Electronic
Research Archive

https://www.aimspress.com/journal/era

ERA, 32(3): 1498–1532.
DOI: 10.3934/era.2024069
Received: 22 January 2024
Revised: 06 February 2024
Accepted: 07 February 2024
Published: 19 February 2024

Research article

MOEA with adaptive operator based on reinforcement learning for weapon target assignment

Shiqi Zou[1,*], Xiaoping Shi[1] and Shenmin Song[2]

[1] Control and Simulation Center, Harbin Institute of Technology, Harbin 150080, China
[2] Center for Control Theory and Guidance Technology, Harbin Institute of Technology, Harbin 150080, China

* **Correspondence:** Email: shiqi_zou@163.com.

Abstract: Weapon target assignment (WTA) is a typical problem in the command and control of modern warfare. Despite the significance of the problem, traditional algorithms still have shortcomings in terms of efficiency, solution quality, and generalization. This paper presents a novel multi-objective evolutionary optimization algorithm (MOEA) that integrates a deep Q-network (DQN)-based adaptive mutation operator and a greedy-based crossover operator, designed to enhance the solution quality for the multi-objective WTA (MO-WTA). Our approach (NSGA-DRL) evolves NSGA-II by embedding these operators to strike a balance between exploration and exploitation. The DQN-based adaptive mutation operator is developed for predicting high-quality solutions, thereby improving the exploration process and maintaining diversity within the population. In parallel, the greedy-based crossover operator employs domain knowledge to minimize ineffective searches, focusing on exploitation and expediting convergence. Ablation studies revealed that our proposed operators significantly boost the algorithm performance. In particular, the DQN mutation operator shows its predictive effectiveness in identifying candidate solutions. The proposed NSGA-DRL outperforms state-and-art MOEAs in solving MO-WTA problems by generating high-quality solutions.

Keywords: weapon target assignment; multi-objective evolutionary algorithm; reinforcement learning; deep Q-network; exploration and exploration

1. Introduction

Weapon target assignment (WTA) plays a crucial role in modern warfare command and control (C2). As military equipment continues to advance, the assignment process must effectively coordinate equipment attributes with offensive and defensive strategies. Missiles, among various combat equipment, hold significant strategic value and therefore have become a focal point of study.

Consequently, missile target assignment (MTA) can be seen as a specialized instance of the broader WTA problem [1]. This has led to the emergence of a new multi-objective WTA problem (MO-WTA) that combines general WTA principles with specific limitations. Moreover, the constrained MO-WTA problem is an important research branch of the WTA problem. One of the key issues that has not yet been resolved for MO-WTA is how to efficiently and accurately determine the optimal allocation.

In recent years, mathematical programming (MP) methods [2, 3], game theory [4], expert systems [5], and evolutionary algorithms (EA) [6, 7] have been widely employed to address the WTA problem. However, these methods are gradually becoming inapplicable to multi-objective optimization, due to conflicts between optimization problems as well as constraints. In particular, the multi-objective evolutionary algorithm (MOEA) has demonstrated outstanding performance in solving the MO-WTA problem [8–11]. For example, Chang et al. [8] designed four heuristic factors and an elite guidance strategy, which were integrated into an improved multi-objective artificial bee colony algorithm (MOABC) to tackle multi-objective DWTA. Similarly, Wang et al. [9] proposed heuristic initialization to expedite exploration, and devised a local search strategy to maintain the exploration-exploitation balance. Furthermore, Zhao et al. [10] enhanced MOEA/D [12] and employed the TOPSIS method [13] for multi-criteria evaluation. Additionally, a study by Chang et al. [11] developed an improved adaptive large-scale neighborhood search (ALNS) algorithm for DWTA, achieving higher-quality solutions in a shorter time compared to exact methods and metaheuristics for most situations.

The emergence of reinforcement learning (RL) has partially addressed challenges in the WTA problem by leveraging the interaction between the agent and the environment, rather than relying on complete system information. In the study [14], the authors proposed a two-stage optimization algorithm, QL-GA, based on Q-learning and genetic algorithms, ensuring improved global optimality of the results. Another study [15] introduced a new algorithm incorporating Monte Carlo (MC) and Q-learning, which, although slower, achieved better performance metrics. The authors [16] developed an incremental search method in the structure of a DQN for target assignment in the radar network, with better performance in multi-target attack and saturation attack scenarios. The research [17] utilized a multi-head Q-value network to enhance the efficiency of the proposed WTA model, addressing search inefficiency. Furthermore, the authors of [18] employed DQN to solve DWTA, demonstrating superior effectiveness compared to traditional Q-learning. Study [19] proposed a new framework that combines DQN with MOABC to solve the MO-WTA problem with strong expansion performance. These studies have shown feasibility in certain states and action spaces, but agents face significant challenges in complex problems. RL methods often encounter the exploration-exploitation dilemma, where problem features become unknown or problem information is not fully utilized. Therefore, the model's generalization ability needs to be strengthened, and feasible learning strategies need to be further adjusted.

The motivation for proposing a new multi-objective optimization framework stems from the recognition of the existing challenges faced by MOEA (e.g., MOEA/D, MOABC, and NSGA-II [20]) in solving multi-objective assignment problems. These challenges include the need to handle constraints in the problem, uncertain convergence, insufficient diversity, and maintaining a balance between exploration and exploitation. Additionally, the superiority of reinforcement learning (RL) in addressing complex sequential decision problems is evident, but it also faces the dilemma of exploration and exploitation. Therefore, this paper aims to complement the strengths of MOEA and

RL to address the above challenges.

In this paper, an adaptive mutation operator mechanism based on the DQN method is proposed to explore new solutions to enhance the algorithmic diversity. Furthermore, the crossover operator that achieves global optimality is improved by a greedy strategy. The main contributions are as follows:

- A new multi-objective weapon target assignment architecture for multi-missiles to intercept multiple targets is proposed. The architecture combines the cumulative cost of a successful intercept with the cumulative survivability of the target. Another new variant in the architecture is proposed to further embed into reinforcement learning.
- An improved NSGA-II with DQN-based adaptive mutation and greedy-based crossover (NSGA-DRL) is proposed to enhance the local update and global optimality in the solution set. The algorithm considers the constraints and properties of the proposed MO-WTA problem to improve performance.
- A DQN-based adaptive mutation operator is proposed to predict higher-quality candidate solutions while ensuring diversity, where the state is the decision variable, the action is the candidate operator, the reward is the new variant designed for the MO-WTA problem, and the environment is the defined MO-WTA problem. Moreover, the agent uses a deep neural network to learn an optimal policy to evaluate the Q-value of the action given the state, where the Q-value means the cumulative reward improvement brought by a better operator.
- A greedy-based crossover operator is proposed to reduce invalid search to speed up convergence, where the rules have been redesigned based on the problem. The rule not only considers global exploration, but also handles the constraints of problems.
- Detailed experiments are conducted for different scales of MO-WTA problems generated, where model training in RL, parameter fine-tuning using the Taguchi method, as well as ablation experiments and comparison experiments are included to validate the performance of the proposed methods. The experimental results show that the proposed NSGA-DRL has a strong expansion performance in the complex MO-WTA problem.

The rest of this paper is organized as follows: Section 2 provides a detailed overview of the related work on WTA problems and RL, including key findings and limitations of existing approaches. Section 3 presents the problem formulation and the constructed models, offering clear explanations of the mathematical aspects. Section 4 introduces the proposed NSGA-DRL algorithm and other proposed methods. Section 5 presents the experimental process, results, and analysis in a clear and structured manner, highlighting the significance of the findings and their implications for the proposed methods.

2. Related work

2.1. WTA

The WTA problem was first formulated in the 1950s and has been established as a combinatorial optimization problem [1]. Furthermore, the WTA problem is categorized into dynamic WTA (DWTA) and static WTA (SWTA) [21], where DWTA consists of several SWTA with time series. The problem is further classified into single-objective optimization WTA (SO-WTA) and multi objective-optimization WTA (MO-WTA) based on the number of solved problems [22].

The existing studies for solving the WTA problem have been conducted as follows. [23] developed an approximate algorithm for a WTA based on a 2-stage convex stochastic program. [2] introduced a linear programming method to obtain optimal solutions for the WTA problem. [24] developed two integer linear programming models for WTA, demonstrating that the proposed models can be solved within a few seconds. [5] proposed an efficient rule-based heuristic for DWTA, providing a new paradigm for solving DWTA with constraints. The authors of [3] introduced a column enumeration algorithm to solve the linearized WTA problem, incorporating bounding and weapon domination concepts in the algorithm. The authors of [25] developed and empirically compared alternative math programming-based and mesh-sampling heuristics for SWTA, identifying a convex polytope as sufficient to characterize all such hierarchies and defining a polytope of interest based on weak dominance criteria.

With the improvement of computational performance, bio-heuristic intelligence algorithms have found wide applications in solving WTA problems. For example, the authors of [6, 26] improved an immunity-based ant colony (ACO) algorithm and a genetic algorithm (GA) with greedy eugenics to explore the SWTA problem, respectively. [27] presented two heuristic methods for solving the general WTA, including the simulated annealing and threshold accepting methods. [28] developed a novel optimization algorithm for assigning weapons to targets based on desired kill probabilities, where execution times remained on the order of milliseconds. [29] improved an adaptive chaos parallel clonal selection algorithm for WTA in warship formation antiaircraft application. The authors of [30] presented an improved artificial fish swarm algorithm-improved harmony search algorithm for WTA, demonstrating the requirement for real-time performance in combat conditions. [31] improved particle swarm optimization (PSO) for large-scale WTA and compared other swarm algorithms, with high efficiency, high quality, and high robustness. [9] proposed an adaptive memetic algorithm with local search for a joint WTA.

The MO-WTA problem suffers from the fact that the optimization problem is conflicting and has constraints such that it is not easy to obtain an optimal solution. Instead, we obtain an approximately optimal Pareto front (PF), which is a collection of optimal solutions. In particular, multi-objective evolutionary algorithms (MOEAs) have strengths in dealing with MO-WTA, some of which are described above. [32] investigated NSGA-II and MOEA/D for solving MO-WTA, with high efficiency. The authors of [7] improved MOEA/D for MO-WTA, which mainly optimized a number of sub-problems. Further, [33] modified MOEA/D with heuristic initialization for the variant of MO-WTA, where the nadir-based Tchebycheff method and the proposed neighbor matching strategy were implemented. [34] presented a shuffled frog leaping algorithm with non-dominated sorting for MO-WTA, with a discrete adaptive mutation based on the function change rate.

2.2. *RL*

RL has evolved significantly in engineering optimization. The application of RL in MOEA is rare for operator selection [35, 36]. RL has the advantage of dynamically finding optimal solutions in finite or infinite environmental interactions. Therefore, it is beneficial to combine RL to operator selection to improve the performance of MOEA.

The goal of RL is to learn a policy π after taking an action a_t under a state s_t, where the action maximizes the expected cumulative reward under the current state [37]. In the learning, the agent takes an action a_t at the state s_t under the policy. Then, a reward r and the next state s_{t+1} will be produced after the interaction between the agent and the environment. The above process is to train the agent, where the tuple (s_t, a_t, r_t, s_{t+1}) will be recorded as a sample.

Deep reinforcement learning (DRL) is the technique of RL combined with deep neural networks, which currently includes the value-based and the policy-based learning approaches. The policy-based approach solves continuous action spaces, learning a policy via an independent function approximator. In this case, the policy is a conditional probability distribution $\pi = p(a_t|s_t;\theta)$, which is used by the gradient to learn the optimal policy parameters.

The value-based method can handle the discrete action spaces by approximating the action-value function (Q-function $Q(s_t, a_t)$). The process calculates the expected cumulative reward ($\mathbb{E}[R_t|s_t, a_t]$), where $R_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$ is the sum of future rewards with the discount. $\gamma \in [0, 1]$ is the discount factor. After the Q-function is approximated, the agent will take the action with the largest Q value under the given state s, i.e., $\pi(s) = \arg \underset{a \in A}{max}\, Q(s, a)$. The above process can be summarized in the Q-learning method.

The classical Q-learning maps the action-value function as a Q-table, where each grid stores the Q value of the action given a state. However, Q-tables can only express discrete and finite state spaces. Therefore, DQN [38] are approximated using deep neural networks instead of Q-tables. The main advantage of DQN is the off-policy learning with a time-difference (TD) method, where the optimal policy can be solved for Markov processes via the Bellman equation. The update strategy for the Q function is expressed as

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha * (r_t + \gamma * \underset{a_{t+1} \in A}{max}\, Q(s_{t+1}, a_{t+1}) - Q(S_t, a_t)) \tag{2.1}$$

where α is the learning rate, and $Q(s_{t+1}, a_{t+1})$ is the state-action pair of the next step. The main innovations of DQN are the use of deep neural networks, the introduction of target networks updated based on the original Q-network, and the use of an experience replay strategy. The parameters θ' of the target network are updated from θ of the original Q-network at fixed iteration intervals, where the loss function L_θ in the training is updated as follows:

$$\begin{cases} L_\theta = E[(Q_{target} - Q(s, a; \theta))^2] \\ Q_{target} = r_t + \gamma * \underset{a_{t+1} \in A}{max}\, Q(s_{t+1}, a_{t+1}; \theta') \end{cases} \tag{2.2}$$

In the proposed adaptive operator mechanism, the state is the decision variable, and the action is the candidate operator. Moreover, the state and action spaces of the mechanism are discrete so that DQN is applicable to the proposed mechanism.

3. Problem formulation and model

3.1. Problem formulation

In this paper, we consider the WTA problem in modern local battlefields with multiple missiles allocating multiple targets. This scenario is shown in Figure 1, where a set of missiles has been assigned to intercept and strike multiple incoming targets detected by radars or satellites. Significantly, the WTA aims to allocate appropriate missiles to hit targets. Generally, the quality of assigned results will directly determine the combat effectiveness and cost of missiles. Therefore, solving WTA means generating an optimal solution that satisfies maximum combat effectiveness and minimum combat cost.

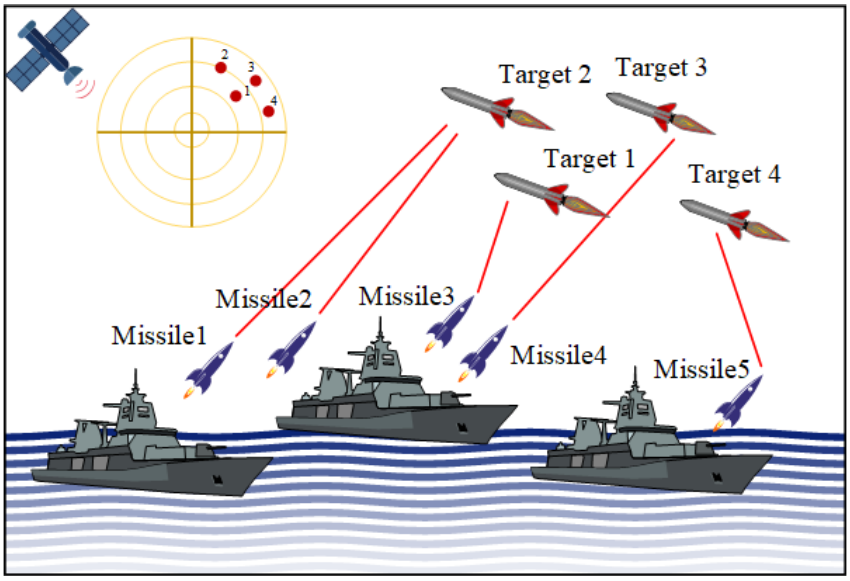


Figure 1. An example of the WTA problem.

3.2. General weapon target assignment model

Let $K = \{m_i | i = 1, 2, \ldots, M\}$ represents the set of missiles, and $T = \{n_j | j = 1, 2, \ldots, N\}$ stands for the set of targets. Crucial parameters of the WTA problem are explained as follows:

- c_i expresses the cost of missile m_i,
- v_j means the threat value of target n_j,
- p_{ij} denotes the kill probability that reflects the extent to which missile m_i destroys target n_j,
- x_{ij} represents the decision variable to illustrate the allocation result of m_i to n_j, defined as follows,

$$x_{ij} = \begin{cases} 1, & \textit{if } m_i \textit{ is allocated to } n_j \\ 0, & \textit{otherwise.} \end{cases}$$

Thus, we build the mathematical model of the WTA as follows, which aims to maximize the combat effectiveness $CE(\mathbf{X})$ of the missiles against all the targets:

$$\begin{array}{lll} \max & CE(\mathbf{X}) = \sum_{j=1}^{N} v_j \times \widetilde{p} = \sum_{j=1}^{N} v_j \times \left[1 - \prod_{i=1}^{M} (1 - p_{ij})^{x_{ij}}\right] & \\ s.t. & \sum_{j=1}^{N} x_{ij} \leq Num_i, & \forall i \in \{1, 2, \ldots M\} \\ & x_{ij} \in \{0, 1\}, & \forall i \in \{1, 2, \ldots M\}, \forall j \in \{1, 2, \ldots N\} \end{array} \tag{3.1}$$

where

- $\widetilde{p}$ stands for the joint kill probability of missiles against target n_j, formulated as

$$\widetilde{p_j} = 1 - \prod_i (1 - p_{ij})^{x_{ij}}, i \in \{1, 2, \ldots M\} \tag{3.2}$$

- Num_i represents the current available number of missile m_i,
- $\mathbf{X} = \left[x_{ij}\right]_{M \times N}$ denotes the decision matrix of the WTA problem.

3.3. Multi-objective weapon target assignment model

In general, the research of MO-WTA mostly focus on maximizing the combat effectiveness and minimizing the combat cost. Therefore, we further build a mathematical model of the combat cost $C(\mathbf{X})$ as follows:

$$\begin{array}{lll} \min & C(\mathbf{X}) = \sum_{i=1}^{M} \sum_{j=1}^{N} c_i \times x_{ij} & \\ s.t. & x_{ij} \in \{0,1\}, & \forall i \in \{1,2,\ldots M\}, \forall j \in \{1,2,\ldots N\} \end{array} \tag{3.3}$$

where

- c_i has been defined as the cost of missile m_i,
- Other notations are consistent with the meaning in Eq (3.3).

Moreover, we transform the max-min optimization into a min-min problem to reduce computational burden. The main argument is that the obtained combat effectiveness cannot exceed the sum of the existing target threats. Therefore, we compute the difference between the two as the combat residual effectiveness, which is used as the objective function F_1. Based on the above definition, the optimization model of MO-WTA is formulated as follows:

$$\begin{array}{llll} \min & F_1 & = \xi - CE(\mathbf{X}) & \\ & & = \xi - \sum_{j=1}^{N} v_j \times \widetilde{p} & \\ & & = \xi - \sum_{j=1}^{N} v_j \times \left[1 - \prod_{i=1}^{M} (1 - p_{ij})^{x_{ij}}\right] & \\ \min & F_2 & = C(\mathbf{X}) & \\ & & = \sum_{i=1}^{M} \sum_{j=1}^{N} c_i \times x_{ij} & \\ s.t. & & \sum_{j=1}^{N} x_{ij} \leq Num_i, & \forall i \in \{1,2,\ldots M\} \\ & & x_{ij} \in \{0,1\}, & \forall i \in \{1,2,\ldots M\}, \forall j \in \{1,2,\ldots N\} \end{array} \tag{3.4}$$

where ξ represents the sum of the target threats for all the targets, defined as

$$\xi = \sum_{i=1}^{M} \sum_{j=1}^{N} v_j, j \in \{1, \ldots N\} \tag{3.5}$$

3.4. A variant of MO-WTA

Here, a variant of the MO-WTA model is constructed as a reward function to participate in DRL. Precisely, the linear weighting method can convert multi-objective optimization into single-objective optimization. The converted model is given by

$$\begin{array}{lll} R & = CE(\mathbf{X}) - C(\mathbf{X}) & \\ & = \sum_{j=1}^{N} (v_j \times \widetilde{p} - \sum_{i=1}^{M} c_i \times x_{ij}) & \\ & = \sum_{j=1}^{N} \left[v_j \times (1 - \prod_{i=1}^{M} (1 - p_{ij})^{x_{ij}}) - \sum_{i=1}^{M} c_i \times x_{ij}\right] & \\ s.t. & x_{ij} \in \{0,1\}, & \forall i \in \{1,2,\ldots M\}, \forall j \in \{1,2,\ldots N\} \end{array} \tag{3.6}$$

where the analysis on Eq (3.6) is as follows:

1) $CE(\mathbf{X})$ represents the combat effectiveness of the missiles,
2) $C(\mathbf{X})$ stands for the combat cost of the missiles,
3) The difference between the two can be expressed as the combat benefit of the missiles,
4) The cumulative reward in DRL needs to be maximized, and the R defined here can meet the requirements,
5) Therefore, this converted formula makes sense.

4. The proposed methods for MO-WTA

4.1. The proposed DQN based adaptive mutation operator

4.1.1. MutOper: DQN-based adaptive mutation operator

The *MutOper* embeds the model learned by DQN to predict higher quality solutions. The proposed operator can enhance the diversity and exploration of solutions, shown in Algorithm 3. The proposed mechanism integrates three mutation operators, shown in Figure 2, including op1 (Swap Operator), op2 (Slide Operator), and op3 (Flip Operator).

- The op1 operator. Randomly select two genes, and swap them.
- The op2 operator. Randomly select two genes (A and B), and move A to the position of B. The other genes all move back in turn.
- The op3 operator. Randomly select one gene, and update a new value based on feasible values.

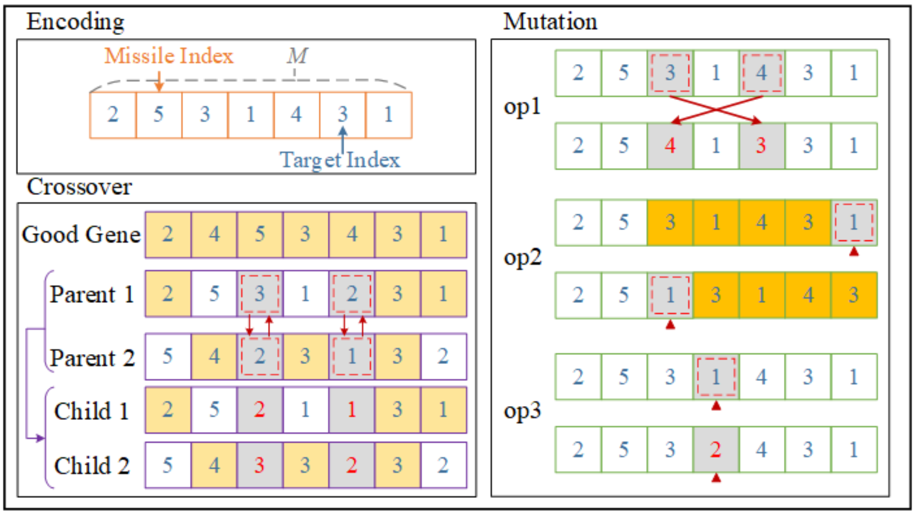
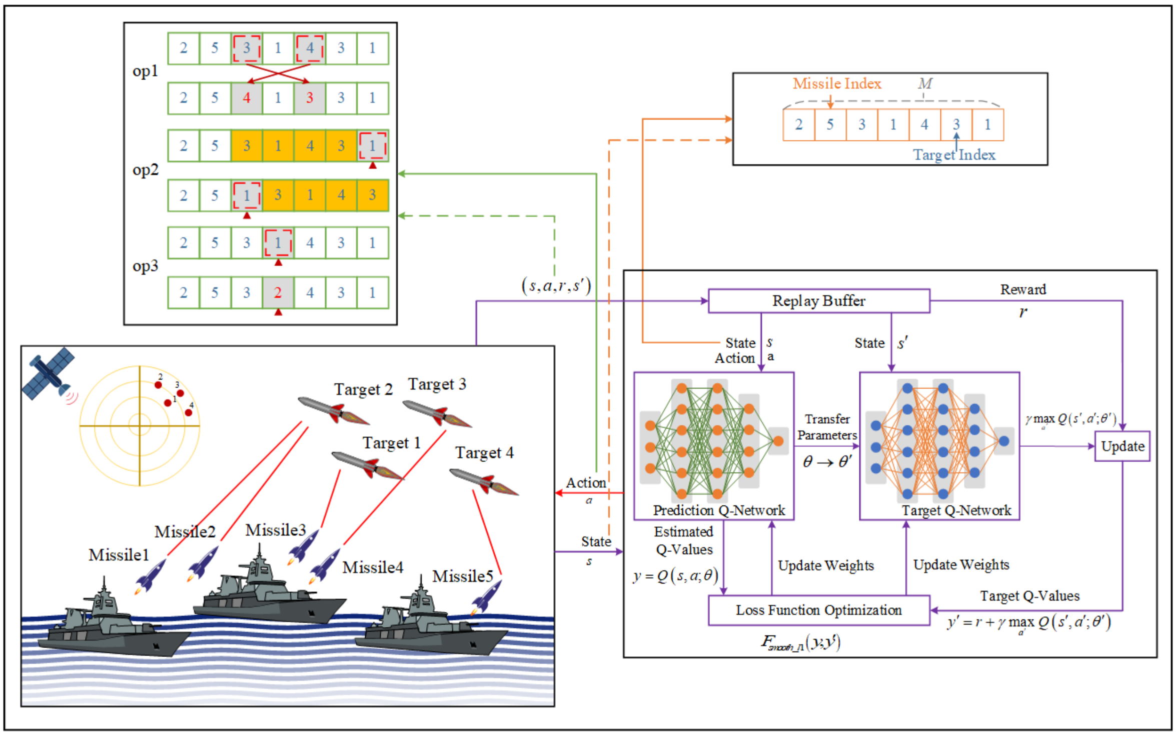


Figure 2. An example of encoding and operators.

Algorithm 1 *MutOper*

```
Input: Offsprings
Output: Pops
Pops ← Offsprings;
$nPop$ ← Pops.shape[0];
for $i$ in range($nPop$) do
    if $rand()$ ¡ $p_m$ then
        Pops ← $DQN$(Pops[$i$]);
        // Shown in Algorithm 1.2.
    end if
end for
return Pops
```



op1
2 5 3 1 4 3 1
2 5 4 1 3 3 1

op2
2 5 3 1 4 3 1
2 5 1 3 1 4 3

op3
2 5 3 1 4 3 1
2 5 3 2 4 3 1

Missile Index M
2 5 3 1 4 3 1
Target Index

Figure 3. An example of DQN learning.

4.1.2. The model learning for MutOper

The DQN learning process and data flow are shown in Figure 3, depicting the interaction between states, actions, and the environment. Furthermore, we define the actions of the agent in the training process, shown in Algorithm 1.1. In particular, the operator selection in the *MutOper* is the action of the agent. In detail, *State*, *Action*, and *Reward* are defined as follows:

- *State*. *State* is denoted as the assignment of MO-WTA, $S = \{s_1, \ldots s_i\}, i \in \{1, \ldots NIND\}$

- *Action. Action* presents flags of candidate operators, $A = \{a_1, \ldots a_j\}, a_j \in \{0, 1, 2\}$
- *Reward. Reward* is defined as the relative change in the R value of Eq. 3.6 after the operation, $Reward = \max\{R_{new} - R_{old}, 0\}$

The whole process uses the ϵ-greedy strategy to select the action and introduces a *Replay Buffer* to store and extract a tuple $(s, a, r, s^{'}, done)$. Where s is the current state, a is the action, r is the reward obtained by executing a, $s^{'}$ is the next state, and *done* is a flag. Moreover, the *Replay Buffer* can eliminate correlations between data. The parameters θ of the target Q-network are periodically updated according to $\theta^{'}$ in the prediction Q-network, which are updated by using the Adam [39] optimizer. Finally, the estimated Q-values and the target Q-values are updated by the loss function F_{smooth_l1} [40], which is the smooth L1 loss.

Algorithm 1.1 The Action Encoding of Model Training for *MutOper*

```
Input: State
Output: Action
// Initialization
for episode in range(Ep) do
    // Agent received env.state
    if $rand \leq \epsilon$ then
        $action \leftarrow rand.sample()$;
    else
        $action \leftarrow model.predict(state)$;
    end if
    // Agent send action to env
    // Update
    $flag \leftarrow action$;
    if $flag == 0$ then
        // Perform the op1.
    else if $flag == 1$ then
        // Perform the op2.
    else if $flag == 2$ then
        // Perform the op3.
    end if
end for
```

4.1.3. Operator selection mechanism in MutOper

We can further obtain the function *DQN*(), which describes in detail the combination of mechanism and model for operator selection. The whole process is shown in Algorithm 1.2. It specifically expresses that the selection mechanism relies on the learned models. The model fully utilizes the problem information to set the global reward and explore the optimal strategy during the training process, which ensures the reliability and diversity of the operator selection mechanism.

Algorithm 1.2 *DQN*(*chr*)

```
Input: $chr$
Output: $assignment$
$assignment \leftarrow chr$;
$model \leftarrow torch.load(path)$;
// $model$ is obtained in Algorihtm 1.1.
$model.eval()$;
$state \leftarrow tensor([assignment])$;
$action \leftarrow select(model, state)$;
$flag \leftarrow decode(action)$;
$mM, mT \leftarrow sample(Missiles, Targets)$;
if $flag == 0$ then
    $new\_assignment \leftarrow assignment$;
    $assignment[mM] \leftarrow new\_assignment[mT]$;
    $assignment[mT] \leftarrow new\_assignment[mM]$;
else if $flag == 1$ then
    $slide \leftarrow assignment[mM]$;
    $assignment \leftarrow delete(assignment, mM)$;
    $assignment \leftarrow insert(assignment, mT, slide)$;
else if $flag == 2$ then
    $assignment[mW] \leftarrow mT$;
end if
return $assignment$
```

4.2. The proposed greedy based crossover operator

4.2.1. RecOper: greedy-based crossover operator

This operator is designed to implement a local update concerning the neighbour information of the candidate solutions. This idea is mainly derived from the methodology of [26], which has limitations in dealing with global computations. We further improve this methodology by considering the global optimum and constraints. *RecOper* is summarized in Algorithm 2.

4.2.2. The greedy strategy in RecOper

Specifically, we apply constraint handles and the greedy strategy to locally update more feasible solutions. The following rules are given to create good solutions of the proposed problem. A "good gene" is defined as $\mathbf{\Omega}$, where each element is ω_i, and the elements are sequences that express the survivability of the target.

$$\omega_i = \arg\min\{v_j \times (1 - p_{ij})\}, \forall i \in \{1, \dots M\} \tag{4.1}$$

Based on $\mathbf{\Omega}$ and the constraints, we can further obtain the "inherited gene" to determine the swap operation. This process ensures local updating through the traditional swap and guarantees global optimality according to the designed rules, elaborated in Algorithm 2.1.

Algorithm 2 *RecOper*

```
Input: Offsprings, P, V
Output: chrPops
chrPops ← Offsprings;
$nPop$ ← chrPops.shape[0];
$\mathbf{\Omega}$ ← Eq (4.1);
// Eq (4.1) calls the parameters P, V.
for $i$ in range(0, $nPop$, 2) do
    if $rand()$ ¡ $p_c$ then
        chrPops ← $do$(chrPops[$i$], chrPops[$i$+1], $\mathbf{\Omega}$);
        // Shown in Algorithm 2.1.
    end if
end for
return chrPops
```

4.3. Procedure of the proposed NSGA-DRL

This paper employs an improved NSGA-II [20] as the main framework (NSGA-DRL), which balances exploration and exploitation in solving MO-WTA. Both crowding distance (*CrowDis*) and binary tournaments [41] (*TourSelect*) participate in the selection mechanism.

The innovative improvement lies in designing an DQN-based adaptive mutation operator and greedy-based crossover operator. Moreover, an efficient non-dominated sorting approach [42] (*ESS-NDSort*) is applied, which has been proven to be more efficient than the method in NSGA-II.

The procedure of NSGA-DRL is shown in Algorithm 3. Here are the key parameters in the pseudocode,

- *NIND* is the size of the population,
- *MaxEva* means the maximum evaluation,
- **P**, **V**, and **C** represent the data in different problem instances,
- **ParetoPops**, **ParetoObjs**, and **ParetoCVs** denote the optimal result of the algorithm,
- Bold symbols stand for variables in matrix or vector form.

A. InitialPop. Starting with a randomly generated initial population containing a missile and target, the population mainly consists of *NIND* initial solution with *N*-dimensional variables. A certain solution is defined as below, whose generation rules are also given:

$$\begin{array}{ll} I_i = [ind_i^1, ind_i^2, \ldots ind_i^N], & i \in \{1, \ldots NIND\}, \\ ind_i^j = ind_{min}^j + (ind_{max}^j - ind_{min}^j) \times rand(), & i \in \{1, \ldots NIND\}, j \in \{1, \ldots N\}. \end{array} \tag{4.2}$$

where

- ind_i^j indicates the *i*-th individual in the population,
- ind_{min}^j and ind_{max}^j are the minimum and maximum values respectively,
- *rand()* is a random integer on [0, 1],
- *N* represents the number of targets.

Algorithm 2.1 *do*($chr1$, $chr2$, $\mathbf{\Omega}$)

```
Input: $chr1$, $chr2$, $\mathbf{\Omega}$
Output: $chr1$, $chr2$
$P1, P2 \leftarrow chr1, chr2$;
$chr1, chr2 \leftarrow P1, P2$;
$inherited\_gene$ = [];
for $i$ in range(length($\mathbf{\Omega}$)) do
    if $P1[i] == P2[i]$ & $P1[i] == \mathbf{\Omega}[i]$ then
        $inherited\_gene.append(i)$
    end if
end for
$gene\_swap \leftarrow$ set(range(length($\mathbf{\Omega}$))) - set($inherited\_gene$);
if length($gene\_swap$) ¡ 2 & != Constraints then
    break;
else
    $swap1, swap2 \leftarrow$ sample($gene\_swap$, 2);
    $chr1[swap1], chr2[swap2] \leftarrow chr2[swap2], chr1[swap1]$;
    $chr1[swap2], chr2[swap1] \leftarrow chr2[swap1], chr1[swap2]$;
end if
return $chr1$, $chr2$
```

B. OptSelect. This part consists of providing individuals with good non-dominance rank and CrowDis, and finally merging and selecting, which integrates *ESS-NDSort*, *CrowDis*, and *TourSelect*. Specifically, *ESS-NDSort* provides a more efficient sorting strategy, which avoids redundant comparisons during operation. Its superior performance in bi-objective optimization problems has been demonstrated in the literature [42]. *CrowDis* and *TourSelect* can guarantee the diversity of solutions, and makes solutions more evenly distributed.

C. Encoding. An integer encoding has been accessed in the algorithm, which can also handle constraints. An example of the encoding is shown in Figure 2, where the length of an individual is the number of missiles in the problem, and each of these genes is the target index assigned to that missile. Further, this encoding satisfies this constraint, i.e., each missile can be assigned to a target.

4.4. Computational complexity of NSGA-DRL

The time complexity of NSGA is mainly determined by the offspring generation. The time complexity of adaptive operator mechanism is $O(D^2)$, where D is the number of decision variables, indicating the size of layers. Thus, the time complexity of neural network training is $O(D^2)$. The time complexity of the whole offspring generation is $O(D)$, which includes the mutation and the crossover. The time complexity of the population update is $O(M)$, where M is the number of objective functions to be calculated. The time complexity of calculating the function improvement is $O(N)$, where N is the size of the population. Therefore, the total time complexity of NSGA-DRL at each offspring generation is $O(N \cdot D^2)$.

Algorithm 3 The framework of NSGA-DRL

```
Input: NIND, MaxEva, P, V, C
Output: ParetoPops, ParetoObjs, ParetoCVs
Pop ← InitialPop(NIND, P, V, C);
// Described in 4.3.A
Objs, CVs ← CalObj(Pop);
rank ← ESS-NDSort(Objs, CVs);
dis ← CrowDis(rank, Objs);
while MaxEva exceeds do
    Offsprings ← TourSelect(Pop, rank, dis, NIND);
    Offsprings ← RecOper(Offsprings, P, V);
    // Algorithm 2
    Offsprings ← MutOper(Offsprings, P);
    // Algorithm 1
    Objs, CVs ← CalObj(Offsprings);
    NewPop ← Pop ∪ Offsprings;
    rank ← ESS-NDSort(NewPop);
    dis ← CrowDis(rank, NewPop);
    Pops ← OptSelect(rank, dis, NewPop, NIND);
    //{ESS-NDSort, CrowDis, TourSelect}
    // Described in 4.3.B
end while
ParetoPops ← Pops;
ParetoObjs,ParetoCVs ← CalObj(ParetoPops);
return ParetoPops, ParetoObjs ParetoCVs
```

5. Experiments and analysis

In this section, we investigate the performance of the proposed NSGA-DRL. First, we give a MO-WTA problem instance, baseline algorithms for comparison, and a detailed simulation environment setup. Then, DQN model training and testing experiments are given. Next, parameter fine-tuning is designed to tune and determine the main parameters of the algorithm for problem size. After that, ablation experiments are conducted to analyze the effectiveness of the proposed methods. Finally, comparative experiments are further performed to investigate the differences between DRL and the selected baseline algorithm.

5.1. General setups

The detailed configuration of the experiment is given here. It is divided into the following sections: MO-WTA problem instance, baseline algorithms for comparison, and simulation environment.

1) MO-WTA problem instance. We build a data generator to provide different simulation environment instances. In addition, we focus on saturation strike scenarios, so there are more missiles than targets. Therefore, the missile-target-ratio ρ is designed to calculate the number of missiles for different numbers of targets. Table 1 shows the specific variables and definitions.

Table 1. Parameters for problem instance.

Variable	Definition	Variable	Definition	Range
N	[6, 12, 18, 24, 48]	p_{ij}	$0.6 + 0.8 \times$ rand()[1]	[0.6, 0.8]
ρ	[2, 2.5, 3, 3.5]	v_j	$1 + 5 \times$ rand()	[1, 5]
M	N $\times\rho$	c_i	$0.1 + 0.5 \times$ rand()	[0.1, 0.5]
Num_M	M / 5			

[1] rand() denotes a random number between [0,1].

We obtained 20 problem instances based on the parameters in Table 1, which were further categorized into three groups based on the number of decision variables, i.e., small, medium, and large scale. Intuitively, we give the relationship between grouping and the number of decision variables in Figure 4. The grouping results and the corresponding parameter settings of the experiment are organized in Table 2.

Table 2. The grouping results and corresponding parameter settings

Problem Group	Number of Decision Variables	Problem Index	Population Size	Number of Evaluations
Small Scale	(0, 50)	1–10, 13	100	10,000
Medium Scale	[50, 100)	11–12, 14–17	150	20,000
Large Scale	[100, $+\infty$)	18-20	200	30,000

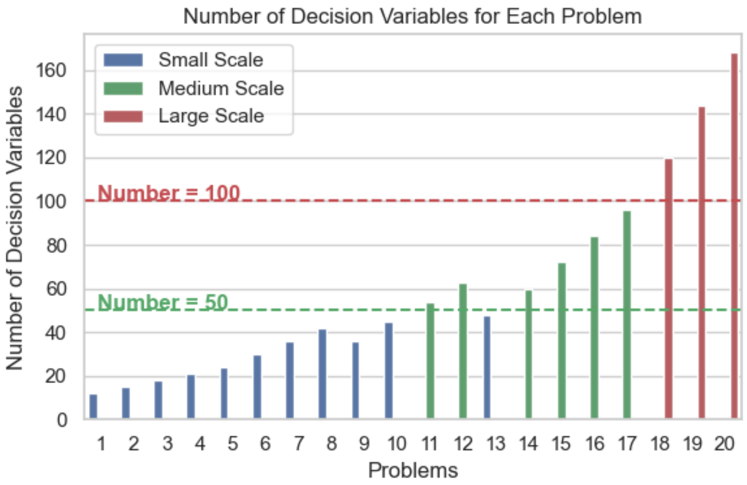


Figure 4. Number of decision variables for each problem.

2) Baseline algorithms for comparison. We compare the proposed NSGA-DRL with C-TSEA [43], AGE-MOEA-II [44], and the original NSGA-II [20].

In detail, C-TSEA provides a new method to deal with the constrained multi-objective optimization problem (CMOP) by introducing two stages. And, it has also been proven to be superior and generalizable. AGE-MOEA-II incorporates Pareto geometry to further extend to new adaptive geometry estimation strategies. In particular, it has been shown to outperform other state-of-the-art MOEAs in large-scale multi-objective optimization problems.

In this paper, the proposed MO-WTA problem is a large-scale multi-objective optimization problem in some scenarios, and has multiple constraints. Therefore, the above two state-of-the-art algorithms are introduced for comparison. In addition, the origin NSGA-II is introduced to verify the improvements in this paper.

3) Performance metric for comparsion. We select the hyper-volume calculation (HV) [45] in MOEA as a performance measure. The detailed definition is shown below.

$$HV(PF, RP) = VOL\left(\bigcup_{b \in P} [b_1, b_1^w] \times [b_2, b_2^w] \times \cdots \times [b_n, b_n^w]\right) \tag{5.1}$$

where

- VOL means the Lebesgue method for measuring the volume.
- $[b_1, \ldots, b_n] \in PF$ stands for the PF obtained.
- $[b_1^w, \ldots, b_n^w] \in RP$ represents the worst points in the hypervolume, defined as the reference point.
- The larger the HV value defined by Eq (5.1), the better the PF obtained.

The reference points RP in this paper are obtained as follows. First, after several independent runs, the algorithms obtain the PF with the highest non-dominated rank. Then, the reference point will be determined according to 1.1 times the maximum value of PF.

4) Simulation environment. The experiments in this paper were conducted on a server with a Windows 11 operating system, an Intel Core i7 CPU, 32 GB of RAM, and a GeForce RTX 3080 GPU. The specific simulation configuration is as follows:

- All simulation programs were developed in Python 3.9 and the Pycharm 2023.2 IDE.
- DQN was implemented in Pytorch 2.0.1 and Gym 0.26.1.
- The Taguchi design used for parameter fine-tuning was analyzed in IBM SPSS Statistics 27.0.1.

5.2. Experiment 1: DQN for model training

In this experiment, we provide details on the neural network configuration, training results, testing results, and a comprehensive summary of the results.

1) Neural network configuration. A neural network architecture was designed to apply DQN to build MO-WTA models. This network consists of multiple layers, including an embedding layer, an input layer, hidden layers, an output layer, activation functions, and dropout layers. We detail the DQN hyperparameters in detail in Table 3.

2) Training results. We conducted training experiments using DQN to explore an optimal policy for MO-WTA. A generalized performance metric represents training results, the cumulative reward curve.

Table 3. Hyperparameters of DQN for MO-WTA

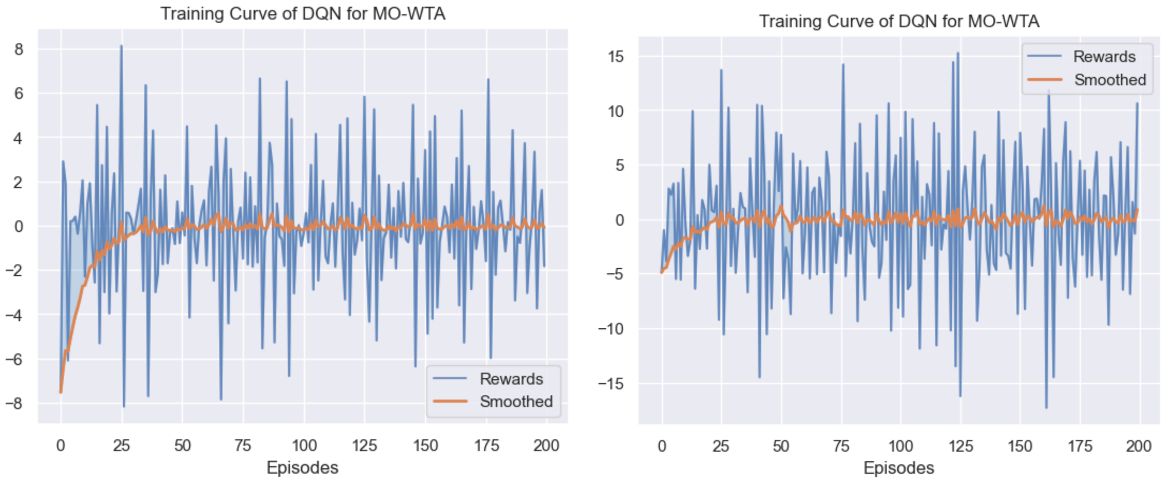
Hyperparameters of DQN		Hyperparameters of DQN Training		Hyperparameters of DQN architecture	
Discounter Factor(γ)	0.95	Learning Rate	10^{-3}	Embedding Layer	(N, M)
Epsilon(ϵ)	0.90	Batch Size	64	Input Layer	M*M
Epsilon Start(ϵ_{start})	0.90	Training Episodes	200	Hidden layer[1]	(128, 256, 128, 64)
Epsilon End(ϵ_{end})	0.05	Maximum Steps[2]	300, 500, 800	Output Layer	3
Epsilon Decay	200	Memory Capacity	10^4	Activation Function	Relu()
Initial Random Seed	10	Optimizer	Adam	Dropout Layer[3]	$p_1 = 0.3$, $p_2 = 0.2$, $p_3 = 0.15$

[1] The hidden layer contains five fully connected layers.

[2] The maximum steps for small, medium, and large scale correspond to 300, 500, and 800, respectively.

[3] The probabilities of the three Dropout layers are 0.3, 0.2, and 0.15 in that order.

The small and large scale results are given here and shown in Figure 5, where MO-WTA 10 and MO-WTA 19 are used as examples, respectively. In the figure, we also smoothed the cumulative rewards of the training to analyze the results visually.



(a) Training Curve on MO-WTA 10

(b) Training Curve on MO-WTA 19

Figure 5. Results of DQN training.

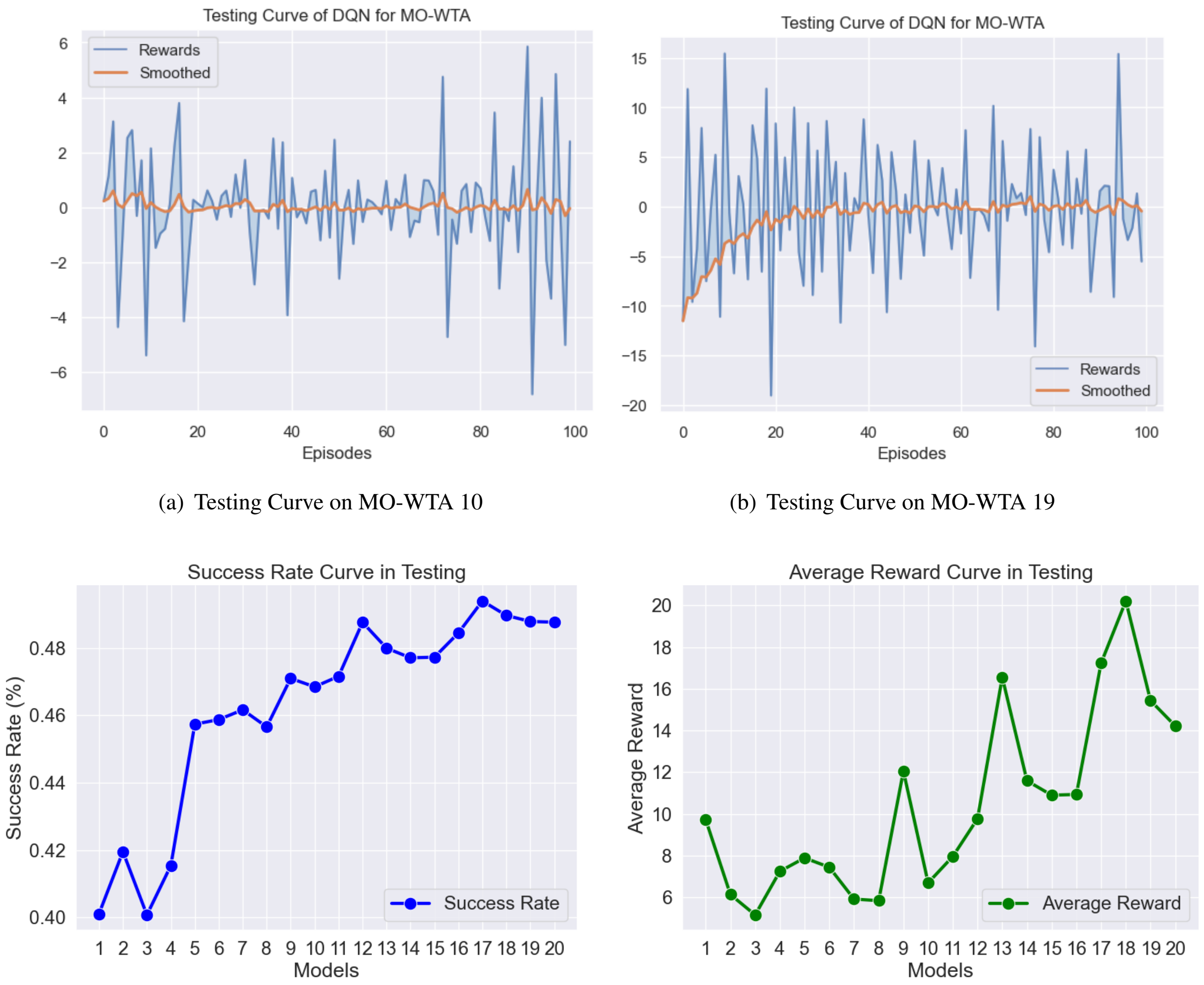
Analyzing the curves in Figure 5, the training curves on MO-WTA 10 and MO-WTA 19 show that the DQN model is learning and improving its performance. The growth from negative in cumulative reward indicates that the agent is exploring and discovering more valuable actions. Finally, the curve stabilizes to 0, indicating that the agent has learned the optimal policy and continues to receive higher rewards.

3) Testing results. We further test the trained models and compute two performance metrics: the success rate and the average reward in testing. The metrics are defined below:

- Success Rate. The success rate is defined as the ratio of positive rewards to the total episode in testing,
- Average Reward. The average reward means the average cumulative reward in testing.

The testing results for all models are shown in Figure 6. The figure contains mainly the success rate and the average reward curve for all trained models, as well as testing curves for MO-WTA 10 and MO-WTA 19 above.

Specifically, the trained models have good convergence in the testing. The overall success rate is around 47%, excluding the models of initial small-scale problem. The average reward expresses an improvement in the agent, where large-scale problems are about 40% of medium and 75% of small-scale.



(a) Testing Curve on MO-WTA 10

(b) Testing Curve on MO-WTA 19

(c) Success Rate in Testing

(d) Average Reward in Testing

Figure 6. Results of DQN testing.

4) Summary of results. Specifically, we summarize the training results of models and the testing results as follows.

A. The training curve shows that the difference between the values produced by the old and new states becomes progressively smaller. With the training of different problem scales, the agent explores the optimal policy and reaches a relatively stable state. In particular, the agent can still learn the optimal policy under large-scale problems.

B. The models trained on MO-WTA 10 and MO-WTA 19 demonstrate that both have learned the optimal policy in testing. Although MO-WTA 19 did not reach the optimum initially, it was fine-tuned and eventually reached the optimal policy.

C. Models 1–5 perform poorly on the metrics, where the success rate is mostly about 41% and the average reward stays at about 7. The models have fewer data so that the agents cannot learn more information. However, too many learning steps tend to cause overfitting.

D. Overall, the trained models had superior performance in testing, which perform better on more large-scale problems, with an upward trend in the metrics curve. In addition, the models demonstrate some generalization ability as the problem scale increases. Moreover, when facing different data, the learned models can be adjusted to the optimal policy in time.

5.3. Experiment 2: parameter tuning

In this paper, the generated problems have been categorized as small, medium, and large-scale problems. The decision variables of different scales interact with the parameters in MOEAs as well. We employed the Taguchi method to fine-tune the main parameters in MOEA. This experiment aims at finding the optimal parameter configuration to ensure the reliability and accuracy of the experimental results.

1) Taguchi method. The Taguchi method [46] is an efficient approach to determining the best combination of parameters using a limited number of experiments. Levels, factors, and signal-to-noise (SN) ratio calculation rules need to be determined in this method. The final configuration is expressed through the plot of the main effects.

2) Parameter tuning process. The crossover probability p_c and the mutation probability p_m have been shown to have the highest uptake on the algorithm performance [47]. Thus, p_c and p_m were defined as factors, while levels are referenced from the literature [9, 48, 49]. These factors and levels are listed below:

A. Factor: p_c, Levels: {0.6, 0.7, 0.8}.

B. Factor: p_m, Levels: {0.01, 0.05, 0.1}.

Therefore, an $L9$ design was adopted to conduct this experiment. Further, we expect a large SN ratio, i.e., response maximization, defined as follows.

$$SN = -10 \times log(\sum(1/Y^2)/n) \tag{5.2}$$

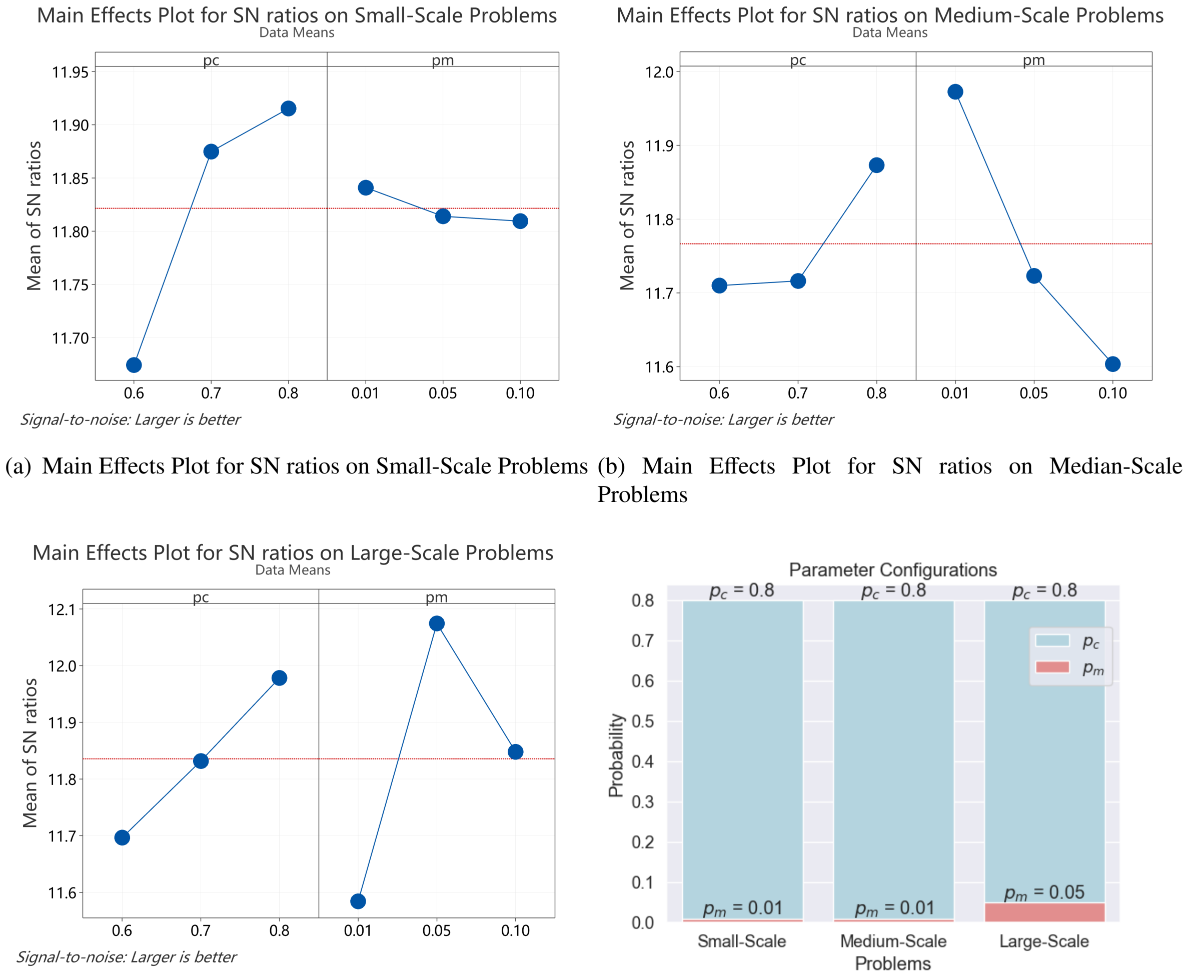
where

- n is the number of Y.
- Y is defined as the R in Eq (3.6) obtained by solutions of NSGA-DRL.

Finally, we sequentially selected MO-WTA10, 15, and 19 to represent small, medium, and large problems to determine parameter configurations.

3) Final parameter configurations. We determined the final experimental parameter configurations according to the parameters corresponding to the largest SN values in each group. The main effects for SN have been displayed in Figure 7, in which the histograms give the parameter configurations. These parameters will be used in subsequent experiments to investigate the performance of NSGA-DRL further, as follows:

A. Small-Scale: $p_c = 0.8$, and $p_m = 0.01$;
B. Medium-Scale: $p_c = 0.8$, and $p_m = 0.01$;
C. Large-Scale: $p_c = 0.8$, and $p_m = 0.05$;



(a) Main Effects Plot for SN ratios on Small-Scale Problems (b) Main Effects Plot for SN ratios on Median-Scale Problems

(c) Main Effects Plot for SN ratios on Large-Scale Problems (d) Parameter Configurations

Figure 7. Results of Taguchi design.

5.4. *Experiment 3: ablation experiments*

In this experiment, we implement the ablation method to confirm the contribution of the proposed variational operator. The HV is used to measure algorithm performance.

1) Experimental setup and process. We use the parameter configurations and performance metrics above to evaluate the proposed variational operator's performance. The algorithms used in this experiment are described as follows.

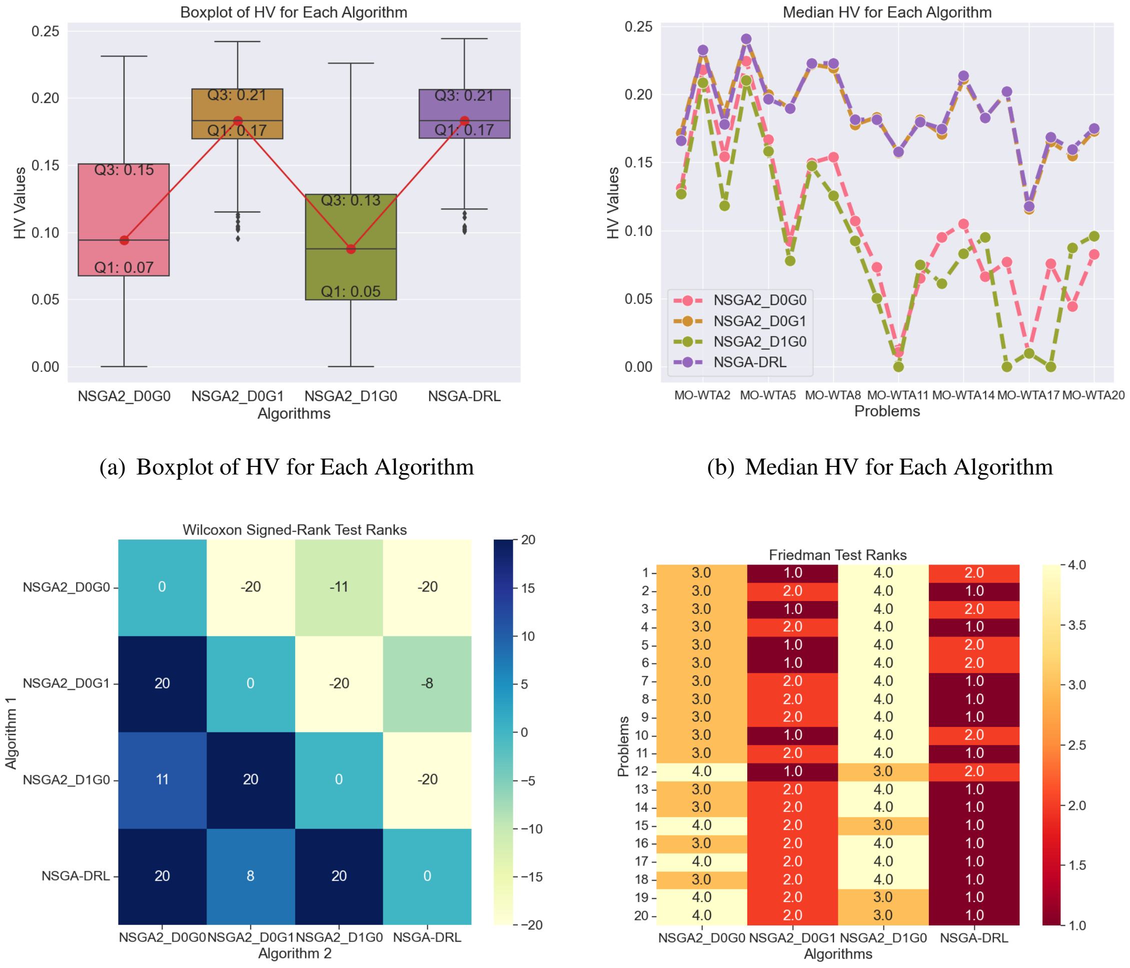
- NSGA2_D0G0. The algorithm uses the main framework in the paper, where single-point crossover and single-point mutation are used as variational operators.
- NSGA2_D0G1. The algorithm uses the main framework in the paper, where the proposed crossover with greedy and single-point mutation are used as variational operators.
- NSGA2_D1G0. The algorithm uses the main framework in the paper, where single-point crossover and the proposed adaptive mutation with DQN are used as variational operators.
- NSGA-DRL. The algorithm uses the main framework in the paper, where the proposed cossover with greedy and the proposed adaptive mutation with DQN are used as variational operators.

To obtain PF, we perform 21 independent runs for the above algorithms in the proposed benchmark problem. As a result, we compute the HV for all algorithms. In conjunction with the HV, we perform the signed Wilconxon rank sum test and the Friedman test for all algorithms in the proposed benchmark problem. Moreover, the algorithms are abstracted into two experimental groups (Algorithm 1 and Algorithm 2) to implement the Wilcoxon signed-rank test. Based on the HV values, we record that the performance of Algorithm 1 compared to Algorithm 2 is statistically better for 1 than 2 as +1, worse for 1 than 2 as -1, and equal for 1 and 2 as 0.

2) Experimental results

A. The HVs of all algorithms in the problem are counted in detail in subfigure (a) of Figure 8. We can conclude that the algorithms applying the proposed operators have a better HV based on the data. Moreover, the outliers in the data obtained by applying the proposed NSGA-DRL are much fewer, mainly concentrated in the interval [0.10, 0.12].
B. The median HV of all algorithms after multiple runs is also given in subfigure (b) of Figure 8. The median distribution can reflect the superiority of our proposed NSGA-DRL over NSGA2_D0G1, where the former has a 3% difference compared to the latter. The proposed adaptive mutation operator with DQN can guide the algorithm to search for new individuals with advantages. Specifically, the proposed NSGA-DRL and NSGA2_D0G1 can obtain larger HVs visualized from MO-WTA 5 to MO-WTA 20, where the large difference between the algorithms and those that do not use the proposed crossover is at most 93%.
C. The results of the Wilcoxon signed-rank test are displayed in subfigure (c) of Figure 8. Comparisons in the test show that NSGA2_D0G1 and NSGA-DRL have a significant advantage. Moreover, the performance of NSGA-DRL over the other algorithms on the proposed problems is statistically dominant (+20, +8 and +20).
D. The results of the Friedman test for all algorithms are given in subfigure (d) of Figure 8. In detail, we rank all algorithms based on the rule of larger HV value, where NSGA-DRL and NSGA2_D0G1 occupy 14 firsts and 6 firsts, respectively. This result further shows that the proposed operators help to explore the solutions.

3) Results analysis. The aim of this experiment is to verify the impact of the proposed crossover and mutation operators on the performance. This experiment combines four variational operators



	NSGA2_D0G0	NSGA2_D0G1	NSGA2_D1G0	NSGA-DRL
NSGA2_D0G0	0	-20	-11	-20
NSGA2_D0G1	20	0	-20	-8
NSGA2_D1G0	11	20	0	-20
NSGA-DRL	20	8	20	0

	NSGA2_D0G0	NSGA2_D0G1	NSGA2_D1G0	NSGA-DRL
1	3.0	1.0	4.0	2.0
2	3.0	2.0	4.0	1.0
3	3.0	1.0	4.0	2.0
4	3.0	2.0	4.0	1.0
5	3.0	1.0	4.0	2.0
6	3.0	1.0	4.0	2.0
7	3.0	2.0	4.0	1.0
8	3.0	2.0	4.0	1.0
9	3.0	2.0	4.0	1.0
10	3.0	1.0	4.0	2.0
11	3.0	2.0	4.0	1.0
12	4.0	1.0	3.0	2.0
13	3.0	2.0	4.0	1.0
14	3.0	2.0	4.0	1.0
15	4.0	2.0	3.0	1.0
16	3.0	2.0	4.0	1.0
17	4.0	2.0	4.0	1.0
18	3.0	2.0	4.0	1.0
19	4.0	2.0	3.0	1.0
20	4.0	2.0	3.0	1.0

(a) Boxplot of HV for Each Algorithm

(b) Median HV for Each Algorithm

(c) Wilcoxon Signed-Rank Test Ranks

(d) Friedman Test Ranks

Figure 8. Results of the ablation experiment.

(single-point crossover, single-point mutation, proposed crossover with greedy, and proposed adaptive mutation with DQN) to perform ablation experiments, where the algorithm in the paper is chosen as the main framework. The HV median can reflect that the proposed operator can explore the optimal solutions and make the distribution of the obtained PF more uniform. Here, the proposed crossover with greedy mechanism can fully utilize the existing individuals, while the adaptive mutation with DQN can explore new individuals to increase the diversity of the population. The results of Friedman's test show that the proposed adaptive mutation with DQN has a weak exploration ability on MO-WTA 1–5, which is also related to the small amount of data for DQN training. However, the proposed mutation demonstrates the advantages of DQN in all the remaining problems, which can be continuously explored in complex decision spaces. Finally, the Wilcoxon signed-rank test results provide a more intuitive representation of the effect of each operator on the algorithm after ablation. The proposed crossover can leverage population information to accelerate convergence, while the proposed mutation can explore new individuals to increase diversity. Therefore, the proposed operators in this paper have empirically shown to be positively gainful to the algorithm's performance. Moreover, the proposed adaptive mutation with DQN is the most prominent contribution to the performance.

5.5. Experiment 4: comparison experiments

5.5.1. Comparison experiment 1

In ablation experiments, we demonstrate that the proposed adaptive mutation with DQN significantly contributes to the algorithm's performance. We further design this comparison experiment to verify the performance under each different operator in the adaptive mechanism.

1) Experimental setup and process. We access each independent operator in the adaptive mechanism and the randomly selected adaptive mechanism into the algorithm, and they are each defined as follows:

- NSGA2-op1. The op1 operator in the mechanism acts as an independent mutation operator in the algorithm.
- NSGA2-op2. The op2 operator in the mechanism acts as an independent mutation operator in the algorithm.
- NSGA2-op3. The op3 operator in the mechanism acts as an independent mutation operator in the algorithm.
- NSGA2-rand. A randomized adaptive mechanism is chosen as the adaptive mutation operator in the algorithm.
- NSGA-DRL. The proposed adaptive operator with DQN is used as a mutation operator in the algorithm.

In detail, we still used 21 independent runs and calculated the HV using the obtained PF. After that, the HV metrics were utilized for statistical analysis. Here, the Wilcoxon signed-rank test and the Friedman test are applied to test for significant differences in performance.

2) Experimental results

A. The HVs of all algorithms in the proposed problems are tallied into a boxplot, shown in subfigure (a) of Figure 9. Moreover, NSGA-DRL still has the properties of uniformity and few outliers in the data distribution. Numerically, NSGA-DRL is close to the median of NSGA2-op1 (about 0.18), but the interval of outliers is much smaller ([0.1, 0.118])
B. The median HV in subfigure (b) of Figure 9 shows that the adaptive operator has a performance advantage, mainly due to constantly switching between different operators for exploration. However, the DQN-based adaptive mechanism outperforms the stochastic-based adaptive mechanism, where NSGA-DRL outperforms the other algorithms by a maximum of about 5.8, 2.9, 4.9, and 1.6%, respectively.
C. Statistically, the Wilcoxon signed-rank test indicates significant differences between algorithms. The adaptive operator with DQN statistically outperforms the other operators, as seen from the y-axis of subfigure (c) in Figure 9. The significant difference advantages of NSGA-DRL over other algorithms are +12, +20, +20, and +18, in that order.
D. However, the op1 operator is superior in small-scale problems as reflected in the Friedman test, where NSGA-DRL still has 16 firsts. It is still statistically inferior to the proposed operator in medium-scale and large-scale problems. The proposed operator can be explored on large-scale problems due to the DQN reward function and its learning mechanism.

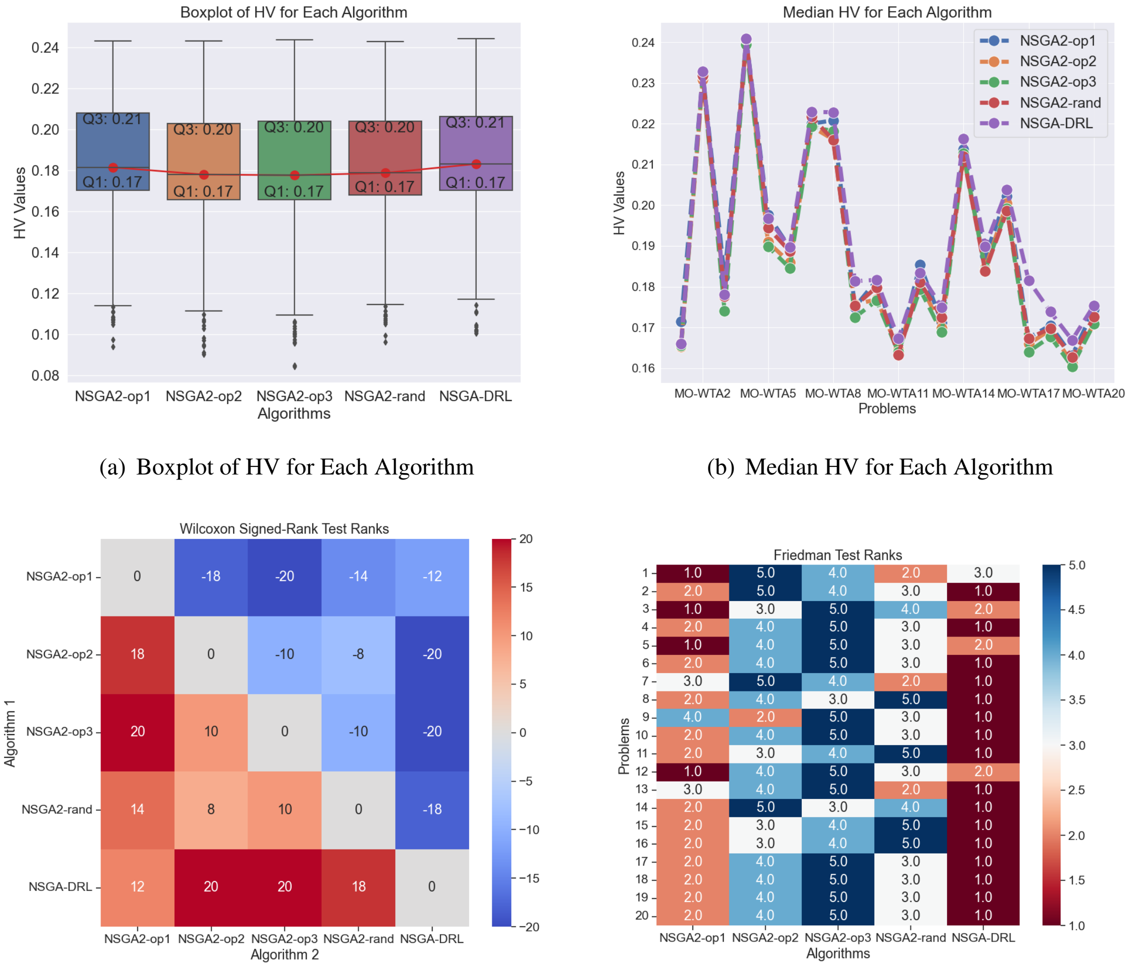
3) Results analysis. This experiment aims to verify the effect of each operator in the adaptive operator on the performance, as well as the effect of the proposed DQN-based and stochastic-based mechanisms on the performance. The proposed adaptive mutation with DQN speeds up algorithm convergence and maintains population diversity. The op1 operator included in the mechanism has a specific contribution, as evidenced by the fact that the algorithm with op1 in Friedman's test can show a top rank ranking in small-scale problems. This phenomenon indicates that the op1 operator in the mechanism has a significant advantage. In contrast, the proposed DQN adaptive operator has global performance to maintain good exploitability and convergence on the problems.

5.5.2. Comparison experiment 2

In the above experiments, we have demonstrated the contribution and effectiveness of the proposed method. We further investigate the performance of NSGA-DRL by comparing it with other baseline algorithms in this experiment.

1) Experimental setup and process. In this experiment, we adopt the parameter settings and the validated method from previously. NSGA-DRL and other baseline algorithms independently are involved in the proposed MO-WTA problems 21 times. Further, we perform HV calculations on the obtained PFs and statistically analyzed the HVs to infer the performance of each algorithm. In detail, we use the Wilcoxon signed-rank test and the Friedman test to verify the performance.

2) Experimental results. Intuitively, we give the experimental results in the form of images, which include the PF of each algorithm on the representative problem, the statistical results of HV, and the



(a) Boxplot of HV for Each Algorithm

(b) Median HV for Each Algorithm

	NSGA2-op1	NSGA2-op2	NSGA2-op3	NSGA2-rand	NSGA-DRL
NSGA2-op1	0	-18	-20	-14	-12
NSGA2-op2	18	0	-10	-8	-20
NSGA2-op3	20	10	0	-10	-20
NSGA2-rand	14	8	10	0	-18
NSGA-DRL	12	20	20	18	0

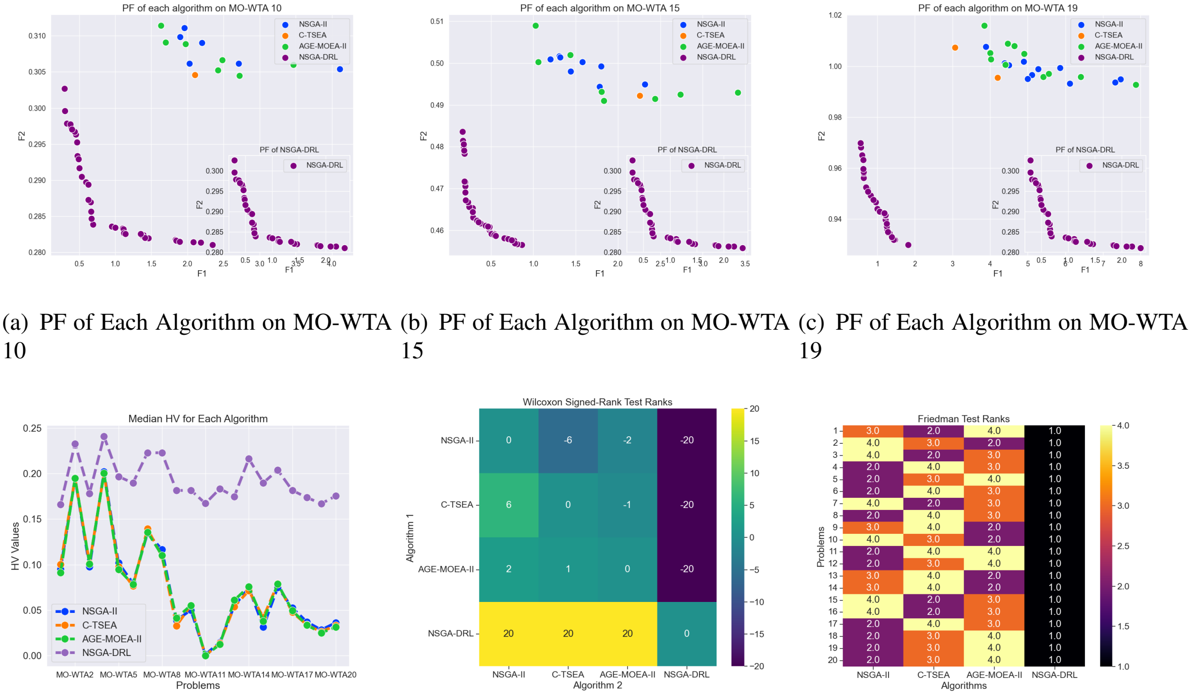
(c) Wilcoxon Signed-Rank Test Ranks

	NSGA2-op1	NSGA2-op2	NSGA2-op3	NSGA2-rand	NSGA-DRL
1	1.0	5.0	4.0	2.0	3.0
2	2.0	5.0	4.0	3.0	1.0
3	1.0	3.0	5.0	4.0	2.0
4	2.0	4.0	5.0	3.0	1.0
5	1.0	4.0	5.0	3.0	2.0
6	2.0	4.0	5.0	3.0	1.0
7	3.0	5.0	4.0	2.0	1.0
8	2.0	4.0	3.0	5.0	1.0
9	4.0	2.0	5.0	3.0	1.0
10	2.0	4.0	5.0	3.0	1.0
11	2.0	3.0	4.0	5.0	1.0
12	1.0	4.0	5.0	3.0	2.0
13	3.0	4.0	5.0	2.0	1.0
14	2.0	5.0	3.0	4.0	1.0
15	2.0	3.0	4.0	5.0	1.0
16	2.0	3.0	4.0	5.0	1.0
17	2.0	4.0	5.0	3.0	1.0
18	2.0	4.0	5.0	3.0	1.0
19	2.0	4.0	5.0	3.0	1.0
20	2.0	4.0	5.0	3.0	1.0

(d) Friedman Test Ranks

Figure 9. Results of comparison experiment 1.

results of the Wilcoxon signed-rank test and Friedman test. In this experiment, we take MO-WTA10, 15, and 19 to represent small-scale, medium-scale, and large-scale problems, respectively. In detail, the PF of each algorithm as well as the statistical results are shown in Figure 10. Based on the obtained data, we embedded NSGA-DRL as a subgraph to represent PF more clearly in Figure 10.



(a) PF of Each Algorithm on MO-WTA 10 (b) PF of Each Algorithm on MO-WTA 15 (c) PF of Each Algorithm on MO-WTA 19

(d) Median HV for Each Algorithm (e) Wilcoxon Signed-Rank Test Ranks (f) Friedman Test Ranks

Figure 10. Results of comparison experiment 2.

A. Subfigures (a)–(c) in Figure 10 show the PF of each algorithm for different problems. The overall distribution of PF is better for all algorithms, but the proposed NSGA-DRL can explore smaller intervals and also has good diversity. In particular, AGE-MOEA-II shows superior diversity and PF distribution on large-scale problems, but the convergence is still not as good as the proposed NSGA-DRL. The above sufficiently shows that NSGA-DRL can guarantee that diversity still converges to a near-optimum under limited evaluation.
B. Subfigures (d) in Figure 10 shows the median HV for each algorithm under 21 runs. It is intuitive to conclude that the HV of NSGA-DRL is overall higher than all other algorithms, where it outperforms NSGA-II, C-TSEA, and AGE-MOEA-II by 64.3%, 64.2%, and 64.4%, respectively.
C. Subfigures (e) in Figure 10 represents the significant difference between the algorithms, which is visualized by the signed rank. Intuitively, NSGA-DRL compares to all other algorithms with a signed-rank sum result of +20, which can illustrate the difference between NSGA-DRL and other algorithms with significant advantages.
D. Subfigures (f) in Figure 10 expresses the ranking of the rank of each algorithm on the problem, where the larger the HV, the higher the ranking. Thus, NSGA-DRL has the highest ranking on the problem, which further accounts for the difference in its significance advantage.

3) Results analysis In the small-scale problem, NSGA-DRL and other algorithms can search the optimal solutions, but the greedy-based crossover operator can quickly guide the algorithm to optimize in a small solution space. With the same number of evaluations in the large-scale problem, the PF distribution of NSGA-DRL is more uniform and maintains diversity. The reason is that the DQN-based adaptive mutation operator in NSGA-DRL can accelerate the convergence, which continuously performs operator selection through the learned model. Besides, AGE-MOEA-II and C-TESA also perform well in large-scale problems, but their complex evaluation mechanism makes it difficult to adapt them to large-scale problems with the constraints in this paper. Therefore, we can conclude that the proposed crossover operator has a great contribution to the global optimization, while the DQN-based adaptive mutation operator shows a local exploration capability for the same number of evaluations. Further, the median HV is able to show the overall performance, where the gap with all other algorithms is statistically above 60%, indicating that the proposed NSGA-DRL performance is superior. Statistically, the performance of NSGA-DRL is evaluated in detail by the Wilcoxon signed-rank test and the Friedman test, where the results of the Wilcoxon test demonstrate the differences between algorithms, while the results of the Friedman test demonstrate the specific differences of the algorithms on each problem.

6. Discussion

The ablation studies conducted herein reveal that the mutation and crossover operators introduced in our algorithm significantly enhance its performance, with the mutation operator exerting a more pronounced influence when used in conjunction with the crossover operator. This finding prompted a deeper examination of the proposed DQN-based mutation operator in comparison experiment 1, which demonstrated its efficacy in forecasting superior candidate solutions. Subsequently, the integration of the DQN-based mutation and greedy-based crossover in NSGA-DRL was scrutinized, highlighting the method's superior performance over contemporary state-of-the-art algorithms.

Contrasting with the work presented in Tian et al. [36], which introduced an operation selection mechanism devoid of real-world application generalization despite its demonstrated feasibility on benchmark problems, our study extends the application of a DQN-based operator mechanism to the real-world MO-WTA problem. This is achieved by integrating several practical operators, with the effectiveness of these methods substantiated through multi-dimensional evaluation in ablation and comparison experiments. Furthermore, Wang et al. [19] proposed an adaptive variational mechanism that conjoins crossover and mutation operators. However, their approach necessitates choosing between crossover and variation, which may impede the balance between exploration and exploitation. In contrast, our research advocates for distinct crossover and mutation designs, with the DQN-enhanced adaptive mutation operator preserving exploration efficacy.

The NSGA-DRL framework, augmented with the proposed DQN-based mutation and greedy-based crossover, is meticulously crafted for real-world MO-WTA problems. The crossover operator imposes regularization on the MO-WTA problem constraints, expediting algorithmic convergence, while the mutation operator leverages neural networks to generalize problem-solving capabilities, thus bolstering solution diversity without compromising quality.

Despite these advancements, this study's limitation stems from the offline learning integration with NSGA-DRL. Future research will pivot towards amalgamating online learning with MOEA to tackle real-world combinatorial optimization challenges more effectively.

7. Conclusions

In this study, we have introduced an innovative multi-objective evolutionary optimization algorithm (MOEA) enhanced by a deep Q-network (DQN)-based mutation operator and a greedy-based crossover operator. The enhanced NSGA-II framework (NSGA-DRL), incorporating these novel operators, has been specifically tailored to address the complexities of the multi-missile target assignment problem. The DQN-based adaptive mutation operator has proven particularly effective in predicting high-quality solutions, significantly bolstering exploration capabilities and promoting diversity within the solution set. Additionally, our greedy-based crossover operator leverages domain-specific knowledge to curtail unproductive searches, primarily focusing on intensifying exploitation and accelerating convergence rates.

The ablation studies and subsequent analyses underscore the substantial impact of the proposed mutation and crossover operators on the algorithm's overall performance. Specifically, the DQN-based mutation operator has demonstrated its predictive prowess in efficiently identifying promising candidate solutions, as evidenced in comparison experiment 1. Furthermore, the proposed NSGA-DRL framework exhibits remarkable proficiency in managing multi-missile target assignment challenges, consistently generating superior candidate solutions.

Looking ahead, our future work will delve into more intricate and dynamic multi-missile target assignment scenarios. We aim to explore real-time interactions for online learning enhancements within the MOEA paradigm. This forward-looking approach is anticipated to further refine the algorithm's robustness and adaptability to real-world applications.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Major Program of Natural Science Foundation of China (61690210).

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. R. A. Murphey, *Target-Based Weapon Target Assignment Problems*, Springer US, 2000.

2. R. K. Ahuja, A. Kumar, K. C. Jha, J. B. Orlin, Exact and heuristic algorithms for the weapon-target assignment problem, *Oper. Res.*, **55** (2007), 1136–1146. https://doi.org/10.1287/opre.1070.0440

3. Y. Lu, D. Z. Chen, A new exact algorithm for the weapon-target assignment problem, *Omega*, **98** (2021), 102138. https://doi.org/10.1016/j.omega.2019.102138

4. C. Leboucher, H. Shin, S. Le Ménec, A. Tsourdos, A. Kotenkoff, P. Siarry, et al., Novel evolutionary game based multi-objective optimisation for dynamic weapon target assignment, *IFAC Proc. Vol.*, **47** (2014), 3936–3941. https://doi.org/10.3182/20140824-6-ZA-1003.02150

5. B. Xin, J. Chen, Z. Peng, L. Dou, J. Zhang, An efficient rule-based constructive heuristic to solve dynamic weapon-target assignment problem, *IEEE Trans. Syst. Man Cybern. Part A*, **41** (2010), 598–606. https://doi.org/10.1109/TSMCA.2010.2089511

6. Z. J. Lee, C. Y. Lee, S. F. Su, An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem, *Appl. Soft Comput.*, **2** (2002), 39–47. https://doi.org/10.1016/S1568-4946(02)00027-3

7. X. Li, D. Zhou, Q. Pan, Y. Tang, J. Huang, Weapon-target assignment problem by multiobjective evolutionary algorithm based on decomposition, *Complexity*, **2018** (2018). https://doi.org/10.1155/2018/8623051

8. T. Chang, D. Kong, N. Hao, K. Xu, G. Yang, Solving the dynamic weapon target assignment problem by an improved artificial bee colony algorithm with heuristic factor initialization, *Appl. Soft Comput.*, **70** (2018), 845–863. https://doi.org/10.1016/j.asoc.2018.06.014

9. Y. Wang, B. Xin, J. Chen, An adaptive memetic algorithm for the joint allocation of heterogeneous stochastic resources, *IEEE Trans. Cybern.*, **52** (2021), 11526–11538. https://doi.org/10.1109/TCYB.2021.3087363

10. L. Zhao, Z. An, B. Wang, Y. Zhang, Y. Hu, A hybrid multi-objective bi-level interactive fuzzy programming method for solving ecm-dwta problem, *Complex Intell. Syst.*, **8** (2022), 4811–4829. https://doi.org/10.1007/s40747-022-00730-9

11. X. Chang, J. Shi, Z. Luo, Y. Liu, Adaptive large neighborhood search algorithm for multi-stage weapon target assignment problem, *Comput. Ind. Eng.*, **181** (2023), 109303. https://doi.org/10.1016/j.cie.2023.109303

12. Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.*, **11** (2007), 712–731. https://doi.org/10.1109/TEVC.2007.892759

13. M. Behzadian, S. K. Otaghsara, M. Yazdani, J. Ignatius, A state-of the-art survey of TOPSIS applications, *Expert Syst. Appl.*, **39** (2012), 13051–13069. https://doi.org/10.1016/j.eswa.2012.05.056

14. Q. Cheng, D. Chen, J. Gong, Weapon-target assignment of ballistic missiles based on q-learning and genetic algorithm, in *2021 IEEE International Conference on Unmanned Systems (ICUS)*, (2021), 908–912. https://doi.org/10.1109/ICUS52573.2021.9641190

15. H. Mouton, H. L. Roux, J. Roodt, Applying reinforcement learning to the weapon assignment problem in air defence, *Sci. Militaria S. Afr. J. Military Stud.*, **39** (2011), 99–116. https://doi.org/10.5787/39-2-115

16. F. Meng, K. Tian, C. Wu, Deep reinforcement learning-based radar network target assignment, *IEEE Sens. J.*, **21** (2021), 16315–16327. https://doi.org/10.1109/JSEN.2021.3074826

17. S. Li, X. He, X. Xu, T. Zhao, C. Song, J. Li, Weapon-target assignment strategy in joint combat decision-making based on multi-head deep reinforcement learning, *IEEE Access*, **11** (2023), 113740–113751. https://doi.org/10.1109/ACCESS.2023.3324193

18. C. Li, B. Xin, Y. He, D. Wang, Y. Li, Dynamic weapon target assignment based on deep q network, in *2023 42nd Chinese Control Conference (CCC)*, (2023), 1773–1778. https://doi.org/10.23919/CCC58697.2023.10240428

19. T. Wang, L. Fu, Z. Wei, Y. Zhou, S. Gao, Unmanned ground weapon target assignment based on deep q-learning network with an improved multi-objective artificial bee colony algorithm, *Eng. Appl. Artif. Intell.*, **117** (2023), 105612. https://doi.org/10.1016/j.engappai.2022.105612

20. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.*, **6** (2002), 182–197. https://doi.org/10.1109/4235.996017

21. H. Cai, J. Liu, Y. Chen, H. Wang, Survey of the research on dynamic weapon-target assignment problem, *J. Syst. Eng. Electron.*, **17** (2006), 559–565. https://doi.org/10.1016/S1004-4132(06)60097-2

22. A. Kline, D. Ahner, R. Hill, The weapon-target assignment problem, *Comput. Oper. Res.*, **105** (2019), 226–236. https://doi.org/10.1016/j.cor.2018.10.015

23. R. A. Murphey, *An Approximate Algorithm For A Weapon Target Assignment Stochastic Program*, Springer US, 2000.

24. O. Karasakal, Air defense missile-target allocation models for a naval task group, *Comput. Oper. Res.*, **35** (2008), 1759–1770. https://doi.org/10.1016/j.cor.2006.09.011

25. M. S. Hughes, B. J. Lunday, The weapon target assignment problem: Rational inference of adversary target utility valuations from observed solutions, *Omega*, **107** (2022), 102562. https://doi.org/10.1016/j.omega.2021.102562

26. Z. J. Lee, S. F. Su, C. Y. Lee, Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics, *IEEE Trans. Syst. Man Cybern. Part B*, **33** (2003), 113–121. https://doi.org/10.1109/TSMCB.2003.808174

27. A. M. Madni, M. Andrecut, Efficient heuristic approach to the weapon-target assignment problem, *J. Aerosp. Comput. Inf. Commun.*, **6** (2009), 405–414. https://doi.org/10.2514/1.34254

28. Z. R. Bogdanowicz, A. Tolano, K. Patel, N. P. Coleman, Optimization of weapon–target pairings based on kill probabilities, *IEEE Trans. Cybern.*, **43** (2012), 1835–1844. https://doi.org/10.1109/TSMCB.2012.2231673

29. H. Liang, F. Kang, Adaptive chaos parallel clonal selection algorithm for objective optimization in WTA application, *Optik*, **127** (2016), 3459–3465. https://doi.org/10.1016/j.ijleo.2015.12.122

30. Z. Li, Y. Chang, Y. Kou, H. Yang, A. Xu, Y. Li, Approach to WTA in air combat using IAFSA-IHS algorithm, *J. Syst. Eng. Electron.*, **29** (2018), 519–529. https://doi.org/10.21629/JSEE.2018.03.09

31. M. Cao, W. Fang, Swarm intelligence algorithms for weapon-target assignment in a multilayer defense scenario: A comparative study, *Symmetry*, **12** (2020), 824. https://doi.org/10.3390/sym12050824

32. J. Li, J. Chen, B. Xin, L. Dou, Solving multi-objective multi-stage weapon target assignment problem via adaptive NSGA-II and adaptive MOEA/D: A comparison study, in *2015 IEEE Congress on Evolutionary Computation (CEC)*, (2015), 3132–3139. https://doi.org/10.1109/CEC.2015.7257280

33. W. Xu, C. Chen, S. Ding, P. M. Pardalos, A bi-objective dynamic collaborative task assignment under uncertainty using modified MOEA/D with heuristic initialization, *Expert Syst. Appl.*, **140** (2020), 112844. https://doi.org/10.1016/j.eswa.2019.112844

34. Y. Zhao, J. Liu, J. Jiang, Z. Zhen, Shuffled frog leaping algorithm with non-dominated sorting for dynamic weapon-target assignment, *J. Syst. Eng. Electron.*, **34** (2023), 1007–1019. https://doi.org/10.23919/JSEE.2023.000102

35. R. Durgut, M. E. Aydin, I. Atli, Adaptive operator selection with reinforcement learning, *Inf. Sci.*, **581** (2021), 773–790. https://doi.org/10.1016/j.ins.2021.10.025 https://doi.org/10.1007/978-3-030-85672-4 https://doi.org/10.1007/978-3-030-85672-4_3

36. 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. Emerging Top. Comput. Intell.*, **7** (2023), 1051–1064. https://doi.org/10.1109/TETCI.2022.3146882

37. M. A. Wiering, M. V. Otterlo, Reinforcement learning, *Adapt. Learn. Optim.*, **12** (2012), 729. https://doi.org/10.1007/978-3-642-27645-3

38. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, et al., Playing atari with deep reinforcement learning, preprint, arXiv:1312.5602. https://doi.org/10.48550/arXiv.1312.5602

39. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, preprint, arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

40. R. Girshick, Fast R-CNN, in *Proceedings of the IEEE international conference on computer vision*, (2015), 1440–1448. https://doi.org/10.1109/ICCV.2015.169

41. T. Blickle, Tournament selection, *Evol. Comput.*, **1** (2000), 181–186. https://doi.org/10.1887/0750308958

42. X. Zhang, Y. Tian, R. Cheng, Y. Jin, An efficient approach to nondominated sorting for evolutionary multiobjective optimization, *IEEE Trans. Evol. Comput.*, **19** (2014), 201–213. https://doi.org/10.1109/TEVC.2014.2308305

43. F. Ming, W. Gong, H. Zhen, S. Li, L. Wang, Z. Liao, A simple two-stage evolutionary algorithm for constrained multi-objective optimization, *Knowl. Based Syst.*, **228** (2021), 107263. https://doi.org/10.1016/j.knosys.2021.107263

44. A. Panichella, An improved pareto front modeling algorithm for large-scale many-objective optimization, in *Proceedings of the Genetic and Evolutionary Computation Conference*, (2022), 565–573. https://doi.org/10.1145/3512290.3528732

45. A. P. Guerreiro, C. M. Fonseca, L. Paquete, The hypervolume indicator: Computational problems and algorithms, *ACM Comput. Surv.*, **54** (2021), 1–42. https://doi.org/10.1145/3453474

46. A. Freddi, M. Salmon, *Introduction to the Taguchi Method*, Springer International Publishing, 2019.

47. W. K. Mashwani, A. Salhi, M. A. Jan, R. A. Khanum, M. Sulaiman, Impact analysis of crossovers in a multi-objective evolutionary algorithm, *Sci. Int.*, **27** (2015), 4943–4956.

48. X. Shi, S. Zou, S. Song, R. Guo, A multi-objective sparse evolutionary framework for large-scale weapon target assignment based on a reward strategy, *J. Intell. Fuzzy Syst.*, **40** (2021), 10043–10061. https://doi.org/10.3233/JIFS-202679

49. S. Zou, X. Shi, S. Song, A multi-objective optimization framework with rule-based initialization for multi-stage missile target allocation, *Math. Biosci. Eng.*, **20** (2023), 7088–7112. https://doi.org/10.3934/mbe.2023306

Appendix

The data and results in the experiments are available at https://github.com/shiqi0404/MOEA_DQN.

A1. Median HV for each algorithm in the experiments

In this section, the HV obtained from the experiments are summarized in tables. In the tables, the data represent the median HV obtained by the algorithm under 21 independent runs, where the highlighted data indicate the best HVs. Furthermore, D denotes the number of decision variables in the problem.

Table A1. Median HV for each algorithm in ablation experiments.

Problem No.	D	NSGA2_D0G0	NSGA2_D0G1	NSGA2_D1G0	NSGA-DRL
1	12	0.1310	**0.1718**	0.1216	0.1659
2	15	0.2181	0.2325	0.2065	**0.2329**
3	18	0.1543	**0.1858**	0.1334	0.1781
4	21	0.2247	0.2407	0.2180	**0.2409**
5	24	0.1671	**0.1999**	0.1233	0.1968
6	30	0.0920	**0.1897**	0.1259	0.1896
7	36	0.1497	0.2220	0.1522	**0.2229**
8	42	0.1542	0.2194	0.0993	**0.2228**
9	36	0.1071	0.1776	0.0855	**0.1813**
10	45	0.0733	**0.1832**	0.0932	0.1817
11	54	0.0109	0.1571	0.0540	**0.1672**
12	63	0.0648	0.1816	0.0788	**0.1833**
13	48	0.0952	0.1706	0.0313	**0.1749**
14	60	0.1052	0.2111	0.0037	**0.2163**
15	72	0.0661	0.1827	0.0905	**0.1899**
16	84	0.0772	0.2019	0.0506	**0.2038**
17	96	0.0121	0.1156	0.0181	**0.1815**
18	120	0.0757	0.1653	0.0948	**0.1738**
19	144	0.0444	0.1549	0.0813	**0.1669**
20	168	0.0827	0.1731	0.0587	**0.1754**
+/=/-		0/0/20	6/6/8	0/0/20	

”+/=/-” indicates that the algorithm is statistically better, worse, or equal to NSGA-DRL under the Wilcoxon signed-rank test, respectively.

Table A2. Median HV for each algorithm in comparison experiment 1.

Problem No.	D	NSGA2-op1	NSGA2-op2	NSGA2-op3	NSGA2-rand	NSGA-DRL
1	12	**0.1714**	0.1653	0.1655	0.1661	0.1659
2	15	0.2322	0.2309	0.2317	0.2318	**0.2329**
3	18	**0.1824**	0.1779	0.1740	0.1777	0.1781
4	21	**0.2409**	0.2406	0.2395	0.2408	**0.2409**
5	24	**0.1976**	0.1909	0.1899	0.1945	**0.1968**
6	30	**0.1896**	0.1859	0.1846	0.1887	**0.1896**
7	36	0.2200	0.2193	0.2194	0.2218	**0.2229**
8	42	0.2208	0.2161	0.2182	0.2160	**0.2228**
9	36	0.1750	0.1756	0.1725	0.1753	**0.1813**
10	45	0.1816	0.1769	0.1766	0.1798	**0.1817**
11	54	0.1647	0.1646	0.1639	0.1632	**0.1672**
12	63	**0.1854**	0.1809	0.1793	0.1810	0.1833
13	48	0.1712	0.1698	0.1688	0.1724	**0.1749**
14	60	0.2137	0.2120	0.2128	0.2121	**0.2163**
15	72	0.1875	0.1845	0.1841	0.1838	**0.1899**
16	84	0.2023	0.2005	0.1992	0.1986	**0.2038**
17	96	0.1675	0.1659	0.1640	0.1673	**0.1815**
18	120	0.1705	0.1694	0.1677	0.1697	**0.1738**
19	144	0.1630	0.1611	0.1604	0.1627	**0.1669**
20	168	0.1744	0.1708	0.1708	0.1725	**0.1754**
+/=/-		6/2/12	0/0/20	0/0/20	0/2/18	

”+/=/-” indicates that the algorithm is statistically better, worse, or equal to NSGA-DRL under the Wilcoxon signed-rank test, respectively.

Table A3. Median HV for Each Algorithm in comparison experiment 2.

Problem No.	D	NSGA-II	C-TSEA	AGE-MOEA-II	NSGA-DRL
1	12	0.0951	0.1002	0.0912	**0.1659**
2	15	0.1945	0.1949	0.1950	**0.2329**
3	18	0.0976	0.1013	0.1007	**0.1781**
4	21	0.2023	0.2004	0.2004	**0.2409**
5	24	0.1018	0.0978	0.0948	**0.1968**
6	30	0.0795	0.0764	0.0782	**0.1896**
7	36	0.1345	0.1393	0.1357	**0.2229**
8	42	0.1166	0.1093	0.1102	**0.2228**
9	36	0.0387	0.0328	0.0412	**0.1813**
10	45	0.0501	0.0551	0.0552	**0.1817**
11	54	0.0011	0.0187	0.0146	**0.1672**
12	63	0.0147	0.0136	0.0123	**0.1833**
13	48	0.0547	0.0540	0.0609	**0.1749**
14	60	0.0745	0.0713	0.0757	**0.2163**
15	72	0.0315	0.0420	0.0380	**0.1899**
16	84	0.0750	0.0791	0.0786	**0.2038**
17	96	0.0526	0.0476	0.0495	**0.1815**
18	120	0.0371	0.0339	0.0336	**0.1738**
19	144	0.0283	0.0261	0.0251	**0.1669**
20	168	0.0361	0.0331	0.0312	**0.1754**
+/=/-		0/0/20	0/0/20	0/0/20	

”+/=/-” indicates that the algorithm is statistically better, worse, or equal to NSGA-DRL under the Wilcoxon signed-rank test, respectively.