

## Reinforcement learning-based differential evolution algorithm for constrained multi-objective optimization problems

Xiaobing Yu<sup>a,b,\*</sup>, Pingping Xu<sup>b</sup>, Feng Wang<sup>b</sup>, Xuming Wang<sup>b</sup>

<sup>a</sup> Collaborative Innovation Center on Forecast and Evaluation of Meteorological Disasters(CIC-FEMD), Nanjing University of Information Science & Technology, Nanjing 210044, China

<sup>b</sup> The Research Institute for Risk Governance and Emergency Decision-Making, School of Management Science and Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China

### ARTICLE INFO

#### Keywords:

Reinforcement learning  
Evolutionary algorithm  
Differential evolution  
Constrained problem

### ABSTRACT

Many real-world problems can be established as Constrained Multi-objective Optimization Problems (CMOPs). It is still challenging to automatically set efficient parameters for Constrained Multi-Objective Evolutionary Algorithms (CMOEAs) to solve these CMOPs. A Reinforcement Learning-based Multi-Objective Differential Evolution (RLMODE) algorithm is proposed, in which the main parameters are dynamically adjusted. During the evolution process, the offspring generated is evaluated and compared with its corresponding parents, the relationship between the offspring and parent can adjust the parameters of RLMODE by the Reinforcement Learning (RL) technique. The feedback mechanism can produce the most appropriate parameters for RLMODE, which pushes the population towards feasible regions. The proposed RLMODE is evaluated on thirty functions and compared with some popular CMOEAs. The performance indicator IGD has revealed that the proposed RLMODE is competitive. Then, they are applied to solve the UAV path planning problem with three objectives and a constraint. The real application has further demonstrated the superiority of the proposed RLMODE.

### 1. Introduction

Many industrial applications involve a set of objectives while meeting some constraints, such as yield strength and ultimate tensile strength are two critical objectives in the micro-alloyed steel compositions (Saini et al., 2023), the total flight path length and the terrain threat of UAVs are two conflicting objectives when planning UAV path (Xiao et al., 2023), energy consumption and trajectory smoothness should be considered simultaneously for arc welding robot path planning (Zhou et al., 2022), maximizing the profit, minimizing the travel time and costs when managing and collecting waste (Hashemi-Amiri et al., 2023), maximizing economic utility and minimizing phosphorus pollution are demanded when dealing with shallow lake problem (Shavazipour et al., 2021). These objectives often conflict with each other. They are defined as Constrained Multi-Objective Optimization Problems (CMOPs). These CMOPs can be summarized as follows:

$$F(x) = (f_1(x), f_2(x), \dots, f_i(x), \dots, f_m(x)) \quad (1)$$

$$\text{s.t.} \begin{cases} g_j(x) < 0, j = 1, 2, \dots, q \\ h_j(x) = 0, j = q + 1, q + 2, \dots, p \\ x \in R^n \end{cases}$$

where  $F(x) \in R^m$  is the objective with  $m$  dimension,  $x \in R^n$  is a vector with  $n$  dimension,  $g_j(x)$ ,  $h_j(x)$  are inequality and equality constraints,  $q$  and  $p - q$  are their amounts. When dealing with the equality constraints, a very small positive number  $\delta$  is often introduced. The primary purpose is to convert the equality into the inequality. The Constraint Violation ( $cv$ ) of each solution  $x$  can be computed as follows:

$$cv(x) = \sum_{j=1}^p cv_j(x) \quad (2)$$

$$cv_j(x) = \begin{cases} \max\{g_j(x), 0\}, j = 1, 2, \dots, q \\ \max\{|h_j(x)| - \delta, 0\}, j = q + 1, q + 2, \dots, p \end{cases}$$

where  $cv_j(x)$  is the violation of each constraint,  $cv$  is the total violation. If  $cv = 0$ , we can define the solution  $x$  as the feasible solution. On the opposite, it is an infeasible one. All the feasible solutions constitute a

\* Corresponding author. Collaborative Innovation Center on Forecast and Evaluation of Meteorological Disasters(CIC-FEMD), Nanjing University of Information Science & Technology, Nanjing 210044, China.

E-mail address: [002257@nuist.edu.cn](mailto:002257@nuist.edu.cn) (X. Yu).

feasible set  $S(S = \{x|cv(x) = 0, x \in R^n\})$ . Let's define two solutions  $x_1 \in S$  and  $x_2 \in S$ . If  $f_i(x_1) \leq f_i(x_2)$  ( $i = 1, 2, \dots, m$ ) and  $f_j(x_1) < f_j(x_2)$  ( $j \in \{1, 2, \dots, m\}$ ), then the solution  $x_1$  dominates  $x_2$ , represented as  $x_1 \leq x_2$ . If no solution in  $S$  can dominate  $x^*$ , we define that  $x^*$  is the Pareto optimal solution. All the Pareto optimal solutions can constitute the Pareto optimal Set  $PS$ . The Pareto front  $PF$  is defined as  $PF = \{F(x)|x \in PS\}$ . These constraints bring more difficulties in obtaining the  $PF$  than the unconstrained problems. They divide the feasible space into different regions so that a large proportion of feasible space becomes infeasible. Constrained  $PF$  is to denote the  $PF$  for CMOPs while unconstrained  $PF$  is to describe the  $PF$  for unconstrained multi-objective optimizations.

When solving CMOPs, minimizing objectives and satisfying constraints are two key issues. Many MOEAs and constraint-handling techniques have been developed. They extensively promoted the development of CMOPs. The former mainly focuses on the selection mechanism, while the latter deals with the constraints. During the evolution process, the parent generates offspring. A combination population is generated with  $2NP$  individuals,  $NP$  is the population size. The mechanism often selects  $NP$  solutions from the combined population for the next evolution (Deb et al., 2002). However, the relationship between the parent and the corresponding offspring is often neglected, which can have some implications for the search direction. For example, if the offspring can dominate the parent, indicating that the search direction and the parameter setting of algorithms are more valid. We can keep and strengthen the search direction. On the opposite, the search direction may not be valid. The feedback mechanism is widely used in single optimization problems with the help of Reinforcement Learning (RL) technique and achieved great success (Cheng-Hung and Chong-Bin, 2018; Hu et al., 2021; Wang et al., 2022).

RL is a valuable technique from machine learning, which can automatically obtain rewards on the basis of the interaction with the environment (Hu et al., 2021). RL can maximize the benefit of its actions through trial and error. The advantages can be effectively used to address CMOPs. Therefore, RL technique is increasingly used in the community of CMOPs. To effectively handle constraints, a novel CMOEA based on deep RL is proposed, in which a dynamic penalty coefficient is dynamically updated according to the training loss (Tang et al., 2023). To provide some guidance strategies for generating a better population, a process knowledge-guided CMOEA is developed. A deep Q-learning network is applied to reflect the process knowledge (Zuo et al., 2023). Five strategies from Differential Evolution (DE) algorithm, DE/best/1, DE/best/2, DE/rand/1, DE/rand/2, and DE/current-to-pbest/1, are used in the process, in which DE/best/1, DE/best/2, and DE/current-to-pbest/1 involve the best solution. It is hard to define the best solution in the context of CMOPs. The best solution involves the whole non-dominated solutions in CMOPs. Deep RL dynamically adjusts two operators, DE and genetic algorithm (GA) to deal with CMOPs as GA has a strong capability of convergence and DE has a strong of exploring (Fei et al., 2024). These algorithms mainly focus on strategies how to generate offspring while neglect to adjust the parameters involved in these strategies. These parameters of CMOEAs are also important to balance local and global search. Motivated by the

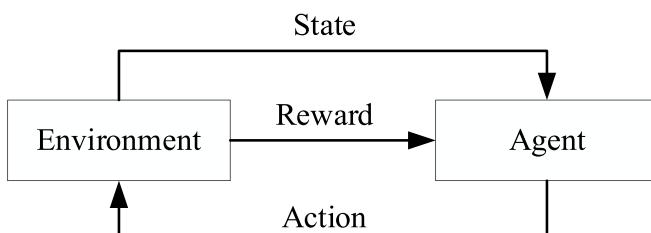


Fig. 1. The relationship among five ingredients.

Table 1

The form of the Q-table.

State	Action			
	$a_1$	$a_2$	...	$a_n$
$s_1$	$Q_{(s_1, a_1)}$	$Q_{(s_1, a_2)}$	...	$Q_{(s_1, a_n)}$
$s_2$	$Q_{(s_2, a_1)}$	$Q_{(s_2, a_2)}$	...	$Q_{(s_2, a_n)}$
...	...	...	...	...
$s_m$	$Q_{(s_m, a_1)}$	$Q_{(s_m, a_2)}$	...	$Q_{(s_m, a_n)}$

Table 2

Eight relationships between the parent and offspring.

State	The parent ( $U_i$ )	The offspring ( $V_i$ )	The state	Reward
1	Feasible	Feasible	$V_i \leq U_i$	1
2			$U_i \leq V_i$	-1
3			$U_i$ and $V_i$ cannot dominate each other	0
4	Feasible	Infeasible	$U_i \leq V_i$	-1
5	Infeasible	Feasible	$V_i \leq U_i$	1
6	Infeasible	Infeasible	$V_i \leq U_i$	1
7			$U_i \leq V_i$	-1
8			$U_i$ and $V_i$ cannot dominate each other	0

observation and research gap, we develop a RL-based Multi-Objective DE (RLMODE) algorithm to solve CMOPs by automatically adjusting the main parameters. The main novelty of the paper can be summarized as follows.

- (1) Reinforcement Learning based Multi-Objective Differential Evolution (RLMODE) algorithm is developed to solve CMOPs.
- (2) The relationship between offspring and its parents is considered in RLMODE so that the main parameters of RLMODE are adaptively adjusted by RL technique, and the most appropriate values can be obtained.
- (3) RLMODE algorithm is used to solve thirty benchmark functions and a real application. The performance indicators have revealed that the RLMODE algorithm is competitive.

The paper is organized as follows: related works are introduced in Section 2; the proposed RLMODE is elaborated in Section 3; the experiments are performed in Section 4; a multi-objective UAV path planning model is solved by RLMODE in Section 5; and the conclusion is made in Section 6.

## 2. Related work

Handling CMOPs is still challenging since minimizing multi-objectives and meeting various constraints are equally important (Liang et al., 2023a). For the former, lots of MOEAs can be available. These MOEAs can be roughly classified into the following types (Yu et al., 2018). The first one is based on decomposition. MOEAs of this type often decompose CMOPs into serial sub-problems and optimize them simultaneously, such as MOEA/D (Qingfu and Hui, 2007). The second type is on the basis of the Pareto dominance principle, in which non-dominated solutions are emphasized, and elites are preserved, such as NSGAI (Deb et al., 2002) and NSGAIII (Deb and Jain, 2014; Jain and Deb, 2014). The last type is built on the performance indicators, such as HypE (Bader and Zitzler, 2011) and IBEA (Zitzler and Künzli, 2004). These algorithms are capable of dealing with unconstrained multi-objective optimization problems and provide well-distributed non-dominated solutions. However, they cannot be directly used to solve CMOPs without constraint handling techniques.

Constraint handling is an indispensable part for CMOEAs when solving CMOPs. Several constraint-handling approaches have been

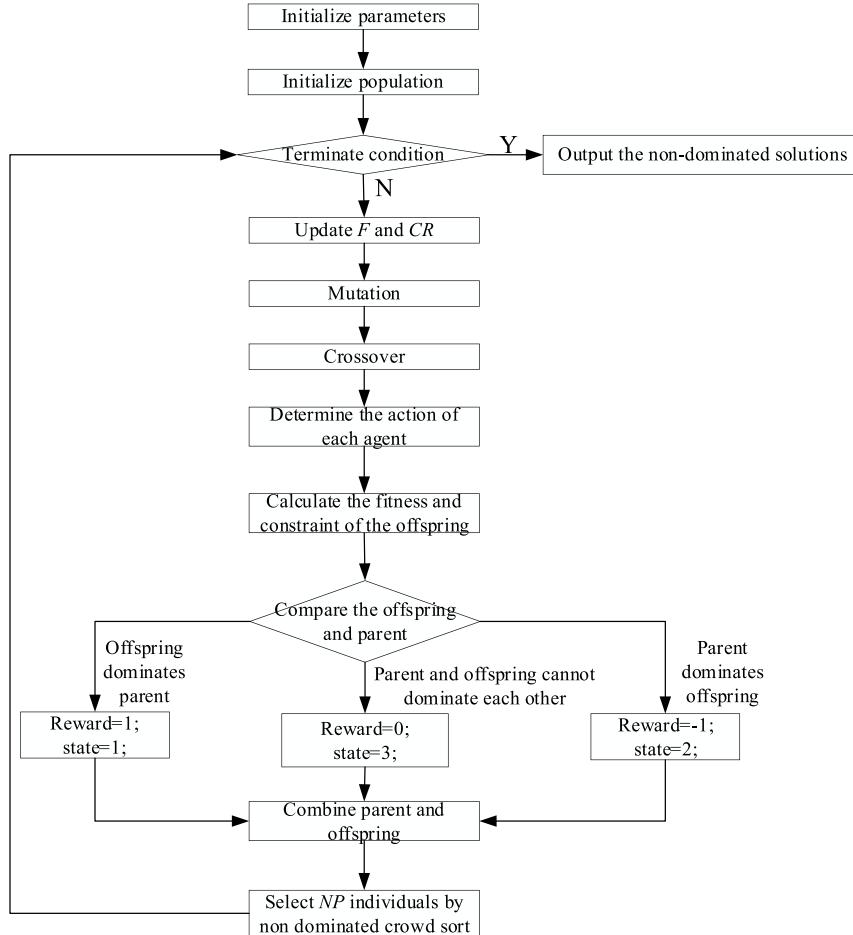


Fig. 2. The flowchart of the proposed RLMODE algorithm.

developed and can be divided into the following categories (Han et al., 2022): (1) Stochastic ranking method. The method introduces a probability parameter  $p$ . If a random number between 0 and 1 is more than  $p$ , the comparison between two individuals is firstly performed based on the  $cv$ . Otherwise, the comparison is on the basis of the objective function (Runarsson and Xin, 2000). (2) Constraint Domination Principle (CDP). This method is very simple and straightforward, which defines the following scenarios: the feasible solution dominates the infeasible solution; the infeasible solution with lower  $cv$  is superior to the infeasible solution with upper  $cv$ . The stochastic ranking method is the same as the CDP when  $p$  is equal to 0. Many MOEAs adopt the method to cope with the constraints, such as NSGAII-CDP (Deb et al., 2002), NGSIII-CDP (Jain and Deb, 2014), etc. MOEA/D also borrows the idea into sub-problems by defining  $cv$  as an alternative to develop MOEA/D-ACDP algorithm (Cheng et al., 2016). Some researchers introduce threshold  $\varepsilon$  – constraint to extend the method. If the  $cv$  of the solution is smaller than the threshold  $\varepsilon$ , the solution is regarded as the feasible (Asafuddoula et al., 2012). MOEA/D uses the  $\varepsilon$  – constraint to obtain solutions which the objective functions are minimized in the feasible regions (Martinez and Coello, 2014). A bi-objective problem is built, in which the scalarizing function and constraint violation degree is considered as an objective function (Zapotecas-Martínez and Ponsich, 2020). MOEA/D- $\varepsilon$  adjusts the  $\varepsilon$  level dynamically based on the ratio of infeasible to total solutions in the population (Fan et al., 2019a). (3) Penalty method. This method often penalizes infeasible solutions by introducing a coefficient or a function to the original objective function. However, it is still difficult to determine the function or the coefficient (Yu et al., 2019). Some adaptive penalty methods have been proposed. A dynamic penalty function with MEOA/D is adopted to promote the

interaction between infeasible and feasible (Maldonado and Zapotecas-Martínez, 2021). The self-adaptive method uses an adaptive penalty and a distance measure function to modify objective functions (Woldesenbet et al., 2009). The adaptive tradeoff method considers three scenarios: all solutions are feasible, infeasible; feasible and

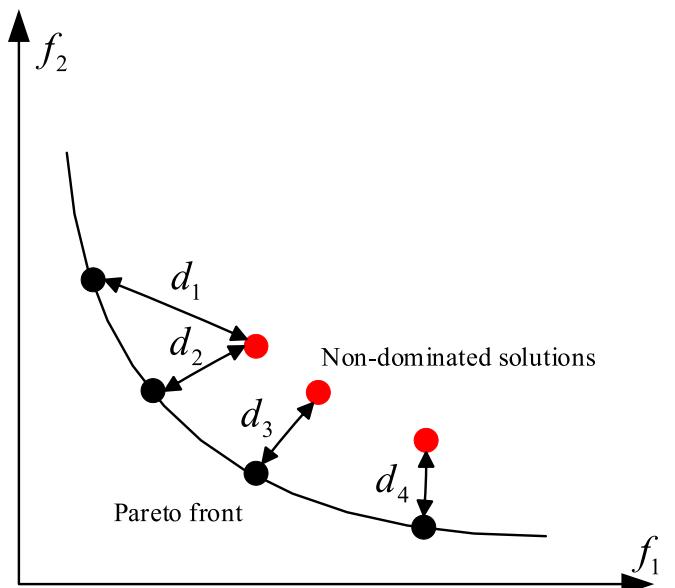
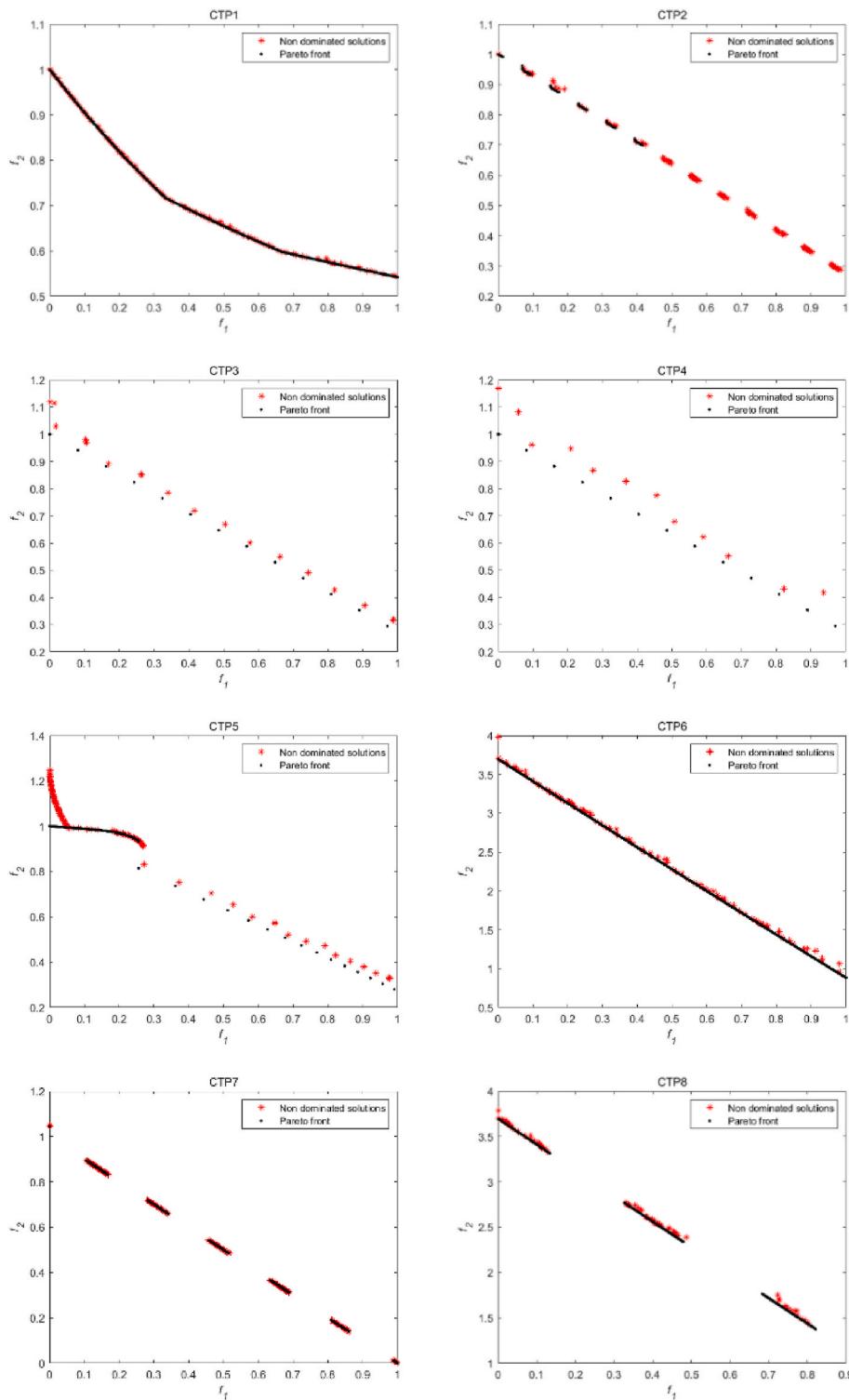


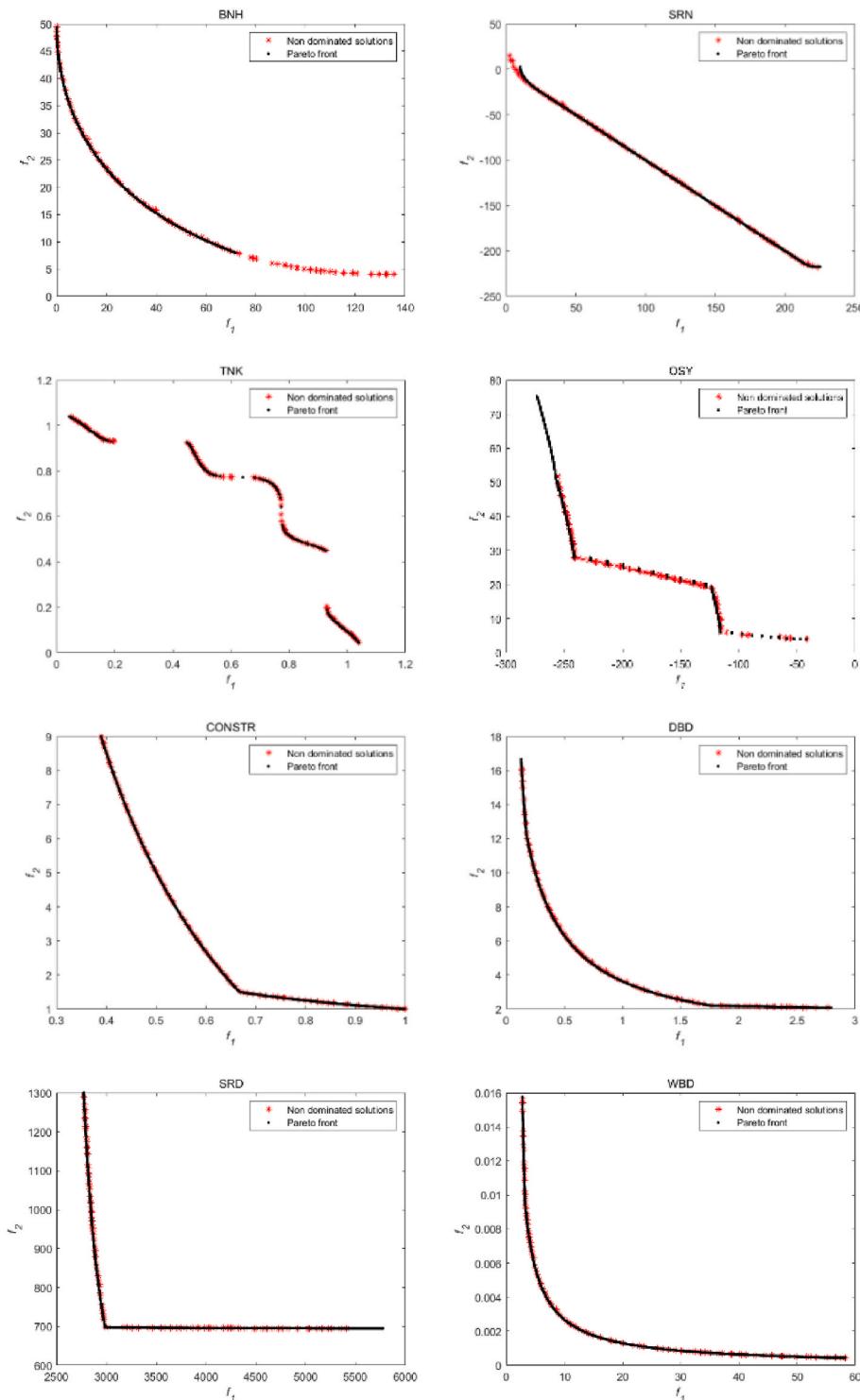
Fig. 3. The illustration of IGD.



**Fig. 4.** The Pareto front and non-dominated solutions obtained by the proposed RLMOEA on CTPs.

infeasible solutions coexist. The method takes corresponding measures to cope with three scenarios (Wang et al., 2008). Experimental results have revealed that the CDP and self-adaptive method can achieve better performance when the dimension of the decision variable is low. Conversely, the adaptive tradeoff method can obtain better results when the dimension is high (Li et al., 2016). Ensemble constraint handling technique often uses some of the above techniques, such as  $\epsilon$ -constraint, self-adaptive method, and the superiority of feasible solutions (Qu and Suganthan, 2011). A dynamic preference-based

constrained MOEA is developed, in which the Pareto dominance and CDP are dynamically adjusted to tradeoff the objective functions and constraints (Yu et al., 2022). (4) Multi-stage and multi-population. This method is increasingly becoming popular and often uses multi-stage, multi-population to evolve the population. The convergence archive and diversity archive are used for CMOPs, in which the first archive pushes the population toward the PF, and the second archive keeps the population diversity (Li et al., 2019). The Push and Pull Search (PPS) algorithm has two stages: the push and pull stages. The push stage does



**Fig. 5.** The Pareto front and non-dominated solutions obtained by the proposed RL MODE on BNH to WBD.

not consider the constraint, and the pull stage adopts the modified  $\epsilon$ -constraint method to handle the constraint (Fan et al., 2019b). A similar method is also implemented in reference (Ming et al., 2021). A two-stage CMOEA is developed, in which the first stage adopts the Pareto domination principle and the unbiased model; the second stage is to push the whole population to PF (Ming et al., 2021). ToP also uses a two-stage method, in which the first stage is to optimize a single optimization problem, and the second stage focuses on the original CMOPs (Liu and Wang, 2019). Motivated by the labor division and cooperation of the team, a dual-population CMOEA is developed for CMOPs, in

which the first objective is the actual function and the second is the degree of cv (Gao et al., 2015). A secondary population can provide useful information to the main population during evolution in dual-population optimization algorithms (Zou et al., 2021). The relationship between unconstrained Pareto front and constrained Pareto front is explored so that the learning stage and evolving stage are designed to perform evolutionary processes (Liang et al., 2023b). Two stages (exploration and exploitation) with two populations are designed to address the CMOPs with large infeasible regions; the first population offers information about the location of feasible region in the

**Table 3**

The mean and std results of IGD from six algorithms on lower dimension.

	NSGAII-CDP	NSGAIID-CDP	CMOEAD	ToP	CCMO	RLMODE
CTP1	8.14E-2 (7E-2) -	6.85E-2 (7E-2) -	1.40E-2 (1E-2) -	1.21E-2 (4E-2) -	1.37E-2 (1E-2) -	<b>3.92E-3(5E-8)</b>
CTP2	<b>3.64E-3(E-3) +</b>	5.94E-3 (3E-3) +	1.26E-2 (4E-3) -	3.42E-2 (2E-2) -	6.36E-3 (3E-3) =	6.14E-3 (2E-6)
CTP3	8.18E-2 (9E-2) -	5.16E-2 (6E-2) -	4.74E-2 (6E-3) -	7.99E-2 (2E-2) -	3.94E-2 (6E-3) -	<b>2.60E-2(2E-5)</b>
CTP4	2.78E-1 (2E-1) -	2.27E-1 (1E-1) -	2.11E-1 (3E-2) -	3.45E-1 (7E-2) -	1.76E-1 (6E-2) -	<b>1.50E-1(5E-4)</b>
CTP5	1.71E-2 (2E-2) -	1.32E-2 (4E-3) -	1.48E-2 (3E-3) -	3.63E-2 (2E-2) -	1.47E-2 (6E-3) -	<b>9.42E-3(4E-6)</b>
CTP6	<b>1.75E-2(1E-2) +</b>	7.19E-2 (8E-2) -	2.73E-2 (9E-3) -	9.28E-2 (1E-1) -	1.52E-2 (2E-3) +	1.79E-2 (1E-5)
CTP7	8.07E-3 (1E-2) -	1.75E-2 (2E-2) -	3.76E-2 (4E-2) -	4.16E-1 (3E-1) -	<b>1.30E-3(2E-4) =</b>	1.33E-3 (E-9)
CTP8	2.87E-1 (3E-1) -	2.56E-1 (2E-1) -	2.42E-1 (2E-1) -	5.04E-1 (8E-1) -	<b>1.11E-2(3.29E-3) +</b>	1.64E-2 (2E-5)
BNH	4.36E-1 (3E-2) -	3.33E-1 (2E-3) +	<b>2.63E-1(7E-3) +</b>	4.16E-1 (3E-2) =	4.22E-1 (1E-2) -	4.07E-1 (5E-4)
SRN	1.13E+0 (5E-2) =	<b>8.00E-1(1E-2) +</b>	1.95E+0 (1E-1) -	1.09E+0 (6E-2) +	8.31E-1 (1E-2) -	1.12E+0 (4E-3)
TNK	4.37E-3 (2E-4) -	5.15E-3 (3E-4) -	8.21E-3 (8E-4) -	3.90E-3 (2E-4) =	5.06E-3 (4E-4) -	<b>3.73E-3(2E-8)</b>
OSY	3.54E+0 (3E+0) -	5.94E+0 (4E+0) -	5.55E+1 (5E+0) -	9.92E+0 (4E+0) -	8.80E+0 (4E+0) -	<b>1.87E+0(1E+0)</b>
CONSTR	2.04E-2 (7E-4) -	2.19E-2 (1E-3) -	6.77E-1 (3E-2) -	<b>1.75E-2(5E-4) +</b>	1.90E-2 (7E-4) -	<b>1.75E-2(2E-7)</b>
DBD	9.91E-2 (6E-2) -	1.53E-1 (4E-2) -	1.69E+0 (2E-1) -	<b>4.89E-2(1E-3) +</b>	9.63E-2 (6E-2) -	7.50E-2 (2E-3)
SRD	<b>1.30E+1(2E+0) +</b>	2.30E+2 (8E+1) -	6.47E+2 (3E-1) -	1.18E+1 (4E+0) +	3.13E+1 (7E+1) -	1.92E+1 (2E+1)
WBD	1.83E-1 (1E-2) -	5.79E-1 (7E-2) -	1.60E+1 (2E-1) -	1.76E-1 (2E-2) -	<b>1.60E-1(3E-2) +</b>	1.69E-1 (3E-4)
+/- =	3/12/1	3/13/0	1/15/0	4/10/2	3/11/2	

exploration stage and the second population is to exploit infeasible solutions (Ming et al., 2022). For highly constrained CMOPs, infeasible solutions are divided into two subpopulations, the first is to find the feasible solution and the second is to focus on the global search for more promising regions (Sun et al., 2023).

During the past decades, identifying the relationships between objective functions and constraints has been studied by evolutionary multitasking. An evolutionary multitasking-based MOEAs for CMOPs is developed, in which one task is for the objectives without constraints, and the other is for original CMOPs(Qiao et al., 2022). A multitasking-constrained MOEA is proposed, and a dynamic auxiliary task is created to solve CMOPs by knowledge transfer (Qiao et al., 2023).

These works have achieved successful results when solving CMOPs. However, most of them neglect the relationship between the parent and offspring. The feedback mechanism can help CMOEAs find a more promising search direction, which can be finished by RL technique. A process knowledge-guided CMOEA is developed. A deep Q-learning network is applied to reflect the process knowledge (Zuo et al., 2023). Five strategies from Differential Evolution (DE) algorithm DE/best/1, DE/best/2, DE/rand/1, DE/rand/2, and DE/current-to-pbest/1 are used in the process. Deep RL dynamically adjusts DE and genetic algorithms to deal with CMOPs (Fei et al., 2024). These algorithms mainly focus on strategies to generate offspring while neglecting to adjust the parameters involved in these strategies. Based on the observation, we develop reinforcement learning-based MOEAs, which mainly focuses on adjusting the main paramters.

### 3. Reinforcement learning Multi-Objective Differential Evolution algorithm (RLMODE)

The proposed RLMODE algorithm is based on DE algorithm and RL technique. Therefore, they are firstly introduced. Then, RLMODE algorithm is elaborately discussed.

#### 3.1. DE algorithm

DE algorithm is an effective and efficient population-based evolutionary algorithm. It was proposed in 1997 (Storn and Price, 1997). At first, DE algorithm was mainly used to solve Chebyshev polynomial coefficients. Then, its robust optimization on continuous problems is identified. It has been adopted to solve various application problems because of its simplicity and flexibility. The main steps of DE are as follows (BilalPant et al., 2020).

DE evolves a population with  $NP$  individuals whose dimension is  $D$ . Each individual is denoted as  $X_i = \{X_i^1, X_i^2, \dots, X_i^D\}$ . The initial population should cover the search space as much as possible. Generally, the population is uniformly randomized in the search space  $[X_{min}, X_{max}]$  ( $X_{min} = \{X_{min}^1, X_{min}^2, \dots, X_{min}^j, \dots, X_{min}^D\}, X_{max} = \{X_{max}^1, X_{max}^2, \dots, X_{max}^j, \dots, X_{max}^D\}\}$ ).  
 $X_i = X_{min} + rand \times (X_{max} - X_{min})$       (3)

where the coefficient  $rand$  is between 0 and 1,  $j$  is the index of dimension ( $j = 1, 2, \dots, D$ ). The mutation is the second step. One of the mutation operators, DE/rand/1, is adopted, as it is robust, fast and one of the most widely used mutation operators in DE literature (BilalPant et al., 2020). The DE/rand/1 is as follows:

**Table 4**

The mean and std results of IGD results from six algorithms on LIRCMOPs.

	NSGAII-CDP	NSGAIID-CDP	CMOEAD	ToP	CCMO	RLMODE
LIRCMOP1	3.21E-1 (3.3E-2) -	3.20E-1 (3.05E-2) -	2.98E-1 (3.4E-2) -	3.51E-1 (1.7E-2) -	3.14E-1 (4.0E-2) -	<b>2.76E-1(4.2E-2)</b>
LIRCMOP2	2.77E-1 (2.9E-2) -	2.69E-1 (2.78E-2) -	2.42E-1 (2.5E-2) =	3.19E-1 (1.3E-2) -	2.62E-1 (3.8E-2) -	<b>2.31E-1(3.2E-2)</b>
LIRCMOP3	3.19E-1 (3.1E-2) =	3.22E-1 (3.47E-2) =	3.88E-1 (3.8E-2) -	3.50E-1 (5.5E-3) -	3.26E-1 (4.1E-2) =	<b>3.16E-1(4.5E-2)</b>
LIRCMOP4	2.99E-1 (2.6E-2) =	2.91E-1 (2.72E-2) =	<b>2.76E-1(3.7E-2) +</b>	3.26E-1 (9.0E-3) -	2.90E-1 (3.5E-2) =	2.95E-1 (4.3E-2)
LIRCMOP5	1.33E+0 (3.4E-1) -	1.23E+0 (7.81E-3) +	1.99E+0 (6.2E-1) -	2.12E+0 (6.1E-1) -	<b>4.17E-1(2.8E-1) +</b>	1.30E+0 (2.42E-1)
LIRCMOP6	1.40E+0 (2.6E-1) -	1.34E+0 (4.75E-4) +	1.66E+0 (5.3E-1) -	1.82E+0 (6.3E-1) -	<b>6.02E-1(3.9E-1) +</b>	1.35E+0 (2.0E-3)
LIRCMOP7	1.19E+0 (7.0E-1) +	1.34E+0 (5.83E-1) -	1.64E+0 (2.5E-1) -	1.69E+0 (4.9E-3) -	<b>1.34E-1(3.7E-2) +</b>	1.33E+0 (5.2E-1)
LIRCMOP8	1.59E+0 (3.4E-1) +	1.68E+0 (4.99E-4) -	1.77E+0 (4.2E-1) -	1.75E+0 (3.2E-1) -	<b>2.01E-1(4.1E-2) +</b>	1.65E+0 (1.8E-1)
LIRCMOP9	1.06E+0 (3.8E-2) -	1.07E+0 (4.43E-2) -	9.44E-1 (9.7E-2) -	7.35E-1 (1.3E-1) -	7.39E-1 (1.4E-1) -	<b>6.54E-1(9.9E-2)</b>
LIRCMOP10	1.00E+0 (5.7E-2) -	1.05E+0 (5.45E-2) -	9.10E-1 (6.9E-2) -	<b>6.29E-1(1.7E-1) =</b>	6.39E-1 (1.6E-1) =	7.06E-1 (1.6E-1)
LIRCMOP11	9.09E-1 (8.5E-2) -	9.79E-1 (6.96E-2) -	8.86E-1 (7.0E-2) -	5.81E-1 (1.2E-1) +	<b>3.84E-1(1.4E-1)=</b>	3.96E-1 (1.3E-1)
LIRCMOP12	9.75E-1 (1.4E-1) -	9.29E-1 (1.33E-1) -	8.07E-1 (1.26E-1) -	4.40E-1 (1.2E-1) =	<b>4.0E-1(1.2E-1) =</b>	4.57E-1 (1.4E-1)
LIRCMOP13	1.33E+0 (2.4E-3) +	1.31E+0 (1.18E-3) +	1.31E+0 (1.3E-3) +	1.41E+0 (1.7E-1) -	<b>1.04E-1(2.3E-3) +</b>	1.33E+0 (3.9E-3)
LIRCMOP14	1.28E+0 (2.4E-3) +	1.27E+0 (1.07E-3) +	1.27E+0 (9.8E-4) +	1.33E+0 (1.49E-2) -	<b>1.00E-1(9.6E-4) +</b>	1.29E+0 (3.3E-3)
+/- =	4/8/2	4/8/2	3/10/1	1/11/2	6/3/5	

**Table 5**

The mean and std results of IGD results from MODE and RLMODE.

	MODE	RLMODE
CTP1	7.22E-2 (7.63E-3)	3.92E-3(5E-8)
CTP2	4.07E-2 (1.71E-4)	6.14E-3(2E-6)
CTP3	8.00E-2 (4.04E-4)	2.60E-2(2E-5)
CTP4	3.49E-1 (3.77E-3)	1.50E-1(5E-4)
CTP5	3.98E-2 (1.58E-4)	9.42E-3(4E-6)
CTP6	9.09E-2 (9.80E-4)	1.79E-2(1E-5)
CTP7	7.06E-1 (5.18E-2)	1.33E-3(9E-9)
CTP8	3.22E-1 (4.52E-2)	1.64E-2(2E-5)
BNH	4.16E-1 (9.11E-4)	4.07E-1(5E-4)
SRN	1.12E+0(3.68E-3)	1.12E+0(4E-3)
TNK	3.55E-3(2.23E-9)	3.73E-3 (2E-8)
OSY	4.79E+0 (1.18E+0)	1.87E+0(1E+0)
CONSTR	1.78E-2 (3.43E-7)	1.75E-2(2E-7)
DBD	8.05E-2 (1.57E-3)	7.50E-2(2E-3)
SRD	7.34E+1 (3.18E+3)	1.92E+1(2E+1)
WBD	1.69E-1(2.52E-4)	1.69E-1(3E-4)

$$V_i = X_{r_1} + F \times (X_{r_2} - X_{r_3}) \quad (4)$$

where three different integers are randomly selected between 1 and  $NP$  ( $i \neq r_1 \neq r_2 \neq r_3$ ),  $F$  is a scalar factor in the range of 0 and 1. The scalar factor amplifies the difference vector  $X_{r_2} - X_{r_3}$ . The crossover operator follows the mutation operator. DE algorithm uses a binomial crossover operator on  $V_i$  and  $X_i$  to generate a trial vector  $U_i = (u_i^1, u_i^2, \dots, u_i^j, \dots, u_i^D)$  as follows:

$$u_i^j = \begin{cases} v_i^j \text{ if } rand_j \leq CR \text{ or } j == j_{rand} \\ x_i^j \text{ otherwise} \end{cases} \quad (5)$$

where the parameter  $CR$  is the crossover rate, and its range is from 0 to 1,  $j_{rand}$  is an integer in the range of 1 and  $NP$  to make sure that at least a component from  $V_i$  can be saved into  $U_i$ . At last, a greedy selection is used to make the comparison between  $U_i$  and  $X_i$  as follows:

$$X_i = \begin{cases} U_i & U_i \leq X_i \\ X_i & \text{otherwise} \end{cases} \quad (6)$$

If the newly generated vector  $U_i$  can dominate  $X_i$ ,  $U_i$  takes the place of  $X_i$ . Otherwise, the individual  $X_i$  is retained.

### 3.2. Reinforcement learning (RL) technique

RL technique is from the machine learning field, which has achieved great success in many applications, such as crop management support (Gautron et al., 2022), building energy efficiency control (Fu et al., 2022), personalized diabetes treatment recommendations (Oh et al., 2022), and so on. RL is to address the management of unknown and dynamic systems, in which agents can interact with the environment so that the reward can be incrementally accumulated. The feedback mechanism is through trial and error. The RL technique has been used to adjust the parameters of EAs and achieved better results through feedback (Hu et al., 2021; Wang et al., 2022). It has been employed to optimize the performance of evolutionary algorithms in single optimization problems, such as economic dispatch problems (Visutarrom et al., 2020) and parameter identification of photovoltaic models (Hu et al., 2021). However, the RL technique is seldom used in MOEAs.

The main ingredients of RL are the environment, learning agent, states, actions, and rewards. The relationship among the five elements is presented in Fig. 1 (Kober et al., 2013). One of the RL techniques is Q-learning. Q-learning is a model-free technique that can use the current

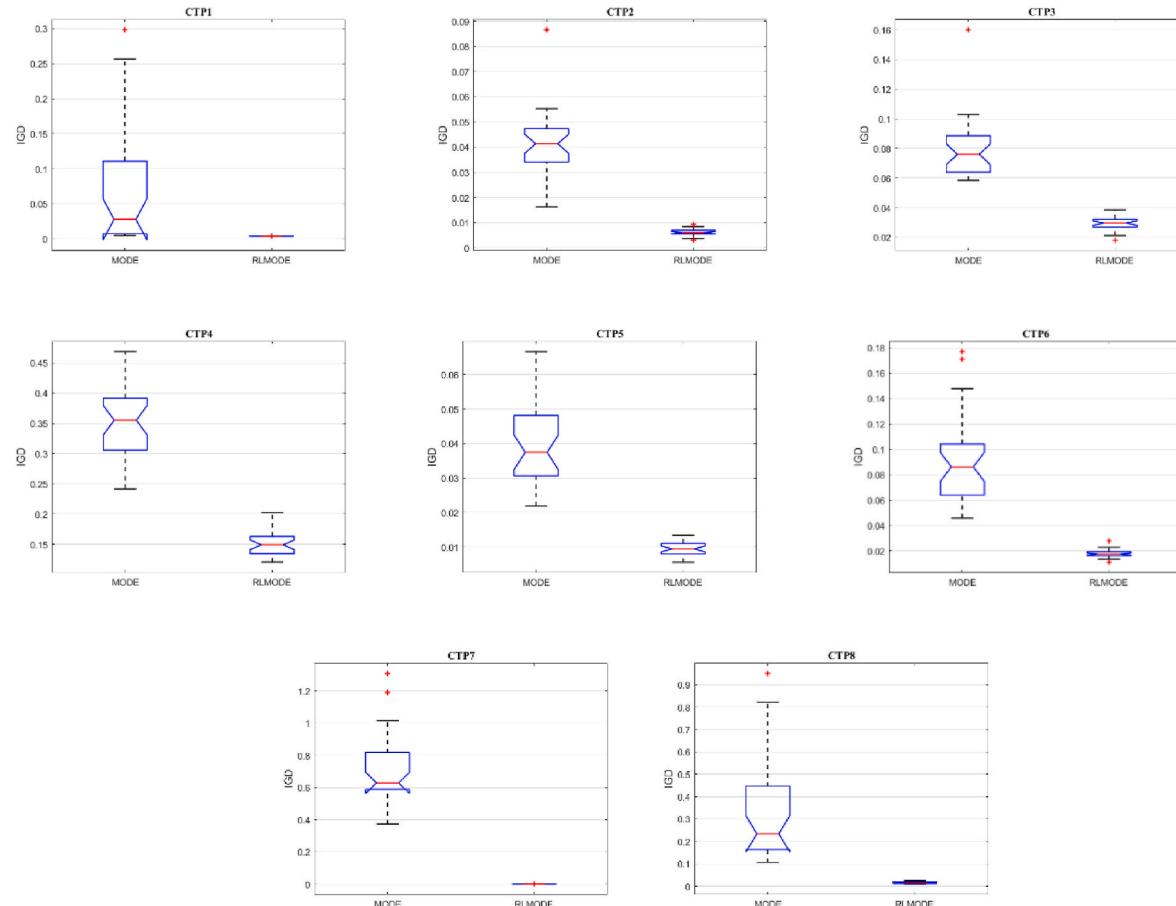
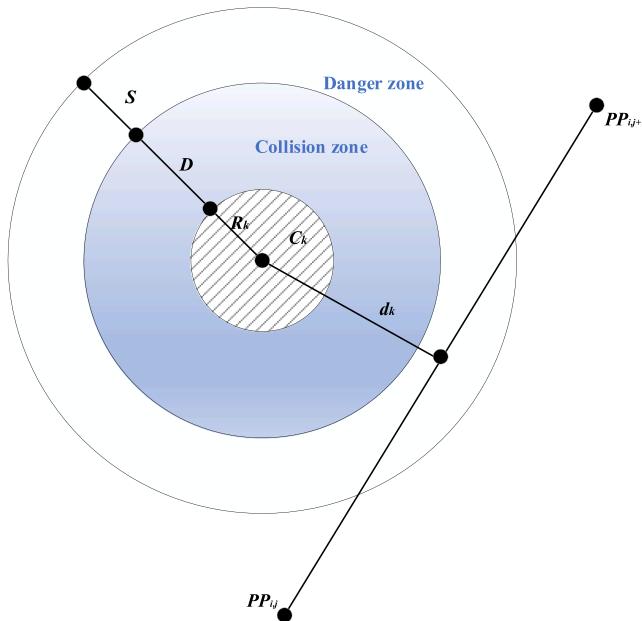


Fig. 6. Box plots of IGD indicator from MODE and RLMODE.



**Fig. 7.** The threat cost of UAV.

Q-value and reward to update the value of the action adaptively. Its main advantages are: Q-learning is simple without a complicated structure; prior knowledge is not demanded when Q-learning is used; the agent can learn and update the information in Q-table immediately. The update process can be mathematically presented as follows:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[\gamma_{t+1} + \gamma \max Q(s_{t+1}, a) - Q_t(s_t, a_t)] \quad (7)$$

where  $s_t$  corresponds to the state of the agent  $a$  in the  $t_{th}$  iteration,  $a_t$  denotes the action that the agent  $a$  takes,  $Q_t(s_t, a_t)$ ,  $Q(s_{t+1}, a)$  and  $Q_{t+1}(s_t, a_t)$  are corresponding Q-values in the Q-table,  $\alpha$  is the learning rate in the range of 0 and 1,  $\gamma_{t+1}$  is the reward,  $\gamma$  is the discount factor between 0 and 1. The agent  $a$  chooses the action according to the Q-table and gets the corresponding reward for the action to update the Q-table. The framework of Q-learning is presented as follows:

#### Algorithm 1. Q-learning

---

**Algorithm 1** Q-learning

---

Input the parameters: $\alpha, \gamma$	Output the state $s$
--	----------------------

---

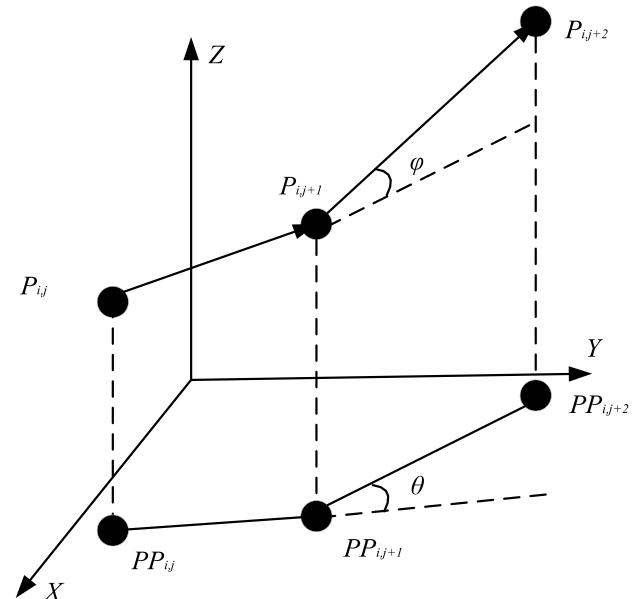
```

1 Initialize the Q-table;
2 Randomly choose a state  $s_t$ ;
3 While(the stop criteria are not met)
4     Choose the best action  $a_t$  for the current state  $s_t$  from Q-table;
5     Perform the action  $a_t$  and get the corresponding reward  $r$ ;
6     Identify the maximal Q-value of the following state  $s_{t+1}$ ;
7     Update the Q-table according to Eq.(7);
8     Update the state  $s_t = s_{t+1}$ ;
9 end

```

---

As discussed in the literature review, there are many MOEAs and constraint-handling techniques to handle CMOPs. For example, NSGAII uses the non-dominated sorting and crowding distance sorting mechanism to select population (Deb and Jain, 2014; Deb et al., 2002). CDP



**Fig. 8.** The demonstration of the turning and climbing angles.

technique gives more opportunities for feasible solutions (Deb et al., 2002). However, most of these MOEAs and constraint-handling techniques do not consider the relationship between the parent and its corresponding offspring. The relationship can reveal some implications concerning evolution and search direction. For example, if the offspring can dominate its corresponding parent, it indicates that the current search direction is more promising. Otherwise, it may be invalid. The feedback mechanism has been adopted in the single optimization problem (Cheng-Hung and Chong-Bin, 2018; Gautron et al., 2022; Hu et al., 2021; Oh et al., 2022; Wang et al., 2022). Therefore, we introduce the method into CMOEAs.

### 3.3. RL MODE

#### (1) Q-learning in RL MODE

In our proposed RL MODE algorithm, Q-table is used. It is a matrix

whose column represents action  $a$  and row represents state  $s$ , as shown in Table 1.

The individual is defined as the agent in RL MODE. For each agent  $a$ , the SoftMax function is often employed to determine which action

**Table 6**

The obstacle of UAVs.

Obstacle	Center	Radius
1	(400,500,100)	70
2	(600,200,150)	60
3	(500,350,150)	70
4	(350,200,150)	60
5	(700,550,150)	60
6	(650,750,150)	70

**Table 7**

The max, min, mean, and std of HV values from five algorithms.

Algorithm	Max	Min	Mean	Std	P - value	Sig.
NSGAII-CDP	0.3436	0	0.254	5.2E-3	0.00195	+
NSGAIID-CDP	0.3303	0	0.212	1.30E-2	0.001953	+
ToP	0.2728	0	0.175	2.9E-3	0.001953	+
CCMO	<b>0.4048</b>	0.3729	<b>0.391</b>	<b>1.1E-4</b>	1.0000	=
RLMODE	0.4040	<b>0.3772</b>	0.390	1.8E-4		

should be taken in the state  $s$ . The SoftMax function is as follows:

$$\pi(s_j, a_j) = \frac{e^{\frac{Q_t(s_j, a_j)}{T}}}{\sum_{j=1}^n e^{\frac{Q_t(s_j, a_j)}{T}}} \quad (8)$$

where  $Q_t(s_j, a_j)$  is the Q-value in the Q-table at the  $t_{th}$  iteration,  $n$  is the amount of the action. Based on values in the Q-table, the agent can compute the probability of the action it takes.

## (2) The feedback mechanism through RL

Through DE algorithm, the offspring  $U_i$  is generated by the parent  $V_i$ . As the constraint is considered, feasible and infeasible solutions may coexist. Hence, we have the following eight scenarios, listed in Table 2.

The eight scenarios can be divided into three categories. The first one includes the first, fifth, and sixth state, in which the offspring is superior to the parent. The observation indicates that the current local search direction is effective. We can further strengthen the local search so the mutant vector can have more opportunities to move towards the Pareto

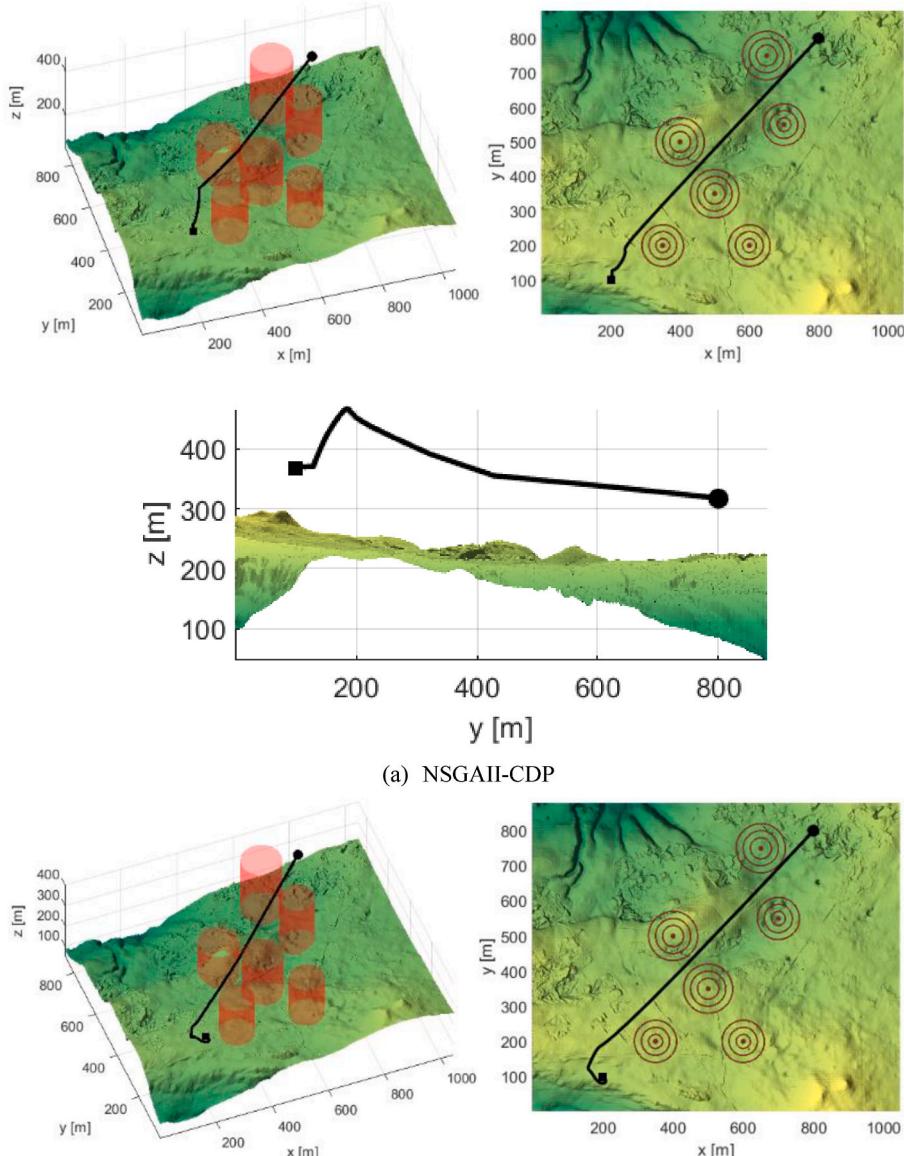


Fig. 9. The 3D view, top view and side view from five algorithms.

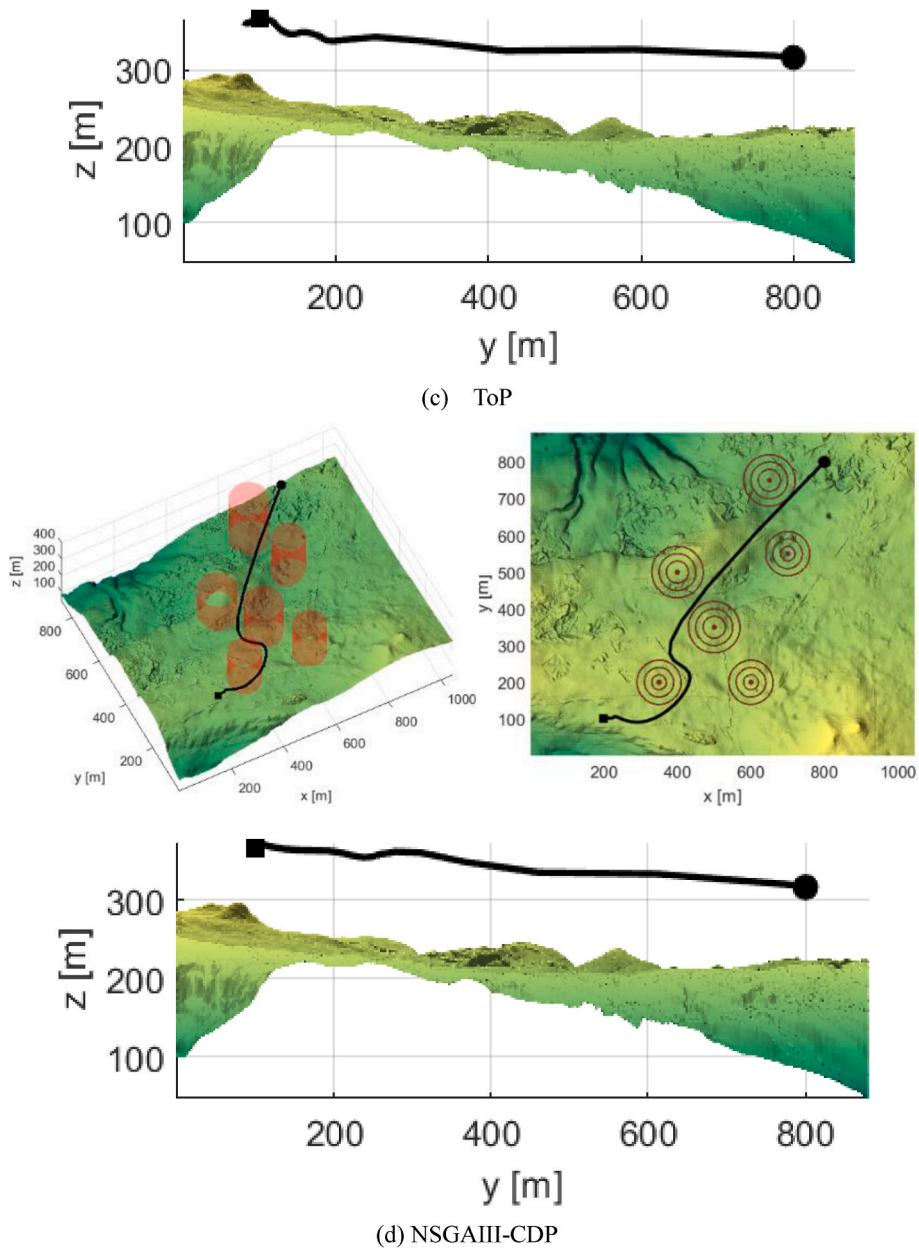


Fig. 9. (continued).

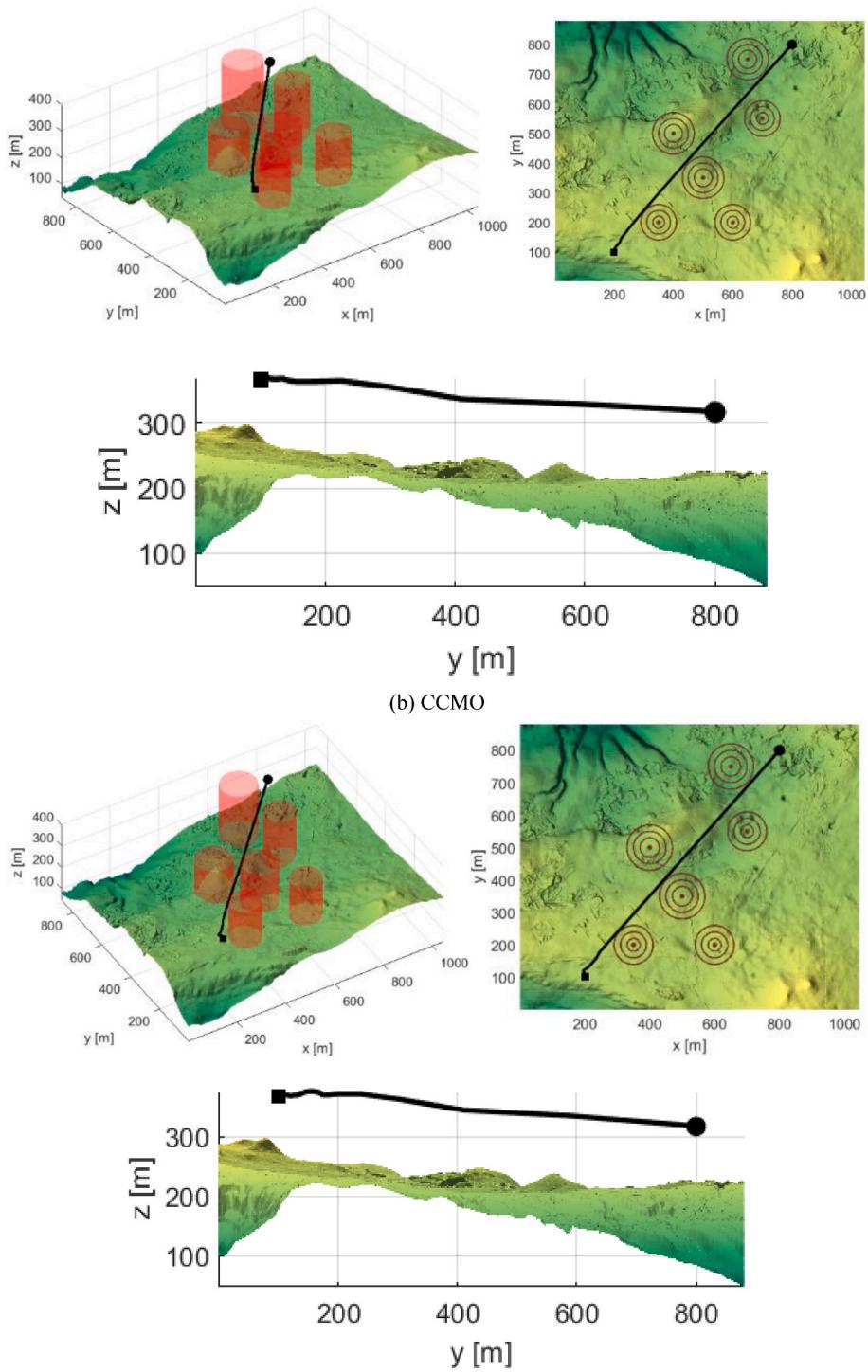


Fig. 9. (continued).

front. As the search direction is correct, the reward value is 1. The mutation scalar factor  $F$  can be further reduced, and the crossover rate  $CR$  can be increased to enhance the local search. So, we set  $F_f = -0.1$ ,  $CR_f = 0.1$ . The second category includes the second, fourth, and seventh state in Tables 2 and in which the offspring is inferior to the parent. The current local search direction may be invalid. We can strengthen the global search by increasing the scalar factor  $F$  and increasing the crossover rate  $CR$  to push the algorithm to jump out of the local optimal. So, we set  $F_f = 0.1$ ,  $CR_f = 0.1$ . As the search direction is incorrect, the

reward value is  $-1$ . The remaining scenarios belong to the third category, in which the parent and offspring cannot dominate each other. They have the same statuses. We set the reward value to 0. The scalar factor  $F$  and crossover rate  $CR$  remain the same,  $F_f = 0$ ,  $CR_f = 0$ .

Therefore, three corresponding actions are:  $F_f = -0.1$ ,  $CR_f = 0.1$ ;  $F_f = 0.1$ ,  $CR_f = 0.1$ ;  $F_f = 0$ ,  $CR_f = 0$ . These actions are converted into the feedback mechanism to update the parameters of RLMODE as follows:

$$F = F_f + F \quad (9)$$

$$CR = CR_f + CR \quad (10)$$

The updated parameters are used to generate the offspring vector  $U_i$ . The SoftMax function is used to determine which action should be taken according to Eq. (8). The objective values and constraints of  $U_i$  can be calculated. As the CDP is simple and robust, we adopt it here to make the comparison between the generation vector  $U_i$  and the original vector  $X_i$ . If the generated vector can dominate the parent, the reward value is set to 1, and the state changes to 1. If the parent can dominate the generated vector, the state changes to 2; if they can't dominate each other, the state is set to 3. At last, we update the Q-table according to Eq. (7), and  $NP$  individuals are selected from the set of offspring  $U_i$  and the parent  $X_i$  by the non-dominated crowd sort. Then, the next iteration begins with the  $NP$  individuals and updated parameters. As each individual may have

different states and take different actions, the values of their  $F$  and  $CR$  are also different during iteration.

### (3) The framework of RLMODE algorithm

The proposed RLMODE algorithm is established based on the RL technique, MOEAs and CDP. At first, the initial population is randomly initialized in the search range according to Eq. (3). Two parameters  $F$  and  $CR$  are initialized in the range of 0 and 1. Then, the population enters into evolves. The  $F$  and  $CR$  are updated, and the offspring are generated by Eqs. (4) and (5). The RL technique is used to update the Q-table, state, and reward. The framework of the proposed RLMODE algorithm is presented in Algorithm 2. The corresponding flowchart is shown in Fig. 2.

#### Algorithm 2. RLMODE

---

Input: Population size and the problem
Output: Non-dominated solutions

---

```

1 Initialize the population  $X_i$  in the search range;  $i = 1, 2, \dots, NP$ 
2 for  $i = 1: NP$ 
3    $F(i) = random(0,1); F_f(i) = 0;$ 
4    $CR(i) = random(0,1); CR_f(i) = 0;$ 
5 end
6 While(the stop criterion is not met)
7   for  $i = 1: NP$ 
8      $F(i) = F(i) + F_f(i);$ 
9      $CR(i) = F(i) + CR_f(i);$ 
10    if  $F(i) > 1$  or  $F(i) < 0$ 
11       $F(i) = rand;$ 
12    end
13    if  $CR(i) > 1$  or  $CR(i) < 0$ 
14       $CR(i) = rand;$ 
15    end
16    Generate the offspring  $U_i$  by Eqs.(4-5);
17    Calculate the selection probability of the action for the agent  $X_i$ ;
18    Evaluate the objective value and constraint of  $U_i$ ;
19    if  $U_i$  dominates  $X_i$ 
20      reward = 1;
21      state = 1;
22      elseif  $X_i$  dominates  $U_i$ 
23        reward=-1;
24        state=2;
25      else
26        reward=0;
27        state=3;
28    end
29  end
30 for  $i = 1: NP$ 
31   Update the Q-table
32 end
33  $X =$  non-dominated crowd sort( $X \cup U$ )
34 end

```

---

#### 4. Experimental study

The experiments are performed on thirty test benchmark functions, which are divided into three types according to their characteristics. The first type is CTPs(CTP1~CTP8) (Deb et al., 2001). The dimension  $D$  of the decision variable is four and the  $g(x)$  is the well-known Rosenbrock in the CTPs as follows:

$$\left\{ \begin{array}{l} f_1(x) = x_1 \\ f_2(x) = g(x)(1 - \frac{f_1(x)}{g(x)}) \\ g(x) = \sum_{i=1}^{D-1} 100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \\ c(x) = \cos(\theta)(f_2(x) - e) - \sin(\theta)f_1(x) \geq \\ a|\sin(b\pi(\sin(\theta)(f_2(x) - e) + \cos(\theta)f_1(x))^c)|^d \end{array} \right. \quad (11)$$

where  $\theta, a, b, c, d, e$  are six parameters. Different values correspond to different CTP problems. The second is mainly from real-world applications like BNH, SRN, TNK, OSY, CONSTR, DBD, SRD, and WBD (Liu et al., 2019). The third group is large infeasible region (LIR) CMOPs (LIRCMOPs). All the problems in the LIRCMOPs have large infeasible regions. The shape and distance functions are used to form the objective functions (Fan et al., 2019a).

Inverted Generational Distance (IGD) is chosen as the performance indicator because it can reflect the diversity and convergence of the  $PF$  simultaneously (Qingfu and Hui, 2007). The definition of IGD is as follows:

$$\left\{ \begin{array}{l} IGD = \frac{\sum_{y^* \in P^*} d(y^*, A)}{|P^*|} \\ d(y^*, A) = \min_{y \in A} \sqrt{\sum_{i=1}^m (y_i^* - y_i)^2} \end{array} \right. \quad (12)$$

where  $P^*$  is the uniformly selected solutions from the  $PF$ ,  $|P^*|$  is the number of the selected solutions,  $A$  is the non-dominated solutions from MOEAs,  $y$  is the single solution in  $A$ ,  $d(y^*, A)$  denotes the minimal distance between the solution  $P^*$  and the solution  $y$ ,  $m$  is the number of the objective, as shown in Fig. 3. Better MOEAs tend to obtain a smaller IGD.

Five CMOEAs, NSGAII-CDP (Deb et al., 2002), NSGAIII-CDP (Deb and Jain, 2014), CMOEA/D (Jain and Deb, 2014), ToP (Liu and Wang, 2019), and CCMO(Tian et al., 2021) are selected as the opponents of the proposed RL MODE. NSGAII-CDP, NSGAIII-CDP, and CMOEA/D are very representative algorithms. CCMO and ToP belong to multi-stage and multi-population, which are state-of-the-art constrained MOEAs.

- (1) NSGAII-CDP is built based on NSGAII and CDP, which is one of the main constraint-handling techniques (Deb et al., 2002).
- (2) NSGAIII follows the framework of NSGAII and introduces reference points to select offspring. NSGAIII-CDP also uses the constraint domination method to deal with constraints (Deb and Jain, 2014).
- (3) CMOEA/D is the extension of MOEA/D, which is one of the decomposition MOEAs (Jain and Deb, 2014).
- (4) ToP considers the constraints in the decision space and objective space simultaneously, which makes the algorithm different from the others (Liu and Wang, 2019).
- (5) CCMO utilizes double populations to solve CMOPs. The first population is based on the original CMOP, while the second population is for a helper problem derived from the original one (Tian et al., 2021).

The parameters are set as follows: the population size = 100 and the maximal iteration = 200. The distribution index of SBX and the

polynomial mutation is 20. These algorithms are run in the PlatEMO, where the parameters of the five algorithms are equal to their original references so that the best performance can be guaranteed (Tian et al., 2017). We run 30 times for each algorithm so that the randomness can be avoided.

##### 4.1. The results of benchmark functions

Non-dominated solutions can be obtained for each algorithm during each run. On the basis of Eq. (12), the performance indicator IGD can be computed. When the thirty runs are finished, the mean and standard deviation of IGD can be obtained. The values of these mean, and standard deviation (std) of IGD from NSGAII-CDP, NSGAIII-CDP, CMOEA/D, ToP, CCMO and RL MODE are shown in Table 3, in which the best results are in bold. Small mean IGD indicates that the non-dominated solutions generated by CMOEAs are diverse with good convergence, which can uniformly distribute along the Pareto front. Small std reveals that the fluctuation of IGD from thirty runs is slight, and the algorithm is very robust.

- (1) For the first group CTPs, the proposed RL MODE algorithm has obtained the best results on four functions among eight functions, i.e., CTP1, CTP3, CTP4, and CTP5. NSGAII-CDP has achieved minimal IGD results on CTP2 and CTP6. CCMO has gotten the best results on CTP7 and CTP8. Therefore, the proposed RL MODE can be ranked as the top one. Both NSGAII-CDP and CCMO are the second. Based on the definition of CTPs, their complexity is controlled by  $\theta, a, b, c, d, e$  and a multi-modal function  $g(x)$ , which can make the Pareto-optimal set to become a collection of several discrete regions. At the extreme, the Pareto-optimal region may change into a set of discrete solutions. With the increasing number of these disconnected regions, CMOEA/D and ToP algorithms have difficulty finding representative solutions in those disconnected regions. However, the proposed RL MODE can move towards these regions with the help of the RL technique, which can adaptively adjust the search direction. When the search direction is valid, the reward is set to 1 so that the search direction can be selected more frequently. On the opposite, the search direction will be changed by penalizing the reward. The non-dominated solutions from the proposed RL MODE are plotted in Fig. 4. It is clear that these solutions are uniformly distributed along the PF, which reveals that the search capability is powerful and reliable.
- (2) For the second group, there are eight functions. The decision dimension of SRD is seven, which is the highest among them. The six algorithms don't achieve better results as the more the decision dimension, the more complex the algorithm encounters. NSGAII-CDP has achieved the best IGD on the SRD. The proposed RL MODE has achieved the best IGD results on three functions, i.e., TNK, OSY, and CONSTR. ToP, NSGAIII-CDP, and CCMO have outperformed the remaining algorithms on two functions (CONSTR and DBD), one function (SRN) and one function (WBD). Therefore, the proposed RL MODE has outperformed the five algorithms in this group. The non-dominated solutions from RL MODE are listed in Fig. 5. Different from CTPs, the Pareto front of these functions is almost continuous, which makes the algorithm find more representative non-dominated solutions. We can see that the proposed RL MODE can obtain well-distributed non-dominated solutions in this group.
- (3) For the third group, there are fourteen LIRCMOPs. The decision dimension of these problems is thirty, and the objective dimension of LIRCMOP1-LIRCMOP12 is two, while the objective dimension of LIRCMOP13 and LIRCMOP14 is three. CCMO and RL MODE have performed better on these LIRCMOPs, especially CCMO. For LIRCMOP1-LIRCMOP4, the feasible region is very small in the entire search space, making it difficult for CMOEAs to

jump out of infeasible regions. Regarding IGD results, RLMODE is significantly superior to its five rivals. For the rest of LIRCMOPS, CCMO has obtained the best IGD on eight functions. CCMO adopts multi-population to solve CMOPs, in which one is to evolve the original CMOPs, and the other is to solve a helper problem derived from the original one. The two populations evolve together and share helpful information. The specially designed mechanism can contribute to good performance. The proposed RLMODE can be ranked second in this group.

The Wilcoxon signed-rank test is employed to validate the difference between the proposed RLMODE and its five opponents. The results are also presented in Tables 3 and 4, in which flag “+” indicates that these opponents are superior to the proposed RLMODE, flag “-” has the opposite meaning, and the “=” denotes that they have similar performances. Based on the statistical results in the last rows of Tables 3 and 4, the proposed RLMODE has achieved the best results in the first two groups and the second place in the third group among the six algorithms, which statistically proves the outstanding search capability of the proposed RLMODE. The RL technique can contribute to the superior performance. As it is still challenging to determine appropriate values of main parameters for CMOEAs, RLMODE can adaptively adjust them to balance the local and global search.

#### 4.2. Discussion

To further validate the effectiveness of the proposed mechanism, we compare our proposed RLMODE with the original Multi-Objective DE (MODE). The main parameters of MODE are  $F = 0.5$  and  $CR = 0.9$ . The environmental setting of the experiment is the same as the above. MODE and RLMODE run thirty times independently. The IGD results from RLMODE and MODE are listed in Table 5. It can be found that RLMODE has obtained better results on fifteen benchmark functions except for TNK, MODE has achieved more minor IGD results on three functions SRN, TNK, and WBD. However, the difference between MODE and the proposed RLMODE is insignificant in TNK function. As the relationship between the offspring and parent is not taken into consideration during evolution, the performance of MODE cannot be guaranteed. Especially for CTPs, the constant parameter setting may deteriorate the performance when CMOPs are very complex.

The box plots are further used to compare the difference between MODE and the proposed RLMODE on CTPs, shown in Fig. 6. The figure demonstrates that the proposed RLMODE algorithm has smaller IGD values for all CTPs. The span distribution of the proposed RLMODE is much smaller. The distance between the top and bottom borders is shorter, which demonstrates the effectiveness of the proposed RLMODE. On the opposite, the span distribution of MODE is more extensive, which indicates that MODE is not robust and reliable.

### 5. The application of UAV path planning

This Section applies the RLMODE algorithm to solve a constrained application problem. Firstly, the model is introduced. Then, steps on how to solve the model are presented. At last, the testing scenario and results of the above algorithms are given.

#### 5.1. Constrained multi-objective UAV path planning model

Various kinds of natural disasters, such as earthquakes and landslides, have given rise to the loss of many lives. When a disaster happens, the most critical issue is to preserve lives. The first 72 h are of great importance. The rescue operations must be conducted immediately. The major problem is the lack of disaster situational awareness. UAVs are very useful tools to learn about situational awareness through monitoring the disaster areas (Erdelj et al., 2017). The path planning is one of the most critical problems when UAVs are used. Several objectives and

constraints have to be considered simultaneously (Phung and Ha, 2021).

Let's  $P_{ij} = (x_{ij}, y_{ij}, z_{ij})$  be the discrete points of the path generated by the above algorithms. In many applications, we hope that the length of the UAV flies should be shorter so that the situation of the destination can be known quickly. Therefore, the first objective of the problem is to minimize the length of UAVs as follows:

$$F_1(x_i) = \sum_{j=1}^{N-1} \left\| \overrightarrow{P_{ij}P_{ij+1}} \right\| \quad (13)$$

where  $N$  is the number of the discrete points along the path, the operator is the Euclidean distance between the two continuous points  $P_{ij}$  and  $P_{ij+1}$ . Generally, there are many obstacles and threats along the path. If we pursue to minimize the length of UAVs, UAVs may encounter some threats. Therefore, we have to consider the safety of UAVs. Let's  $K$  be the number of the dangers, and each of them is assumed to be a cylinder with the center  $C_k$  and the radius  $R_k$  in Fig. 7. In Fig. 7, the line  $\left\| \overrightarrow{PP_{ij}PP_{ij+1}} \right\|$  is the projection of the segment  $\left\| \overrightarrow{P_{ij}P_{ij+1}} \right\|$ ,  $D$  is the diameter of the collision zone,  $R$  is the diameter of the danger zone,  $d_k$  is the distance between the center  $C_k$  to the line  $\left\| \overrightarrow{PP_{ij}PP_{ij+1}} \right\|$ . The threat  $C_k$  to the line  $\left\| \overrightarrow{PP_{ij}PP_{ij+1}} \right\|$  can be computed as follows:

$$T_k \left( \left\| \overrightarrow{P_{ij}P_{ij+1}} \right\| \right) = \begin{cases} \infty & \text{if } d_k \leq d + R_k \\ 0 & \text{if } d_k > S + d + R_k \\ S + d + R_k - d_k & \text{others} \end{cases} \quad (14)$$

The second objective function, i.e., total threats of UAVs, can be summarized as follows:

$$F_2(x_i) = \sum_{j=1}^{N-1} \sum_{k=1}^K T_k \left( \left\| \overrightarrow{P_{ij}P_{ij+1}} \right\| \right) \quad (15)$$

where  $K$  is the number of threats, and  $N$  is the amount of discrete points to denote the UAV path.

Generally speaking, UAVs should fly at a certain height. If the height is too high, it may be difficult to obtain high-resolution images. On the opposite, too low height may result in a collision with the obstacle (Phung and Ha, 2021). We suppose that the lower and upper heights of UAVs are  $h_{min}$  and  $h_{max}$ . It is expected that UAVs should fly at the height of  $(h_{min} + h_{max})/2$ . The altitude cost can be computed as follows:

$$H_{ij} = \begin{cases} |z_{ij} - (h_{min} + h_{max})/2| & h_{min} \leq z_{ij} \leq h_{max} \\ \infty & \text{others} \end{cases} \quad (16)$$

where  $z_{ij}$  is the  $z$ -coordinate of the point  $P_{ij}$ . The third objective function can be summarized as follows:

$$F_3(x_i) = \sum_{j=1}^N H_{ij} \quad (17)$$

Finally, the operation of UAVs should be noted. As the mechanics of UAVs are limited, the turning and climbing angles should be less than a predefined angle  $\varphi$ . The turning angle  $\theta_{ij}$  and climbing angle  $\varphi_{ij}$  can be demonstrated in Fig. 8, in which  $PP_{ij}$ ,  $PP_{ij+1}$ , and  $PP_{ij+2}$  are the projection of the three continuous-discrete points  $P_{ij}$ ,  $P_{ij+1}$ , and  $P_{ij+2}$ . The turning angle and climbing angle can be calculated as follows:

$$\theta_{ij} = \arctan \left( \frac{\overrightarrow{PP_{ij}PP_{ij+1}} \times \overrightarrow{PP_{ij+1}PP_{ij+2}}}{\overrightarrow{PP_{ij}PP_{ij+1}} \bullet \overrightarrow{PP_{ij+1}PP_{ij+2}}} \right) \quad (18)$$

$$\varphi_{ij} = \arctan \left( \frac{z_{ij+1} - z_{ij}}{\overrightarrow{PP_{ij}PP_{ij+1}}} \right) \quad (19)$$

where the  $z_{ij}$  and  $z_{ij+1}$  are  $z$ -axis of points  $PP_{ij}$  and  $PP_{ij+1}$ . The corresponding constraints can be obtained as follows:

$$h_{ij}^1 = \begin{cases} 0 & \text{abs}(\theta_{ij}) \leq \varphi \\ \text{abs}(\theta_{ij}) & \text{others} \end{cases} \quad (20)$$

$$h_{ij}^2 = \begin{cases} 0 & \text{abs}(\varphi_{ij+1} - \varphi_{ij}) \leq \varphi \\ \text{abs}(\varphi_{ij+1} - \varphi_{ij}) & \text{others} \end{cases} \quad (21)$$

$$h(x_i) = \sum_{j=1}^{N-2} (h_{ij}^1 + h_{ij}^2) \quad (22)$$

where  $\varphi$  is the predefined angle,  $g(x_i)$  is the total constraint. Finally, the problem can be formatted as follows:

$$\begin{cases} F(x_i) = (F_1(x_i), F_2(x_i), F_3(x_i)) \\ h(x_i) = 0 \end{cases} \quad (23)$$

### 5.2. Solving the constrained multi-objective UAV path planning model by RLMODE

The proposed RLMODE and the five MOEAs are adopted to deal with the path planning of UAVs. The main steps are often needed as follows.

- (1) Define the test scenario. The position of the start and destination of UAVs should be predefined. We use 3-D coordinates to represent them, which makes the simulation more practical.
- (2) Represent the UAV path. Generally, a UAV path is encoded as a set of vectors. Each denotes the route of the UAV from the start point to the destination. We adopt the spherical coordinate system with three components:  $\rho \in (0, \text{path\_length})$ , elevation angle  $\psi \in (-\pi/2, \pi/2)$ , and the azimuth angle  $\varphi \in (-\pi, \pi)$ . The path can be encoded as follows:  $(\rho_{i1}, \psi_{i1}, \varphi_{i1}, \rho_{i2}, \psi_{i2}, \varphi_{i2}, \dots, \rho_{iN-2}, \psi_{iN-2}, \varphi_{iN-2})$ , where  $N$  is the amount of the discrete points along the path. As the start and destination points are predefined,  $N - 2$  discrete points have to be generated by algorithms.
- (3) Establish the constrained model. In the previous Section, the model has been established with three objectives and a constraint.
- (4) Optimize the model by RLMODE. We define the number of discrete points. Serial discrete points can represent the UAV path. The objective values and corresponding constraints can be computed by Eq. (23).
- (5) Output the non-dominated solutions. When the stop condition is met, algorithms can produce some non-dominated solutions. We can use these solutions to generate UAV paths.

### 5.3. Testing scenario and result analysis

As the Pareto front of the model of Eq. (23) is unavailable, the IGD cannot be used in this problem. We adopt Hypervolume ( $HV$ ) as the performance indicator, which is also widely used in MOEAs.  $HV$  is to calculate the volume covered by non-dominated solutions from MOEAs, as shown in Eq. (24) (Zitzler and Thiele, 1999).

$$HV = \text{volume} \left( \bigcup_{i=1}^{|P_A|} v_i \right) \quad (24)$$

where  $|P_A|$  is the number of non-dominated solutions. The larger the  $HV$ , the better the MOEAs. We use a real scenario to test the model and the above six algorithms, which is from Christmas Island in Austria (Phung and Ha, 2021). There are six obstacles for the model in Table 6. We select ten 3-D points plus the start and ending points to generate serial continuous-discrete 3-D points representing the UAV path through the spherical coordinate system.

The start and ending points are [200, 100, 150] and [800, 800, 150]. Then, the objective values and corresponding constraints can be computed through Eq. (23). The maximal iteration is 400, and the remaining parameters are the same as in the above experiments. Each algorithm runs thirty times independently.

The maximal (Max), minimal (Min), mean, and Std of HV from each algorithm are demonstrated in Table 7. Note that the Max HV value can

verify whether the algorithm is able to find a better set of non-dominated solutions. In terms of the Max HV values, CCMO has achieved the best result as the two populations of the algorithm can balance the optimization objectives and handling constraints. The proposed RLMODE can be ranked second place. With regard to the Mean and Min HV, it can be noted that the proposed RLMODE can achieve the best result. As the comparison information between the parent and the offspring is taken into consideration during evolution, the  $F$  and  $CR$  of RLMODE are adaptively adjusted by the RL technique. The feedback mechanism can help individuals move towards the Pareto front. As the min HV of NSGAI-CDP, ToP, and NSGAI-III-CDP is 0, it indicates that they cannot find any feasible solutions during evolution. The constraint of the problem makes the feasible region much smaller, which causes much trouble to these CMOEAs. For the Std of HV values, CCMO and RLMODE have achieved the top two performances, which reveals that the two algorithms are very robust. Based on the statistical results and the above observations, CCMO and RLMODE are the most competitive algorithms.

Then, the Wilcoxon signed Ranks test based on these HV values is also performed to test the significant difference between RLMODE and its four opponents. The comparison results are shown in Table 7,  $P$ -value is to represent the significance of whether the hypothesis should be abandoned. The flags “=” and “+” indicate that the proposed RLMODE is similar to and superior to its opponents. From Tables 7 and it can be observed that the proposed RLMODE is equal to CCMO, and better than the remaining algorithms. However, the framework of CCMO is more complicated than the proposed RLMODE, which has two populations. The first is to solve the original CMOPs, while the second is to address a helper problem derived from the original one. On the contrary, the proposed RLMODE is simple and easily implemented without any extra parameters. The information between the parent and offspring is taken into consideration. This feedback mechanism is realized by the RL technique.

We select each solution from the non-dominated set obtained by NSGAI-CDP, NSGAI-III-CDP, ToP, CCMO, and RLMODE, respectively. We use these solutions to generate the UAV path. The 3D, top, and side views are plotted in Fig. 9. It can be found that the UAV paths generated by CCMO and the proposed RLMODE are much smoother than NSGAI-CDP, NSGAI-III-CDP, and ToP in terms of the three views. The length of two paths from the two algorithms is shorter than that of three. The finding is consistent with the HV result, which further proves that the proposed RLMODE is a reliable and effective algorithm.

## 6. Conclusion

In this paper, a RL-based MODE algorithm, RLMODE, is developed. The algorithm uses the information between the parent and its offspring to adjust the search direction adaptively with the help of RL technique. Eight different scenarios are considered between the parent and offspring. The RL technique implements the process, which can incrementally and autonomously learn knowledge by interacting with the environment. The main characteristic of the proposed RLMODE is straightforward without setting values of scalar factor  $F$  and crossover rate  $CR$ . The proposed RLMODE and five representative MOEAs are used to deal with thirty benchmark functions. According to the performance indicator IGD, the proposed RLMODE is superior to its five rivals in the first two groups. Then, we apply the proposed RLMODE algorithm to solve the UAV path problem, in which three objectives and a constraint are involved. RLMODE has achieved better UAV paths, which indicates that the algorithm is competitive. Therefore, RLMODE is a reliable and alternative algorithm when solving CMOPs.

To further develop the performance of RLMODE, we can conduct the following research: (1) two main parameters are automatically adjusted by RL technique in our algorithm. Efficient strategies are also critical for CMOEAs when generating offspring. We also need to select efficient strategies for CMOEAs through RL technique automatically. (2) Deep RL approach is a new emerging technique that has been applied in

optimization problems. We can use the approach to reestablish the framework of our CMOEAs.

### CRediT authorship contribution statement

**Xiaobing Yu:** Data curation, Investigation, Writing – original draft.  
**Pingping Xu:** Reviewing and Editing. **Feng Wang:** Reviewing and Editing. **Xuming Wang:** Reviewing and Editing.

### 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

Data will be made available on request.

### Acknowledgements

This research was funded by National Natural Science Foundation of China (No. 72274099, No. 71974100), Humanities and Social Sciences Fund of the Ministry of Education, China (No. 22YJC630144), Qing Lan Project of Jiangsu province (R2023Q08).

### References

- Asafuddoula, M., Ray, T., Sarker, R., Alam, K., 2012. An adaptive constraint handling approach embedded MOEA/D. In: 2012 IEEE Congress on Evolutionary Computation, pp. 1–8.
- Bader, J., Zitzler, E., 2011. HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.* 19, 45–76.
- Bilal, Pant, M., Zaheer, H., Garcia-Hernandez, L., Abraham, A., 2020. Differential Evolution: a review of more than two decades of research. *Eng. Appl. Artif. Intell.* 90.
- Cheng-Hung, C., Chong-Bin, L., 2018. Reinforcement learning-based differential evolution with cooperative coevolution for a compensatory neuro-fuzzy controller. *IEEE Transact. Neural Networks Learn. Syst.* 29, 4719–4729.
- Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B., 2016. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* 20, 773–791.
- Deb, K., Jain, H., 2014. 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, 577–601.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multi-objective genetic algorithm: nsga-II. *IEEE Trans. Evol. Comput.* 6, 182–197.
- Deb, K., Pratap, A., Meyarivan, T., 2001. Constrained test problems for multi-objective evolutionary optimization. In: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization. Springer-Verlag, pp. 284–298.
- Erdejlj, M., Natalizio, E., Chowdhury, K.R., Akyildiz, I.F., 2017. Help from the sky: leveraging UAVs for disaster management. *IEEE Pervasive Computing* 16, 24–32.
- Fan, Z., Li, W., Cai, X., Huang, H., Fang, Y., You, Y., Mo, J., Wei, C., Goodman, E., 2019a. An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. *Soft Comput.* 23, 12491–12510.
- Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., Goodman, E., 2019b. Push and pull search for solving constrained multi-objective optimization problems. *Swarm Evol. Comput.* 44, 665–679.
- Fei, M., Wenying, G., Ling, W., Yaochu, J., 2024. Constrained multi-objective optimization with deep reinforcement learning assisted operator selection, pp. 1–13. *IEEE/CAA J Automatica Sinica*.
- Fu, Q., Han, Z., Chen, J., Lu, Y., Wu, H., Wang, Y., 2022. Applications of reinforcement learning for building energy efficiency control: a review. *J. Build. Eng.* 50.
- Gao, W.F., Yen, G.G., Liu, S.Y., 2015. A dual-population differential evolution with coevolution for constrained optimization. *IEEE Trans. Cybern.* 45, 1094–1107.
- Gautron, R., Maillard, O.-A., Preux, P., Corbeels, M., Sabbadin, R., 2022. Reinforcement learning for crop management support: review, prospects and challenges. *Comput. Electron. Agric.* 200.
- Han, D., Du, W., Jin, Y., Du, W., Yu, G., 2022. A fuzzy constraint handling technique for decomposition-based constrained multi- and many-objective optimization. *Inf. Sci.* 597, 318–340.
- Hashemi-Amiri, O., Mohammadi, M., Rahmanifar, G., Hajighaei-Keshteli, M., Fusco, G., Colombaroni, C., 2023. An allocation-routing optimization model for integrated solid waste management. *Expert Syst. Appl.* 227.
- Hu, Z., Gong, W., Li, S., 2021. Reinforcement learning-based differential evolution for parameters extraction of photovoltaic models. *Energy Rep.* 7, 916–928.
- Jain, H., Deb, K., 2014. 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, 602–622.
- Kober, J., Bagnell, J.A., Peters, J., 2013. Reinforcement learning in robotics: a survey. *Int. J. Robot. Res.* 32, 1238–1274.
- Li, J.P., Wang, Y., Yang, S., Cai, Z., 2016. A comparative study of constraint-handling techniques in evolutionary constrained multi-objective optimization. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 4175–4182.
- Li, K., Chen, R., Fu, G., Yao, X., 2019. Two-archive evolutionary algorithm for constrained multi-objective optimization. *IEEE Trans. Evol. Comput.* 23, 303–315.
- Liang, J., Ban, X., Yu, K., Qu, B., Qiao, K., Yue, C., Chen, K., Tan, K.C., 2023a. A survey on evolutionary constrained multi-objective optimization. *IEEE Trans. Evol. Comput.* 27, 201–221.
- Liang, J., Qiao, K., Yu, K., Qu, B., Yue, C., Guo, W., Wang, L., 2023b. Utilizing the relationship between unconstrained and constrained Pareto fronts for constrained multi-objective optimization. *IEEE Trans. Cybern.* 53, 3873–3886.
- Liu, Z.-Z., Wang, Y., 2019. Handling constrained multi-objective optimization problems with constraints in both the decision and objective spaces. *IEEE Trans. Evol. Comput.* 23, 870–884.
- Liu, Z., Wang, Y., Wang, B., 2019. Indicator-based constrained multi-objective evolutionary algorithms. *IEEE Transac. Syst., Man, and Cybernetics: Systems* 1–13.
- Maldonado, H.M., Zapotecas-Martinez, S., 2021. A dynamic penalty function within MOEA/D for constrained multi-objective optimization problems. In: 2021 IEEE Congress on Evolutionary Computation (CEC), pp. 1470–1477.
- Martinez, S.Z., Coello, C.A.C., 2014. A multi-objective evolutionary algorithm based on decomposition for constrained multi-objective optimization, 2014 IEEE Congress on Evolutionary Computation (CEC) 429–436.
- Ming, F., Gong, W., Zhen, H., Li, S., Wang, L., Liao, Z., 2021. A simple two-stage evolutionary algorithm for constrained multi-objective optimization. *Knowl. Base Syst.* 228.
- Ming, M., Wang, R., Ishibuchi, H., Zhang, T., 2022. A novel dual-stage dual-population evolutionary algorithm for constrained multi-objective optimization. *IEEE Trans. Evol. Comput.* 26, 1129–1143.
- Oh, S.H., Park, J., Lee, S.J., Kang, S., Mo, J., 2022. Reinforcement learning-based expanded personalized diabetes treatment recommendation using South Korean electronic health records. *Expert Syst. Appl.* 206.
- Phung, M.D., Ha, Q.P., 2021. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput.* 107.
- Qiao, K., Yu, K., Qu, B., Liang, J., Song, H., Yue, C., 2022. An evolutionary multitasking optimization framework for constrained multi-objective optimization problems. *IEEE Trans. Evol. Comput.* 26, 263–277.
- Qiao, K., Yu, K., Qu, B., Liang, J., Song, H., Yue, C., Lin, H., Tan, K.C., 2023. Dynamic auxiliary task-based evolutionary multitasking for constrained multi-objective optimization. *IEEE Trans. Evol. Comput.* 27, 642–656.
- Qingfu, Z., Hui, L., 2007. MOEA/D: a multi-objective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* 11, 712–731.
- Qu, B.Y., Suganthan, P.N., 2011. Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods. *Eng. Optim.* 43, 403–416.
- Runarsson, T.P., Xin, Y., 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* 4, 284–294.
- Saini, B.S., Chakrabarti, D., Chakrabarti, N., Shavazipour, B., Miettinen, K., 2023. Interactive data-driven multi-objective optimization of metallurgical properties of microalloyed steels using the DESDEO framework. *Eng. Appl. Artif. Intell.* 120.
- Shavazipour, B., Kwakkel, J.H., Miettinen, K., 2021. Multi-scenario multi-objective robust optimization under deep uncertainty: a posteriori approach. *Environ. Model. Software* 144.
- Storn, R., Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* 11, 341–359.
- Sun, Z., Ren, H., Yen, G.G., Chen, T., Wu, J., An, H., Yang, J., 2023. An evolutionary algorithm with constraint relaxation strategy for highly constrained multi-objective optimization. *IEEE Trans. Cybern.* 53, 3190–3204.
- Tang, F., Feng, Z., Li, Y., Yang, C., Sun, B., 2023. A constrained multi-objective deep reinforcement learning approach for temperature field optimization of zinc oxide rotary volatile kiln. *Adv. Eng. Inf.* 58.
- Tian, Y., Cheng, R., Zhang, X., Jin, Y., 2017. PlatEMO: a matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput. Intell. Mag.* 12, 73–87.
- Tian, Y., Zhang, T., Xiao, J., Zhang, X., Jin, Y., 2021. A coevolutionary framework for constrained multi-objective optimization problems. *IEEE Trans. Evol. Comput.* 25, 102–116.
- Visutarrom, T., Chiang, T.C., Konak, A., Kulturel-Konak, S., 2020. Reinforcement learning-based differential evolution for solving economic dispatch problems. In: 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 913–917.
- Wang, F., Wang, X., Sun, S., 2022. A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. *Inf. Sci.* 602, 298–312.
- Wang, Y., Cai, Z., Zhou, Y., Zeng, W., 2008. An adaptive tradeoff model for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.* 12, 80–92.
- Woldesenbet, Y.G., Yen, G.G., Tessema, B.G., 2009. Constraint handling in multi-objective evolutionary optimization. *IEEE Trans. Evol. Comput.* 13, 514–525.
- Xiao, Y., Yang, H., Liu, H., Wu, K., Wu, G., 2023. UAV 3-D Path Planning Based on MOEA/D with Adaptive Areal Weight Adjustment.
- Yu, K., Liang, J., Qu, B., Luo, Y., Yue, C., 2022. Dynamic selection preference-assisted constrained multi-objective differential evolution. *IEEE Transac. Syst., Man, and Cybernetics: Systems* 52, 2954–2965.

- Yu, X., Lu, Y., Wang, X., Luo, X., Cai, M., 2019. An effective improved differential evolution algorithm to solve constrained optimization problems. *Soft Comput.* 23, 2409–2427.
- Yu, X., Yu, X., Lu, Y., Yen, G.G., Cai, M., 2018. Differential evolution mutation operators for constrained multi-objective optimization. *Appl. Soft Comput.* 67, 452–466.
- Zapotecas-Martínez, S., Ponsich, A., 2020. Constraint handling within MOEA/D through an additional scalarizing function. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference.
- Zhou, X., Wang, X., Gu, X., 2022. An approach for solving the three-objective arc welding robot path planning problem. *Eng. Optim.* 55, 650–667.
- Zitzler, E., Künzli, S., 2004. Indicator-based selection in multi-objective search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiño, P., Kabán, A., Schwefel, H.-P. (Eds.), *Parallel Problem Solving from Nature - PPSN VIII*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 832–842.
- Zitzler, E., Thiele, L., 1999. Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* 3, 257–271.
- Zou, J., Sun, R., Yang, S., Zheng, J., 2021. A dual-population algorithm based on alternative evolution and degeneration for solving constrained multi-objective optimization problems. *Inf. Sci.* 579, 89–102.
- Zuo, M., Gong, D., Wang, Y., Ye, X., Zeng, B., Meng, F., 2023. Process knowledge-guided autonomous evolutionary optimization for constrained multi-objective problems. *IEEE Trans. Evol. Comput.* 1-1.