Knowledge-Based Systems 277 (2023) 110801

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Multi-strategy multi-objective differential evolutionary algorithm with reinforcement learning

Yupeng Han [a,b], Hu Peng [a,b,*], Changrong Mei [a], Lianglin Cao [a], Changshou Deng [a], Hui Wang [c], Zhijian Wu [d]

[a] School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, PR China
[b] School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330013, PR China
[c] School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, PR China
[d] School of Computer Science, Wuhan University, Wuhan 430072, PR China

ARTICLE INFO

Article history:
Received 7 June 2022
Received in revised form 10 July 2023
Accepted 11 July 2023
Available online 26 July 2023

Dataset link: https://whuph.github.io/

Keywords:
Multiobjective optimization
Differential evolution
Multistrategy
Reinforcement learning
Reference point adaptation

ABSTRACT

Multiobjective evolutionary algorithms (MOEAs) have gained much attention due to their high effectiveness and efficiency in solving multiobjective optimization problems (MOPs). However, when solving MOPs, it is important but difficult to maintain a good balance of exploration and exploitation. In addition, some reference point based MOEAs with fixed reference points perform poorly on MOPs with irregular frontiers. Therefore, this paper proposes a new multistrategy multiobjective differential evolutionary (DE) algorithm, named RLMMDE. In RLMMDE, a multistrategy and multicrossover DE optimizer is utilized to alleviate the exploration and exploitation dilemma. An adaptive reference point activation mechanism based on RL is proposed to activate the adaptive adjustment of reference points. Moreover, a reference point adaptation method is proposed to improve the performance of RLMMDE on irregular frontier problems. Experimental results of RLMMDE tested on some benchmark test suites (i.e., ZDT, DTLZ, UF, WFG, and LSMOP) and two practical mixed-variable optimization problems show that the algorithm outperforms some advanced MOEAs.

1. Introduction

Multiobjective optimization problems (MOPs), as an important optimization problem in the real world, do not have a single optimal solution that can optimize all objectives. There is a natural conflict within multiple objective functions. Multiobjective evolutionary algorithms (MOEAs), as population-based heuristic search algorithms, develop a set of solutions that do not require the use of problem-related information (e.g., gradients). Due to this search paradigm, MOEAs can solve different types of MOPs [1]. However, how to obtain a set of solutions that have better diversity and convergence and are more approximate to the Pareto front (PF) for MOPs is still a challenge. In recent years, the application of reinforcement learning (RL) to evolutionary algorithms (EAs) to improve their performance has received increased attention.

RL, as a branch of artificial intelligence (AI), is focused on how an artificial agent interacts with its environment in different states to obtain the maximum reward [2]. RL can be represented as a Markov decision process consisting of a transfer function, a reward function, and a set of states and actions [3]. By carefully designing the states, actions and reward or punishment mechanism in RL, it can interact well with the environment to solve problems. RL takes into account the influence of historical information and does not require expert experience, allowing it to take more robust actions and apply them to various types of problems [4]. This also means that RL, when applied to EAs, can enable the selection of optimal actions in different states for different types of problems during evolution. Due to these properties of RL, an increasing number of different models, for instance, Q-learning (QL) [5], policy gradient (PG) [6], deep Q-network (DQN) [7] and double deep Q-learning (DDQN) [8] are embedded in EAs.

In RL-based single-objective EAs, applying RL to the adaptation of parameters, mutation strategies or variational operators has been considered as an effective approach [2,9,10]. For parameters adaptation, AMA [11], RLDE [12] and LDE [13] introduced RL as a parameter adaptive mechanism to learn optimization experience during evolution as a way to automatically update the key parameters in the algorithm. For the adaptation of strategies or operators, hybrid algorithms [14], DEDQN [15], DEDDQN [16] and RLBSO [10] introduced different RL models to achieve strategy adaptation in the evolutionary process. The above studies show

* Corresponding author at: School of Computer and Big Data Science, Jiujiang University, Jiujiang 332005, PR China.
E-mail address: hu_peng@whu.edu.cn (H. Peng).

https://doi.org/10.1016/j.knosys.2023.110801
0950-7051/

that RL can improve the search performance of EAs under different types of problems by eliminating the uncertainty of manual selection and help EAs deal with tricky parameter or strategy selection.

However, in most RL-based EAs, RL is applied to single objective EAs and rarely applied to MOEAs. In RL-based MOEAs, Liu et al. [17] proposed a Q-learning-based MOEA, named QLPSO, in which RL is used for the adaptive selection of parameters in the algorithm. Tian et al. [18] proposed a deep RL-based operator selection method, named MOEA/D-DQN, in which deep RL is used to control the adaptive selection of operators to balance exploration and exploitation in the evolutionary process. Ning et al. [19] proposed a new RL based MOEA/D, named RL-MOEA/D, in which RL is used to control both the differential evolution (DE) operators and neighborhood size parameter. In addition to applications to parameter adaptation and operator selection, RL has also been applied to other aspects of MOEAs [20,21]. The introduction of RL into MOEAs has received increasing attention and has been applied to various aspects of MOEAs. However, RL has not received sufficient attention in reference point adaptation. Therefore, an RL-based activation mechanism for the reference point adaptation method is designed in this work.

In the offspring generation optimizer of MOEAs, most of them use the traditional single GA operator to generate the next generation. Although they have also achieved good performance, many studies have indicated that the superiority of multiple operators is evident and more suitable for solving complex optimization problems [18,22]. Therefore, multistrategy may be a good choice when solving MOPs, which are more complex than single-objective optimization problems (SOPs), and a good choice to balance exploration and exploitation. In some MOEAs with reference point adaptation, the performance of the algorithm is affected by the frequency of reference point adjustment. Therefore, it is very important to adaptively execute the reference point adaptation method. In addition, reference point adjustment during evolution is more beneficial for MOEAs to solve problems with different PF shapes, especially irregular PFs. Based on the above considerations, this paper proposes a new multistrategy multiobjective DE algorithm, named RLMMDE. The main contributions of this work are summarized as follows:

- To better balance exploration and exploitation and adaptation to complex multiobjective environments, a DE optimizer consisting of multiple mutation strategies and crossover operators with different preferences is proposed.
- An activation mechanism is designed to determine whether reference point adjustment is needed, and Q-learning is employed to decide on the appropriate time to perform reference point adjustment based on the current generation of reference point utilization, which better reduces the negative impact due to reference point adjustment.
- A reference point adaptation method, where the new reference point set consists of candidate solutions and the original associated reference points, generates a more uniform distribution of the obtained solutions and improves the performance of the algorithm in solving irregular PF problems.

The remainder of the paper is structured as follows. The related introduction of multistrategy and multicrossover DE, RL-based MOEAs, and the reference point adaptation method is illustrated in Section 2. In Section 3, the motivation of this work is presented. The proposed RLMMDE is shown in detail in Section 4. The performance of RLMMDE in benchmark test suites is evaluated in Section 5. Finally, in Section 6, the conclusion is summarized.

2. Related work

Multiobjective (MO) optimization, which aims to optimize problems with multiple optimization objectives, is a well-known and popular area of research. The main involved areas are production planning and scheduling, industrial design and image recognition. In recent years, many different evolutionary algorithm (EA)-based multiobjective evolutionary (MOE) optimization methods have been successfully applied to address MOPs, such as production scheduling [23], detection systems [24], classification [25], association rule mining [26,27]. When solving MOPs, it is important but difficult to make MOEAs maintain a good balance between exploration and exploitation during evolution. The dilemma of exploration and exploitation is not well balanced by a single operator. Multistrategy or multioperator is a good choice due to their different preferences. In recent years, the use of reinforcement learning to enhance evolutionary algorithms has achieved good results. Reference point adjustment is more beneficial for MOEAs to solve problems with different Pareto front shapes (PFs), especially for irregular fronts. The related work of these three directions is briefly described below.

2.1. Multistrategy and multicrossover DE

Evolutionary computation uses computational models of evolutionary processes as a key element in the design and implementation of computer based problem solving systems. A variety of computational models of evolution have been proposed and studied, which are referred to as EAs [28]. EAs maintain a population in which individuals evolve according to selection rules and other operators i.e., recombination and mutation. Each individual in the population receives its degree of adaptation in the environment. Selection focuses attention on individuals with high fitness levels, thus making use of the available fitness information. Perturbation of these individuals by crossover and mutation can provide robust and powerful adaptive search mechanisms. Multiple mutation strategy has been the focus and hot topic in the research of EAs. Whether in single-objective EAs or MOEAs, many multistrategy based algorithms have been proposed and achieved good performance. The crossover operator is another important operation in evolutionary computation in addition to the mutation strategy, but the crossover operator has been given much less attention than the mutation strategy. This is related to the fact that it is more difficult to propose crossover operators with certain characteristics that are different from traditional crossover operators. Some of the recent research papers on multistrategy and multicrossover are listed below.

In multistrategy-based single objective EAs, Sallam et al. [22] proposed an improved multistrategy DE, named IMODE, in which three competitively varying strategies are used. The algorithm focuses more on the operator with the optimal performance in the evolutionary process. In addition, a local search is used to accelerate the convergence of the algorithm. Yu et al. [29] proposed a data-driven DE, named DDEA-MESS, in which three sampling strategies with different characteristics are combined. In addition, a mechanism for multistrategy selection and a method for updating the database are proposed to avoid falling into a local optimum. Zhou et al. [30] proposed a hierarchical multistrategy based DE, named DEHM, in which different strategies are used for different subpopulations to maintain the convergence and diversity of populations. In addition, a new selection strategy in which potential individuals with superior genes are not lost is proposed.

In multistrategy based MOEAs, Lin et al. [31] proposed a new multistrategy operator selection method based on decomposed MOEA, named B-AOS, in which four different DE strategies are used to balance exploration and exploitation. In addition, two criteria emphasizing different properties are proposed to assist in strategy selection. Yan et al. [32] proposed a multistrategy approach based on leader recommendation, called LROS, in which simulated binary crossover (SBX) and three DE strategies guide the evolution of the population. In addition, the evolutionary process is divided into several stages to guide the selection of strategies. Li et al. [33] proposed a multistrategy approach based on fitness rank, named FRRMAB, in which four different characteristics of DE strategies are utilized. The fitness improvement rate and a decay mechanism are used to assist in selecting strategies. Sun et al. [34] proposed a new multistrategy approach based on dynamic Thompson sampling, named DYTS, in which four different characteristics of DE strategies are utilized. In addition, a reward allocation mechanism is used to select the better operator during evolution. Besides, many multistrategy approaches of other heuristic and swarm intelligence algorithms have been used in MOEAs to enhance the performance of the algorithms [35–37].

The crossover operator in DE is not as much hot and proposed as the mutation strategy, but there are some excellent crossover operators with different features being proposed. Reynoso-Meza et al. [38] proposed a hybrid DE algorithm with adaptive crossover operator. Due to binomial crossover (for crossover probability <1) is not a rotationally invariant operator, it has poor performance in dealing with functions with high dependence between search variables. In order to solve this issue, two different crossover operators, traditional binomial crossover and rotationally invariant line recombination operator, are used for different problems. Bujok et al. [39] studied the efficiency and complexity of four different crossover operators in DE. An experimental comparison of binomial crossover, exponential crossover, Eigenvector crossover and exponential crossover with randomly ordered continuously selected coordinates shows that the different operators have different preferences for different problems and differ in time complexity and problem solving efficiency. Different crossover operators are applicable to different problems, and the characteristics of problems in a test suite are also different. Using multiple crossover operators is a good choice to improve the performance and robustness of the algorithm, especially for more complex multi-objective environments.

2.2. Reinforcement learning based EAs

Reinforcement learning (RL) is a field of machine learning and belongs to a branch of AI. As shown in Fig. 1, RL focuses on how an artificial agent interacts with its environment in different states. Based on the environment's response to previous interactions, the artificial agent takes action to maximize the cumulative reward. Due to this property of reinforcement learning, it is widely used in EAs. Out of the variety of RL-based EAs, it is mainly used to adaptively adjust the control parameters and the adaptive selection of strategies or operators [2,9,10,40–44].

However, multiobjective algorithms based on RL have rarely been proposed. On the basis of NSGA-II [45], Kaur et al. [46] proposed an NSGA-II based MOEA named NSGA-RL, in which RL is used for parameter adaptation in the GA. Huang et al. [3] proposed a multiobjective DE algorithm with an RL strategy (LRMODE). In LRMODE, the results of the analysis of the topology of the local landscape are combined with the RL strategy to determine the optimal probability distribution of the algorithm's mutation strategy, which controls the adaptive selection of strategies. Ma et al. [47] proposed a MOEA based on a two-engine coevolution mechanism named MOBFA. In MOBFA, an RL based neighbor-discount-information mechanism (NDI) is used to adjust the position of individuals.

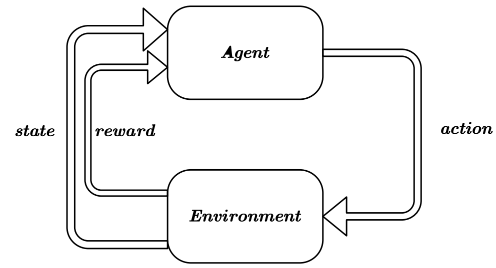


Fig. 1. Illustration of reinforcement learning. Agent interacts with its environment in different states. Based on the environment's response to the action of agent, the agent should take which action to maximize the cumulative reward.

2.3. Existing reference point adaptation methods

The true PF is unknown in practical applications, so the evolution of the population is guided by sampling a set of reference points based on the approximate PF obtained from MOEAs. This means that the selection and adjustment of the reference points becomes crucial. The predefined fixed reference point does not perform well on some MOPs. To address this issue, some reference point adaptation methods have recently been proposed to adjust the distribution of reference points to make it as close as possible to the approximate PF [48–52]. It is worth noting that reference points are also referred to as reference vectors or weight vectors in some decompositions based on MOEAs. Based on the way these methods adaptively adjust their reference points, they can be divided into two categories. One adaptive adjustment of reference points is based on the distribution of candidate solutions stored in an external archive, while the other is based on the distribution of candidate solutions in the current population for each generation [48]. To describe these reference point adaptation methods more clearly, we will briefly introduce relevant papers on these two methods.

ANSGA-III [50] is a representative item in the second category. During each generation of the population evolution process in ANSGA-III, the reference points are adjusted according to the better candidate solutions in the current generation of the population to obtain better population diversity. Similar to the adjustment of the reference points described above, the same two types are used. A niche count of reference points is used to discriminate the method of reference point adjustment. If a reference point has empty niche counts, then the added reference point will be deleted. However, when the reference point has a larger niche count, then a new reference point will be added randomly.

MOEA/D-AWA [51] is a representative one of the first category. In MOEA/D-AWA, after the population has converged to a certain extent, the reference points are adjusted according to the current Pareto optimal solutions, i.e., the candidate solutions in the extra archive for obtaining better population diversity. The reference points are adjusted in two ways, i.e., addition and deletion. Specifically, if a nondominated solution in the archive is located in a crowded region of the PF, then the reference points associated with the solution are deleted. However, when a nondominated solution in the archive lies in a sparse region of a PF, then the reference points associated with the solution will be regenerated.

3. Motivation

Among the MOEAs proposed to solve MOPs in recent years, GA and DE are the most common optimizers for offspring generation. There are also some MOEAs using other optimizers [35,37,53]. However, while the algorithms are used to handle complex PFs problems, GA optimizers may lose the diversity of populations in the early stages of the search, which is a very important factor in solving MOPs [54]. Moreover, according to [55,56], GA optimizer often generates inferior solutions during the optimization process. DE is a simple, excellent optimizer with two main operations, mutation strategy and crossover operator. Moreover, it has strong exploratory properties in the early stages of the search. This means that it is more advantageous when solving complex MOPs. Therefore, this paper uses the DE operator as an optimizer for MOEA. Furthermore, there is no omnipotent operator that can effectively balance exploration and exploitation. The performance of the operators depends on the characteristics of problems. Therefore, multiple mutation strategy and crossover operators with different characteristics are used in this paper to balance exploration and exploitation during evolution, improve the performance and robustness of the algorithm.

As stated in Section 2.2, RL is an important area of machine learning. The properties of RL make it widely used in various RL-based EAs for adaptively adjusting control parameters and adaptive selection of strategies or operators. A variety of RL models are used in these RL-based EAs [15,16,40]. Depending on whether these models use neural networks, they are roughly classified into deep RL models and basic RL models. The EAs based on deep RL models achieve good performance. However, they are time-consuming and demanding on computing devices when the number of dimensions increases. Their time complexity is much larger than that of the basic RL models [13]. This situation will be even more complicated in complex MOPs. However, for the basic RL models, they are relatively simpler and less expensive. The Q-learning model is one of the basic RL models, first proposed by Watkins and Dayan [5]. Due to its simple application, fast convergence, and low computational cost, an adaptive reference point activation mechanism based on Q-learning is proposed in this paper for activating the adaptive adjustment of reference points.

According to the description in Section 2.3, in reference point based MOEAs, a set of predefined reference points is used to guide the overall evolution so that the obtained candidate solutions are distributed as uniformly as possible along the PF. However, these predefined reference points are fixed throughout the evolutionary process, and they do not change once they are constructed. This may lead to the poor performance of the algorithms on some MOPs, especially on some problems with irregular PFs [48]. For these problems, only a few reference points are associated with the solution, which loses the diversity of the solutions. Based on the above considerations, the starting point of our algorithm is to improve the performance of the algorithm in solving problems with irregular PFs. Therefore, in this paper, a reference point adaptive approach is proposed for increasing the performance of RLMMDE over irregular PF problems.

4. The proposed RLMMDE

In this section, a multistrategy multiobjective DE algorithm, named RLMMDE, is proposed. In this algorithm, a multi-strategy and multicrossover DE optimizer is utilized to better balance exploration and exploitation and improve the performance of the algorithm in solving MOPs. In addition, to control the adaptive adjustment of reference points, an adaptive reference point activation mechanism based on RL is proposed. Moreover, a reference point adaptation method is also proposed for improving the performance of the algorithm in solving irregular frontier problems. In the following subsections, more details about RLMMDE can be found.

4.1. Multistrategy and multicrossover DE optimizer

The balance between exploration and exploitation has been a crucial research topic in MOEAs. One of the key factors that influences this balance is the evolutionary strategy adopted by these algorithms. However, there exists an inherent trade-off between a strategy's ability to explore and exploit. If a strategy tends to prioritize global search, its exploitation capabilities may be compromised, and vice versa [37]. Using multiple evolutionary strategies with distinct characteristics can effectively balance exploration and exploitation requirements [18,34,36]. Although GA is currently the most widely used optimizer, it suffers from inadequate diversity in the early stages of the search, and tends to produce suboptimal solutions during the evolutionary process [54]. Therefore, a DE optimizer with multiple strategies and crossover operators [57,58] is utilized in this paper. In this optimizer, six evolutionary strategies consisting of three mutation strategies with different characteristics and two crossover operators are used to generate offspring. In addition, the current most suitable evolutionary strategy is selected for each individual by a roulette wheel selection method. However, these evolutionary strategies were originally designed for single-objective optimization problems and in some aspects are not suitable for MOPs. Therefore, some important modifications are made to the original method.

In the multistrategy and multicrossover DE optimizer, three strategies with different characteristics are used, namely DE/rand/1, DE/best/1 and DE/target-to-rand/1. For the choice of crossover operator, the classical binomial crossover with better problem versatility and the multiple exponential recombination [59] with better dealing with the dependent variable are used. The mutation strategies with different characteristics are presented in the following manner:

(1) DE/rand/1:

$$\mathbf{V}_{i,G} = \mathbf{X}_{r1,G} + F \cdot \left(\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G}\right) \tag{1}$$

(2) DE/best/1:

$$\mathbf{V}_{i,G} = \mathbf{X}_{best,G} + F \cdot \left(\mathbf{X}_{r1,G} - \mathbf{X}_{r2,G}\right) \tag{2}$$

(3) DE/target-to-rand/1:

$$\mathbf{U}_{i,G} = \mathbf{X}_{i,G} + rand \cdot \left(\mathbf{X}_{r1,G} - \mathbf{X}_{i,G}\right) + F \cdot \left(\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G}\right) \tag{3}$$

where F is the scaling factor that amplifies the difference vectors. $\mathbf{X}_{r1,G}$, ..., $\mathbf{X}_{r3,G}$ are mutually different individuals randomly chosen from the population that are all different from the current individual $\mathbf{X}_{i,G}$ at generation G. $\mathbf{X}_{best,G}$ is the superior individual in the population.

The classical binomial crossover and the multiple exponential recombination [59] are presented as Eq. (4) and Algorithm 1.

$$\mathbf{U}_{i,j,G} = \begin{cases} \mathbf{V}_{i,j,G}, & if\ rand_j \leqslant CR\ or\ j = j_{rand} \\ \mathbf{X}_{i,j,G}, & otherwise \end{cases} \tag{4}$$

where $rand_j$ is a uniformly distributed random number between 0 and 1, the crossover probability $CR \in [0, 1]$ is a parameter that greatly affects the performance of DE, j denotes the jth dimension of the vector. To ensure that $\mathbf{U}_{i,G}$ gets at least one component from $\mathbf{V}_{i,G}$, a random integer j_{rand} is chosen from the interval [1,D].

In this optimizer, DE/rand/1 is the most widely used, exploratory and robust to many problems. DE/best/1 performs better in convergence rate and prefers exploitation. In multiobjective optimization, there does not exist an optimal individual in a population, so the individuals from the first level after the nondominated sorting [60] are randomly selected as the optimal individual. DE/target-to-rand/1 is a modified version of DE/current-to-rand/1 that is suitable for the rotation problem [61]. In addition to

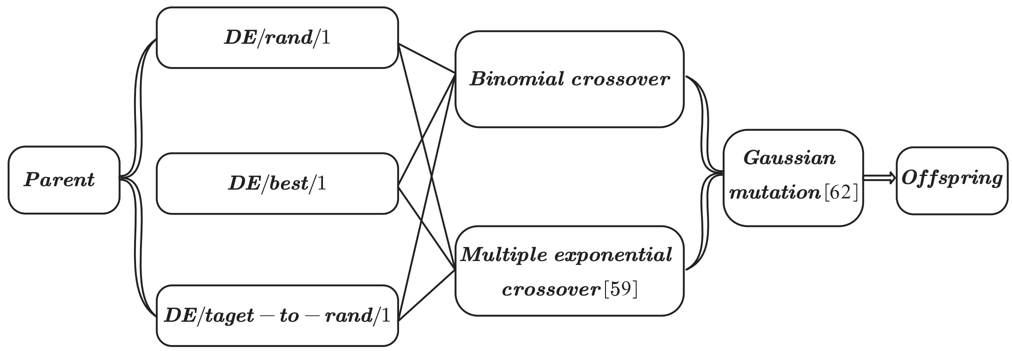


Fig. 2. Process of the generation of offspring. The parent generate offspring by six evolutionary strategies with three different characteristic mutation strategies and two different characteristic crossover operators.

Algorithm 1 Mutation and multiple exponential recombination[59]

/*Mutation and multiple exponential recombination phases*/
$Crm = Em/(Em+1)$, where $Em = T * Cr$; $Crs = Es/(Es+1)$, where $Es = T*(1-Cr)$; Set n as an integer randomly generated from the interval $[1, D]$; k = 1, $MutationEnable$ = 1;
while $k \leqslant D$ **do**
  if $MutationEnable$ = 1 **then**
    while $k \leqslant D$ and rand(0,1) $\leqslant Cr_m$ **do**
      $d = (n)_D$, where $(n)_D$ equals to n if $n \leqslant D$, equals to $n-D$ if $n > D$;
      $\mathbf{U}_{i,j} = \mathbf{X}_{r1,j} + F \cdot \left(\mathbf{X}_{r2,j} - \mathbf{X}_{r3,j}\right)$;
      $n = n+1$, $k = k+1$;
    end while
    $MutationEnable$ = 0
  else
    while $k \leqslant D$ and rand(0,1) $\leqslant Cr_s$ **do**
      $d = (n)_D$, where $(n)_D$ equals to n if $n \leqslant D$, equals to $n-D$ if $n > D$;
      $\mathbf{U}_{i,j}$=$\mathbf{X}_{i,j}$;
      $n = n+1$, $k = k+1$;
    end while
    $MutationEnable$ = 1
  end if
end while

the binomial crossover operation in the traditional DE algorithm, several other excellent crossover operations have been proposed, including exponential crossover and multiple exponential recombination [59]. It is worth noting that the multiple exponential recombination operator has been shown to be more robust than the traditional binomial when dealing with the dependent variable. Moreover, the design mechanism of this crossover operator retains the main advantages of the traditional crossover operator. Therefore, we choose these three mutation strategies and two crossover operators with different characteristics to combine into six evolutionary strategies to balance exploration and exploitation and adapt to a more complex multiobjective optimization environment. In addition, to further enhance the local search performance of the algorithm on MOPs, Gaussian mutation [62] is used with a certain probability. After generating offspring, the environment selection method in [60] is utilized to select better solution for the next generation. Based on the above description, the illustration and pseudocode of the multistrategy and multi-crossover DE optimizer are presented in Fig. 2 and Algorithm 2, respectively.

Algorithm 2 Multi-strategy and multi-crossover DE optimizer

Input: Population P, scaling factor F, crossover probability CR, mutation strategy candidate pool, crossover operator candidate pool
for each $P_i \in P$ **do**
  $F = N(F, \sigma)$, $CR = N(CR, \sigma)$
  Randomly select a mutation strategy from Eq. (1), Eq. (2) or Eq. (3) to generate $\mathbf{V}_i$;
  Randomly select one between the two crossover operator to generate the new solution $\mathbf{U}_i$;
  if $rand < Prob$ **then**
    Execute gaussian mutation method[62];
  end if
  $FEs = FEs + 1$;
end for
Output: Offspring $P^{'}$

4.2. Adaptive reference point activation mechanism based on reinforcement learning

Q-learning, as a model-free reinforcement learning algorithm, is considered one of the most productive and efficient algorithms [2,63]. In Q-learning, Q stands for Quality. To acquire future rewards, the quality of each executed action is learned by the agent. The set of executed actions and the set of environmental states can be denoted as $S = \{s_1, s_2, \dots, s_m\}$ and $A = \{a_1, a_2, \dots, a_n\}$, respectively. In a given environment, the agent is given a state and asked to select an action with the maximum Q value to execute in the current state at each iteration. The agent selects the best action based on a collected table of information, i.e., the Q-table. The environment then gives the agent a reward r_{t+1} and a new state S_{t+1} after the action has occurred. The collected information is used by the agent to create evaluations of the predicted values $Q(s_t, a_t)$ gained by each action.

As a value-based learning algorithm, Q-learning updates its value function through a Bellman equation. The following Eq. (5) is an expression for the update equation:

$$Q_{s_t,a_t}^{t+1} = Q\left(s_t, a_t\right) + \alpha\left[r_t + \gamma max_a Q\left(s_{t+1}, a_t\right) - Q\left(s_t, a_t\right)\right] \tag{5}$$

where α is the learning rate and γ is the discount factor, both of which take values from (0,1). r_t is the value of the reward or punishment after taking action a_t.

The reward or punishment decision is determined by the environment. If positive feedback is received, a reward is obtained, otherwise a penalty is assigned. In the early stage of evolution,

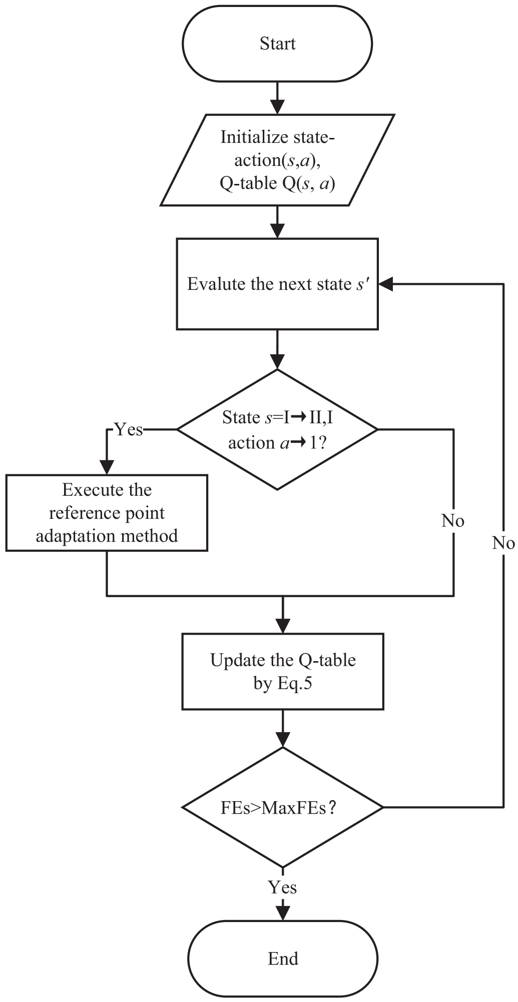


Fig. 3. Process of the adaptive reference point activation mechanism.

the value of α is set high, but as the population evolves, α should become smaller ensure the convergence of the policy for optimal actions. The discount factor γ defines whether the agent is biased toward future rewards or is more focused on current rewards. The lower the value, the greater the agent will favor current rewards, and vice versa. In this paper, the values of α and γ are the same as in [2]. γ is set to the commonly used value of 0.8. The value of α is set as follows:

$$\alpha = 1 - 0.9\left(\frac{FEs}{MaxFEs}\right) \tag{6}$$

As described in Section 3, the most common role of RL is to be used for adaptive adjustment of parameters and adaptive selection of strategies. In this paper, we make the first attempt to design a reference point activation mechanism using Q-learning. The states, actions, reward and the setting of rewards or penalties in Q-learning are very important to its performance. Next, the relevant settings and principle of these components in the reference point activation mechanism will be briefly described.

To discriminate the association of reference points in the current generation, the association operation in NSGA-III [60] is used. The association operation refers to the linking of individuals in the population to a reference point. After connecting the reference point and the origin to establish a reference line, the vertical distance of each individual in the population from each reference line is calculated. The reference point of the reference line closest to an individual of the population is considered to be associated with that individual. ρ_j denotes the total number of all individuals in the population associated with the jth reference point. Here, those reference points that are associated with individuals are considered useful and those that are not associated with individuals are considered useless. The ideal case is that each reference point has an individual in the population associated with it.

Table 1
The settings of reward and penalty. Depending on the transition of the reference point state, it is determined whether the execution of the action (reference point adaptation method) is rewarded or penalized. Positive numbers represent rewards and negative numbers represent penalties.

State	I	II
I	−10	5
II	−10	5

Based on these considerations, the number of reference points that are not associated with individuals of the population *count* is used as a criterion for classification, and a predefined *threshold* is set to classify the states. When the total number of reference points not associated with individuals in the population is more than the set *threshold*, the current reference point utilization is considered low. Then, it is necessary to perform the reference point adaptation method. This state is denoted as *state I*. Otherwise, it is considered that the current reference point utilization is good and adaptive adjustment of reference points is not necessary. This state is denoted as *state II*. In *state I*, the current state requires adaptive adjustment due to insufficient utilization of reference points, so there are two actions in this state, which are executed ($a\to 1$) or not executed ($a\to 2$). In *state II*, there is no action to take in this state because the current state does not require no adaptive adjustment of reference points due to high utilization.

In addition, the setting of rewards or penalties is another important factor that affects the number of reference point executions. The specific settings are shown in Table 1. The reason for this setting is as follows: when the state of the current population reference point changes from I to II or remains at II after the execution of reference point adaptation, it indicates that the reference point utilization becomes high at this time, so the action is valid and will be rewarded. Otherwise, it indicates that the reference point utilization becomes low at this time and the action is invalid and will be penalized. Based on the above description, the process and pseudocode of the adaptive reference point activation mechanism based on RL are shown in Fig. 3 and Algorithm 3.

Fig. 3 and Algorithm 3 illustrate the overall process of the activation mechanism. First, the state action pairs and Q-table are initialized. The utilization of the reference point is determined based on the association operation, and subsequently the state of the reference point is determined. Then, the reward and punishment of the current action is determined based on the transition of the state. Finally, the current state action is used to determine whether to execute the reference point adaptation method. Each execution needs to update the Q-table according to Eq. (5).

4.3. A reference point adaptation method

Reference point based MOEAs are known to use a predetermined set of reference points to guide the overall evolutionary stage. Such a set plays a significant role in ensuring that the candidate solutions obtained are as uniformly distributed along the Pareto front as possible. However, keeping the predefined

Algorithm 3 Adaptive reference point activation mechanism based on reinforcement learning

Input: State-action pair (s, a), Q-table $Q(s, a)$, current number of function evaluations *FEs*, the number of reference points that are not associated with individuals of the population *count*, maximum number of function evaluations *MaxFEs*

Initialize the current state s, action a;
Execute the action a with the maximum Q value in the current state;
if $count > threshold$ **then**
  The next state of reference point $s' \leftarrow$I;
else
  The next state of reference point $s' \leftarrow$II;
end if
if State s I$\rightarrow$II or s II$\rightarrow$II **then**
  Reward the current action a according to Table 1;
else
  Punish the current action a according to Table 1;
end if
if State $s \rightarrow$I and action $a \rightarrow 1$ **then**
  Execute reference point adaptation method in Algorithm 4;
  Update the Q-table by Eq. (5);
else
  Update the Q-table by Eq. (5);
end if
Update the current state $s \leftarrow$ the next origin state s';

Output: An action a with the maximum Q value in the current state

reference points unchanged throughout the evolution process can lead to suboptimal algorithm performance on some MOPs. NSGA-III [60] has lower-than-expected efficiency when handling problems featuring irregular fronts. This issue also affects the proposed algorithm. To improve the performance of the proposed algorithm on irregular frontier problems, a reference point adaptation method is proposed.

It is important to note that excessive frequent changes of reference points may negatively impact the algorithm's convergence. To mitigate this issue, the reference point adaptation method is not employed during the initial stage of evolution [49, 64]. In this study, reference point adjustments only begin after $0.8 * MaxFEs$. Although performing reference point adaptation only occurs in the later stages, it was found in some preexperiments that using reference point adaptation all the time in the later stages could lead to worse results on some problems. Therefore, an RL based reference point activation mechanism in the last section is proposed to control the number of times of the reference points to be executed.

In this method, candidate solutions in the population are incorporated into the set of reference points to guide the evolution of the population. The reference point set composed of candidate solutions and the original already associated reference points not only maintains the uniform distribution provided by the original reference points, but also reflects the shape of the approximate PF in the population [48].

Based on the above considerations, a reference point adaptation method is proposed. Algorithm 4 presents the pseudocode of the method. First, the association operation [60] is used to count the number of individuals associated with each reference point, *rho*. All reference points with and without individuals associated with them are identified, *index*, *Noindex*. Those reference points without individuals associated with them are removed from the original reference point set Z. Next, the number of new reference points to be added is calculated based on the reference points associated with individuals, *num*, ensuring that the number of reference points removed is less than *num*. The individuals set S in the population furthest away from each associated reference point are found, and those already selected are eliminated, ensuring that the same individuals are not selected again. Finally, randomly select the individuals with the same number of reference points as the deleted ones from the set S to add to the new set of reference points, Z'.

Algorithm 4 Reference point adaptation method

Input: Population P, the original reference points set Z, associated reference points and solution *rho*, *FEs*, *MaxFEs*

/*Late stages of evolution*/
if $FEs > MaxFEs \times 0.8$ **then**
  if the activation condition in Algorithm 3 is satisfied **then**
    Find the index of reference points that are not associated with individual, $Noindex = find(rho = 0)$;
    Find the index of reference points that are associated with individual, $index = find(rho)$;
    Delete the reference points in Z that are not individually associated;
    Calculate the number of reference points to be generated based on the reference point associated with the individual, $num = \lceil rho(index)/sum(rho) * length(Noindex) \rceil$;
    The number of associated reference points set $Associate\_Z\_num = length(index)$;
    while $Associate\_Z\_num < lenth(Z)$ **do**
      for $i = 1 : length(index)$ **do**
        for $j = 1 : num(i)$ **do**
          Find the individual set S farthest from the current reference point ;
        end for
        if $Associate\_Z\_num > lenth(Z)$ **then**
          Break;
        end if
      end for
    end while
    if $lenth(S) > lenth(Z) - lenth(index)$ **then**
      Randomly select $lenth(Z) - lenth(index)$ individuals from the set S to put into a new set of reference points;
    end if
  end if
end if

Output: New reference points set Z'

4.4. The framework of RLMMDE

A multistrategy multiobjective DE algorithm, named RLMMDE, is proposed. First, a DE optimizer with multiple strategies and crossover operators is proposed to improve the performance and better balance exploration and exploitation. In addition, to adaptively use the reference point adaptation method, an activation mechanism based on RL is proposed. Finally, to improve the performance of the algorithm in solving irregular problems, a reference point adaptation method is proposed. Based on the above description, the framework of RLMMDE can be presented as Algorithm 5.

5. Experimental studies

In the experimental section, benchmark suites are used to test the performance of RLMMDE. Different types of advanced

Algorithm 5 The framework of RLMMDE

Input: Population size N, reference points Z, the predefined *threshold*, the empty Q-table, the discount factor γ, current generation *FEs*, maximum number of iterations *MaxFEs*

Randomly initialize the population P;
Generate the original set of uniformly distributed reference points Z;
Compute ideal point Z^{min};
$FEs = N$;
while $FEs < MaxFEs$ **do**
  $P^{'}$ ←Execute the multi-strategy and multi-crossover DE optimizer according to Algorithm 2;
  Update ideal point Z^{min};
  Execute the environmental selection according to NSGA-III[60];
  if $FEs > MaxFEs \times 0.8$ **then**
    Execute adaptive reference point activation mechanism according to Algorithm 3 ;
    if the activation condition is satisfied **then**
      Execute the reference point adaptation method according to Algorithm 4 ;
    end if
  end if
  $FEs = FEs + N$;
end while

Output: The final population P'

algorithms are used to verify the performance of the proposed algorithm. To further validate the effectiveness of the algorithm components, different algorithm variants are used to verify the effectiveness of multistrategy and multicrossover DE optimizer, adaptive reference point activation mechanism based on RL and reference point adaptation method. In addition, two practical mixed-variable optimization problems are also used to test the performance of RLMMDE.

5.1. Experimental design

5.1.1. Test problems and performance metrics

To test the performance of RLMMDE, different types of test suites, ZDT [65], DTLZ [66] and UF [67] are used. The ZDT test suite contains problems with many different PFs, such as many local optimal, convex, discontinuous convex and nonconvex PFs. The DTLZ test suite also contains problems with many different characteristics, such as discontinuous nonconvex, and multimodal PFs. The UF test suite contains unconstrained test problems with complex variable linkages. In addition, WFG [68], LSMOP [69], SDTLZ1, SDTLZ2, CDTLZ2 [60] and IDTLZ1, IDTLZ2 [50] are used to further analyze the performance of RLMMDE. The detailed settings of population size (N), number of objectives (M), dimensions (D) and maximum number of function evaluations (MaxFEs) for these functions are given in Table 2.

It is known that the evaluation criteria of the final outcome set are determined by the convergence and diversity of the Pareto solution set. Thus, in this paper, two different performance metrics, inverted generational distance (IGD) [70] and averaged Hausdorff distance (Δ_p) [71], are used to evaluate the final outcome set obtained by RLMMDE and the compared algorithms. The IGD metric is one of the most commonly used performance metrics in MOEAs. It represents the average distance of the minimum distance from each reference point to the nearest solution. The smaller the value, the better the overall performance of the algorithm. The Δ_p metric is a frequently used performance metric in MOEAs. It represents the "average Hausdorff distance" between the solution set and the true PF, and consists of the indicators generational distance (GD) [72] and IGD. The smaller its value is, the better the overall performance of the algorithm. In the calculation of these two indicators, 10 000 uniformly distributed points on the PF are sampled by Das and Dennis's method [73].

Table 2
Setting of benchmark test suites. N, M, D and MaxFEs denote population size, number of objectives, dimensions and maximum number of function evaluations, respectively.

Problems	N	M	D	MaxFEs
ZDT1∼ZDT3	100	2	30	10 000
ZDT4, ZDT6	100	2	10	10 000
DTLZ1	105	3	7	30 000
DTLZ2∼DTLZ6	105	3	12	30 000
DTLZ7	105	3	22	30 000
UF1∼UF7	600	2	30	300 000
UF8∼UF10	600	3	30	300 000
WFG1∼WFG9	105	3	12	30 000
LSMOP1∼LSMOP9	105	3	300	30 000
IDTLZ1, SDTLZ1	105	3	7	30 000
IDTLZ2, SDTLZ2, CDTLZ2	105	3	12	30 000

5.1.2. Comparison algorithms and parameter setting

Different types of advanced algorithms are used for comparison to verify the performance of the proposed algorithm. The first type has 9 algorithms used for comparison. They are the popular traditional MOEAs, which are MOEA/D [74], NSGA-II [45], NSGA-III [60], GrEA [75], NSGA-II/SDR [76], MaOEA-DDFC [77], MOPSO [78], PESA-II [79], and IBEA [80].

The second type has 3 algorithms for comparison, namely MOEA/D-DRA [81], IM-MOEA [82] and MOEA/D-FRRAB [33]. None of them use the classical GA operator, but other variation operators, where MOEA/D-DRA is a variation of MOEA/D, which uses an offspring generation method based on a DE operator, which performs better on the UF test problem. IM-MOEA uses a Gaussian process based offspring generation method, which also performs better on UF test problems. MOEA/D-FRRMAB is a variant of MOEA/D based on adaptive operator selection, which has several different offspring generation strategies.

The third type has 4 algorithms for comparison, namely ANSGA-III [50], MOEA/D-AWA [51], RVEA [83] and RVEA* [83]. They all use reference point adaptation method. RVEA* is an improved version of RVEA for addressing problems with irregular PFs. In addition, the initial reference points of MOEAs based on reference points are generated using the systematic approach of Das and Dennis [73].

The parameter settings of the compared algorithms are set to the settings of their original papers. In this paper, the scaling factor F and crossover probability CR are generated from a Gaussian distribution with mean 0.9, variance 0.1 and mean 0.1, variance 0.1, respectively. The discount factor γ is set to 0.8. The predefined *threshold* is set to 50.

The statistical confidence of pairwise comparisons between RLMMDE and the compared algorithm are evaluated using the Wilcoxon rank sum test [84] at a significance level of 0.05 and the Friedman test [85]. The results for each test instance are expressed as the mean value and standard deviation obtained over 30 independent runs. The optimal results for each test instance are shown in **bold**. The symbols "+", "−" and "≈" after the experimental results indicate that the results of other MOEAs are significantly better, worse and statistically similar to the results of RLMMDE, respectively. All algorithms are executed on PlatEMO [86], MATLAB2016b.

Table 3
The comparisons of IGD values obtained by RLMMDE and the 9 compared algorithms on ZDT, DTLZ, UF test suites.

Problem	MOEA/D	NSGA-II	NSGA-III	PESA-II	RLMMDE
ZDT1	1.4283e−1 (4.94e−2) −	1.1863e−2 (1.86e−3) −	2.6443e−2 (5.68e−3) −	4.8625e−2 (4.89e−2) −	**6.6623e−3 (8.56e−4)**
ZDT2	5.3334e−1 (6.86e−2) −	2.5794e−2 (3.06e−2) −	4.5494e−2 (2.35e−2) −	1.0753e−1 (8.97e−2) −	**6.2902e−3 (1.19e−3)**
ZDT3	1.4808e−1 (4.03e−2) −	1.1827e−2 (5.52e−3) =	2.5280e−2 (9.90e−3) −	3.3700e−2 (2.70e−2) −	**1.1207e−2 (1.38e−3)**
ZDT4	5.0262e−1 (1.71e−1) =	2.8961e−1 (1.70e−1) +	6.4792e−1 (3.81e−1) =	**1.8555e−1 (1.32e−1)** +	5.0678e−1 (3.50e−1)
ZDT6	8.3480e−2 (2.64e−2) −	5.4567e−2 (3.11e−2) −	2.9578e−1 (1.47e−1) −	2.3978e−2 (7.47e−3) −	**3.3455e−3 (3.08e−4)**
DTLZ1	1.9635e−2 (5.30e−4) +	3.9414e−2 (5.27e−2) −	**1.9519e−2 (5.11e−4)** +	2.4357e−2 (1.29e−3) +	3.2247e−2 (5.15e−2)
DTLZ2	**5.0306e−2 (2.42e−6)** +	6.7041e−2 (2.54e−3) −	5.0329e−2 (2.45e−5) +	6.4213e−2 (3.18e−3) −	5.0509e−2 (3.94e−5)
DTLZ3	6.6466e−1 (8.45e−1) +	2.9282e−1 (4.74e−1) +	6.7287e−1 (9.97e−1) +	3.8722e−1 (5.29e−1) +	1.2904e+0 (1.07e+0)
DTLZ4	4.2637e−1 (3.46e−1) =	6.6102e−2 (2.50e−3) −	6.6728e−2 (8.96e−2) −	6.2103e−2 (1.56e−3) −	**5.0792e−2 (8.48e−5)**
DTLZ5	3.1149e−2 (7.85e−5) −	5.4665e−3 (3.13e−4) +	1.1390e−2 (1.23e−3) −	1.1127e−2 (1.31e−3) −	8.9480e−3 (1.51e−3)
DTLZ6	3.1244e−2 (5.42e−5) −	**5.6055e−3 (1.66e−4)** +	1.7754e−2 (2.06e−3) −	1.4043e−2 (2.12e−3) =	1.5235e−2 (5.80e−3)
DTLZ7	1.3705e−1 (3.52e−3) +	1.1030e−1 (9.53e−2) +	9.1683e−2 (7.66e−2) +	1.5748e−1 (1.35e−1) +	1.3633e+0 (8.43e−2)
UF1	2.6270e−1 (1.17e−1) −	9.2505e−2 (1.30e−2) −	7.4707e−2 (1.25e−2) −	9.0702e−2 (2.67e−2) −	3.3484e−2 (1.25e−2)
UF2	1.1943e−1 (5.03e−2) −	2.3860e−2 (5.11e−3) +	2.2085e−2 (1.72e−3) +	2.6641e−2 (8.35e−3) −	2.5602e−2 (3.96e−2)
UF3	3.0719e−1 (2.87e−2) −	1.6460e−1 (4.98e−2) +	1.3769e−1 (3.44e−2) +	1.7944e−1 (6.92e−2) +	1.9739e−1 (1.04e−2)
UF4	4.7377e−2 (2.52e−3) −	4.0382e−2 (5.11e−4) −	4.1315e−2 (5.78e−4) −	4.0366e−2 (1.02e−3) −	**3.8114e−2 (8.43e−4)**
UF5	5.0388e−1 (1.22e−1) −	2.1315e−1 (5.10e−2) −	2.1893e−1 (5.19e−2) −	2.9034e−1 (1.04e−1) −	**1.7418e−1 (1.72e−2)**
UF6	4.2402e−1 (6.70e−2) −	1.2737e−1 (2.11e−2) −	1.0997e−1 (1.31e−2) =	1.4961e−1 (7.51e−2) =	1.1161e−1 (2.05e−2)
UF7	3.9101e−1 (1.60e−1) −	4.7654e−2 (5.40e−2) −	3.4218e−2 (7.12e−3) −	8.3472e−2 (1.10e−1) −	**2.3055e−2 (4.69e−3)**
UF8	3.3582e−1 (2.67e−1) −	2.3917e−1 (1.63e−2) −	4.8116e−1 (4.01e−2) −	1.5490e−1 (7.71e−2) −	1.1282e−1 (7.78e−2)
UF9	2.7027e−1 (3.88e−2) −	1.6410e−1 (7.39e−2) −	1.8409e−1 (6.05e−2) −	2.0211e−1 (7.48e−2) −	**1.0514e−1 (1.83e−2)**
UF10	7.3023e−1 (1.27e−1) −	3.2186e−1 (3.65e−2) =	3.1234e−1 (5.34e−2) =	3.3096e−1 (5.61e−2) =	3.2815e−1 (6.95e−2)
+/−/=	4/16/2	7/13/2	6/13/3	5/14/3	

Table 4
The comparisons of IGD values obtained by RLMMDE and the 9 compared algorithms on ZDT, DTLZ, UF test suites.

Problem	NSGA-II/SDR	MOPSO	GrEA	IBEA	MaOEA-DDFC	RLMMDE
ZDT1	3.7074e−2 (1.80e−2) −	1.0210e+0 (2.67e−1) −	1.3411e−2 (2.96e−3) −	6.9970e−3 (8.40e−4) =	6.3321e−2 (2.06e−2) −	**6.6623e−3 (8.56e−4)**
ZDT2	5.1276e−2 (4.36e−2) −	1.7738e+0 (3.79e−1) −	2.0501e−2 (1.83e−2) −	1.6404e−1 (1.24e−1) −	3.1553e−1 (1.46e−1) −	**6.2902e−3 (1.19e−3)**
ZDT3	3.2021e−2 (1.27e−2) −	1.0759e+0 (2.32e−1) −	1.8956e−2 (8.60e−3) −	2.1077e−2 (9.61e−3) −	3.5324e−2 (1.72e−2) −	**1.1207e−2 (1.38e−3)**
ZDT4	4.4286e−1 (2.30e−1) =	1.8229e+1 (6.99e+0) −	2.5423e−1 (1.68e−1) +	4.7560e−1 (1.81e−1) =	5.5457e−1 (2.20e−1) =	5.0678e−1 (3.50e−1)
ZDT6	2.1156e−1 (7.26e−2) −	1.1217e+0 (1.98e+0) −	5.2300e−2 (2.47e−2) −	4.4158e−2 (1.78e−2) −	1.6369e−1 (5.93e−2) −	**3.3455e−3 (3.08e−4)**
DTLZ1	4.3002e−2 (2.63e−2) −	8.8036e+0 (2.90e+0) −	9.7754e−2 (8.46e−2) −	1.5967e−1 (2.53e−2) −	3.9904e−2 (1.25e−2) −	3.2247e−2 (5.15e−2)
DTLZ2	4.7229e−1 (1.68e−2) −	1.3602e−1 (1.63e−2) −	6.3339e−2 (7.94e−4) −	7.9169e−2 (2.52e−3) −	5.5607e−2 (5.39e−4) −	5.0509e−2 (3.94e−5)
DTLZ3	4.7919e−1 (1.57e−1) +	9.3254e+1 (5.54e+1) −	**2.6746e−1 (3.43e−1)** +	5.5679e−1 (2.35e−1) +	5.7342e−1 (6.30e−1) +	1.2904e+0 (1.07e+0)
DTLZ4	5.0643e−1 (8.53e−2) −	3.4522e−1 (1.93e−1) −	2.0571e−1 (2.48e−1) −	7.8076e−2 (2.76e−3) −	2.6627e−1 (2.44e−1) −	**5.0792e−2 (8.48e−5)**
DTLZ5	3.0730e−2 (5.87e−3) −	1.1891e−2 (2.53e−3) −	2.1579e−2 (1.16e−3) −	1.5385e−2 (1.24e−3) −	**5.2899e−3 (2.42e−4)** +	8.9480e−3 (1.51e−3)
DTLZ6	6.2604e−2 (1.56e−2) −	2.5145e+0 (9.79e−1) −	2.2281e−2 (1.13e−4) −	2.5021e−2 (3.57e−3) −	2.0513e−2 (1.02e−2) −	1.5235e−2 (5.80e−3)
DTLZ7	8.2668e−2 (4.74e−3) +	3.8571e+0 (1.16e+0) −	9.0432e−2 (5.35e−2) +	**8.2281e−2 (5.59e−2)** +	1.3194e−1 (1.23e−1) +	1.3633e+0 (8.43e−2)
UF1	5.7794e−2 (5.42e−3) −	2.5480e−1 (6.51e−2) −	7.4268e−2 (1.33e−2) −	9.6078e−2 (1.54e−2) −	1.5814e−1 (8.11e−2) −	3.3484e−2 (1.25e−2)
UF2	2.4263e−2 (3.84e−3) +	6.4692e−2 (6.50e−3) −	**2.0744e−2 (4.12e−3)** +	2.8297e−2 (5.03e−3) −	5.8964e−2 (5.85e−3) −	2.5602e−2 (3.96e−2)
UF3	1.4701e−1 (2.79e−2) +	4.0726e−1 (4.14e−2) −	**1.1377e−1 (2.11e−2)** +	2.1951e−1 (5.19e−2) =	1.6226e−1 (1.76e−2) +	1.9739e−1 (1.04e−2)
UF4	4.2330e−2 (4.25e−4) −	7.8349e−2 (1.01e−2) −	4.0099e−2 (5.02e−4) −	4.1602e−2 (9.55e−4) −	5.5702e−2 (2.44e−3) −	**3.8114e−2 (8.43e−4)**
UF5	2.2219e−1 (4.48e−2) −	2.4422e+0 (3.72e−1) −	2.2339e−1 (5.02e−2) −	2.4559e−1 (5.05e−2) −	3.2627e−1 (1.02e−1) −	**1.7418e−1 (1.72e−2)**
UF6	1.1481e−1 (2.67e−2) =	1.2417e+0 (2.82e−1) −	**1.0480e−1 (2.97e−2)** +	1.3158e−1 (1.51e−2) −	1.5530e−1 (6.96e−2) −	1.1161e−1 (2.05e−2)
UF7	2.9112e−2 (3.84e−3) −	2.6779e−1 (7.70e−2) −	3.6791e−2 (9.99e−3) −	4.2388e−2 (9.07e−3) −	7.0612e−2 (3.49e−2) −	**2.3055e−2 (4.69e−3)**
UF8	**1.0463e−1 (3.38e−2)** +	2.8023e−1 (2.62e−2) −	2.2712e−1 (1.20e−1) −	2.6714e−1 (4.54e−2) −	2.8233e−1 (6.04e−2) −	1.1282e−1 (7.78e−2)
UF9	1.7985e−1 (1.04e−1) =	4.9186e−1 (5.66e−2) −	1.2991e−1 (6.40e−2) =	1.7542e−1 (8.12e−2) −	5.8523e−1 (4.66e−2) −	**1.0514e−1 (1.83e−2)**
UF10	3.1501e−1 (4.23e−2) =	1.4117e+0 (1.97e−1) −	**3.0371e−1 (4.96e−2)** =	5.1228e−1 (1.77e−1) −	3.1186e−1 (6.82e−2) =	3.2815e−1 (6.95e−2)
+/−/=	5/13/4	0/22/0	6/14/2	2/17/3	4/16/2	

5.2. Comparison with traditional MOEAs

To verify the performance of the proposed algorithm, the results of RLMMDE are compared with those of the first type of algorithms, i.e., popular traditional MOEAs, for three different types of test suites. The specific test problems and related parameter settings can be found in Section 5.1. Detailed test results are shown in Tables 3–6. In these tables, the IGD and Δ_p values obtained by solving these test suites with RLMMDE and conventional MOEAs are given, respectively.

As shown in Tables 3 and 4 for the experimental results in IGD metrics, the performance of RLMMDE significantly outperforms the other 9 compared algorithms from the last row of the table. On these 22 test problems from 3 different test suites, RLMMDE obtains 10 optimal solutions, while the other algorithms (two tables from left to right) obtain 1, 1, 1, 1, 1, 0, 5, 1 and 1 optimal solutions, respectively. Since RLMMDE uses the environment selection strategy in NSGA-III, one might think that it is the effect of this strategy. However, it can be seen from the results that the proposed algorithm obtains better results than NSGA-III in the case of the same parameter setting. In addition, RLMMDE obtains 9 better results compared to NSGA-III. This may be because the proposed multistrategy and multicrossover DE optimizer plays a good role in balancing the convergence and diversity of these problems. Moreover, the proposed reference point adaptation method further improves the performance of RLMMDE in dealing with problems with irregular PFs. Since NSGA-III uses a single operator and a fixed reference point, this may be a reason that affects its performance.

As seen from Tables 5 and 6 for the experimental results in Δ_p metrics, RLMMDE obtains 9 optimal solutions, while the other algorithms (two tables from left to right) obtain 1, 2, 1, 1, 1, 0, 4, 2 and 1 optimal solutions on these 22 test problems from 3 different test suites, respectively. From the last row of the table, it can be seen that RLMMDE has at least 11 better results than the other compared algorithms. It is worth noting that the IGD and Δ_p indicators obtained very similar results for some test problems. This may be because the Δ_p metric is composed of GD and IGD metrics, which has similar properties to the IGD metric.

In addition, to further present the experimental results of RLMMDE and the compared 9 algorithms on the 22 test problems, the average ranking after the Friedman test and stacked histogram of the mean IGD and Δ_p value ranking are shown in Table 7 and Fig. 4. From Table 7, it can be seen that RLMMDE obtains the best ranking for both IGD and Δ_p values, followed by GrEA, NSGA-II and NSGA-III. From Fig. 4, RLMMDE accounts for the most squares in green, which indicates that it obtains the most optimal results and has the best performance, followed by

Table 5
The comparisons of Δ_p values obtained by RLMMDE and the 9 compared algorithms on ZDT, DTLZ, UF test suites.

Problem	MOEA/D	NSGA-II	NSGA-III	PESA-II	RLMMDE
ZDT1	1.4283e−1 (4.94e−2) −	1.1863e−2 (1.86e−3) −	2.6650e−2 (5.78e−3) −	4.8625e−2 (4.89e−2) −	**6.6623e−3 (8.56e−4)**
ZDT2	5.3334e−1 (6.86e−2) −	2.5805e−2 (3.06e−2) −	4.5561e−2 (2.35e−2) −	1.0753e−1 (8.97e−2) −	**6.2902e−3 (1.19e−3)**
ZDT3	1.4808e−1 (4.03e−2) −	1.1827e−2 (5.52e−3) =	2.5280e−2 (9.90e−3) −	3.3700e−2 (2.70e−2) −	**1.1207e−2 (1.38e−3)**
ZDT4	5.6136e−1 (2.37e−1) =	2.9873e−1 (1.77e−1) +	6.9529e−1 (3.80e−1) =	**1.8555e−1 (1.32e−1)** +	5.7941e−1 (4.16e−1)
ZDT6	8.3669e−2 (2.63e−2) −	5.8508e−2 (3.32e−2) −	3.1366e−1 (1.50e−1) −	3.0204e−2 (2.04e−2) −	**3.3455e−3 (3.08e−4)**
DTLZ1	1.9635e−2 (5.30e−4) +	4.8575e−2 (8.27e−2) −	**1.9519e−2 (5.11e−4)** +	8.9409e−2 (3.51e−1) −	3.3747e−2 (5.87e−2)
DTLZ2	**5.0306e−2 (2.42e−6)** +	6.7041e−2 (2.54e−3) −	5.0329e−2 (2.45e−5) +	6.4213e−2 (3.18e−3) −	5.0509e−2 (3.94e−5)
DTLZ3	7.6559e−1 (9.90e−1) +	**3.5493e−1 (5.62e−1)** +	1.1567e+0 (1.54e+0) =	7.9025e−1 (1.20e+0) +	1.5358e+0 (1.41e+0)
DTLZ4	4.2637e−1 (3.46e−1) =	6.6102e−2 (2.50e−3) −	6.6728e−2 (8.96e−2) −	6.2103e−2 (1.56e−3) −	**5.0792e−2 (8.48e−5)**
DTLZ5	3.1149e−2 (7.85e−5) −	5.4665e−3 (3.13e−4) +	1.1390e−2 (1.23e−3) −	1.1127e−2 (1.31e−3) −	**8.9480e−3 (1.51e−3)**
DTLZ6	3.1244e−2 (5.42e−5) −	**5.6055e−3 (1.66e−4)** +	1.7754e−2 (2.06e−3) −	1.4099e−2 (2.14e−3) =	1.5235e−2 (5.80e−3)
DTLZ7	1.3705e−1 (3.52e−3) +	1.1030e−1 (9.53e−2) +	9.1683e−2 (7.66e−2) +	1.5748e−1 (1.35e−1) +	1.3633e+0 (8.43e−2)
UF1	2.6270e−1 (1.17e−1) −	9.2505e−2 (1.30e−2) −	7.4707e−2 (1.25e−2) −	9.0702e−2 (2.67e−2) −	3.3484e−2 (1.25e−2)
UF2	1.1943e−1 (5.03e−2) −	2.3860e−2 (5.11e−3) +	2.2085e−2 (1.72e−3) +	2.6641e−2 (8.35e−3) −	2.5602e−2 (3.96e−2)
UF3	3.0719e−1 (2.87e−2) −	1.6460e−1 (4.98e−2) +	1.3769e−1 (3.44e−2) +	1.7944e−1 (6.92e−2) +	1.9739e−1 (1.04e−2)
UF4	4.8028e−2 (2.58e−3) −	4.1950e−2 (2.55e−4) −	4.2745e−2 (2.85e−4) −	4.1766e−2 (7.24e−4) −	**3.9100e−2 (4.72e−4)**
UF5	5.0388e−1 (1.22e−1) −	2.1315e−1 (5.10e−2) −	2.1893e−1 (5.19e−2) −	2.9034e−1 (1.04e−1) −	**1.7418e−1 (1.72e−2)**
UF6	4.2402e−1 (6.70e−2) −	1.2737e−1 (2.11e−2) =	1.1093e−1 (1.29e−2) =	1.4961e−1 (7.51e−2) =	1.1657e−1 (1.71e−2)
UF7	3.9101e−1 (1.60e−1) −	4.7654e−2 (5.40e−2) −	3.4218e−2 (7.12e−3) −	8.3472e−2 (1.10e−1) −	**2.3055e−2 (4.69e−3)**
UF8	4.3943e−1 (2.57e−1) −	2.0497e+0 (4.51e−1) −	4.8116e−1 (4.01e−2) −	1.1506e+0 (1.04e+0) −	**1.1282e−1 (7.78e−2)**
UF9	2.7027e−1 (3.88e−2) −	1.6769e−1 (8.40e−2) =	1.8787e−1 (6.24e−2) −	1.5037e+0 (7.33e−1) −	1.4314e−1 (4.16e−2)
UF10	7.3315e−1 (1.16e−1) −	4.1224e+0 (4.50e+0) =	3.1234e−1 (5.34e−2) +	2.7889e+0 (3.36e+0) −	4.5963e−1 (1.95e−1)
+/−/=	4/16/2	7/11/4	6/13/3	4/16/2	

Table 6
The comparisons of Δ_p values obtained by RLMMDE and the 9 compared algorithms on ZDT, DTLZ, UF test suites.

Problem	NSGA-II/SDR	MOPSO	GrEA	IBEA	MaOEA-DDFC	RLMMDE
ZDT1	3.7187e−2 (1.80e−2) −	1.0548e+0 (2.74e−1) −	1.3411e−2 (2.96e−3) −	6.9970e−3 (8.40e−4) =	6.3321e−2 (2.06e−2) −	**6.6623e−3 (8.56e−4)**
ZDT2	5.1685e−2 (4.35e−2) −	1.7738e+0 (3.79e−1) −	2.0501e−2 (1.83e−2) −	1.6404e−1 (1.24e−1) −	3.1553e−1 (1.46e−1) −	**6.2902e−3 (1.19e−3)**
ZDT3	3.2021e−2 (1.27e−2) −	1.2662e+0 (2.41e−1) −	1.8956e−2 (8.60e−3) −	2.1077e−2 (9.61e−3) −	3.5324e−2 (1.72e−2) −	**1.1207e−2 (1.38e−3)**
ZDT4	4.5867e−1 (2.28e−1) =	1.9068e+1 (7.51e+0) −	2.5806e−1 (1.66e−1) +	4.7560e−1 (1.81e−1) =	5.5457e−1 (2.20e−1) =	5.7941e−1 (4.16e−1)
ZDT6	2.1672e−1 (7.08e−2) −	1.3799e+0 (2.03e+0) −	6.0474e−2 (2.95e−2) −	4.5765e−2 (1.91e−2) −	1.8168e−1 (8.01e−2) −	**3.3455e−3 (3.08e−4)**
DTLZ1	4.3002e−2 (2.63e−2) −	2.9872e+1 (6.28e+0) −	2.3280e−1 (3.69e−1) −	1.5967e−1 (2.53e−2) −	2.7968e−1 (4.95e−1) −	3.3747e−2 (5.87e−2)
DTLZ2	4.7229e−1 (1.68e−2) −	1.4273e−1 (2.83e−2) −	6.3339e−2 (7.94e−4) −	7.9169e−2 (2.52e−3) −	5.5607e−2 (5.39e−4) −	5.0509e−2 (3.94e−5)
DTLZ3	4.7919e−1 (1.57e−1) +	2.0839e+2 (1.89e+1) −	6.5298e−1 (1.21e+0) +	5.5679e−1 (2.35e−1) +	7.3576e−1 (7.62e−1) +	1.5358e+0 (1.41e+0)
DTLZ4	5.0643e−1 (8.53e−2) −	3.6750e−1 (1.88e−1) −	2.0571e−1 (2.48e−1) −	7.8076e−2 (2.76e−3) −	2.6627e−1 (2.44e−1) −	**5.0792e−2 (8.48e−5)**
DTLZ5	3.0730e−2 (5.87e−3) −	1.1891e−2 (2.53e−3) −	2.1579e−2 (1.16e−3) −	1.5385e−2 (1.24e−3) −	**5.2899e−3 (2.42e−4)** +	8.9480e−3 (1.51e−3)
DTLZ6	6.2604e−2 (1.56e−2) −	3.2922e+0 (1.01e+0) −	2.2281e−2 (1.13e−4) −	2.5021e−2 (3.57e−3) −	2.0513e−2 (1.02e−2) −	1.5235e−2 (5.80e−3)
DTLZ7	8.2668e−2 (4.74e−3) +	3.8724e+0 (1.15e+0) −	9.0432e−2 (5.35e−2) +	**8.2281e−2 (5.59e−2)** +	1.3194e−1 (1.23e−1) +	1.3633e+0 (8.43e−2)
UF1	**5.7794e−2 (5.42e−3)** −	2.9550e−1 (8.89e−2) −	7.4268e−2 (1.33e−2) −	9.6078e−2 (1.54e−2) −	1.5814e−1 (8.11e−2) −	3.3484e−2 (1.25e−2)
UF2	2.4263e−2 (3.84e−3) +	6.4692e−2 (6.50e−3) −	2.0744e−2 (4.12e−3) +	**2.8297e−2 (5.03e−3)** −	5.8964e−2 (5.85e−3) −	2.5602e−2 (3.96e−2)
UF3	1.4701e−1 (2.79e−2) +	6.8782e−1 (3.40e−1) −	**1.1377e−1 (2.11e−2)** +	2.1951e−1 (5.19e−2) =	1.6226e−1 (1.76e−2) +	1.9739e−1 (1.04e−2)
UF4	4.4174e−2 (3.21e−4) −	8.4758e−2 (1.01e−2) −	4.1234e−2 (3.16e−4) −	4.1606e−2 (9.51e−4) −	5.5702e−2 (2.44e−3) −	**3.9100e−2 (4.72e−4)**
UF5	2.2278e−1 (4.46e−2) −	2.7012e+0 (3.95e−1) −	2.2339e−1 (5.02e−2) −	2.4559e−1 (5.05e−2) −	3.3179e−1 (1.21e−1) −	**1.7418e−1 (1.72e−2)**
UF6	1.2063e−1 (3.04e−2) =	1.5703e+0 (4.88e−1) −	**1.0480e−1 (2.97e−2)** +	1.3158e−1 (1.51e−2) −	1.5895e−1 (7.48e−2) =	1.1657e−1 (1.71e−2)
UF7	2.9112e−2 (3.84e−3) −	3.2093e−1 (9.34e−2) −	3.6791e−2 (9.99e−3) −	4.2388e−2 (9.07e−3) −	7.0612e−2 (3.49e−2) −	**2.3055e−2 (4.69e−3)**
UF8	2.5743e−1 (4.35e−1) −	1.5515e+0 (1.15e+0) −	2.2712e−1 (1.20e−1) −	2.6762e−1 (4.53e−2) −	4.1432e−1 (3.64e−1) −	**1.1282e−1 (7.78e−2)**
UF9	7.9833e−1 (7.46e−1) −	3.8319e+0 (1.33e+0) −	**1.4056e−1 (7.76e−2)** =	1.7542e−1 (8.12e−2) =	5.8523e−1 (4.66e−2) −	1.4314e−1 (4.16e−2)
UF10	2.2633e+0 (8.35e−1) −	1.7378e+0 (2.36e−1) −	**3.0371e−1 (4.96e−2)** +	5.1228e−1 (1.77e−1) =	4.8161e−1 (3.75e−1) =	4.5963e−1 (1.95e−1)
+/−/=	4/16/2	0/22/0	7/14/1	2/15/5	4/15/3	

Table 7
Average Friedman rankings for RLMMDE and the 9 compared algorithm on ZDT, DTLZ, UF test suites.

IGD		Δ_p	
Algorithm	Rank	Algorithm	Rank
RLMMDE	**3.2273**	RLMMDE	**3.0909**
MOEA/D	7.9318	MOEA/D	7.5227
NSGA-II	3.8409	NSGA-II	4.2727
NSGA-III	4.5	NSGA-III	4.3409
PESA-II	5.0227	PESA-II	5.7727
NSGA-II/SDR	5.2727	NSGA-II/SDR	5.4773
MOPSO	9.3636	MOPSO	9.4091
GrEA	3.4545	GrEA	3.4773
IBEA	5.7273	IBEA	5.1818
MaOEA-DDFC	6.6591	MaOEA-DDFC	6.4545

GrEA. Finally, the proposed RLMMDE is more competitive among the compared algorithms both in average ranking and mean IGD and Δ_P values ranking.

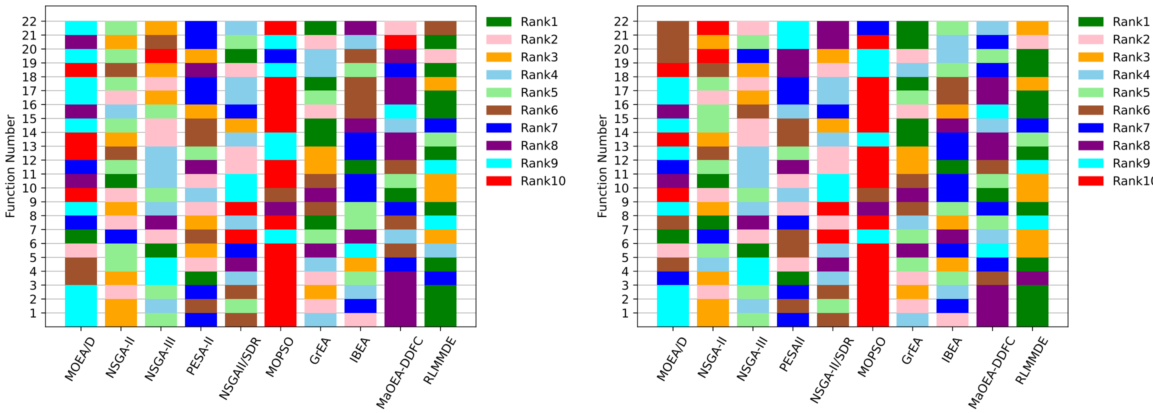
5.3. Comparison with MOEAs with different optimizers

In order to compare the performance of RLMMDE with MOEAs having different types of optimizers, the results of RLMMDE are compared with the second type of algorithms, i.e., MOEA/D-D-DRA, MOEA/D-FRRMAB and IM-MOEA, under three different types of test suites. The specific test problems and related parameter settings are described in Section 5.1. Detailed test results are given in Tables 8 and 9. In these two tables, the IGD and Δ_P values obtained by solving MOPs with RLMMDE and MOEAs with different types of optimizers are given, respectively.

The experimental results for the IGD metrics are shown in Table 8, and it can be seen from the last row of the table that RLMMDE significantly outperforms the other 3 compared algorithms. On 22 test problems, RLMMDE obtains 14 optimal solutions, while MOEA/D-DRA, MOEA/D-FRRMAB and IM-MOEA obtain 2, 4 and 2 optimal solutions, respectively. In general, RLMMDE outperforms the compared algorithms on 15, 15, and 16 test functions, respectively.

As seen from Table 9 for the experimental results in Δ_p metrics, RLMMDE obtains 16 optimal solutions, while MOEA/D-DRA, MOEA/D-FRRMAB and IM-MOEA obtain 2, 3 and 1 optimal solutions respectively on these 22 test problems. As seen in the last row of the table, RLMMDE has at least 16 better results than the other compared algorithms.

Moreover, to further analyze the experimental results of RLMMDE and the three compared algorithms on the 22 test problems,



(a) IGD (b) Δ_p

Fig. 4. Stacked histogram of mean IGD and Δ_p value ranking for RLMMDE and the 9 compared algorithm on ZDT, DTLZ, UF test suites.

Table 8
The comparisons of IGD values obtained by MOEA/D-DRA, MOEA/D-FRRMAB, IM-MOEA and RLMMDE on ZDT, DTLZ, UF test suites.

Problem	MOEA/D-DRA	MOEA/D-FRRMAB	IM-MOEA	RLMMDE
ZDT1	1.9582e−1 (5.86e−2) −	4.0737e−2 (2.01e−2) −	6.2827e−2 (1.00e−2) −	**6.6623e−3 (8.56e−4)**
ZDT2	2.9388e−1 (9.37e−2) −	1.1988e−1 (5.75e−2) −	6.5957e−2 (1.03e−2) −	**6.2902e−3 (1.19e−3)**
ZDT3	3.1706e−1 (4.44e−2) −	2.0010e−1 (6.37e−2) −	7.8062e−2 (1.36e−2) −	**1.1207e−2 (1.38e−3)**
ZDT4	1.0728e+1 (6.90e+0) −	9.9357e+0 (4.58e+0) −	**1.0500e−2 (1.16e−3)** +	5.0678e−1 (3.50e−1)
ZDT6	1.0743e−1 (1.25e−1) −	4.4955e−3 (4.41e−3) =	1.5141e+0 (7.36e−2) −	**3.3455e−3 (3.08e−4)**
DTLZ1	6.8937e−1 (1.17e+0) −	5.7660e−1 (1.46e+0) −	3.6296e+0 (1.02e+0) −	**3.2247e−2 (5.15e−2)**
DTLZ2	6.9276e−2 (6.51e−4) −	6.9502e−2 (8.21e−4) −	9.1352e−2 (4.49e−3) −	**5.0509e−2 (3.94e−5)**
DTLZ3	2.9546e+1 (2.44e+1) −	1.4456e+1 (1.92e+1) −	5.8190e+1 (1.12e+1) −	**1.2904e+0 (1.07e+0)**
DTLZ4	2.2957e−1 (1.96e−1) −	1.2118e−1 (7.09e−2) −	7.6259e−2 (3.77e−3) −	**5.0792e−2 (8.48e−5)**
DTLZ5	1.2086e−2 (1.20e−4) −	1.2175e−2 (1.83e−4) −	2.1688e−2 (2.70e−3) −	**8.9480e−3 (1.51e−3)**
DTLZ6	**1.2259e−2 (6.79e−5)** =	1.2315e−2 (4.05e−5) =	4.8608e+0 (1.46e−1) −	1.5235e−2 (5.80e−3)
DTLZ7	2.1661e−1 (7.21e−2) +	**1.7913e−1 (1.50e−2)** +	3.4388e−1 (4.62e−2) +	1.3633e+0 (8.43e−2)
UF1	**1.8715e−3 (5.93e−4)** +	5.2806e−3 (1.04e−2) +	5.1621e−2 (8.53e−3) −	3.3484e−2 (1.25e−2)
UF2	3.4021e−3 (1.30e−3) +	**2.9022e−3 (8.32e−4)** +	2.1601e−2 (5.22e−3) +	2.5602e−2 (3.96e−2)
UF3	2.5675e−2 (3.44e−2) +	**1.7344e−2 (1.40e−2)** +	7.1777e−2 (8.82e−3) +	1.9739e−1 (1.04e−2)
UF4	6.1359e−2 (4.34e−3) −	5.0825e−2 (2.30e−3) −	5.8686e−2 (2.26e−3) −	**3.8114e−2 (8.43e−4)**
UF5	5.6750e−1 (1.46e−1) −	5.4904e−1 (1.40e−1) −	5.4523e−1 (8.92e−2) −	**1.7418e−1 (1.72e−2)**
UF6	3.8481e−1 (9.66e−2) −	4.3097e−1 (1.04e−1) −	1.4395e−1 (3.44e−2) −	**1.1161e−1 (2.05e−2)**
UF7	2.7869e−1 (2.82e−1) =	1.9414e−1 (2.56e−1) −	4.2634e−2 (6.00e−2) =	**2.3055e−2 (4.69e−3)**
UF8	1.2798e−1 (6.22e−2) =	**1.0797e−1 (4.66e−2)** =	1.7016e−1 (5.79e−2) −	1.1282e−1 (7.78e−2)
UF9	1.7851e−1 (2.17e−2) −	1.7449e−1 (4.08e−2) −	1.6441e−1 (5.67e−2) −	**1.0514e−1 (1.83e−2)**
UF10	4.9053e−1 (6.58e−2) −	8.4947e−1 (1.90e−1) −	**2.3729e−1 (5.72e−4)** +	3.2815e−1 (6.95e−2)
+/−/=	4/15/3	4/15/3	5/16/1	

the average ranking after the Friedman test and stacked histogram of mean IGD and Δ_p values ranking are shown in Table 10 and Fig. 5. As seen from Table 10, RLMMDE ranks first for both IGD and Δ_p values, followed by MOEA/D-FRMAB. As shown in Fig. 5, RLMMDE accounts for the most squares in green, which indicates that it obtains the most optimal results and has the best performance. Finally, the proposed RLMMDE is more competitive among the compared algorithms in terms of average ranking, mean IGD and Δ_p values ranking.

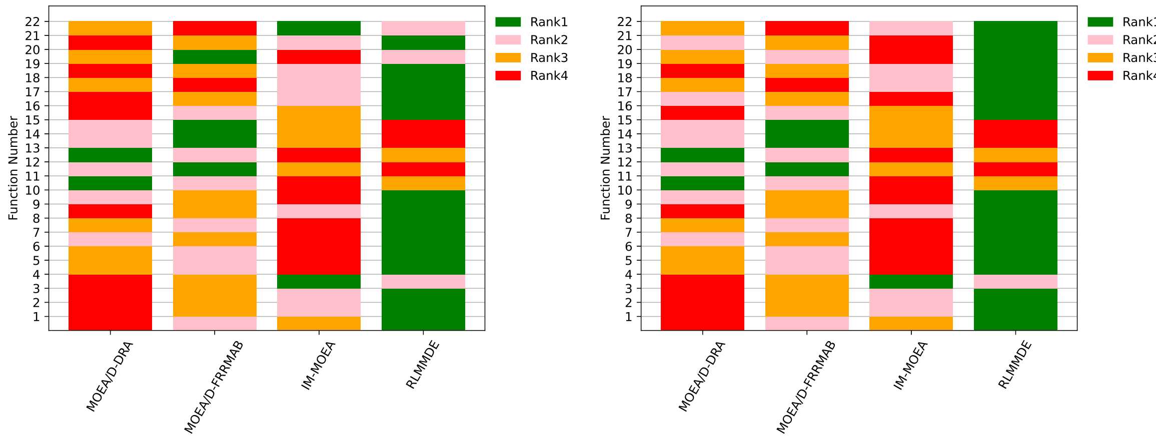
5.4. Comparison with MOEAs with reference point adaptation

In this section, three MOEAs with reference point adaptation methods, namely ANSGA-III, RVEA, RVEA* and MOEA/D-AWA, are used for comparison with the proposed algorithm in three different types of test suites. The specific test problems and the related parameter settings can be found in Section 5.1. Detailed test results are given in Tables 11 and 12. In these tables, the IGD and Δ_p values obtained by solving these suites with RLMMDE and conventional MOEAs are given, respectively.

As shown in Table 11, RLMMDE performs significantly better than the other 4 compared algorithms in the experimental results for the IGD metric. On these 22 test problems, RLMMDE obtains 12 optimal solutions, while the other algorithms (from left to right of the table) obtain 1, 1, 5 and 2 optimal solutions, respectively. Overall, RLMMDE outperforms the compared algorithms on the results of 14, 18, 14 and 17 test instances, respectively.

As seen in Table 12 for the experimental results of the Δ_p metric, RLMMDE obtains 12 optimal solutions, while the other algorithms (table from left to right) obtain 3, 1, 3, and 3 optimal solutions on these 22 test problems, respectively. As seen from the last row of the table, RLMMDE has at least 14 better results than the other compared algorithms.

In addition, to further demonstrate the experimental results of RLMMDE and the four compared algorithms on the 22 test problems, the average ranking after the Friedman test and stacked histogram of the IGD and Δ_p ranking are shown in Table 13 and Fig. 6. From Table 13, it can be seen that RLMMDE obtained the best ranking for IGD and Δ_p values, followed by RVEA* on the average IGD ranking and ANSGA-III on the average Δ_p ranking. From Fig. 6, RLMMDE has the most green squares, indicating that it obtains the most optimal results and has the best performance. Finally, the proposed RLMMDE is more competitive among the compared algorithms, both in the average ranking and mean IGD and Δ_p values ranking.



(a) IGD (b) Δ_p

Fig. 5. Stacked histogram of mean IGD and Δ_p value ranking for MOEA/D-DRA, MOEA/D-FRRMAB, IM-MOEA and RLMMDE on ZDT, DTLZ, UF test suites.

Table 9
The comparisons of Δ_p values obtained by MOEA/D-DRA, MOEA/D-FRRMAB, IM-MOEA and RLMMDE on ZDT, DTLZ, UF test suites.

Problem	MOEA/D-DRA	MOEA/D-FRRMAB	IM-MOEA	RLMMDE
ZDT1	3.5988e−1 (1.12e−1) −	5.9976e−2 (3.74e−2) −	6.3420e−2 (9.94e−3) −	**6.6623e−3 (8.56e−4)**
ZDT2	4.5765e−1 (2.11e−1) −	1.4455e−1 (7.73e−2) −	7.0788e−2 (1.09e−2) −	**6.2902e−3 (1.19e−3)**
ZDT3	4.6037e−1 (1.23e−1) −	2.0662e−1 (7.16e−2) −	7.8062e−2 (1.36e−2) −	**1.1207e−2 (1.38e−3)**
ZDT4	3.1691e+1 (9.49e+0) −	2.3934e+1 (7.18e+0) −	**7.7083e−2 (9.99e−2)** +	5.7941e−1 (4.16e−1)
ZDT6	2.7936e−1 (2.79e−1) −	4.0850e−2 (3.64e−2) −	1.5660e+0 (8.35e−2) −	**3.3455e−3 (3.08e−4)**
DTLZ1	2.6896e+0 (2.54e+0) −	1.0791e+0 (2.41e+0) −	4.3575e+1 (5.09e+0) −	**3.3747e−2 (5.87e−2)**
DTLZ2	6.9276e−2 (6.51e−4) −	6.9502e−2 (8.21e−4) −	9.3459e−2 (7.54e−3) −	**5.0509e−2 (3.94e−5)**
DTLZ3	6.7426e+1 (3.81e+1) −	4.7362e+1 (3.79e+1) −	2.8362e+2 (3.43e+1) −	**1.5358e+0 (1.41e+0)**
DTLZ4	2.2957e−1 (1.96e−1) −	1.2118e−1 (7.09e−2) −	7.6259e−2 (3.77e−3) −	**5.0792e−2 (8.48e−5)**
DTLZ5	1.2086e−2 (1.20e−4) −	1.2175e−2 (1.83e−4) −	2.1884e−2 (2.83e−3) −	**8.9480e−3 (1.51e−3)**
DTLZ6	**1.2259e−2 (6.79e−5)** =	1.2315e−2 (4.05e−5) =	5.7976e+0 (1.07e−1) −	1.5235e−2 (5.80e−3)
DTLZ7	2.1661e−1 (7.21e−2) +	**1.7913e−1 (1.50e−2)** +	3.4388e−1 (4.62e−2) +	1.3633e+0 (8.43e−2)
UF1	**1.8883e−3 (6.55e−4)** +	5.4515e−3 (1.03e−2) +	5.1621e−2 (8.53e−3) −	3.3484e−2 (1.25e−2)
UF2	3.4511e−3 (1.28e−3) +	**3.0705e−3 (8.74e−4)** +	2.1601e−2 (5.22e−3) +	2.5602e−2 (3.96e−2)
UF3	2.5829e−2 (3.44e−2) +	**1.7344e−2 (1.40e−2)** +	7.9066e−2 (1.78e−2) +	1.9739e−1 (1.04e−2)
UF4	6.5154e−2 (4.16e−3) −	5.4105e−2 (2.42e−3) −	6.0072e−2 (2.34e−3) −	**3.9100e−2 (4.72e−4)**
UF5	5.7462e−1 (1.37e−1) −	6.9840e−1 (2.75e−1) −	7.7792e−1 (1.29e−1) −	**1.7418e−1 (1.72e−2)**
UF6	3.8481e−1 (9.66e−2) −	4.3097e−1 (1.04e−1) −	3.3290e−1 (1.20e−1) −	**1.1657e−1 (1.71e−2)**
UF7	2.7869e−1 (2.82e−1) =	1.9416e−1 (2.56e−1) −	4.2757e−2 (5.99e−2) =	**2.3055e−2 (4.69e−3)**
UF8	1.3822e−1 (6.09e−2) −	1.2010e−1 (6.16e−2) =	3.1490e−1 (1.31e−1) −	**1.1282e−1 (7.78e−2)**
UF9	3.8748e−1 (1.72e−1) −	4.3609e−1 (1.58e−1) −	5.2631e−1 (1.41e−1) −	**1.4314e−1 (4.16e−2)**
UF10	1.4114e+0 (5.84e−1) −	2.0360e+0 (5.71e−1) −	4.7792e−1 (3.63e−2) −	**4.5963e−1 (1.95e−1)**
+/−/=	4/16/2	4/16/2	4/17/1	

Table 10
Average Friedman rankings for MOEA/D-DRA, MOEA/D-FRRMAB, IM-MOEA and RLMMDE on ZDT, DTLZ, UF test suites.

IGD		Δ_p	
Algorithm	Rank	Algorithm	Rank
RLMMDE	**1.7273**	RLMMDE	**1.6364**
MOEA/D-DRA	3.0227	MOEA/D-DRA	2.8409
MOEA/D-FRRMAB	2.3864	MOEA/D-FRRMAB	2.4318
IM-MOEA	2.8636	IM-MOEA	3.0909

5.5. Performance on other test instances

To further evaluate the performance of RLMMDE, four classical MOEAs, namely NSGA-II [45], PESA-II [79], IBEA [80], MOEA/D-DRA [81], in WFG [68], LSMOP [69], SDTLZ1, SDTLZ2, CDTLZ2 [60] and IDTLZ1, IDTLZ2 [50] test instances are compared. The specific test problems and the related parameter settings can be found in Section 5.1. Detailed test results are given in Table 14. In the table, the IGD values obtained by RLMMDE and the compared algorithms for these test problems are given.

As shown in Table 14, RLMMDE significantly outperformed the other four comparison algorithms in the experimental IGD metric results. On these 23 test problems, RLMMDE obtains 9 optimal solutions, while the other algorithms obtain 9, 0, 3, and 2 optimal solutions, respectively. In general, RLMMDE significantly outperforms the other compared algorithms in terms of results for 14, 15, 15, and 14 test instances. It is worth noting that the proposed algorithm performs better on the WFG test suite but relatively worse on the LSMOP test suite.

5.6. Effectiveness analysis of the components in RLMMDE

To verify the multistrategy and multicrossover DE optimizer and adaptive reference point activation mechanism based on RL, three different variants are proposed for comparison with RLMMDE. In addition, three MOEAs with the reference point adaptation method are used to verify the proposed reference point adaptation method.

Table 11
The comparisons of IGD values obtained by ANSGA-III, RVEA, RVEA*, MOEA/D-AWA and RLMMDE on ZDT, DTLZ, UF test suites.

Problem	ANSGA-III	RVEA	RVEA*	MOEA/D-AWA	RLMMDE
ZDT1	2.8122e−2 (8.63e−3) −	1.2783e−1 (2.06e−2) −	1.8070e−2 (1.65e−2) −	2.9287e−2 (1.04e−2) −	**6.6623e−3 (8.56e−4)**
ZDT2	6.4231e−2 (6.11e−2) −	1.7080e−1 (3.78e−2) −	3.2645e−1 (2.93e−1) −	1.1915e−1 (1.55e−1) −	**6.2902e−3 (1.19e−3)**
ZDT3	2.6591e−2 (9.20e−3) −	1.6504e−1 (2.72e−2) −	2.0684e−2 (1.18e−2) −	2.4911e−2 (1.20e−2) −	**1.1207e−2 (1.38e−3)**
ZDT4	7.3899e−1 (2.58e−1) −	1.4071e+0 (6.39e−1) −	**2.9117e−1 (2.31e−1)** +	4.7006e−1 (2.08e−1) =	5.0678e−1 (3.50e−1)
ZDT6	2.8452e−1 (1.16e−1) −	3.4203e−1 (7.31e−2) −	2.5802e−2 (9.56e−3) −	2.6599e−2 (1.50e−2) −	**3.3455e−3 (3.08e−4)**
DTLZ1	2.4722e−2 (7.75e−3) +	2.0128e−2 (3.77e−3) +	2.3364e−2 (1.63e−2) +	**1.9879e−2 (4.68e−4)** +	3.2247e−2 (5.15e−2)
DTLZ2	5.2313e−2 (1.47e−3) −	**5.0370e−2 (1.46e−4)** +	5.1158e−2 (2.71e−4) −	5.0890e−2 (2.92e−4) −	5.0509e−2 (3.94e−5)
DTLZ3	4.0948e−1 (4.69e−1) +	8.7987e−1 (7.19e−1) =	3.3367e−1 (5.74e−1) +	**1.9108e−1 (2.92e−1)** +	1.2904e+0 (1.07e+0)
DTLZ4	1.0050e−1 (1.49e−1) −	6.6760e−2 (8.97e−2) −	2.6053e−1 (2.71e−1) −	2.1559e−1 (2.34e−1) −	**5.0792e−2 (8.48e−5)**
DTLZ5	9.5252e−3 (8.61e−4) −	7.4255e−2 (1.57e−2) −	**8.8262e−3 (2.37e−3)** =	9.5057e−3 (2.34e−4) −	8.9480e−3 (1.51e−3)
DTLZ6	1.2826e−2 (1.45e−3) =	7.1538e−2 (1.07e−2) −	**8.8044e−3 (2.06e−3)** +	9.5833e−3 (3.44e−4) +	1.5235e−2 (5.80e−3)
DTLZ7	6.9291e−2 (3.32e−3) +	1.0478e−1 (4.84e−3) +	**6.4776e−2 (5.38e−2)** +	1.3857e−1 (7.67e−2) +	1.3633e+0 (8.43e−2)
UF1	7.2923e−2 (1.59e−2) −	8.0468e−2 (6.92e−3) −	9.8988e−2 (2.49e−2) −	1.2394e−1 (1.05e−1) −	**3.3484e−2 (1.25e−2)**
UF2	**2.3837e−2 (2.65e−3)** +	5.5851e−2 (5.05e−3) −	3.7873e−2 (7.68e−3) −	4.0476e−2 (2.98e−2) −	2.5602e−2 (3.96e−2)
UF3	**1.4485e−1 (3.58e−2)** +	2.5338e−1 (4.29e−2) −	2.0641e−1 (5.45e−2) =	3.4325e−1 (6.57e−2) −	1.9739e−1 (1.04e−2)
UF4	4.1067e−2 (6.21e−4) −	8.5777e−2 (3.88e−3) −	4.1481e−2 (1.21e−3) −	4.8880e−2 (6.41e−3) −	**3.8114e−2 (8.43e−4)**
UF5	2.0606e−1 (3.76e−2) −	2.3487e−1 (3.33e−2) −	2.7462e−1 (9.05e−2) −	5.2548e−1 (1.11e−1) −	**1.7418e−1 (1.72e−2)**
UF6	1.1268e−1 (1.74e−2) =	1.5195e−1 (6.41e−2) −	2.0685e−1 (1.32e−1) −	4.5666e−1 (2.19e−1) −	**1.1161e−1 (2.05e−2)**
UF7	3.3528e−2 (7.07e−3) −	4.7626e−2 (6.88e−3) −	1.3676e−1 (1.16e−1) −	2.1006e−1 (1.52e−1) −	**2.3055e−2 (4.69e−3)**
UF8	4.8701e−1 (3.28e−2) −	2.9666e−1 (2.59e−2) −	2.1570e−1 (1.87e−2) −	1.1569e−1 (8.12e−2) −	**1.1282e−1 (7.78e−2)**
UF9	2.0902e−1 (4.75e−2) −	2.3839e−1 (9.42e−2) −	1.9795e−1 (6.64e−2) −	2.2874e−1 (3.49e−2) −	**1.0514e−1 (1.83e−2)**
UF10	3.2916e−1 (4.36e−2) =	3.7754e−1 (2.04e−2) −	**2.9840e−1 (5.26e−2)** =	5.1075e−1 (7.34e−2) −	3.2815e−1 (6.95e−2)
+/−/=	5/14/3	3/18/1	5/14/3	4/17/1	

Table 12
The comparisons of Δ_p values obtained by ANSGA-III, RVEA, RVEA*, MOEA/D-AWA and RLMMDE on ZDT, DTLZ, UF test suites.

Problem	ANSGA-III	RVEA	RVEA*	MOEA/D-AWA	RLMMDE
ZDT1	2.8191e−2 (8.68e−3) −	1.3133e−1 (2.03e−2) −	1.8070e−2 (1.65e−2) −	2.9287e−2 (1.04e−2) −	**6.6623e−3 (8.56e−4)**
ZDT2	6.4350e−2 (6.11e−2) −	1.7622e−1 (3.86e−2) −	3.2646e−1 (2.93e−1) −	1.1915e−1 (1.55e−1) −	**6.2902e−3 (1.19e−3)**
ZDT3	2.6591e−2 (9.20e−3) −	1.7005e−1 (3.36e−2) −	2.0684e−2 (1.18e−2) −	2.4911e−2 (1.20e−2) −	**1.1207e−2 (1.38e−3)**
ZDT4	7.7157e−1 (2.57e−1) −	1.4715e+0 (6.81e−1) −	**2.9117e−1 (2.31e−1)** +	4.7006e−1 (2.08e−1) =	5.7941e−1 (4.16e−1)
ZDT6	3.0727e−1 (1.22e−1) −	3.6121e−1 (6.89e−2) −	2.7368e−2 (9.86e−3) −	2.7303e−2 (1.64e−2) −	**3.3455e−3 (3.08e−4)**
DTLZ1	1.0580e−1 (2.87e−1) −	2.2182e−2 (1.50e−2) +	3.0937e−2 (5.78e−2) +	**1.9879e−2 (4.68e−4)** +	3.3747e−2 (5.87e−2)
DTLZ2	5.2313e−2 (1.47e−3) −	**5.0370e−2 (1.46e−4)** +	5.1158e−2 (2.71e−4) −	5.0890e−2 (2.92e−4) −	5.0509e−2 (3.94e−5)
DTLZ3	7.1648e−1 (9.68e−1) +	1.0525e+0 (8.52e−1) =	3.5694e−1 (6.21e−1) +	**1.9267e−1 (2.97e−1)** +	1.5358e+0 (1.41e+0)
DTLZ4	1.0050e−1 (1.49e−1) −	6.6760e−2 (8.97e−2) −	2.6053e−1 (2.71e−1) −	2.1559e−1 (2.34e−1) −	**5.0792e−2 (8.48e−5)**
DTLZ5	9.5252e−3 (8.61e−4) −	1.0004e−1 (2.91e−2) −	**8.8262e−3 (2.37e−3)** =	9.5057e−3 (2.34e−4) −	8.9480e−3 (1.51e−3)
DTLZ6	1.3420e−2 (3.23e−3) =	1.2265e−1 (5.54e−2) −	1.1189e−2 (6.90e−3) +	**9.5833e−3 (3.44e−4)** +	1.5235e−2 (5.80e−3)
DTLZ7	6.9291e−2 (3.32e−3) +	1.0478e−1 (4.84e−3) +	**6.4776e−2 (5.38e−2)** +	1.3857e−1 (7.67e−2) +	1.3633e+0 (8.43e−2)
UF1	7.2923e−2 (1.59e−2) −	8.3033e−2 (1.69e−2) −	9.8988e−2 (2.49e−2) −	1.2394e−1 (1.05e−1) −	**3.3484e−2 (1.25e−2)**
UF2	**2.3837e−2 (2.65e−3)** +	5.5851e−2 (5.05e−3) −	3.7873e−2 (7.68e−3) −	4.0476e−2 (2.98e−2) −	2.5602e−2 (3.96e−2)
UF3	**1.4505e−1 (3.55e−2)** +	2.5338e−1 (4.29e−2) −	2.0641e−1 (5.45e−2) =	3.4325e−1 (6.57e−2) −	1.9739e−1 (1.04e−2)
UF4	4.2795e−2 (3.14e−4) −	9.1690e−2 (4.10e−3) −	4.2412e−2 (5.62e−4) −	4.8880e−2 (6.41e−3) −	**3.9100e−2 (4.72e−4)**
UF5	2.0611e−1 (3.76e−2) −	2.3583e−1 (3.28e−2) −	2.7462e−1 (9.05e−2) −	5.2548e−1 (1.11e−1) −	**1.7418e−1 (1.72e−2)**
UF6	1.1268e−1 (1.74e−2) =	1.7029e−1 (7.39e−2) −	2.0685e−1 (1.32e−1) −	4.5666e−1 (2.19e−1) −	**1.1657e−1 (1.71e−2)**
UF7	3.3528e−2 (7.07e−3) −	4.7626e−2 (6.88e−3) −	1.3676e−1 (1.16e−1) −	2.1006e−1 (1.52e−1) −	**2.3055e−2 (4.69e−3)**
UF8	4.8701e−1 (3.28e−2) −	2.3084e+0 (6.41e−1) −	3.8308e−1 (1.05e−1) −	1.1569e−1 (8.12e−2) −	**1.1282e−1 (7.78e−2)**
UF9	2.1000e−1 (4.65e−2) −	6.9874e−1 (3.91e−1) −	2.0067e−1 (6.39e−2) −	2.2874e−1 (3.49e−2) −	**1.4314e−1 (4.16e−2)**
UF10	**3.2916e−1 (4.36e−2)** +	2.9549e+0 (1.12e+0) −	5.9614e−1 (3.86e−1) =	5.1075e−1 (7.34e−2) −	4.5963e−1 (1.95e−1)
+/−/=	5/15/2	3/18/1	5/14/3	4/17/1	

Table 13
Average Friedman rankings for ANSGA-III, RVEA, RVEA*, MOEA/D-AWA and RLMMDE on ZDT, DTLZ, UF test suites.

IGD		Δ_p	
Algorithm	Rank	Algorithm	Rank
RLMMDE	**2**	RLMMDE	**2**
ANSGA-III	2.9091	ANSGA-III	2.8182
RVEA	3.8636	RVEA	3.9545
RVEA*	2.7273	RVEA*	2.9091
MOEA/D-AWA	3.5	MOEA/D-AWA	3.3182

5.6.1. Validation of the optimizer and activation mechanism

Three different variants are compared with RLMMDE to verify the optimizer and adaptive reference point activation mechanism. The three variations are described as follows:

- Variant 1: The offspring generation uses the GA optimizer in NSGA-II [45].
- Variant 2: The offspring generation uses the PSO optimizer in MOPSO [78].
- Variant 3: As in RVEA [83], the frequency of employing the reference point adaptation is controlled by a parameter *fr*. *fr* is set to 0.1.

The specific test instances and related parameter settings are the same as those described in Section 5.1. Detailed test results can be found in Table 15. The IGD metric is used to test the performance of the three variants and RLMMDE. The results for each test instance are presented as the mean and standard deviation after 30 independent runs.

As shown in Table 15, RLMMDE obviously obtains better performance compared to variants 1 and 2. In general, RLMMDE obtains 15 and 22 better results than variants 1 and 2, respectively. In terms of the number of obtained best IGD values, the proposed algorithm obtains 14, while variant 1 and variant 2 obtain only 5 and 0, respectively. This shows the effectiveness of the proposed multistrategy and multicrossover DE optimizer.

Table 14
The comparisons of IGD values obtained by IBEA, NSGA-II, PESA-II, MOEA/D-DRA and RLMMDE on WFG, LSMOP and DTLZ variants test instances.

Problem	IBEA	NSGA-II	PESA-II	MOEA/D-DRA	RLMMDE
WFG1	**1.8982e−1 (1.24e−2)** +	3.5767e−1 (4.06e−2) +	4.4296e−1 (7.49e−2) +	1.4267e+0 (8.33e−2) −	1.1447e+0 (1.26e−1)
WFG2	2.9019e−1 (6.35e−3) −	2.1826e−1 (1.15e−2) −	1.8815e−1 (8.88e−3) −	3.4260e−1 (2.14e−2) −	**1.5329e−1 (2.12e−3)**
WFG3	**3.7351e−2 (1.58e−3)** +	9.4376e−2 (1.59e−2) +	4.3148e−1 (2.03e−1) −	1.5768e−1 (2.42e−2) +	3.1170e−1 (8.37e−2)
WFG4	3.1209e−1 (1.35e−2) −	2.6403e−1 (9.72e−3) −	2.8178e−1 (1.24e−2) −	3.6959e−1 (1.45e−2) −	**2.1125e−1 (2.11e−3)**
WFG5	3.1522e−1 (1.05e−2) −	2.7175e−1 (9.44e−3) −	2.7680e−1 (1.01e−2) −	3.1757e−1 (6.58e−3) −	**2.1836e−1 (1.70e−3)**
WFG6	3.2221e−1 (1.42e−2) −	3.0707e−1 (1.50e−2) −	3.1151e−1 (1.50e−2) −	4.1024e−1 (3.60e−2) −	**2.4223e−1 (2.09e−2)**
WFG7	3.1475e−1 (1.22e−2) −	2.7530e−1 (9.77e−3) −	2.8199e−1 (1.38e−2) −	3.3710e−1 (7.89e−3) −	**2.1768e−1 (2.29e−3)**
WFG8	3.3463e−1 (8.86e−3) −	3.5897e−1 (1.29e−2) −	3.7158e−1 (1.34e−2) −	4.2538e−1 (5.49e−2) −	**2.8552e−1 (5.60e−3)**
WFG9	2.9099e−1 (9.21e−3) −	2.7703e−1 (2.24e−2) −	2.7613e−1 (3.19e−2) −	3.1534e−1 (1.97e−2) −	**2.5304e−1 (5.19e−2)**
LSMOP1	**2.4096e+0 (3.77e−1)** +	3.4375e+0 (3.76e−1) =	3.5548e+0 (4.31e−1) =	2.4422e+0 (3.38e−1) +	3.7313e+0 (1.07e+0)
LSMOP2	**8.2495e−2 (1.20e−3)** +	1.0652e−1 (3.31e−3) −	9.6763e−2 (2.86e−3) −	9.2170e−2 (2.11e−3) +	9.5639e−2 (2.07e−4)
LSMOP3	**9.6073e+0 (5.94e+0)** +	1.2138e+1 (1.81e+0) +	1.1108e+1 (1.39e+0) +	1.1443e+1 (1.41e+0) +	1.3436e+1 (9.71e−1)
LSMOP4	**1.9502e−1 (4.31e−3)** +	2.7800e−1 (5.90e−3) +	2.7048e−1 (5.40e−3) +	2.4835e−1 (6.01e−3) +	2.8508e−1 (2.70e−3)
LSMOP5	**4.4014e+0 (7.36e−1)** +	8.3581e+0 (8.65e−1) +	1.0486e+1 (1.03e+0) =	4.7523e+0 (6.21e−1) +	1.2748e+1 (3.91e+0)
LSMOP6	**2.0146e+2 (2.06e+2)** =	7.7103e+2 (8.28e+2) −	3.3793e+3 (1.62e+3) −	9.6076e+2 (7.83e+2) −	4.1084e+2 (5.28e+2)
LSMOP7	9.5382e+1 (1.48e+2) −	1.5974e+0 (4.26e−2) −	2.2995e+3 (1.51e+3) −	**1.3589e+0 (5.50e−2)** +	1.4575e+0 (4.59e−2)
LSMOP8	**5.8481e−1 (6.04e−2)** +	9.5582e−1 (6.01e−2) −	2.2030e+0 (7.65e−1) −	6.1387e−1 (3.25e−2) +	9.1505e−1 (8.23e−2)
LSMOP9	4.6222e+0 (4.46e−1) −	6.5480e+0 (9.11e−1) −	1.1891e+1 (2.25e+0) −	**1.3665e+0 (2.22e−1)** +	3.2390e+0 (5.02e−1)
IDTLZ1	2.4461e−1 (7.83e−3) −	2.7502e−2 (1.35e−3) +	**2.3457e−2 (1.13e−3)** +	4.7059e−2 (2.24e−2) −	4.6081e−2 (9.95e−2)
IDTLZ2	9.3745e−2 (7.35e−3) −	6.6673e−2 (2.06e−3) +	**6.1806e−2 (3.01e−3)** +	8.0259e−2 (1.01e−3) −	7.4380e−2 (4.27e−3)
SDTLZ1	4.1325e−1 (6.00e−2) −	8.7431e−2 (1.00e−1) −	**5.8598e−2 (4.66e−3)** +	6.5816e−1 (6.85e−1) −	7.5892e−2 (1.00e−1)
SDTLZ2	1.8235e−1 (7.31e−3) −	1.4636e−1 (5.56e−3) −	1.4268e−1 (4.76e−3) −	2.1436e−1 (1.74e−3) −	**1.2003e−1 (1.48e−4)**
CDTLZ2	4.5976e−2 (2.96e−3) −	4.7650e−2 (3.69e−3) −	4.5275e−2 (2.61e−3) −	9.9103e−2 (8.75e−4) −	**4.3098e−2 (4.18e−4)**
+/−/=	8/14/1	7/15/1	6/15/2	9/14/0	

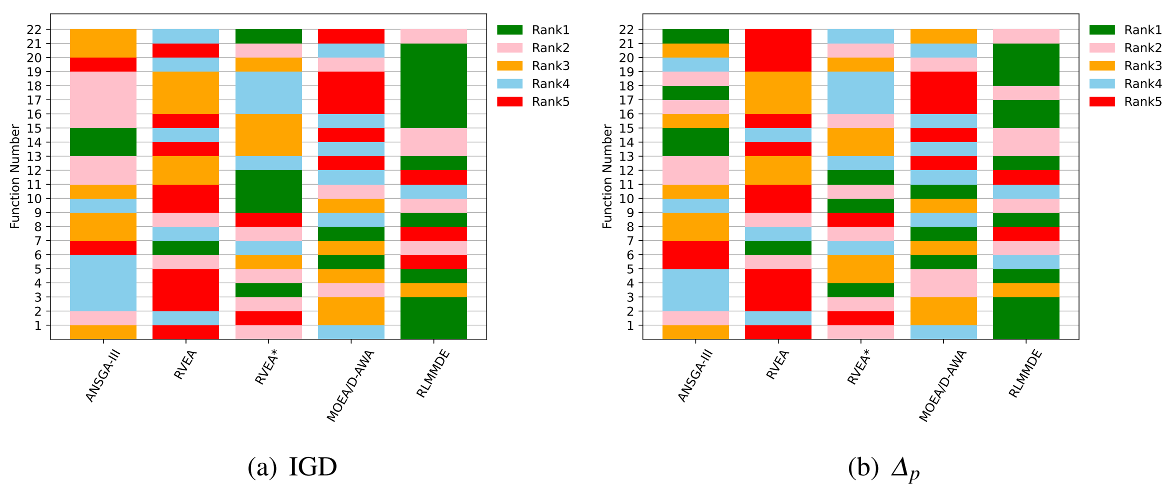


Fig. 6. Stacked histogram of mean IGD and Δ_p value ranking for ANSGA-III, RVEA, RVEA*, MOEA/D-AWA and RLMMDE on ZDT, DTLZ, UF test suites.

In addition, the proposed adaptive reference point activation mechanism based on RL achieves 8 better results than variant 3, which set a frequency to control reference point adaptation on all 22 test instances. In conclusion, the optimizer and activation mechanism proposed in this paper are necessary and effective.

5.6.2. Validation of the reference point adaptation method

To verify the reference point adaptation method, one variant and two MOEAs, ANSGA-III [50] and MOEA/D-AWA [51], are used for comparison with RLMMDE. The one variant and two MOEAs are described as follows:

- Variant 4: The reference point adaptation method in RLMMDE are removed.
- ANSGA-III: The parameter settings are the same as in the original paper.
- MOEA/D-AWA: The parameter settings are the same as in the original paper.

It is worth noting that the starting point of the reference point adaptive method proposed in this paper is to improve the ability of the algorithm to handle irregular fronts. Thus, ZDT3, DTLZ5 and DTLZ6 with irregular PFs are used to test the performance. The three test instances and related parameter settings are the same as those described in Section 5.1. Detailed test results can be found in Table 16. The IGD metric is used to test the performance of variant 4, ANSGA-III, MOEA/D-AWA and RLMMDE.

From Table 16, it can be seen that RLMMDE obtains significantly better results on DTLZ5 and DTLZ6 compared to variant 4 which does not use the reference point adaptation method. Compared to ANSGA-III and MOEA/D-AWA, RLMMDE obtains significantly better results on ZDT3 and DTLZ5. These experimental results show the effectiveness of the proposed reference point adaptive method. It also shows that the proposed reference point adaptation method improves the performance of the algorithm in dealing with irregular frontier problems.

5.7. Multiobjective mixed-variable optimization problem

In this section, the proposed algorithm is used for two practical multiobjective mixed-variable optimization problems (MOMVOPs). The first one is an analytical MINLP problem and the other is a disc brake design problem. These two practical problems contain multiple objectives, multiple complex constraints,

Table 15
The comparisons of IGD values obtained by Variant 1, Variant 2, Variant 3 and RLMMDE on ZDT, DTLZ, UF test suites.

Problem	Variant 1	Variant 2	Variant 3	RLMMDE
ZDT1	1.4578e−2 (2.11e−3) −	2.1646e+0 (5.72e−1) −	8.0721e−3 (8.20e−4) −	**6.5178e−3 (1.14e−3)**
ZDT2	2.0945e−2 (8.83e−3) −	2.2768e+0 (3.95e−1) −	8.4179e−3 (9.00e−4) −	**6.7258e−3 (1.28e−3)**
ZDT3	1.8140e−2 (5.47e−3) −	1.6264e+0 (4.41e−1) −	1.2135e−2 (3.00e−3) =	**1.1902e−2 (2.26e−3)**
ZDT4	**4.6718e−1 (2.58e−1)** =	2.1588e+1 (7.49e+0) −	6.0856e−1 (3.59e−1) =	5.5835e−1 (2.25e−1)
ZDT6	1.4909e−1 (7.27e−2) −	9.8018e−1 (1.56e+0) −	3.9038e−2 (2.27e−2) −	**3.3096e−3 (2.07e−4)**
DTLZ1	**1.9312e−2 (3.47e−4)** +	1.1825e+1 (6.61e+0) −	5.4208e−2 (6.79e−2) −	1.9866e−2 (4.45e−3)
DTLZ2	**5.0324e−2 (2.37e−5)** +	6.7608e−2 (8.16e−3) −	7.3008e−2 (4.58e−3) −	5.0511e−2 (9.28e−5)
DTLZ3	**3.9426e−1 (4.88e−1)** +	9.7757e+1 (5.91e+1) −	1.9008e+0 (1.75e+0) =	2.0291e+0 (3.15e+0)
DTLZ4	1.8138e−1 (2.21e−1) −	5.1599e−1 (2.16e−1) −	1.0505e−1 (1.23e−2) −	**5.0828e−2 (8.32e−5)**
DTLZ5	2.2953e−2 (4.88e−2) −	8.4698e−2 (1.75e−1) −	**8.8591e−3 (1.31e−3)** =	9.3123e−3 (2.41e−3)
DTLZ6	2.0342e−2 (1.13e−2) −	1.9286e+0 (8.97e−1) −	3.6397e−2 (2.61e−2) −	**1.5297e−2 (5.38e−3)**
DTLZ7	**1.1245e+0 (1.72e−1)** +	5.8912e+0 (1.48e+0) −	1.3907e+0 (8.01e−2) =	1.3657e+0 (8.17e−2)
UF1	9.2477e−2 (1.37e−2) −	3.8634e−1 (1.78e−1) −	3.8816e−2 (1.35e−2) =	**3.4724e−2 (1.16e−2)**
UF2	2.2018e−1 (2.22e−1) −	6.2335e−1 (2.49e−1) −	**1.6913e−2 (5.57e−3)** =	2.2218e−2 (2.58e−2)
UF3	2.0102e−1 (4.47e−2) =	6.5452e−1 (5.46e−1) −	2.0487e−1 (1.59e−2) =	**1.9802e−1 (1.37e−2)**
UF4	4.9658e−2 (6.00e−3) −	8.9282e−2 (1.39e−2) −	**3.7788e−2 (1.01e−3)** =	3.8079e−2 (9.64e−4)
UF5	2.1339e−1 (3.29e−2) −	2.6717e+0 (4.28e−1) −	1.7276e−1 (2.13e−2) =	**1.7046e−1 (1.59e−2)**
UF6	1.1810e−1 (1.74e−2) =	1.9119e+0 (6.68e−1) −	1.2049e−1 (1.08e−2) =	**1.1398e−1 (1.76e−2)**
UF7	4.9855e−2 (1.27e−2) −	5.7116e−1 (2.63e−1) −	2.4598e−2 (3.93e−3) =	**2.3519e−2 (5.84e−3)**
UF8	5.2177e−1 (2.81e−2) −	6.0639e−1 (1.66e−1) −	1.9150e−1 (1.73e−1) =	**1.7443e−1 (1.66e−1)**
UF9	3.7621e−1 (1.35e−1) −	8.4460e−1 (6.57e−1) −	9.6029e−2 (1.89e−2) =	**9.4933e−2 (1.63e−2)**
UF10	3.4631e−1 (5.96e−2) −	2.5108e+0 (3.99e−1) −	3.3431e−1 (5.82e−2) −	**2.9774e−1 (4.11e−2)**
+/−/=	4/15/3	0/22/0	0/8/14	

Table 16
The comparisons of IGD values obtained by Variant 4, ANSGA-III, MOEA/D-AWA and RLMMDE on ZDT, DTLZ, UF test suites.

Problem	Variant4	ANSGA-III	MOEA/D-AWA	RLMMDE
ZDT3	**9.0590e−3 (9.72e−4)** +	2.6591e−2 (9.20e−3) −	2.4911e−2 (1.20e−2) −	1.1207e−2 (1.38e−3)
DTLZ5	1.1115e−2 (1.48e−3) −	9.5252e−3 (8.61e−4) −	9.5057e−3 (2.34e−4) −	**8.9480e−3 (1.51e−3)**
DTLZ6	2.0264e−2 (2.49e−3) −	1.2826e−2 (1.45e−3) =	**9.5833e−3 (3.44e−4)** +	1.5235e−2 (5.80e−3)

Table 17
Definition of analytical MINLP problem.

Objective function	min	$f_1 = x_1^2 - x_2 + x_3 + 3y_1 + 2y_2 + y_3$ $f_2 = 2x_1^2 + x_2 - 3x_3 - 2y_1 + y_2 - 2y_3$
Constraints	g_1	$3x_1 - x_2 + x_3 + 2y_1 \le 0$
	g_2	$4x_1^2 + 2x_1 + x_2 + x_3 + y_1 + 7y_2 - 40 \le 0$
	g_3	$-x_1 - 2x_2 + 3x_3 + 7y_3 \le 0$
	g_4	$-x_1 + 12y_1 - 10 \le 0$
	g_5	$x_1 - 2y_1 - 5 \le 0$
	g_6	$-x_2 + y_2 - 20 \le 0$
	g_7	$x_2 - y_2 - 40 \le 0$
	g_8	$-x_3 + y_3 - 17 \le 0$
	g_9	$x_3 - y_3 - 25 \le 0$
Where		$-100 \leqslant x_1, x_2, x_3 \leqslant 100$ $y_1, y_2, y_3 \in \{0, 1\}$

and mixed types of decision variables, which makes it difficult for the algorithms to obtain an optimal set of trade-off designs or solutions. In addition, it is noteworthy that the actual PF of these two practical optimal design problems is unknown.

5.7.1. Analytical MINLP problem

The analytical MINLP problem is a classical MO-MVOPs. It was first proposed by Dimkou in [87]. since it contains two objectives, nine constraints and mixed types of decision variables, MOEAs have difficulty solving this problem. The decision variables of this problem include three continuous variables (x_1, x_2, x_3) and three binary discrete variables ($y_1, y_2, y_3 \in \{0, 1\}$). The definition of the problem is shown as Table 17.

To analyze the performance of the proposed algorithm on this problem, the diversity metric (PD) [88] is used to test the diversity of the obtained solutions. The larger the value of this metric is, the better the diversity of the solutions obtained by the algorithms. This metric does not require any parameters and does not require information on the true PF. The metric values obtained by RLMMDE are compared with those obtained by ARMOEA [48], CMOEA/D [50], RVEA [83], MOEA/D-DAE [89] and MOEA/DD [90]. However, although each of these algorithms has its own constraint handling method, none of them has a method to handle discrete variables. Thus, the specific variables are discretized before each calculation of the objective function. In RLMMDE, the discrete variable handling method we previously proposed in [57] is used.

Table 18
The comparisons of PD values obtained by ARMOEA, CMOEA/D, RVEA, MOEA/D-DAE, MOEA/DD and RLMMDE on analytical MINLP problem.

Algorithm	Mean	Std	MaxFEs
RLMMDE	**1.1994E+05**	1.31E+04	20 000
ARMOEA	1.1716E+05	1.34E+04	20 000
CMOEA/D	1.1975E+05	3.49E+04	20 000
RVEA	7.9188E+04	1.29E+04	20 000
MOEA/D-DAE	9.8508E+04	2.47E+04	20 000
MOEA/DD	4.8779E+04	2.30E+04	20 000

As shown in Table 18, the average values and standard deviation of the PD metric obtained by all algorithms after 30 runs with the population size set to 100 and MaxFEs set to 20 000 are presented. The results show that the diversity of the Pareto solution obtained by RLMMDE is competitive.

5.7.2. Disc brake design problem

The disc brake design problem was originally discussed by Osyczka in [91]. The minimum of the mass of the brake and stopping time are the problem's two objectives for optimal design. The disc brake design problem contains two objectives, five complex constraints, and mixed types of decision variables. Its decision variables contain 3 continuous variables, the inner radius of the disc (x_1), the outer radius of the disc (x_2), the engaging force (x_3) and a discrete variable, the number of friction surfaces (x_4). Notably, the value of x_4 must be an integer. The definition of the problem is shown as Table 19.

Table 19
Definition of disc brake design problem.

Objective function	min	$f_1 = 4.9 \times 10^{-5}\left(x_2^2 - x_1^2\right)(x_4 - 1)$ $f_2 = \frac{9.82\times 10^6\left(x_2^2-x_1^2\right)}{x_3x_3\left(x_2^3-x_1^3\right)}$
Constraints	g_1	$20 - (x_2 - x_1) \leq 0$
	g_2	$2.5\,(x_4 + 1) - 30 \leq 0$
	g_3	$\frac{x_3}{\pi\left(x_2^2-x_1^2\right)} - 0.4 \leq 0$
	g_4	$\frac{2.22\times 10^{-3}x_3\left(x_2^3-x_1^3\right)}{\left(x_2^2-x_1^2\right)^2} \leq 0$
	g_5	$\frac{900-2.66\times 10^{-2}x_3x_4\left(x_2^3-x_1^3\right)}{\left(x_2^2-x_1^2\right)} \leq 0$
Where		$55 \leq x_1 \leq 80$ $75 \leq x_2 \leq 110$ $1000 \leq x_3 \leq 3000$ $2 \leq x_4 \leq 20$, *Integer*

Table 20
The comparisons of PD values obtained by NSGA-III, ANSGA-III, CMOEA/D, RVEA, MOEA/DD and RLMMDE on disc brake design problem.

Algorithm	Mean	Std	MaxFEs
RLMMDE	**6.8531E+03**	6.64E+02	10 000
NSGA-III	6.5788E+03	8.47E+02	10 000
ANSGA-III	6.5274E+03	9.80E+02	10 000
CMOEA/D	3.8084E+03	7.86E+02	10 000
RVEA	6.4080E+03	1.06E+03	10 000
MOEA/DD	5.7404E+03	8.68E+02	10 000

In this problem, the diversity metric (PD) is also used to test the diversity of the obtained solutions. RLMMDE are compared with NSGA-III [60], ANSGA-III [50], CMOEA/D [50], RVEA [83] and MOEA/DD [90]. The discrete variables are treated in the same way as in the analytical MINLP problem. As shown in Table 20, the average values and standard deviation of PD metric after 30 runs for all algorithms with the population size set to 100 and MaxFEs set to 10 000 are given. The results show that RLMMDE obtains better PD values than the other compared algorithms for the same FEs. This also indicates that the diversity of the obtained Pareto solutions is optimal.

6. Conclusion

In this paper, a new multistrategy multiobjective DE algorithm named RLMMDE is proposed. In RLMMDE, a multistrategy and multicrossover DE optimizer is proposed to better balance exploration and exploitation and improve the performance of the algorithm in solving MOPs. To control the frequency and activation of the reference point adaptation method, an RL-based activation mechanism is proposed. In addition, a reference point adaptation method is proposed to further improve the performance of the algorithm in solving irregular PF problems.

In the experimental study, three different types of test suites are used to test the performance of RLMMDE. Compared with 9 conventional MOEAs, three MOEAs with different optimizers, and four MOEAs with reference point adaptation methods, the performance of the proposed algorithm is significantly competitive. In addition, four different variants and two algorithms with the ability to handle irregular fronts are constructed to verify the effectiveness of the components of RLMMDE. Furthermore, experimental results on the WFG, LSMOP and DTLZ variants also show that RLMMDE is better than the compared algorithms. Finally, the proposed algorithms are applied to two practical multiobjective mixed-variable optimization problems. The experimental results show that RLMMDE obtained the optimal solution design.

However, the feedback from the activation mechanism is not very timely. The performance becomes poor when dealing with some problems after executing the reference point adaptation method, e.g., the DTLZ7 problem. In the future, we will focus more on monitoring the state of reference points after adjustment and propose a reference point adaptation method with better problem versatility.

CRediT authorship contribution statement

Yupeng Han: Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hu Peng:** Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis. **Changrong Mei:** Supervision, Resources, Investigation, Formal analysis, Conceptualization. **Lianglin Cao:** Validation, Project administration, Investigation, Formal analysis. **Changshou Deng:** Validation, Supervision, Methodology, Investigation, Formal analysis. **Hui Wang:** Validation, Supervision, Formal analysis. **Zhijian Wu:** Supervision, Investigation.

Declaration of competing interest

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.

Data availability

The Matlab source code of RLMMDE can be downloaded from Hu Peng's homepage: https://whuph.github.io/.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62266024, 61763019, 62166027), the Science and Technology Foundation of Jiangxi Province, PR China (20202BABL202019).

References

[1] Y. Tian, L. Si, X. Zhang, R. Cheng, C. He, K.C. Tan, Y. Jin, Evolutionary large-scale multi-objective optimization: A survey, ACM Comput. Surv. 54 (8) (2021) 1–34.
[2] T.N. Huynh, D.T. Do, J. Lee, Q-learning-based parameter control in differential evolution for structural optimization, Appl. Soft Comput. 107 (2021) 107464.
[3] Y. Huang, W. Li, F. Tian, X. Meng, A fitness landscape ruggedness multiobjective differential evolution algorithm with a reinforcement learning strategy, Appl. Soft Comput. 96 (2020) 106693.
[4] F. Wang, X. Wang, S. Sun, A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization, Inform. Sci. 602 (2022) 298–312.
[5] C.J. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (3) (1992) 279–292.
[6] R.S. Sutton, D. McAllester, S. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: S. Solla, T. Leen, K. Müller (Eds.), Advances in Neural Information Processing Systems, Vol. 12, MIT Press, 1999.
[7] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
[8] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double q-learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016.
[9] H. Zhang, J. Sun, Z. Xu, Learning to mutate for differential evolution, in: 2021 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2021, pp. 1–8.
[10] F. Zhao, X. Hu, L. Wang, J. Zhao, J. Tang, et al., A reinforcement learning brain storm optimization algorithm (BSO) with learning mechanism, Knowl.-Based Syst. 235 (2022) 107645.
[11] P. Rakshit, A. Konar, P. Bhowmik, I. Goswami, S. Das, L.C. Jain, A.K. Nagar, Realization of an adaptive memetic algorithm using differential evolution and Q-learning: A case study in multirobot path planning, IEEE Trans. Syst. Man, Cybern.: Syst. 43 (4) (2013) 814–831.

[12] Z. Hu, W. Gong, S. Li, Reinforcement learning-based differential evolution for parameters extraction of photovoltaic models, Energy Rep. 7 (2021) 916–928.
[13] J. Sun, X. Liu, T. Bäck, Z. Xu, Learning adaptive differential evolution algorithm from optimization experiences by policy gradient, IEEE Trans. Evol. Comput. 25 (4) (2021) 666–680.
[14] A. Seyyedabbasi, R. Aliyev, F. Kiani, M.U. Gulle, H. Basyildiz, M.A. Shah, Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems, Knowl.-Based Syst. 223 (2021) 107044.
[15] Z. Tan, K. Li, Differential evolution with mixed mutation strategy based on deep reinforcement learning, Appl. Soft Comput. 111 (2021) 107678.
[16] M. Sharma, A. Komninos, M. López-Ibáñez, D. Kazakov, Deep reinforcement learning based parameter control in differential evolution, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp. 709–717.
[17] 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, IEEE, 2019, pp. 815–822.
[18] 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 Trans. Emerg. Top. Comput. Intell. (2022) 1–14.
[19] W. Ning, B. Guo, X. Guo, C. Li, Y. Yan, Reinforcement learning aided parameter control in multi-objective evolutionary algorithm based on decomposition, Prog. Artif. Intell. 7 (4) (2018) 385–398.
[20] K.V. Moffaert, M.M. Drugan, A. Nowé, Hypervolume-based multi-objective reinforcement learning, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2013, pp. 352–366.
[21] K. Li, T. Zhang, R. Wang, Deep reinforcement learning for multiobjective optimization, IEEE Trans. Cybern. 51 (6) (2020) 3103–3114.
[22] K.M. Sallam, S.M. Elsayed, R.K. Chakrabortty, M.J. Ryan, Improved multi-operator differential evolution algorithm for solving unconstrained problems, in: 2020 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2020, pp. 1–8.
[23] R. Ojstersek, M. Brezocnik, B. Buchmeister, Multi-objective optimization of production scheduling with evolutionary computation: A review, Int. J. Ind. Eng. Comput. 11 (3) (2020) 359–376.
[24] S. Elhag, A. Fernández, A. Altalhi, S. Alshomrani, F. Herrera, A multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems, Soft Comput. 23 (4) (2019) 1321–1336.
[25] M. Fazzolari, R. Alcalá, F. Herrera, A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm, Appl. Soft Comput. 24 (2014) 470–481.
[26] J.M. Luna, J.R. Romero, S. Ventura, Grammar-based multi-objective algorithms for mining association rules, Data Knowl. Eng. 86 (2013) 19–37.
[27] J.L. Olmo, J.M. Luna, J.R. Romero, S. Ventura, Mining association rules with single and multi-objective grammar guided ant programming, Integr. Comput.-Aided Eng. 20 (3) (2013) 217–234.
[28] W.M. Spears, K.A. De Jong, T. Bäck, D.B. Fogel, H. De Garis, An overview of evolutionary computation, in: Machine Learning: ECML-93: European Conference on Machine Learning Vienna, Austria, April 5–7, 1993 Proceedings, Springer, 2005, pp. 442–459.
[29] F. Yu, W. Gong, H. Zhen, A data-driven evolutionary algorithm with multi-evolutionary sampling strategy for expensive optimization, Knowl.-Based Syst. 242 (2022) 108436.
[30] Z. Zhou, J. Abawajy, M. Shojafar, M. Chowdhury, DEHM: An improved differential evolution algorithm using hierarchical multistrategy in a cybertwin 6G network, IEEE Trans. Ind. Inform. 18 (7) (2022) 4944–4953.
[31] W. Lin, Q. Lin, J. Ji, Z. Zhu, C.A.C. Coello, K.-C. Wong, Decomposition-based multiobjective optimization with bicriteria assisted adaptive operator selection, Swarm Evol. Comput. 60 (2021) 100790.
[32] Z. Yan, Y. Tan, W. Zheng, L. Meng, H. Zhang, Leader recommend operators selection strategy for a multiobjective evolutionary algorithm based on decomposition, Inform. Sci. 550 (2021) 166–188.
[33] K. Li, A. Fialho, S. Kwong, Q. Zhang, Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 18 (1) (2013) 114–130.
[34] L. Sun, K. Li, Adaptive operator selection based on dynamic thompson sampling for MOEA/D, in: International Conference on Parallel Problem Solving from Nature, Springer, 2020, pp. 271–284.
[35] W. Huang, W. Zhang, Adaptive multi-objective particle swarm optimization with multi-strategy based on energy conversion and explosive mutation, Appl. Soft Comput. 113 (2021) 107937.
[36] Z. Yong, H. Chun-lin, S. Xian-fang, S. Xiao-yan, A multi-strategy integrated multi-objective artificial bee colony for unsupervised band selection of hyperspectral images, Swarm Evol. Comput. 60 (2021) 100806.
[37] 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 Gener. Comput. Syst. 131 (2022) 59–74.
[38] G. Reynoso-Meza, J. Sanchis, X. Blasco, J.M. Herrero, Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems, in: 2011 IEEE Congress of Evolutionary Computation, CEC, 2011, pp. 1551–1556.
[39] P. Bujok, On the performance and complexity of crossover in differential evolution algorithm, in: Artificial Intelligence and Soft Computing: 19th International Conference, ICAISC 2020, Zakopane, Poland, October 12-14, 2020, Proceedings, Part I 19, Springer, 2020, pp. 363–375.
[40] A.K. Sadhu, A. Konar, T. Bhattacharjee, S. Das, Synergism of firefly algorithm and Q-learning for robot arm path planning, Swarm Evol. Comput. 43 (2018) 50–68.
[41] C.-H. Chen, C.-B. Liu, Reinforcement learning-based differential evolution with cooperative coevolution for a compensatory neuro-fuzzy controller, IEEE Trans. Neural Netw. Learn. Syst. 29 (10) (2017) 4719–4729.
[42] 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.
[43] D. Jia, H. Guo, Z. Song, L. Shi, X. Deng, M. Perc, Z. Wang, Local and global stimuli in reinforcement learning, New J. Phys. 23 (8) (2021) 083020.
[44] I. Fister, P.N. Suganthan, I. Fister, S.M. Kamal, F.M. Al-Marzouki, M. Perc, D. Strnad, Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution, Nonlinear Dynam. 84 (2016) 895–914.
[45] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[46] A. Kaur, K. Kumar, A reinforcement learning based evolutionary multi-objective optimization algorithm for spectrum allocation in cognitive radio networks, Phys. Commun. 43 (2020) 101196.
[47] L. Ma, S. Cheng, X. Wang, M. Huang, H. Shen, X. He, Y. Shi, Cooperative two-engine multi-objective bee foraging algorithm with reinforcement learning, Knowl.-Based Syst. 133 (2017) 278–293.
[48] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Trans. Evol. Comput. 22 (4) (2017) 609–622.
[49] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791.
[50] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach, IEEE Trans. Evol. Comput. 18 (4) (2013) 602–622.
[51] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, MOEA/D with adaptive weight adjustment, Evol. Comput. 22 (2) (2014) 231–264.
[52] S. Jiang, H. Li, J. Guo, M. Zhong, S. Yang, M. Kaiser, N. Krasnogor, AREA: An adaptive reference-set based evolutionary algorithm for multiobjective optimisation, Inform. Sci. 515 (2020) 365–387.
[53] S. Xian-fang, S. Xiao-yan, Z. Yong, H. Chun-lin, A multi-strategy integrated multi-objective artificial bee colony for unsupervised band selection of hyperspectral images, Swarm Evol. Comput. 60 (2021) 100806.
[54] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2009) 284–302.
[55] K. Deb, A. Sinha, S. Kukkonen, Multi-objective test problems, linkages, and evolutionary methodologies, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, 2006, pp. 1141–1148.
[56] A.W. Iorio, X. Li, Solving rotated multi-objective optimization problems using differential evolution, in: Australasian Joint Conference on Artificial Intelligence, Springer, 2004, pp. 861–872.
[57] H. Peng, Y. Han, C. Deng, J. Wang, Z. Wu, Multi-strategy co-evolutionary differential evolution for mixed-variable optimization, Knowl.-Based Syst. 229 (2021) 107366.
[58] Y. Han, H. Peng, A. Jiang, C. Wang, F. Kong, M. Li, A differential evolution algorithm for multi-objective mixed-variable optimization problems, in: Bio-Inspired Computing: Theories and Applications: 16th International Conference (BIC-TA 2021), Springer, 2022, pp. 145–159.
[59] X. Qiu, K.C. Tan, J. Xu, Multiple exponential recombination for differential evolution, IEEE Trans. Cybern. 47 (4) (2016) 995–1006.
[60] 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 Trans. Evol. Comput. 18 (4) (2014) 577–601.
[61] H. Peng, Z. Wu, C. Deng, Enhancing differential evolution with commensal learning and uniform local search, Chin. J. Electron. 26 (4) (2017) 725–733.
[62] R. Hinterding, Gaussian mutation and self-adaption for numeric genetic algorithms, in: Proceedings of 1995 IEEE International Conference on Evolutionary Computation, Vol. 1, IEEE, 1995, p. 384.

[63] S. Mahadevan, Average reward reinforcement learning: Foundations, algorithms, and empirical results, Mach. Learn. 22 (1) (1996) 159–195.
[64] I. Giagkiozis, R.C. Purshouse, P.J. Fleming, Towards understanding the cost of adaptation in decomposition-based optimization algorithms, in: 2013 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 2013, pp. 615–620.
[65] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[66] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multi-objective optimization, Comput. Eng. KanGAL Rep(1990) 112 (2001) 1–27.
[67] Q. Zhang, A. Zhou, S. Zhao, P. Suganthan, W. Liu, S. Tiwari, Multiobjective optimization test instances for the CEC 2009 special session and competition, Mech. eng. (New York, N.Y.: 1919) 487 (2008) 1–30.
[68] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Trans. Evol. Comput. 10 (5) (2006) 477–506.
[69] R. Cheng, Y. Jin, M. Olhofer, et al., Test problems for large-scale multiobjective and many-objective optimization, IEEE Trans. Cybern. 47 (12) (2016) 4108–4121.
[70] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V. Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132.
[71] O. Schutze, X. Esquivel, A. Lara, C.A.C. Coello, Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization, IEEE Trans. Evol. Comput. 16 (4) (2012) 504–522.
[72] D.A. Van Veldhuizen, Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations, (Ph. D. thesis), Department of Electrical and Computer Engineering. Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1999, 9928483.
[73] I. Das, J. Dennis, Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (3) (2000) 631–657.
[74] Q. Zhang, L. Hui, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2008) 712–731.
[75] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 17 (5) (2013) 721–736.
[76] Y. Tian, R. Cheng, X. Zhang, Y. Su, Y. Jin, A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization, IEEE Trans. Evol. Comput. 23 (2) (2018) 331–345.
[77] J. Cheng, G.G. Yen, G. Zhang, A many-objective evolutionary algorithm with enhanced mating and environmental selections, IEEE Trans. Evol. Comput. 19 (4) (2015) 592–605.
[78] C.C. Coello, M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 2, IEEE, 2002, pp. 1051–1056.
[79] D.W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, PESA-II: Region-based selection in evolutionary multiobjective optimization, in: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc., 2001, pp. 283–290.
[80] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, Conference on Parallel Problem Solving from Nature (PPSN VIII), 2004, pp. 832–842.
[81] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in: 2009 IEEE Congress on Evolutionary Computation, IEEE, 2009, pp. 203–208.
[82] R. Cheng, Y. Jin, K. Narukawa, B. Sendhoff, A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling, IEEE Trans. Evol. Comput. 19 (6) (2015) 838–856.
[83] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791.
[84] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[85] J. Alcalá-Fdez, L. Sánchez, S. Garcia, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, et al., KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (3) (2009) 307–318.
[86] Y. Tian, R. Cheng, X. Zhang, Y. Jin, Platemo: A MATLAB platform for evolutionary multi-objective optimization, IEEE Comput. Intell. Mag. 12 (2017) 73–87.
[87] T. Dimkou, K. Papalexandri, A parametric optimization approach for multiobjective engineering problems involving discrete decisions, Comput. Chem. Eng. 22 (1998) 951–954.
[88] H. Wang, Y. Jin, X. Yao, Diversity assessment in many-objective optimization, IEEE Trans. Cybern. 47 (6) (2016) 1510–1522.
[89] Q. Zhu, Q. Zhang, Q. Lin, A constrained multiobjective evolutionary algorithm with detect-and-escape strategy, IEEE Trans. Evol. Comput. 24 (5) (2020) 938–947.
[90] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput. 19 (5) (2014) 694–716.
[91] A. Osyczka, S. Kundu, A genetic algorithm-based multicriteria optimization method, in: 1st World Congresses of Structural and Multidisciplinary Optimization, Elsevier, 1995, pp. 909–914.