximate PF solution set obtained through the algorithm, and $d(PF_i, P_j)$ represents the Euclidean distance between the true solution PF_i and the approximate solution P_j. In other words, for each solution i in the true PF solution set, the solution i with the closest Euclidean distance among the approximate PF solutions obtained by the algorithm is identified and averaged. This average value represents the final value of the IGD.

The HV indicator represents the sum of the hypercubic volumes enclosed by the solution set of the approximate PF obtained by the algorithm and the predefined reference point, typically set as the maximum objective function value. A larger HV value indicates a larger hypercubic volume occupied by the solution set of the approximate PF in the objective space, closer proximity of the algorithm's generated approximate PF to the true PF, and better comprehensive performance of the algorithm.

$$\mathrm{HV(P,Z^{nad})} = \cup_{\mathrm{p\in P}}^{\mathrm{P}}\mathrm{V(p,Z^{nad})} \tag{10}$$

Where P represents the approximate PF solution obtained by the algorithm, Z^{nad} denotes the predefined reference point which is dominated by all individuals in P, and $V(p, Z^{nad})$ represents the Lebesgue measure, which quantifies the hypervolume of the set P. Therefore, the HV value can assess both the convergence and diversity of the solution set obtained by the algorithm.

5.1.3. Experimental parameter setting

This summary focuses on the experimental parameters. Different test problems require different settings for the number of objective functions (m) and population size (N). For the DTLZ and WFG series test problems, the values of N are set to 91, 210, 156, and 275 when the values of m are 3, 5, 8, and 10, respectively. The number of iterations is set to 1000 generations, and each algorithm is executed independently 30 times. For the GA operator, the crossover probability (p_c), the SBX distribution index (d_c) [43], the polynomial mutation distribution index (d_m) [44], and the probability of mutation (p_m) [45] were set to 1, 20, 20, and $1/D$, respectively. For the DE operator, the crossover rate (CR) was set to 1, differential weight (F) was set to 0.5, and neighborhood size (T) was set to $N/10$. Additionally, the parameters for the other algorithms are set according to the configurations specified in their respective original papers where they were proposed.

5.2. Experimental analysis

This section presents simulation test results for two types of problems: MOPs and Many-Objective Problems (MaOPs). A problem is classified as MaOPs when the number of objective functions (m) exceeds 3. Tables 2 and 3 display the test results for the MOPs, while Tables 5 and 6 present the test results for the MaOPs. The symbols in the tables indicate the performance of the compared algorithms, with "+" indicating better performance, "-" indicating worse performance, and "=" indicating comparable performance. The highlighted section represents the optimal result for each test instance. The experiment is described in detail in the following two parts.

5.2.1. Performance on MOPs

This section presents the performance of MRL-MOEA on MOPs, using the proposed algorithm to compare it with five other comparison algorithms on the DTLZ and WFG test instances. The IGD and HV values of the algorithms are given in Tables 2 and 3, respectively. In these tables, the entries outside the parentheses represent the mean of 30 trials, while the entries inside represent the standard deviation.

Based on the IGD values presented in Table 2, it can be observed that the seven test problem instances, namely DTLZ1-DTLZ4, WFG4, WFG6, and WFG7, are optimally handled by the proposed MRL-MOEA algorithm when compared with MOEA/D-DQN [34], RLMMDE [29], IDBEA [46], ARMOEA [47], and MSEA [48]. The subsequent best performer is AR-MOEA, followed by MSEA, IDBEA, RLMMDE, and MOEA/D-DQN, with best values of 4, 2, 0, 0, and 0, respectively. The RLMMDE may focus more on solving irregularly shaped PF, whereas the emphasis of the MRL-MOEA algorithm proposed in this paper lies in the convergence and diversity maintenance of PF with regular shapes. Overall, the convergence of the proposed MRL-MOEA algorithm surpasses that of the other comparative algorithms.

Table. 2. The IGD values obtained by different algorithms on the DTLZ1-DTLZ4 and WFG1-WFG9 test suites.

Problem	MOEADDQN	RLMMDE	IDBEA	ARMOEA	MSEA	MRL-MOEA
DTLZ1	2.0572e-2 $_{1.20e-5}$ -	2.0559e-2 $_{1.35e-6}$ -	6.2338e-1 $_{5.99e-1}$ -	2.0573e-2 $_{2.21e-5}$ -	2.0874e-2 $_{1.06e-4}$ -	**2.0410e-2** $_{\mathbf{1.05e-4}}$
DTLZ2	5.4639e-2 $_{7.91e-5}$ -	5.4492e-2 $_{1.23e-5}$ -	5.4493e-2 $_{1.57e-4}$ -	5.4470e-2 $_{1.73e-5}$ -	5.5417e-2 $_{4.46e-4}$ -	**5.4464e-2** $_{\mathbf{3.69e-7}}$
DTLZ3	5.5843e-2 $_{9.71e-4}$ -	6.0790e-2 $_{2.48e-2}$ -	5.7059e-2 $_{4.48e-3}$ -	5.4789e-2 $_{2.21e-4}$ -	5.5576e-2 $_{4.39e-4}$ -	**5.4469e-2** $_{\mathbf{1.02e-5}}$
DTLZ4	5.6823e-2 $_{2.02e-3}$ -	5.4564e-2 $_{2.79e-5}$ -	3.3188e-1 $_{2.74e-1}$ -	4.1653e-1 $_{3.10e-1}$ -	1.4486e-1 $_{2.72e-1}$ -	**5.4528e-2** $_{\mathbf{2.18e-4}}$
WFG1	1.8792e-1 $_{9.61e-3}$ -	4.1920e-1 $_{1.39e-1}$ -	1.6304e-1 $_{6.66e-3}$ +	**1.4931e-1** $_{\mathbf{2.77e-3}}$ **+**	2.7540e-1 $_{1.03e-1}$ -	1.8193e-1 $_{2.62e-2}$
WFG2	2.4504e-1 $_{2.98e-2}$ -	1.6443e-1 $_{1.33e-3}$ +	1.7157e-1 $_{2.15e-3}$ +	**1.6397e-1** $_{\mathbf{1.07e-3}}$ **+**	1.6544e-1 $_{3.64e-3}$ +	2.2057e-1 $_{1.70e-2}$
WFG3	1.3529e-1 $_{8.93e-4}$ =	3.6340e-1 $_{2.23e-1}$ -	1.0685e-1 $_{1.39e-3}$ +	1.1807e-1 $_{6.71e-3}$ +	**7.9932e-2** $_{\mathbf{2.95e-3}}$ **+**	1.4084e-1 $_{2.64e-2}$
WFG4	3.1524e-1 $_{2.75e-2}$ -	2.2152e-1 $_{3.43e-4}$ -	2.4060e-1 $_{4.66e-3}$ -	2.2083e-1 $_{1.26e-4}$ -	2.2238e-1 $_{1.79e-3}$ -	**2.2012e-1** $_{\mathbf{1.46e-3}}$
WFG5	2.9034e-1 $_{1.78e-2}$ -	2.3137e-1 $_{2.05e-3}$ =	2.3675e-1 $_{2.70e-3}$ -	**2.2983e-1** $_{\mathbf{7.88e-5}}$ **=**	2.3023e-1 $_{1.33e-3}$ =	2.3065e-1 $_{1.82e-3}$
WFG6	3.1589e-1 $_{2.70e-2}$ -	2.3864e-1 $_{3.42e-2}$ -	2.7930e-1 $_{1.25e-2}$ -	2.3993e-1 $_{9.09e-3}$ =	2.4313e-1 $_{9.86e-3}$ -	**2.3685e-1** $_{\mathbf{9.60e-3}}$
WFG7	2.6689e-1 $_{2.69e-2}$ -	2.2319e-1 $_{8.80e-4}$ -	2.6444e-1 $_{1.02e-2}$ -	2.2120e-1 $_{1.68e-4}$ -	2.2130e-1 $_{1.81e-3}$ -	**2.1955e-1** $_{\mathbf{1.17e-3}}$
WFG8	3.5025e-1 $_{4.28e-3}$ -	2.7949e-1 $_{9.08e-3}$ =	3.0292e-1 $_{5.52e-3}$ -	2.7437e-1 $_{2.67e-3}$ =	**2.6940e-1** $_{\mathbf{2.94e-3}}$ **+**	2.7526e-1 $_{3.10e-3}$
WFG9	3.0931e-1 $_{3.38e-2}$ -	2.5648e-1 $_{5.25e-2}$ -	2.3539e-1 $_{5.56e-3}$ -	**2.2089e-1** $_{\mathbf{1.08e-3}}$ **=**	2.2418e-1 $_{2.27e-2}$ =	2.2128e-1 $_{2.09e-3}$
+/-/=	0/12/1	1/10/2	3/10/0	3/6/4	3/8/2	

Additionally, the performance of MRL-MOEA is also reflected in the HV values presented in Table 3. It is evident that MRL-MOEA exhibits superior performance in terms of HV values. Moreover, when considering the WFG5-WFG9 test problems, the proposed MRL-MOEA demonstrates competitive performance. This can be attributed to the strategy of adaptively adjusting weight vectors based on the contributions of candidate solutions in the external archive. This strategy effectively enhances the homogeneity of the PF by adjusting the weight vectors. Overall, MRL-MOEA consistently outperforms other algorithms on the test problems of the WFG series.

Table. 3. The HV values obtained by different algorithms on the DTLZ1-DTLZ4 and WFG1-WFG9 test suites.

Problem	MOEADDQN	RLMMDE	IDBEA	ARMOEA	MSEA	MRL-MOEA
DTLZ1	8.4145e-1 $_{1.86e-4}$ +	**8.4173e-1** $_{\mathbf{1.17e-6}}$ **+**	3.1649e-1 $_{4.03e-1}$ -	8.4151e-1 $_{2.92e-4}$ +	8.4143e-1 $_{2.21e-4}$ +	8.3813e-1 $_{4.02e-4}$
DTLZ2	5.5875e-1 $_{4.62e-4}$ -	5.5959e-1 $_{3.83e-5}$ -	5.5961e-1 $_{1.42e-5}$ =	5.5960e-1 $_{1.82e-5}$ -	5.5955e-1 $_{8.74e-4}$ =	**5.5961e-1** $_{\mathbf{2.23e-6}}$
DTLZ3	5.5659e-1 $_{1.95e-3}$ -	5.5330e-1 $_{2.78e-2}$ =	5.4652e-1 $_{1.00e-2}$ -	5.5532e-1 $_{2.16e-3}$ -	5.5616e-1 $_{2.18e-3}$ -	**5.5948e-1** $_{\mathbf{1.94e-4}}$
DTLZ4	5.5918e-1 $_{3.55e-4}$ -	5.5951e-1 $_{8.46e-5}$ -	4.1819e-1 $_{1.42e-1}$ -	3.9068e-1 $_{1.56e-1}$ -	5.1154e-1 $_{1.44e-1}$ -	**5.5961e-1** $_{\mathbf{5.30e-4}}$
WFG1	9.3572e-1 $_{2.38e-3}$ +	8.9075e-1 $_{2.42e-2}$ -	9.3924e-1 $_{4.95e-3}$ +	**9.4255e-1** $_{\mathbf{7.27e-4}}$ **+**	9.1615e-1 $_{2.14e-2}$ =	9.1218e-1 $_{9.04e-3}$
WFG2	8.9320e-1 $_{2.36e-2}$ -	9.3064e-1 $_{7.47e-4}$ +	9.2459e-1 $_{1.53e-3}$ +	9.3066e-1 $_{7.39e-4}$ +	**9.3753e-1** $_{\mathbf{7.74e-4}}$ **+**	9.1874e-1 $_{4.28e-3}$
WFG3	3.7478e-1 $_{7.48e-4}$ +	3.5926e-1 $_{2.03e-2}$ =	3.9216e-1 $_{1.38e-3}$ +	3.8079e-1 $_{3.14e-3}$ +	**3.9905e-1** $_{\mathbf{2.00e-3}}$ **+**	3.6215e-1 $_{1.17e-2}$
WFG4	5.1568e-1 $_{1.60e-2}$ -	5.5784e-1 $_{3.19e-4}$ =	5.4464e-1 $_{1.65e-3}$ -	**5.5890e-1** $_{\mathbf{2.27e-4}}$ **+**	5.5842e-1 $_{6.94e-4}$ =	5.5810e-1 $_{7.55e-4}$
WFG5	4.7705e-1 $_{1.48e-2}$ -	5.1064e-1 $_{4.36e-3}$ -	5.1487e-1 $_{1.25e-3}$ -	5.1831e-1 $_{8.29e-5}$ -	5.1830e-1 $_{7.09e-4}$ -	**5.1962e-1** $_{\mathbf{1.16e-3}}$
WFG6	4.6955e-1 $_{4.44e-2}$ -	**5.2061e-1** $_{\mathbf{3.69e-2}}$ **+**	4.8324e-1 $_{1.35e-2}$ -	5.0455e-1 $_{1.31e-2}$ -	5.0202e-1 $_{1.35e-2}$ -	5.1857e-1 $_{1.42e-2}$
WFG7	5.3196e-1 $_{2.21e-2}$ -	5.5361e-1 $_{7.77e-4}$ -	5.3667e-1 $_{2.32e-3}$ -	5.5802e-1 $_{2.99e-4}$ -	5.5984e-1 $_{3.83e-4}$ -	**5.6015e-1** $_{\mathbf{3.11e-4}}$
WFG8	4.3057e-1 $_{4.66e-3}$ -	4.7407e-1 $_{6.09e-3}$ -	4.6206e-1 $_{2.51e-3}$ -	4.7751e-1 $_{1.48e-3}$ -	4.8155e-1 $_{1.26e-3}$ -	**4.8252e-1** $_{\mathbf{1.45e-3}}$
WFG9	4.7800e-1 $_{3.89e-2}$ -	4.9813e-1 $_{5.38e-2}$ -	5.2468e-1 $_{3.95e-3}$ -	5.3721e-1 $_{2.64e-3}$ =	5.3674e-1 $_{2.38e-2}$ -	**5.3805e-1** $_{\mathbf{3.24e-3}}$
+/-/=	3/10/0	3/7/3	3/9/1	5/7/1	3/7/3	

These algorithms were ranked based on the analysis of their obtained HV and IGD values. Table 4 lists the rankings of these algorithms in terms of IGD and HV values, as determined by the Friedman test [41]. MRL-MOEA achieves average rankings of 2.2692 and 2.6923 for IGD and HV values, respectively, indicating its superiority over the other four algorithms, with the exception of MSEA. To better comprehend the rankings of these algorithms on IGD and HV

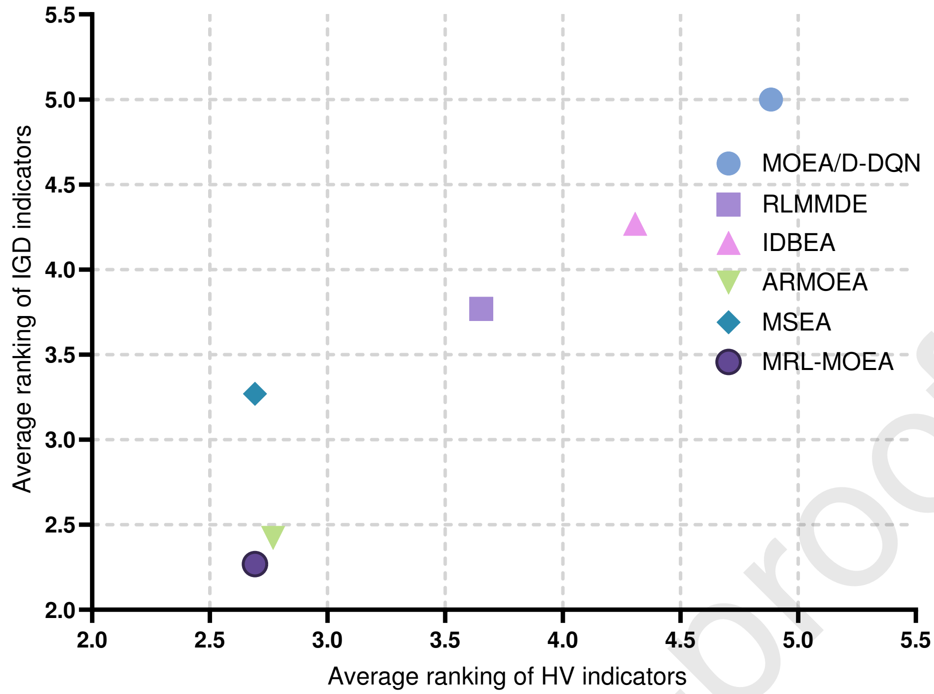


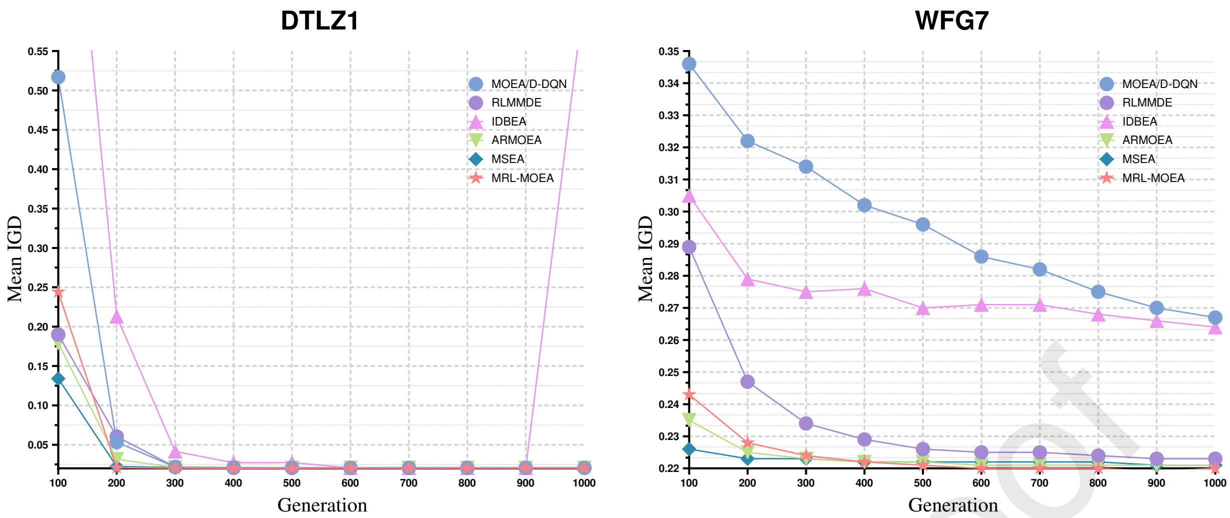
Fig. 6. Average ranking plot of the six algorithms on IGD and HV values. The horizontal and vertical axes show the average ranking of HV and IGD values, respectively.

indicators, the data from Table 4 is visualized in Fig. 6. In this figure, MRL-MOEA demonstrates the lowest ranking in the two-dimensional space, suggesting that it outperforms the other algorithms and possesses a competitive edge.

Table. 4. The Friedman ranking of the six algorithms. The left column represents the ranking based on the IGD values, while the right column represents the ranking based on the HV values.

	IGD	HV
Algorithm	Ranking	Ranking
MOEA/D-DQN	5	4.8846
RLMMDE	3.7692	3.6538
IDBEA	4.2692	4.3077
ARMOEA	2.4231	2.7692
MSEA	3.2692	**2.6923**
MRL-MOEA	**2.2692**	**2.6923**

To examine the evolutionary behaviors of these algorithms, Fig. 7 presents their average IGD trajectories. The IGD values described in Fig. 7 are obtained by sampling every 100 generations and averaging over 30 such samples. As observed in Fig. 7(a), on the DTLZ1 test instance, MRL-MOEA exhibits superior convergence performance compared to its competitors. The convergence rates of the other comparative algorithms do not match that of MRL-MOEA. Moreover, for the WGF7 test instance, Fig. 7(b) indicates that although MRL-MOEA's convergence is slower in the early stages of population evolution, it continues to evolve in the later stages. When other algorithms decelerate or get trapped in local optima, the reinforcement learning-based adaptive operator selection strategy can compel individuals to discover new directions for evolution.



(a) MOP-IGD-DTLZ

(b) MOP-IGD-WFG

Fig. 7. The average IGD trajectories of the six algorithms on the DTLZ1 and WFG7 test instances. (a) The average IGD trajectories of six algorithms on DTLZ1 test problem with three objectives. (b) The average IGD trajectories of the six algorithms on WFG7 test instance with three objectives.

The obtained approximate PF for all algorithms on the DTLZ2 test instance is shown in Fig. 8. By combining this figure with Tables 2 and 3 can be found that the MRL-MOEA algorithm outperforms similar comparative algorithms on the tested problems in DTLZ1-DTLZ4. Based on the convergence trends of all algorithms shown in Fig. 7, it can be observed from Fig. 8 that the MRL-MOEA algorithm not only demonstrates good convergence but also achieves a uniformly distributed approximate PF. This fully illustrates the competitiveness of the proposed algorithm in solving MOPs and further validates the effectiveness of the strategy proposed in this paper.

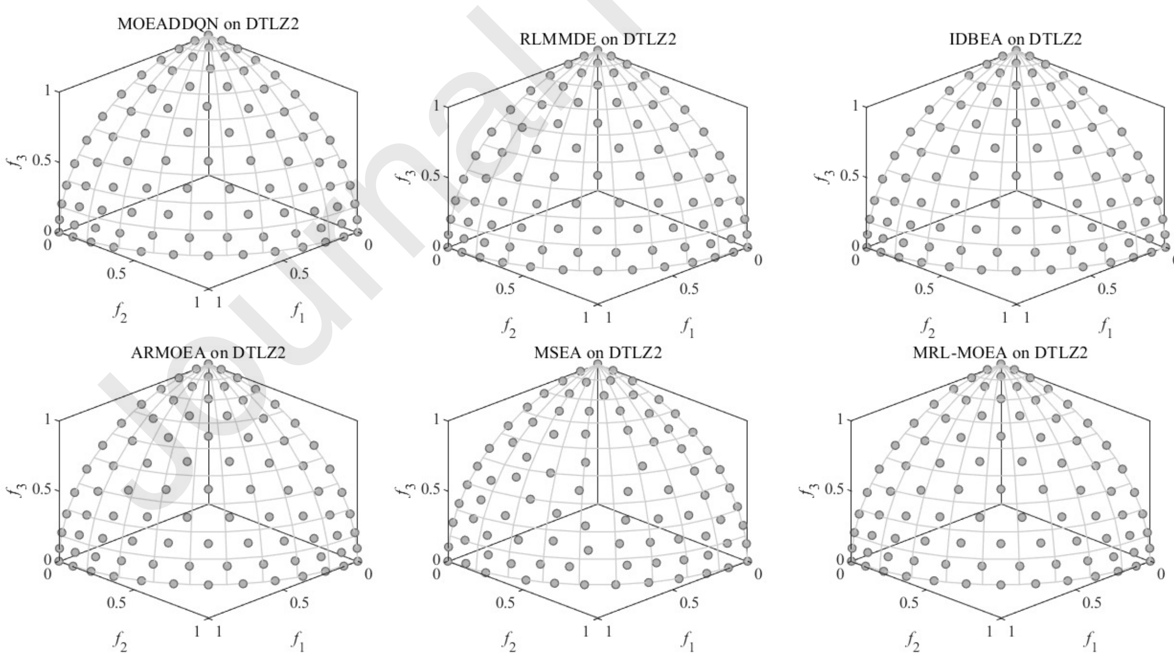


Fig. 8. The approximate PF obtained by MRL-MOEA and the comparison algorithms on the DTLZ2 test problem.

5.2.2. Performance on MaOPs

In this section, the primary objective is to evaluate the performance of MRL-MOEA on MaOPs. Consequently, MRL-MOEA is subjected to further testing on the DTLZ1-DTLZ4 and WFG1-WFG9 test suites with 5, 8, and 10 objectives. Additionally, to verify the effectiveness of the proposed strategy for solving MaOPs, some of the comparative algorithms previously used for solving MOPs were replaced. The comparative algorithms employed include MOEA/D-DQN, RLMMDE, IDBEA, onebyoneEA [49], and MaOEAIGD [50]. Tables 5 and 6 showcase the comparative statistical performance of MRL-MOEA against other comparison algorithms, as measured by IGD and HV indicators for MaOPs.

According to the Wilcoxon rank sum test outcomes presented in Table 5, MRL-MOEA may not always secure the top position for certain eight-objective test instances. However, MRL-MOEA consistently demonstrates superior performance over the comparative algorithms, particularly excelling on the WFG4-WFG9 test suites. In addition, the HV values outlined in Table 6 reveal that MRL-MOEA's performance is not as robust, with IDBEA exhibiting stronger results. This could potentially be ascribed to IDBEA's environmental selection mechanism, which is adept at preserving population diversity. In contrast to the IGD indicator, the value of HV is evaluated using the approximate PF obtained by the algorithm. Therefore, when the true PF information is available, the IGD indicator more accurately reflects the comprehensive performance of the algorithm. Despite this, the novel MRL-MOEA still surpasses the other four algorithms in the comparison, affirming its efficacy. In sum, the MRL-MOEA proves to be a more potent tool for solving MaOPs, reinforcing its status as an effective algorithm in the field of MOEAs.

Table. 5. The IGD values obtained by MaOEAs for MaOPs with 5, 8, and 10 objectives.

Problem	M	MOEADDQN	RLMMDE	IDBEA	onebyoneEA	MaOEAIGD	MRL-MOEA
WFG1	5	5.0219e-1 $_{2.05e-2}$ +	8.0708e-1 $_{9.98e-2}$ +	**3.7226e-1** $_{\mathbf{3.47e-3}}$ +	6.9670e-1 $_{3.50e-2}$ +	3.9271e+0 $_{8.08e-1}$ -	9.5324e-1 $_{2.88e-1}$
	8	1.4356e+0 $_{1.53e-1}$ +	2.0587e+0 $_{5.15e-1}$ =	**1.0949e+0** $_{\mathbf{3.41e-1}}$ +	1.6657e+0 $_{4.22e-2}$ +	7.9571e+0 $_{2.28e+0}$ -	2.1809e+0 $_{5.14e-1}$
	10	1.5074e+0 $_{1.14e-1}$ +	2.3559e+0 $_{6.31e-1}$ -	**1.1756e+0** $_{\mathbf{2.90e-1}}$ +	1.7587e+0 $_{4.32e-2}$ +	8.6997e+0 $_{3.47e+0}$ -	1.8527e+0 $_{2.11e-1}$
WFG2	5	6.0077e-1 $_{4.22e-2}$ -	9.1601e-1 $_{6.72e-2}$ -	**4.0090e-1** $_{\mathbf{2.87e-3}}$ +	6.6367e-1 $_{5.09e-2}$ -	1.7110e+0 $_{1.60e-1}$ -	5.6662e-1 $_{3.95e-2}$
	8	1.6160e+0 $_{1.29e-1}$ +	1.1624e+0 $_{2.23e-1}$ +	**9.6170e-1** $_{\mathbf{8.89e-2}}$ +	1.7896e+0 $_{6.41e-2}$ +	2.4077e+0 $_{5.02e-1}$ =	2.3474e+0 $_{5.08e-1}$
	10	**1.5353e+0** $_{\mathbf{8.79e-2}}$ +	1.6121e+0 $_{6.31e-1}$ +	2.0469e+0 $_{3.73e+0}$ +	1.8444e+0 $_{5.90e-2}$ +	2.3152e+0 $_{3.82e-1}$ +	2.3339e+0 $_{8.02e-1}$
WFG3	5	2.0238e+0 $_{4.35e-2}$ -	**3.5084e-1** $_{\mathbf{2.63e-1}}$ +	4.0113e-1 $_{2.23e-2}$ +	1.2056e+0 $_{1.36e-1}$ -	5.4304e+0 $_{1.79e-2}$ -	5.2962e-1 $_{7.75e-2}$
	8	2.6357e+0 $_{7.14e-2}$ -	**5.1006e-1** $_{\mathbf{2.74e-1}}$ +	8.8853e+0 $_{2.38e-3}$ -	3.4700e+0 $_{2.77e-1}$ -	8.8717e+0 $_{3.66e-2}$ -	8.4484e-1 $_{1.07e-1}$
	10	3.4885e+0 $_{8.76e-2}$ -	**5.0538e-1** $_{\mathbf{3.25e-1}}$ +	1.1188e+1 $_{2.61e-3}$ -	4.7550e+0 $_{4.30e-1}$ -	1.0866e+1 $_{1.74e+0}$ -	9.0768e-1 $_{9.07e-2}$
WFG4	5	1.2966e+0 $_{1.20e-1}$ -	2.0147e+0 $_{1.33e-1}$ -	9.6829e-1 $_{7.79e-4}$ =	1.4896e+0 $_{1.01e-1}$ -	6.4827e+0 $_{6.86e-1}$ -	**9.6713e-1** $_{\mathbf{2.32e-2}}$
	8	4.0137e+0 $_{1.49e-1}$ -	3.9891e+0 $_{1.20e-1}$ -	**2.9787e+0** $_{\mathbf{6.62e-3}}$ +	4.4132e+0 $_{1.44e-1}$ -	9.7433e+0 $_{1.96e-1}$ -	3.0997e+0 $_{1.40e-1}$
	10	5.3759e+0 $_{1.92e-1}$ -	5.4227e+0 $_{1.01e-1}$ -	4.5472e+0 $_{6.12e-3}$ -	5.9738e+0 $_{1.82e-1}$ -	1.2100e+1 $_{8.43e-2}$ -	**4.3169e+0** $_{\mathbf{7.75e-2}}$
WFG5	5	1.6809e+0 $_{1.90e-1}$ -	2.0253e+0 $_{1.25e-1}$ -	9.6229e-1 $_{8.02e-4}$ -	1.4277e+0 $_{1.09e-1}$ -	3.8952e+0 $_{2.44e+0}$ -	**9.5849e-1** $_{\mathbf{5.99e-3}}$
	8	3.8331e+0 $_{2.22e-1}$ -	3.9290e+0 $_{6.99e-1}$ -	**2.9710e+0** $_{\mathbf{5.44e-3}}$ +	4.2964e+0 $_{1.50e-1}$ -	1.3689e+1 $_{1.51e+0}$ -	3.0199e+0 $_{1.02e-1}$
	10	5.2912e+0 $_{1.96e-1}$ -	7.3034e+0 $_{1.24e+0}$ -	4.5459e+0 $_{3.49e-3}$ -	5.9026e+0 $_{1.95e-1}$ -	1.7505e+1 $_{2.81e+0}$ -	**4.3524e+0** $_{\mathbf{1.20e-1}}$
WFG6	5	1.0889e+0 $_{8.87e-2}$ -	1.9432e+0 $_{1.42e-1}$ -	9.6709e-1 $_{2.06e-3}$ -	1.7185e+0 $_{8.75e-2}$ -	5.0474e+0 $_{1.06e+0}$ -	**9.5989e-1** $_{\mathbf{9.58e-3}}$
	8	3.5045e+0 $_{3.26e-1}$ -	5.0055e+0 $_{8.54e-1}$ -	**2.9919e+0** $_{\mathbf{5.78e-3}}$ =	4.9689e+0 $_{1.06e-1}$ -	8.8743e+0 $_{4.08e+0}$ -	3.0483e+0 $_{1.28e-1}$
	10	4.7647e+0 $_{4.92e-1}$ -	6.9286e+0 $_{5.88e-1}$ -	4.5496e+0 $_{4.50e-3}$ -	6.8067e+0 $_{1.16e-1}$ -	9.7812e+0 $_{5.22e+0}$ -	**4.4699e+0** $_{\mathbf{9.82e-2}}$
WFG7	5	1.1365e+0 $_{2.88e-2}$ -	1.9962e+0 $_{1.15e-1}$ -	9.6855e-1 $_{7.64e-4}$ -	1.9258e+0 $_{1.15e-1}$ -	5.3456e+0 $_{4.10e-1}$ -	**9.6457e-1** $_{\mathbf{7.85e-3}}$
	8	3.6920e+0 $_{1.01e-1}$ -	4.0063e+0 $_{1.28e-1}$ -	**2.9954e+0** $_{\mathbf{1.00e-2}}$ +	4.7620e+0 $_{1.89e-1}$ -	1.0447e+1 $_{1.78e+0}$ -	3.0706e+0 $_{1.20e-1}$
	10	4.7198e+0 $_{1.34e-1}$ -	5.4841e+0 $_{7.92e-2}$ -	4.5528e+0 $_{8.16e-3}$ -	6.1972e+0 $_{2.49e-1}$ -	1.3240e+1 $_{2.08e+0}$ -	**4.3432e+0** $_{\mathbf{7.10e-2}}$
WFG8	5	2.6539e+0 $_{1.91e-1}$ -	1.6542e+0 $_{1.09e-1}$ -	9.8002e-1 $_{1.18e-3}$ -	1.5519e+0 $_{9.52e-2}$ -	4.8172e+0 $_{1.29e+0}$ -	**9.6833e-1** $_{\mathbf{6.38e-3}}$
	8	4.1703e+0 $_{2.12e-1}$ -	4.0649e+0 $_{5.89e-1}$ -	**3.0590e+0** $_{\mathbf{4.87e-2}}$ =	4.5392e+0 $_{3.19e-1}$ -	1.1056e+1 $_{1.38e+0}$ -	3.0849e+0 $_{8.01e-2}$
	10	5.7665e+0 $_{2.80e-1}$ -	5.8349e+0 $_{6.90e-1}$ -	5.3022e+0 $_{3.48e+0}$ -	6.4551e+0 $_{3.54e-1}$ -	1.3441e+1 $_{2.21e+0}$ -	**4.5159e+0** $_{\mathbf{4.43e-2}}$
WFG9	5	2.2425e+0 $_{3.23e-1}$ -	2.2957e+0 $_{4.73e-1}$ -	9.4147e-1 $_{4.87e-3}$ =	1.3960e+0 $_{1.10e-1}$ -	4.7418e+0 $_{9.05e-1}$ -	**9.3968e-1** $_{\mathbf{2.02e-2}}$
	8	3.9516e+0 $_{1.18e-1}$ -	5.7907e+0 $_{7.94e-1}$ -	**2.9476e+0** $_{\mathbf{7.38e-3}}$ +	4.1540e+0 $_{1.90e-1}$ -	9.6459e+0 $_{3.24e+0}$ -	3.0618e+0 $_{1.72e-1}$
	10	5.4393e+0 $_{1.50e-1}$ -	7.1190e+0 $_{7.74e-1}$ -	4.4458e+0 $_{1.67e-2}$ -	5.4985e+0 $_{1.50e-1}$ -	1.3582e+1 $_{4.04e+0}$ -	**4.1988e+0** $_{\mathbf{6.82e-2}}$
DTLZ1	5	5.3191e-2 $_{1.79e-4}$ -	5.2688e-2 $_{3.97e-6}$ -	3.2812e-1 $_{4.23e-1}$ -	5.8343e-2 $_{1.51e-3}$ -	2.9086e-1 $_{2.25e-1}$ -	**5.1118e-2** $_{\mathbf{1.90e-4}}$
	8	1.0182e-1 $_{2.99e-3}$ +	**9.7006e-2** $_{\mathbf{3.03e-5}}$ +	7.2760e-1 $_{3.92e-1}$ -	1.1350e-1 $_{1.78e-3}$ +	3.8252e-1 $_{3.29e-1}$ =	2.5337e-1 $_{4.87e-2}$
	10	1.1954e-1 $_{1.98e-3}$ +	**1.0856e-1** $_{\mathbf{3.36e-4}}$ +	4.6631e-1 $_{1.74e-1}$ -	1.1285e-1 $_{1.17e-3}$ +	1.4713e-1 $_{9.99e-2}$ +	2.4473e-1 $_{4.31e-2}$
DTLZ2	5	1.6647e-1 $_{9.82e-4}$ -	1.6505e-1 $_{1.63e-5}$ +	1.6513e-1 $_{2.94e-5}$ -	**1.6191e-1** $_{\mathbf{9.09e-4}}$ +	1.6835e-1 $_{6.66e-4}$ -	1.6513e-1 $_{1.68e-6}$
	8	3.3686e-1 $_{6.65e-3}$ -	3.1504e-1 $_{7.96e-5}$ -	3.1524e-1 $_{8.54e-5}$ -	3.5073e-1 $_{2.56e-3}$ -	3.4330e-1 $_{6.49e-3}$ -	**3.1491e-1** $_{\mathbf{1.67e-5}}$
	10	4.4907e-1 $_{8.50e-3}$ -	4.3336e-1 $_{3.89e-2}$ -	4.2171e-1 $_{3.38e-4}$ -	**3.9669e-1** $_{\mathbf{1.68e-3}}$ +	4.3927e-1 $_{4.68e-3}$ -	4.2088e-1 $_{2.84e-3}$
DTLZ3	5	1.7348e-1 $_{3.93e-3}$ -	1.6494e-1 $_{3.19e-5}$ +	1.6572e-1 $_{1.24e-3}$ -	**1.6429e-1** $_{\mathbf{1.41e-3}}$ +	1.1128e+1 $_{5.62e+0}$ -	1.6516e-1 $_{2.44e-4}$
	8	3.5869e-1 $_{1.41e-2}$ +	**3.2250e-1** $_{\mathbf{2.14e-2}}$ +	1.5967e+0 $_{2.31e+0}$ -	3.5285e-1 $_{2.64e-3}$ +	1.1882e+1 $_{8.11e+0}$ -	6.7626e-1 $_{4.04e-1}$
	10	4.5879e-1 $_{3.34e-3}$ +	4.2871e-1 $_{4.91e-2}$ +	1.2417e+0 $_{3.94e-5}$ -	**3.9886e-1** $_{\mathbf{1.94e-3}}$ +	4.8832e+0 $_{2.68e+0}$ -	7.7896e-1 $_{2.26e-1}$
DTLZ4	5	1.7374e-1 $_{4.73e-3}$ -	**1.6481e-1** $_{\mathbf{4.77e-5}}$ +	3.2842e-1 $_{1.55e-1}$ -	1.6520e-1 $_{1.25e-3}$ =	1.8382e-1 $_{6.08e-2}$ -	1.6513e-1 $_{2.36e-6}$
	8	3.4902e-1 $_{6.63e-3}$ -	**3.2218e-1** $_{\mathbf{9.86e-4}}$ +	5.6680e-1 $_{7.02e-2}$ -	3.5332e-1 $_{1.58e-2}$ -	3.6050e-1 $_{4.99e-2}$ -	3.2292e-1 $_{3.04e-2}$
	10	4.5755e-1 $_{1.45e-2}$ -	4.2646e-1 $_{2.02e-3}$ -	5.5766e-1 $_{6.59e-2}$ -	**4.0062e-1** $_{\mathbf{1.13e-3}}$ +	4.3453e-1 $_{1.39e-2}$ -	4.2268e-1 $_{1.17e-2}$
=/-/=		9/30/0	14/24/1	11/24/4	13/25/1	2/35/2	

Table. 6. The HV values obtained by MaOEAs for MaOPs with 5, 8, and 10 objectives.

Problem	M	MOEADDQN	RLMMDE	IDBEA	onebyoneEA	MaOEAIGD	MRL-MOEA
WFG1	5	9.9681e-1 $_{3.29e-4}$ +	9.8558e-1 $_{5.67e-3}$ +	**9.9865e-1** $_{\mathbf{8.28e-5}}$ +	9.8961e-1 $_{2.29e-3}$ +	3.5519e-1 $_{1.27e-1}$ -	9.2453e-1 $_{3.75e-2}$
	8	**9.9861e-1** $_{\mathbf{1.21e-3}}$ +	9.8478e-1 $_{1.67e-2}$ +	9.9318e-1 $_{9.77e-3}$ +	9.9530e-1 $_{9.74e-4}$ +	3.3413e-1 $_{1.72e-1}$ -	9.2232e-1 $_{5.60e-2}$
	10	**9.9994e-1** $_{\mathbf{3.60e-5}}$ +	9.8178e-1 $_{2.73e-2}$ +	9.9575e-1 $_{5.08e-3}$ +	9.9784e-1 $_{6.37e-4}$ +	5.4187e-1 $_{2.73e-1}$ -	9.7361e-1 $_{2.02e-2}$
WFG2	5	**9.9684e-1** $_{\mathbf{3.38e-4}}$ +	9.8375e-1 $_{3.01e-3}$ +	9.9633e-1 $_{4.30e-4}$ +	9.8150e-1 $_{2.73e-3}$ +	9.3297e-1 $_{3.71e-2}$ =	9.4305e-1 $_{4.58e-3}$
	8	9.8789e-1 $_{9.11e-3}$ +	9.9412e-1 $_{1.14e-3}$ +	9.9548e-1 $_{3.20e-3}$ +	9.8844e-1 $_{5.17e-3}$ +	9.4966e-1 $_{6.61e-2}$ +	8.8019e-1 $_{5.96e-2}$
	10	**9.9983e-1** $_{\mathbf{1.60e-4}}$ +	9.9539e-1 $_{1.76e-3}$ +	9.4691e-1 $_{1.91e-1}$ +	9.9412e-1 $_{2.03e-3}$ +	9.8581e-1 $_{3.26e-2}$ +	9.3215e-1 $_{6.39e-2}$
WFG3	5	9.0756e-2 $_{4.85e-4}$ =	**2.0572e-1** $_{\mathbf{2.77e-2}}$ +	2.0457e-1 $_{1.85e-2}$ +	7.8824e-2 $_{8.96e-3}$ =	8.5340e-2 $_{2.30e-3}$ =	9.5193e-2 $_{4.07e-2}$
	8	4.1376e-2 $_{2.82e-2}$ =	4.0671e-2 $_{3.53e-2}$ =	**8.4231e-2** $_{\mathbf{2.76e-3}}$ +	1.7696e-3 $_{9.69e-3}$ -	1.7299e-2 $_{2.16e-2}$ -	3.7109e-2 $_{3.16e-2}$
	10	2.1934e-2 $_{2.40e-2}$ +	7.4957e-5 $_{4.11e-4}$ =	**7.5162e-2** $_{\mathbf{8.50e-3}}$ +	0.0000e+0 $_{0.00e+0}$ =	1.1903e-2 $_{1.49e-2}$ +	1.3448e-3 $_{4.35e-3}$
WFG4	5	7.7197e-1 $_{1.75e-2}$ -	6.3347e-1 $_{2.20e-2}$ -	**8.0453e-1** $_{\mathbf{1.21e-3}}$ +	6.8247e-1 $_{1.35e-2}$ -	1.0708e-1 $_{3.88e-2}$ -	8.0187e-1 $_{4.36e-3}$
	8	7.7185e-1 $_{2.73e-2}$ -	8.3696e-1 $_{1.14e-2}$ -	**9.1554e-1** $_{\mathbf{1.23e-3}}$ +	7.4713e-1 $_{1.62e-2}$ -	1.0466e-1 $_{3.13e-2}$ -	8.6572e-1 $_{4.76e-2}$
	10	8.7816e-1 $_{2.62e-2}$ -	9.0278e-1 $_{4.97e-3}$ -	**9.6466e-1** $_{\mathbf{2.32e-3}}$ +	8.1123e-1 $_{1.41e-2}$ -	9.9155e-2 $_{2.52e-2}$ -	9.3792e-1 $_{1.93e-2}$
WFG5	5	6.3846e-1 $_{3.31e-2}$ -	5.3293e-1 $_{2.03e-2}$ -	**7.6085e-1** $_{\mathbf{6.13e-4}}$ =	6.4628e-1 $_{1.38e-2}$ -	3.6118e-1 $_{2.41e-1}$ -	7.6068e-1 $_{9.94e-4}$
	8	6.9956e-1 $_{3.97e-2}$ -	7.6300e-1 $_{6.81e-2}$ -	**8.6164e-1** $_{\mathbf{6.16e-4}}$ +	7.0072e-1 $_{1.39e-2}$ -	9.1574e-2 $_{2.31e-2}$ -	8.3992e-1 $_{2.94e-2}$
	10	7.7278e-1 $_{2.26e-2}$ -	6.8077e-1 $_{9.01e-2}$ -	**9.0323e-1** $_{\mathbf{3.05e-4}}$ +	7.6304e-1 $_{1.25e-2}$ -	1.1172e-1 $_{1.17e-1}$ -	8.9984e-1 $_{2.69e-3}$
WFG6	5	7.0106e-1 $_{6.60e-2}$ -	5.2666e-1 $_{6.84e-2}$ -	7.3893e-1 $_{1.30e-2}$ -	5.9328e-1 $_{2.22e-2}$ -	1.8153e-1 $_{8.04e-2}$ -	**7.5623e-1** $_{\mathbf{1.61e-2}}$
	8	6.8701e-1 $_{9.97e-2}$ -	5.7072e-1 $_{1.47e-1}$ -	**8.3169e-1** $_{\mathbf{1.81e-2}}$ =	6.0490e-1 $_{2.64e-2}$ -	2.6383e-1 $_{1.52e-1}$ -	8.2987e-1 $_{2.79e-2}$
	10	7.2914e-1 $_{1.30e-1}$ -	5.8442e-1 $_{7.21e-2}$ -	8.7876e-1 $_{1.40e-2}$ -	6.6934e-1 $_{2.10e-2}$ -	3.7743e-1 $_{2.10e-1}$ -	**8.9534e-1** $_{\mathbf{1.89e-2}}$
WFG7	5	7.9098e-1 $_{1.57e-3}$ -	6.2799e-1 $_{1.71e-2}$ -	8.0724e-1 $_{6.15e-4}$ -	6.3287e-1 $_{1.91e-2}$ -	2.4792e-1 $_{3.25e-2}$ -	**8.1122e-1** $_{\mathbf{5.59e-4}}$
	8	8.4635e-1 $_{1.73e-2}$ -	8.3366e-1 $_{1.16e-2}$ -	**9.1737e-1** $_{\mathbf{9.20e-4}}$ +	6.9283e-1 $_{1.44e-2}$ -	2.1112e-1 $_{6.77e-2}$ -	8.9857e-1 $_{3.14e-2}$
	10	9.4477e-1 $_{4.28e-3}$ -	8.9653e-1 $_{4.42e-3}$ -	**9.6519e-1** $_{\mathbf{4.33e-4}}$ +	7.7886e-1 $_{1.12e-2}$ -	2.2626e-1 $_{6.76e-2}$ -	9.5947e-1 $_{1.10e-2}$
WFG8	5	3.6220e-1 $_{6.63e-2}$ -	5.5606e-1 $_{2.01e-2}$ -	6.9921e-1 $_{1.19e-3}$ -	5.4034e-1 $_{1.79e-2}$ -	7.3174e-2 $_{6.99e-2}$ -	**7.1610e-1** $_{\mathbf{1.98e-2}}$
	8	6.1092e-1 $_{4.86e-2}$ -	7.3385e-1 $_{5.75e-2}$ -	8.0599e-1 $_{2.01e-2}$ -	5.0429e-1 $_{7.60e-2}$ -	1.6753e-1 $_{5.17e-2}$ -	**8.4399e-1** $_{\mathbf{3.13e-2}}$
	10	7.0317e-1 $_{4.28e-2}$ -	8.2178e-1 $_{5.11e-2}$ -	8.1735e-1 $_{1.98e-1}$ -	5.9672e-1 $_{5.90e-2}$ -	2.2516e-1 $_{7.83e-2}$ -	**9.3718e-1** $_{\mathbf{1.66e-2}}$
WFG9	5	4.9023e-1 $_{6.44e-2}$ -	4.6345e-1 $_{8.42e-2}$ -	7.6717e-1 $_{3.34e-3}$ -	6.5560e-1 $_{1.42e-2}$ -	2.2554e-1 $_{8.89e-2}$ -	**7.6997e-1** $_{\mathbf{4.30e-3}}$
	8	5.9028e-1 $_{5.50e-2}$ -	4.4223e-1 $_{4.25e-2}$ -	**8.5393e-1** $_{\mathbf{8.59e-3}}$ +	6.9691e-1 $_{3.36e-2}$ -	2.5109e-1 $_{1.68e-1}$ -	8.0168e-1 $_{4.98e-2}$
	10	6.1615e-1 $_{5.28e-2}$ -	5.3894e-1 $_{4.32e-2}$ -	**9.0236e-1** $_{\mathbf{4.81e-3}}$ +	7.7089e-1 $_{9.31e-3}$ -	2.4460e-1 $_{1.67e-1}$ -	8.7213e-1 $_{4.12e-2}$
DTLZ1	5	9.7914e-1 $_{2.36e-4}$ +	**9.7988e-1** $_{\mathbf{1.48e-4}}$ +	5.8546e-1 $_{4.65e-1}$ =	9.4222e-1 $_{4.47e-3}$ -	5.2799e-1 $_{3.13e-1}$ -	9.5216e-1 $_{2.63e-3}$
	8	9.9701e-1 $_{4.17e-4}$ +	**9.9762e-1** $_{\mathbf{4.53e-5}}$ +	1.5339e-1 $_{2.31e-1}$ -	9.7400e-1 $_{4.75e-3}$ +	4.6966e-1 $_{4.23e-1}$ =	6.5442e-1 $_{6.36e-2}$
	10	9.9959e-1 $_{4.00e-5}$ +	**9.9969e-1** $_{\mathbf{1.75e-5}}$ +	3.5885e-1 $_{2.98e-1}$ -	9.9162e-1 $_{1.61e-3}$ +	8.8204e-1 $_{2.51e-1}$ +	6.7067e-1 $_{6.64e-2}$
DTLZ2	5	8.0840e-1 $_{1.36e-3}$ -	8.1279e-1 $_{4.18e-4}$ =	8.1264e-1 $_{4.65e-4}$ =	8.0157e-1 $_{2.24e-3}$ -	**8.1287e-1** $_{\mathbf{3.66e-4}}$ =	8.1271e-1 $_{3.47e-4}$
	8	9.2256e-1 $_{1.16e-3}$ -	**9.2417e-1** $_{\mathbf{2.39e-4}}$ =	9.2365e-1 $_{2.40e-4}$ -	9.0721e-1 $_{3.67e-3}$ -	9.2005e-1 $_{2.52e-3}$ -	9.2408e-1 $_{2.24e-4}$
	10	9.6909e-1 $_{8.76e-4}$ -	9.6425e-1 $_{1.69e-2}$ =	9.6955e-1 $_{1.64e-4}$ -	9.5461e-1 $_{2.70e-3}$ -	9.6847e-1 $_{1.06e-3}$ -	**9.6968e-1** $_{\mathbf{5.15e-4}}$
DTLZ3	5	8.0204e-1 $_{3.74e-3}$ -	**8.1311e-1** $_{\mathbf{4.48e-4}}$ +	8.0873e-1 $_{2.32e-3}$ -	7.9982e-1 $_{2.32e-3}$ -	0.0000e+0 $_{0.00e+0}$ -	8.1242e-1 $_{7.14e-4}$
	8	9.1542e-1 $_{7.11e-3}$ +	**9.2178e-1** $_{\mathbf{9.79e-3}}$ +	1.2809e-1 $_{1.47e-1}$ -	9.0183e-1 $_{4.90e-3}$ +	0.0000e+0 $_{0.00e+0}$ -	6.7647e-1 $_{2.56e-1}$
	10	**9.6574e-1** $_{\mathbf{2.24e-3}}$ +	9.5925e-1 $_{5.40e-2}$ +	9.0854e-2 $_{5.77e-5}$ -	9.5527e-1 $_{2.30e-3}$ +	4.5553e-3 $_{1.75e-2}$ -	6.7294e-1 $_{2.42e-1}$
DTLZ4	5	8.0979e-1 $_{1.50e-3}$ -	**8.1326e-1** $_{\mathbf{4.32e-4}}$ +	7.1844e-1 $_{9.31e-2}$ -	8.0843e-1 $_{1.50e-3}$ -	8.0628e-1 $_{2.52e-2}$ =	8.1268e-1 $_{4.08e-4}$
	8	**9.3021e-1** $_{\mathbf{1.20e-3}}$ +	9.2526e-1 $_{3.59e-4}$ +	8.2033e-1 $_{4.18e-2}$ -	9.1747e-1 $_{5.29e-3}$ -	9.1775e-1 $_{1.16e-2}$ -	9.2230e-1 $_{6.66e-3}$
	10	**9.7238e-1** $_{\mathbf{3.48e-3}}$ +	9.7069e-1 $_{2.05e-4}$ +	9.2956e-1 $_{2.78e-2}$ -	9.6484e-1 $_{1.03e-3}$ -	9.6966e-1 $_{2.88e-3}$ +	9.6940e-1 $_{2.31e-3}$
+/-/=		14/23/2	16/18/5	18/17/4	10/27/2	5/29/5	

Table. 7. The Friedman ranking of the six algorithms. The left column represents the average ranking based on the IGD values, while the right column represents the average ranking based on the HV values.

	IGD	HV
Algorithm	Ranking	Ranking
MOEA/D-DQN	3.3333	2.8974
RLMMDE	3.3205	3.2179
IDBEA	2.8462	**2.5513**
onebyoneEA	3.4872	4.1667
MaOEA-IGD	5.6667	5.3974
MRL-MOEA	**2.3462**	2.7692

After performing a Friedman ranking on the IGD and HV data obtained by the six algorithms recorded in Tables 5 and 6, the results are presented in Table 7. This table more intuitively demonstrates the competitiveness of each algorithm. According to the data presented in Table 7, MRL-MOEA has achieved an impressive average rank of 2.4872 based on IGD values and 2.9231 on HV values. These rankings are on par with those of the well-regarded IDBEA algorithm.

A further illustration of these findings is provided in Fig. 9, which plots the Friedman ranking of MRL-MOEA in a bi-dimensional space. It can be observed from the figure that, in terms of HV values, MRL-MOEA performs slightly worse than IDBEA, yet it significantly surpasses other competing algorithms. In terms of IGD indicators, MRL-MOEA significantly outperforms all comparative algorithms. Regarding this result, as previously mentioned, compared to the HV indicator, the IGD indicator is calculated by sampling the true PF. Therefore, the ranking of IGD values more accurately reflects the convergence and diversity of the algorithms.

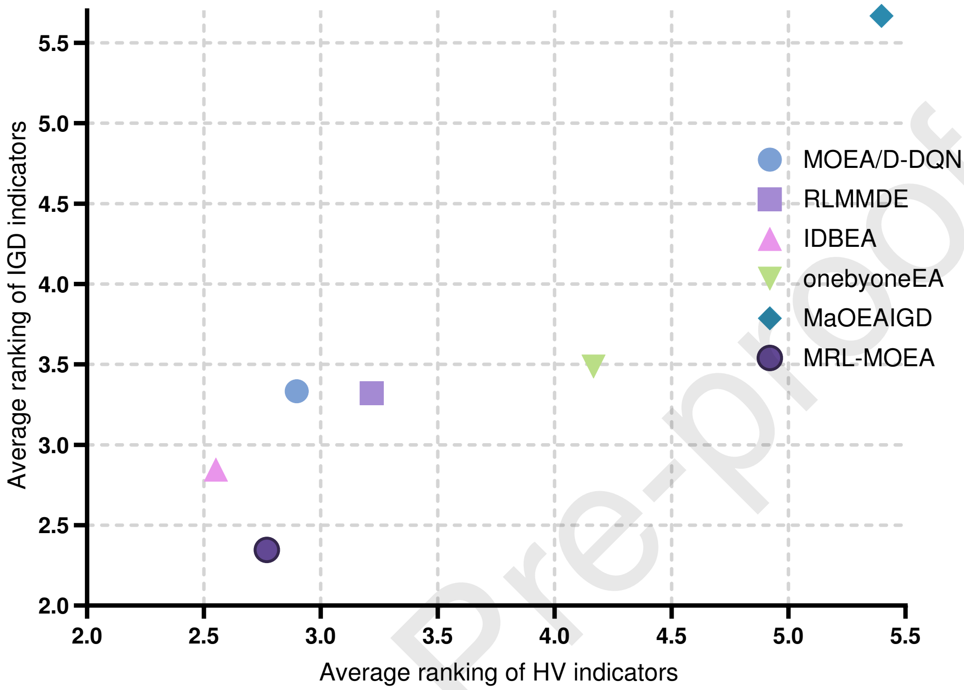
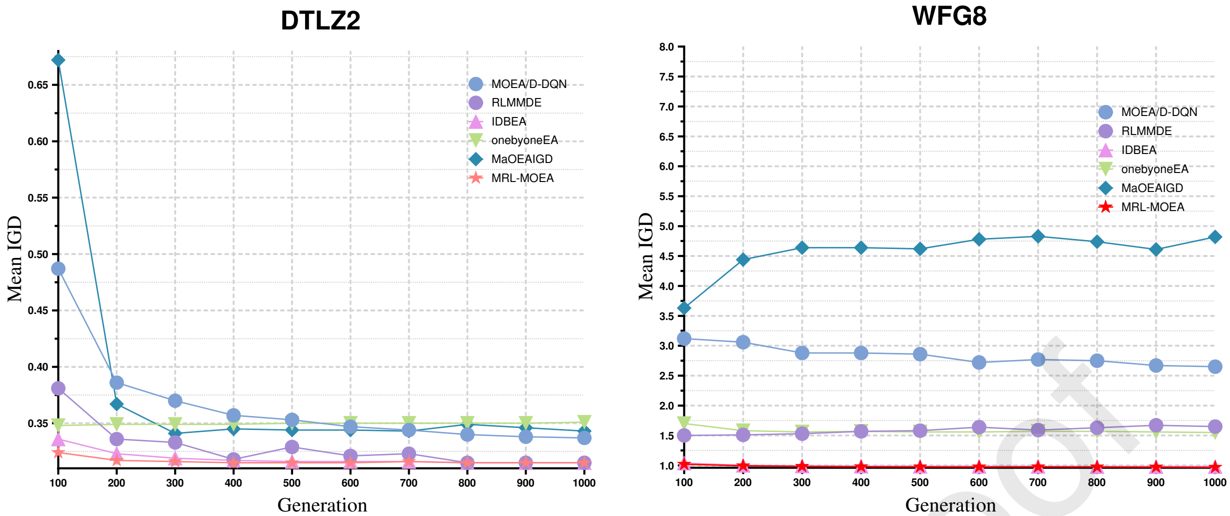


Fig. 9. Average ranking plot of the five algorithms on IGD and HV. The horizontal and vertical axes show the average ranking of HV and IGD, respectively.

In order to assess the convergence of the algorithms on MaOPs, the average IGD trajectories of these algorithms are displayed in Fig. 10. The IGD values in Fig. 10 are derived by sampling every 100 generations and averaging over 30 runs. It can be observed from Fig. 10(a) that the MRL-MOEA has significantly higher convergence speed and stability than the other five MOEAs on the eight-objective DTLZ2 test instance. Moreover, MRL-MOEA also shows a great advantage in convergence over the other algorithms on WGF8 test instance with five objectives, as shown in Fig. 10(b). In contrast, the other five algorithms have lower convergence performance, some of them being trapped in local optima or diverging in the opposite direction. This indicates that the strategy of adaptively selecting operators based on Q-learning can effectively drive individuals to evolve in a good direction.

Fig. 11 presents the PF obtained by MRL-MOEA and five other MaOEAs on the DTLZ4 test instance with 8 objectives. As shown in Fig. 11, the dimensions of the target space are represented on the horizontal axis, while the objective function value for each dimension is represented on the vertical axis. A more regular and lower line indicates better algorithm performance in these test instances. Notably, the line associated with MRL-MOEA demonstrates the highest regularity and the lowest values among the various lines. Overall, MRL-MOEA remains competitive among other algorithms for solving MaOPs.



(a) MaOP-IGD-DTLZ (b) MaOP-IGD-WFG

Fig. 10. Trajectories of the six algorithms on DTLZ2 and WFG8 test instances. (a) Average IGD trajectories of six algorithms on DTLZ2 test problem with eight objectives. (b) IGD trajectories averaged across six algorithms on WFG8 test problem with five objectives.

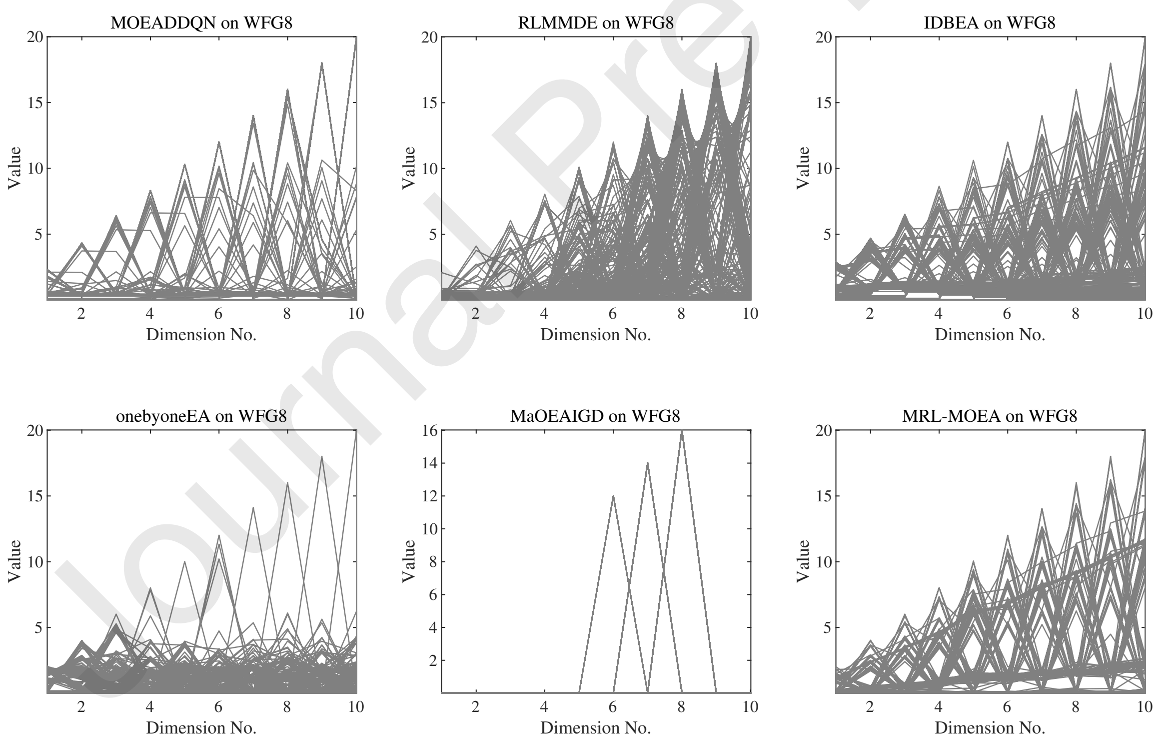


Fig. 11. The approximate PF obtained by MRL-MOEA and the comparison algorithms on the WFG8 test instance with ten objectives. The horizontal axis and the vertical axis represent the dimensions and the objective function values obtained by the algorithms, respectively.

5.2.3. *Effectiveness of different components of MRL-MOEA*

After a detailed analysis of the performance of the algorithm in the previous sections, this section will clarify the impact of the various strategies proposed in this paper and the combination of these strategies on the overall performance of the algorithm. To this end, three variant algorithms were designed based on the MRL-MOEA algorithm framework. The performance of different variant algorithms was tested using the DTLZ1-DTLZ4 test problems with 3, 5, 8, and 10 objectives. The specific strategies and purposes of these variants are as follows.

- Variant-1: Only GA operators are used to generate individuals, without the use of adaptive weight vector adjustment strategy and adaptive crossover operator selection strategy.
- Variant-2: GA operators are used to generate offspring, and the adaptive weight vector adjustment strategy is allowed.
- Variant-3: This variant solely employs the adaptive crossover operator selection strategy, without the use of the adaptive weight vector adjustment strategy.

Tables 8 and9 present the performance results of MRL-MOEA and its three variants on the IGD and HV indicators. Overall, MRL-MOEA outperforms the variants, indicating that combining its strategies is necessary and effective. A comparison between Variant-1 and Variant-2 reveals minimal differences in performance, which might be attributed to the DTLZ test problems not encompassing discontinuous or complex PF, thus diminishing the apparent impact of the weight adjustment strategy. However, the results of Variant-3 suggest a slight improvement over the first two variants and a performance close to the original algorithm, hinting at the potential of the adaptive operator selection strategy in enhancing algorithmic performance. The small performance gap between the variants and MRL-MOEA on the DTLZ test suite could be ascribed to the algorithm's inherent robustness or the test set's limitations.

It is worth mentioning that the matching strategy, as the first strategy proposed in this paper, serves as a prerequisite for the other strategies. This strategy provides the foundational information necessary for the execution of other strategies; therefore, it cannot be implemented in isolation. All the variant algorithms adopted the matching strategy. However, by comparing the IGD and HV values of Variant-1 with those of other algorithms like IDBEA, as shown in previous tables, it is evident that the variant algorithm employing only the matching strategy, although inferior to the MRL-MOEA, still outperforms these comparative algorithms. This leads to the indirect conclusion that Variant-1 surpasses MOEA/D, thereby verifying the effectiveness of the matching strategy proposed in this paper.

Table. 8. The IGD values obtained by MRL-MOEA and other variants.

Problem	M	Variant-1	Variant-2	Variant-3	MRL-MOEA
DTLZ1	3	**2.0283e-2 $_{7.56e-5}$ +**	2.0319e-2 $_{9.84e-5}$ +	2.0389e-2 $_{8.52e-5}$ =	**2.0411e-2 $_{7.86e-5}$**
	5	5.1386e-2 $_{2.21e-4}$ -	**5.1403e-2 $_{1.84e-4}$ -**	5.1044e-2 $_{2.16e-4}$ =	**5.1036e-2 $_{1.37e-4}$**
	8	2.3308e-1 $_{7.53e-2}$ =	**2.2296e-1 $_{8.14e-2}$ =**	2.5065e-1 $_{4.41e-2}$ =	2.4692e-1 $_{6.60e-2}$
	10	2.6579e-1 $_{6.94e-2}$ -	2.3592e-1 $_{6.49e-2}$ =	2.6900e-1 $_{4.59e-2}$ -	**2.3355e-1 $_{4.46e-2}$**
DTLZ2	3	**5.4464e-2 $_{2.11e-7}$ =**	5.4464e-2 $_{2.55e-7}$ =	5.4464e-2 $_{5.20e-7}$ =	**5.4464e-2 $_{5.39e-7}$**
	5	1.6513e-1 $_{6.42e-7}$ -	1.6513e-1 $_{7.50e-7}$ -	1.6513e-1 $_{2.04e-6}$ =	**1.6513e-1 $_{2.71e-6}$**
	8	3.1488e-1 $_{6.33e-6}$ +	**3.1488e-1 $_{5.95e-6}$ +**	3.1493e-1 $_{2.00e-5}$ =	3.1634e-1 $_{7.69e-3}$
	10	**4.2160e-1 $_{1.52e-4}$ +**	4.2168e-1 $_{1.17e-4}$ +	4.2115e-1 $_{1.84e-4}$ =	4.2266e-1 $_{8.71e-3}$
DTLZ3	3	5.4749e-2 $_{3.75e-4}$ -	5.4738e-2 $_{2.87e-4}$ -	5.4475e-2 $_{7.86e-6}$ =	**5.4474e-2 $_{1.38e-5}$**
	5	1.8843e-1 $_{1.27e-1}$ -	1.9387e-1 $_{1.57e-1}$ -	1.6510e-1 $_{1.39e-5}$ =	**1.6509e-1 $_{2.73e-5}$**
	8	6.4931e-1 $_{2.82e-1}$ =	6.6009e-1 $_{2.87e-1}$ =	6.5581e-1 $_{2.71e-1}$ =	**5.8844e-1 $_{2.87e-1}$**
	10	**6.2490e-1 $_{2.37e-1}$ +**	7.6979e-1 $_{2.47e-1}$ =	7.4256e-1 $_{2.25e-1}$ =	7.8963e-1 $_{1.90e-1}$
DTLZ4	3	**5.4464e-2 $_{3.03e-7}$ =**	8.4179e-2 $_{1.63e-1}$ =	5.4464e-2 $_{1.94e-6}$ =	**5.4464e-2 $_{5.81e-7}$**
	5	**1.6513e-1 $_{3.31e-5}$ +**	1.7320e-1 $_{4.42e-2}$ -	1.7321e-1 $_{4.43e-2}$ -	1.6514e-1 $_{7.37e-5}$
	8	3.4632e-1 $_{5.31e-2}$ -	**3.3410e-1 $_{4.37e-2}$ +**	3.4234e-1 $_{5.06e-2}$ =	**3.3593e-1 $_{4.36e-2}$**
	10	4.2688e-1 $_{1.77e-2}$ -	4.2482e-1 $_{1.45e-2}$ -	**4.2395e-1 $_{1.42e-2}$ =**	**4.2422e-1 $_{1.51e-2}$**
+/-/=		5/7/4	4/6/6	0/2/14	

To more clearly illustrate the impact of strategies on the overall performance of the algorithm and the effectiveness of each strategy, in conjunction with the descriptions of these variant algorithms mentioned above, a Friedman combined ranking is conducted on the data in Tables 8 and 9. It can be observed from Table 10 that these algorithms have consistent rankings in both the IGD and HV indicators, with the MRL-MOEA algorithm performing excellently, followed by Variant-3 in second place, then Variant-1, and Variant-2.

Table. 9. The HV values obtained by MRL-MOEA and other variants.

Problem	M	Variant-1	Variant-2	Variant-3	MRL-MOEA
DTLZ1	3	8.3755e-1 $_{5.22e-4}$ -	8.3759e-1 $_{4.83e-4}$ -	8.3806e-1 $_{4.35e-4}$ =	**8.3818e-1 $_{3.22e-4}$**
	5	9.5341e-1 $_{1.39e-2}$ +	**9.5608e-1 $_{1.66e-3}$** +	9.5211e-1 $_{2.70e-3}$ =	9.5185e-1 $_{2.43e-3}$
	8	6.5086e-1 $_{1.20e-1}$ =	**6.5882e-1 $_{1.14e-1}$** =	6.5548e-1 $_{5.34e-2}$ =	6.4865e-1 $_{1.16e-1}$
	10	6.3131e-1 $_{1.32e-1}$ -	6.7688e-1 $_{1.14e-1}$ =	6.1177e-1 $_{1.01e-1}$ -	**6.7875e-1 $_{6.71e-2}$**
DTLZ2	3	**5.5962e-1 $_{8.33e-7}$** +	5.5962e-1 $_{1.09e-6}$ +	5.5961e-1 $_{2.74e-6}$ =	5.5961e-1 $_{3.88e-6}$
	5	8.1266e-1 $_{4.14e-4}$ =	8.1265e-1 $_{3.78e-4}$ =	8.1262e-1 $_{5.58e-4}$ =	8.1256e-1 $_{4.18e-4}$
	8	9.2413e-1 $_{2.69e-4}$ =	**9.2418e-1 $_{2.51e-4}$** +	9.2406e-1 $_{2.75e-4}$ =	9.2342e-1 $_{3.35e-3}$
	10	**9.6984e-1 $_{1.84e-4}$** =	9.6981e-1 $_{2.17e-4}$ =	9.6975e-1 $_{1.61e-4}$ =	9.6928e-1 $_{2.73e-3}$
DTLZ3	3	5.5569e-1 $_{2.82e-3}$ -	5.5587e-1 $_{2.54e-3}$ -	5.5937e-1 $_{1.33e-4}$ =	**5.5942e-1 $_{1.12e-4}$**
	5	7.9014e-1 $_{1.11e-1}$ -	7.8723e-1 $_{1.31e-1}$ -	8.1257e-1 $_{3.34e-4}$ =	**8.1254e-1 $_{4.07e-4}$**
	8	6.5943e-1 $_{2.68e-1}$ =	6.4701e-1 $_{2.77e-1}$ =	6.7087e-1 $_{2.53e-1}$ =	**7.1875e-1 $_{2.49e-1}$**
	10	**8.0261e-1 $_{2.45e-1}$** +	6.5716e-1 $_{2.58e-1}$ =	7.1174e-1 $_{2.42e-1}$ =	6.6454e-1 $_{2.23e-1}$
DTLZ4	3	**5.5962e-1 $_{1.05e-6}$** +	5.4399e-1 $_{8.56e-2}$ -	5.5961e-1 $_{7.14e-6}$ =	5.5961e-1 $_{3.40e-6}$
	5	**8.1266e-1 $_{4.17e-4}$** =	8.0927e-1 $_{1.86e-2}$ =	8.0902e-1 $_{1.96e-2}$ =	8.1255e-1 $_{5.68e-4}$
	8	9.1591e-1 $_{1.39e-2}$ =	9.1906e-1 $_{1.16e-2}$ =	9.1724e-1 $_{1.25e-2}$ =	**9.1912e-1 $_{1.13e-2}$**
	10	9.6866e-1 $_{3.36e-3}$ =	9.6900e-1 $_{3.11e-3}$ =	**9.6913e-1 $_{2.48e-3}$** =	9.6901e-1 $_{3.14e-3}$
+/-/=		4/4/8	3/4/9	0/1/15	

The comparison of the rankings between the MRL-MOEA and Variant-1 reveals that the two strategies of adaptive crossover operator selection and adaptive weight vector adjustment significantly enhance algorithm performance. The rankings of Variant-1 and Variant-2 indicate that, on the basis of uniformly using the adaptive weight vector adjustment strategy, the adaptive crossover operator selection strategy more effectively balances the diversity and convergence of the population. If there is a need to readjust the weight vectors, using a single crossover operator to generate offspring cannot guarantee population convergence.

Moreover, when strategies related to enhancing population convergence are ineffective, merely controlling the diversity of the population may often be counterproductive. At the same time, the comparison between Variant-1 and Variant-3 further proves the effectiveness of this strategy. Similarly, the MRL-MOEA and Variant-3 comparison concludes that the adaptive weight vector adjustment strategy can further improve algorithm performance. Overall, the strategies proposed in this paper are effective for the MRL-MOEA, and designing multiple interdependent strategies contributes more than simply stacking multiple strategies together.

Table. 10. The Friedman ranking of the MRL-MOEA and other variants. The left column represents the ranking based on the IGD values, while the right column represents the ranking based on the HV values.

	IGD	HV
Algorithm	Ranking	Ranking
Variant-1	2.5312	2.5
Variant-2	2.7188	2.6562
Variant-3	2.4375	2.4375
MRL-MOEA	**2.3125**	**2.4062**

6. Conclusion and future work

This paper proposes a novel multi-state reinforcement learning-based multi-objective evolutionary algorithm called MRL-MOEA to solve MOPs and MaOPs. Firstly, it introduces a novel matching relationship between weight vectors and individuals. This strategy prevents the elimination of offspring due to their distance from the weight vectors, thereby avoiding the waste of computational resources. Moreover, in the adaptive crossover operator selection strategy, a state model integrates population information into the Q-learning framework. Six actions are designed by combining the DE and GA operators, each with different search characteristics. To prevent over-reliance on the Q-table that could lead to the population falling into local optima, the concept of reverse learning updates the data in the Q-table. Finally, to address the shortcomings of the matching strategy, a widely used adaptive weight vector adjustment strategy is introduced.

In the experiment, tests on benchmark suites with varying numbers of objectives were conducted to assess the effectiveness of MRL-MOEA. The results demonstrated that MRL-MOEA outperforms other state-of-the-art algorithms in most test instances. The IGD trajectories of these algorithms on MOPs and MaOPs prove that MRL-MOEA's strategy effectively ensures that the transfer group continues evolving toward the true PF. Additionally, three variant algorithms were designed, each employing different strategies proposed in this paper. These variants were used to validate the effectiveness of each individual strategy as well as to assess the overall impact of combining different strategies on the algorithm's performance. Overall, the experimental results largely align with the objectives anticipated during the design of the various strategies within the algorithm.

However, although the matching strategy can assign appropriate individuals to each weight vector, some subproblems may not receive good individuals due to differences in their difficulty levels. We have alleviated this problem by adopting an improved adaptive weight vector strategy, but it still increases the algorithm's complexity. This work demonstrates the promising potential of incorporating RL into MOEAs. In future research, the main direction will be to further combine RL and evolutionary algorithms to propose a new algorithm framework for optimization problems with unknown PF and high dimensions. Additionally, some parameter values in the algorithm are fixed, which, according to the no-free-lunch theorem, will not be conducive to the robustness of the algorithm. Therefore, developing a completely parameter-free algorithm framework is also an important direction for future research.

Acknowledgements

The authors would like to thank anonymous reviewers for their detailed and constructive comments, which helped us improve the quality of this work. This work was supported by the National Natural Science Foundation of China (Nos.62266024, 61866014, and 61962024) and the 18th Student Research Project of Jiangxi University of Finance and Economics (No.20231016154549407).

References

[1] A. Slowik, H. Kwasnicka, Evolutionary algorithms and their applications to engineering problems, Neural Computing and Applications 32 (2020) 12363–12379.

[2] Y. Liu, Y. Du, S. Dou, L. Peng, X. Su, Research summary of intelligent optimization algorithm for warehouse agv path planning, in: LISS 2021: Proceedings of the 11th International Conference on Logistics, Informatics and Service Sciences, Springer, 2022, pp. 96–110.

[3] Z. Pan, D. Lei, L. Wang, A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling, IEEE Transactions on Cybernetics 52 (2020) 5051–5063.
URL https://api.semanticscholar.org/CorpusID:226204143

[4] H. Peng, C. Wang, Y. Han, W. Xiao, X. Zhou, Z. Wu, Micro multi-strategy multi-objective artificial bee colony algorithm for microgrid energy optimization, Future Generation Computer Systems 131 (2022) 59–74.

[5] F. C. Fernandez, W. Caarls, Parameters tuning and optimization for reinforcement learning algorithms using evolutionary computing, in: 2018 International Conference on Information Systems and Computer Science (INCISCOS), IEEE, 2018, pp. 301–305.

[6] S. Yang, Y. Tian, C. He, X. Zhang, K. C. Tan, Y. Jin, A gradient-guided evolutionary approach to training deep neural networks, IEEE Transactions on Neural Networks and Learning Systems 33 (9) (2021) 4861–4875.

[7] P. Kerschke, H. H. Hoos, F. Neumann, H. Trautmann, Automated algorithm selection: Survey and perspectives, Evolutionary Computation 27 (1) (2019) 3–45. doi:10.1162/evco_a_00242.

[8] Y. Tian, S. Peng, T. Rodemann, X. Zhang, Y. Jin, Automated selection of evolutionary multi-objective optimization algorithms, in: 2019 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2019, pp. 3225–3232.

[9] K. McClymont, E. C. Keedwell, Markov chain hyper-heuristic (mchh) an online selective hyper-heuristic for multi-objective continuous problems, in: Proceedings of the 13th annual conference on Genetic and evolutionary computation, 2011, pp. 2003–2010.

[10] K. Li, Ã. Fialho, S. Kwong, Q. Zhang, Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 18 (1) (2014) 114–130. doi:10.1109/TEVC.2013.2239648.

[11] K. Van Moffaert, M. M. Drugan, A. Nowé, Hypervolume-based multi-objective reinforcement learning, in: Evolutionary Multi-Criterion Optimization: 7th International Conference, EMO 2013, Sheffield, UK, March 19-22, 2013. Proceedings 7, Springer, 2013, pp. 352–366.

[12] G. Karafotias, A. E. Eiben, M. Hoogendoorn, Generic parameter control with reinforcement learning, in: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 1319–1326. doi:10.1145/2576768.2598360.
URL https://doi.org/10.1145/2576768.2598360

[13] F. Zou, G. G. Yen, L. Tang, C. Wang, A reinforcement learning approach for dynamic multi-objective optimization, Information Sciences 546 (2021) 815–834. doi:https://doi.org/10.1016/j.ins.2020.08.101.
URL https://www.sciencedirect.com/science/article/pii/S0020025520308677

[14] J. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm, in: 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM '02. Proceedings., Vol. 1, 2002, pp. 606–611 vol.1. doi:10.1109/TENCON.2002.1181348.

[15] A. Qin, P. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: 2005 IEEE Congress on Evolutionary Computation, Vol. 2, 2005, pp. 1785–1791 Vol. 2. doi:10.1109/CEC.2005.1554904.

[16] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Transactions on Evolutionary Computation 10 (6) (2006) 646–657. doi:10.1109/TEVC.2006.872133.

[17] X. Qiu, J.-X. Xu, K. C. Tan, H. A. Abbass, Adaptive cross-generation differential evolution operators for multiobjective optimization, IEEE Transactions on Evolutionary Computation 20 (2) (2016) 232–244. doi:10.1109/TEVC.2015.2433672.

[18] Y. Xie, J. Qiao, D. Wang, B. Yin, A novel decomposition-based multiobjective evolutionary algorithm using improved multiple adaptive dynamic selection strategies, Information Sciences 556 (2021) 472–494. doi:https://doi.org/10.1016/j.ins.2020.08.070. URL https://www.sciencedirect.com/science/article/pii/S0020025520308380

[19] Z. Yan, Y. Tan, B. Wang, L. Liu, H. Zhang, A dual-operator strategy for a multiobjective evolutionary algorithm based on decomposition, Knowledge-Based Systems 240 (2022) 108141. doi:https://doi.org/10.1016/j.knosys.2022.108141. URL https://www.sciencedirect.com/science/article/pii/S0950705122000156

[20] Z. Yan, Y. Tan, H. Chen, L. Meng, H. Zhang, An operator pre-selection strategy for multiobjective evolutionary algorithm based on decomposition, Information Sciences 610 (2022) 887–915. doi:https://doi.org/10.1016/j.ins.2022.08.039. URL https://www.sciencedirect.com/science/article/pii/S0020025522009288

[21] J. qing Li, M. xian Song, L. Wang, P. yong Duan, Y. yan Han, H. Sang, Q. ke Pan, Hybrid artificial bee colony algorithm for a parallel batching distributed flow-shop problem with deteriorating jobs, IEEE Transactions on Cybernetics 50 (2020) 2425–2439. URL https://api.semanticscholar.org/CorpusID:204331114

[22] H. L. Liao, Q. H. Wu, Multi-objective optimisation by reinforcement learning, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8. doi:10.1109/CEC.2010.5585972.

[23] H. L. Liao, Q. H. Wu, L. Jiang, Multi-objective optimization by reinforcement learning for power system dispatch and voltage stability, in: 2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010, pp. 1–8. doi:10.1109/ISGTEUROPE.2010.5638914.

[24] J. Deng, Q. Zhang, Approximating hypervolume and hypervolume contributions using polar coordinate, IEEE Transactions on Evolutionary Computation 23 (5) (2019) 913–918.

[25] W. Ning, B. Guo, X. Guo, C. Li, Y. Yan, Reinforcement learning aided parameter control in multi-objective evolutionary algorithm based on decomposition, Progress in Artificial Intelligence 7 (2018) 385–398.

[26] Q. Zhang, H. Li, Moea/d: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on evolutionary computation 11 (6) (2007) 712–731.

[27] X. Cheng, W. N. Browne, M. Zhang, Decomposition based multi-objective evolutionary algorithm in xcs for multi-objective reinforcement learning, in: 2018 IEEE Congress on Evolutionary Computation (CEC), 2018, pp. 1–8. doi:10.1109/CEC.2018.8477931.

[28] Y. Liu, H. Lu, S. Cheng, Y. Shi, An adaptive online parameter control algorithm for particle swarm optimization based on reinforcement learning, in: 2019 IEEE Congress on Evolutionary Computation (CEC), 2019, pp. 815–822. doi:10.1109/CEC.2019.8790035.

[29] Y. Han, H. Peng, C. Mei, L. Cao, C. Deng, H. Wang, Z. Wu, Multi-strategy multi-objective differential evolutionary algorithm with reinforcement learning, Knowledge-Based Systems 277 (2023) 110801. doi:https://doi.org/10.1016/j.knosys.2023.110801. URL https://www.sciencedirect.com/science/article/pii/S0950705123005518

[30] K. M. Sallam, S. M. Elsayed, R. K. Chakrabortty, M. J. Ryan, Evolutionary framework with reinforcement learning-based mutation adaptation, IEEE Access 8 (2020) 194045–194071. doi:10.1109/ACCESS.2020.3033593.

[31] A. Kaur, K. Kumar, A reinforcement learning based evolutionary multi-objective optimization algorithm for spectrum allocation in cognitive radio networks, Physical Communication 43 (2020) 101196. doi:https://doi.org/10.1016/j.phycom.2020.101196. URL https://www.sciencedirect.com/science/article/pii/S1874490720302731

[32] Y. Huang, W. Li, F. Tian, X. Meng, A fitness landscape ruggedness multiobjective differential evolution algorithm with a reinforcement learning strategy, Applied Soft Computing 96 (2020) 106693. doi:https://doi.org/10.1016/j.asoc.2020.106693. URL https://www.sciencedirect.com/science/article/pii/S1568494620306311

[33] K. Jiao, J. Chen, B. Xin, L. Li, A reference vector based multiobjective evolutionary algorithm with q-learning for operator adaptation, Swarm and Evolutionary Computation 76 (2023) 101225. doi:https://doi.org/10.1016/j.swevo.2022.101225. URL https://www.sciencedirect.com/science/article/pii/S2210650222001912

[34] Y. Tian, X. Li, H. Ma, X. Zhang, K. C. Tan, Y. Jin, Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization, IEEE Transactions on Emerging Topics in Computational Intelligence 7 (4) (2023) 1051–1064. doi:10.1109/TETCI.2022.3146882.

[35] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints, IEEE transactions on evolutionary computation 18 (4) (2013) 577–601.

[36] G. Jiale, Q. Xing, C. Fan, Z. Liang, Double adaptive selection strategy for moea/d, Journal of Systems Engineering and Electronics (2019). URL https://api.semanticscholar.org/CorpusID:145828543

[37] Y. Tian, R. Cheng, X. Zhang, Y. Jin, Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum], IEEE Computational Intelligence Magazine 12 (2017) 73–87.

[38] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 1, IEEE, 2002, pp. 825–830.

[39] S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, in: Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings 3, Springer, 2005, pp. 280–295.

[40] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on evolutionary computation 7 (2) (2003) 117–132.

[41] D. W. Zimmerman, B. D. Zumbo, Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks, The Journal of Experimental Education 62 (1) (1993) 75–86.

[42] Y. Tian, X. Xiang, X. Zhang, R. Cheng, Y. Jin, Sampling reference points on the pareto fronts of benchmark multi-objective optimization problems, in: 2018 IEEE congress on evolutionary computation (CEC), IEEE, 2018, pp. 1–6.

[43] K. Deb, R. B. Agrawal, et al., Simulated binary crossover for continuous search space, Complex systems 9 (2) (1995) 115–148.
[44] K. Deb, M. Goyal, et al., A combined genetic adaptive search (geneas) for engineering design, Computer Science and informatics 26 (1996) 30–45.
[45] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii, IEEE transactions on evolutionary computation 13 (2) (2008) 284–302.
[46] M. Asafuddoula, T. Ray, R. Sarker, A decomposition-based evolutionary algorithm for many objective optimization, IEEE Transactions on Evolutionary Computation 19 (3) (2014) 445–460.
[47] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Transactions on Evolutionary Computation 22 (4) (2017) 609–622.
[48] Y. Tian, C. He, R. Cheng, X. Zhang, A multistage evolutionary algorithm for better diversity preservation in multiobjective optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems 51 (9) (2019) 5880–5894.
[49] Y. Liu, D. Gong, J. Sun, Y. Jin, A many-objective evolutionary algorithm using a one-by-one selection strategy, IEEE Transactions on Cybernetics 47 (9) (2017) 2689–2702.
[50] Y. Sun, G. G. Yen, Z. Yi, Igd indicator-based evolutionary algorithm for many-objective optimization problems, IEEE Transactions on Evolutionary Computation 23 (2) (2018) 173–187.

Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jing Wang and Hu Peng report financial support was provided by National Natural Science Foundation of China.
Ziyun Zhang report financial support was provided by Student Research Project of Jiangxi University of Finance and Economics.

Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jing Wang and Hu Peng report financial support was provided by National Natural Science Foundation of China.
Ziyun Zhang report financial support was provided by Student Research Project of Jiangxi University of Finance and Economics.