[1]Kainuo Chen,
[1]Fuguang Zhang,
[1]Yantao Yin,
[1]Guangchuan Du,
[2]Han Zhang

Missile Life Extension Maintenance Scheduling Optimization: Multi-Objective Differential Evolution Algorithm Based on Reinforcement Learning

Journal of Electrical Systems

Abstract: **-** Missile life extension maintenance is a key component of equipment storage and life extension engineering, involving complex maintenance task scheduling problems. This paper proposes a multi-objective differential evolution algorithm based on reinforcement learning to solve the multi-objective optimization problem of how to minimize the overall maintenance time and prioritize the completion of key maintenance tasks under limited resources and strict logical constraints. First, the algorithm constructs a multi-strategy framework to transform the goals of minimizing the overall maintenance time and completing the priority tasks into solving the Pareto optimal solution set. Then, this paper integrates reinforcement learning to adaptively adjust the parameters of the differential evolution algorithm to optimize the search efficiency and accuracy. Finally, experiments are conducted on the missile maintenance benchmark dataset to verify the effectiveness and superiority of the proposed algorithm. Simulation results show that compared with the existing algorithms, this algorithm has an improvement advantage in maintenance scheduling effect. In addition, the ablation experiment further confirms the practical effectiveness of the optimization measures in improving the performance of the algorithm.

Keywords: Differential Evolution Algorithm (DE); Multi-Objective (MO); Reinforcement Learning (RL); Missile Life Extension and Maintenance (MLEM); Flexible Operation Scheduling (FOS).

I. INTRODUCTION

Missile equipment maintenance task scheduling decision involves unified control and arrangement of all maintenance tasks and resources. The purpose is to rationally select and assign tasks to appropriate maintenance organizations, and optimize the order and scheduling of task execution. The complexity of this process mainly stems from the combined influence of multiple dynamic factors such as the complexity and multi-professional nature of missile equipment, the time-consuming maintenance work, and the limited maintenance resources **Error! Reference source not found.**. Missile life extension maintenance scheduling is part of missile equipment maintenance task scheduling, which involves decomposing each equipment maintenance project into multiple operation stages and dividing them according to the different maintenance functional components. Each maintenance operation stage is carried out in different maintenance operation rooms, equipped with corresponding maintenance equipment and operators 0. Missile life extension maintenance support includes multiple operation stages, and the completion time of each stage should have a theoretical value and be numbered in sequence. When grassroots maintenance organizations receive missile equipment maintenance plan instructions, they need to complete the maintenance support work of all missiles in the shortest possible time, while ensuring that the maintenance tasks of some key missiles are given priority. The goal of missile life extension maintenance scheduling is to apply reliability and maintainability theory under specified conditions, tap the technical potential of the product, and take design, maintenance and management measures for key parts that affect reliability to achieve an extension of the storage period. However, current engineering practice has found that there is a serious time lag in both life extension testing and implementation of life extension measures [3]. For example, the service life of a certain model of product is 10 years. After the specified service life of the product, it is necessary to start the life extension work, including life extension project demonstration, technical research, formulation of life extension test plan, accelerated life test, and life extension repair of the remaining products [3]. However, sometimes after the life extension repair is completed, the time point for the second life extension has arrived before it is put into use. Therefore, two prerequisites must be put forward for the missile life extension project: one is to complete the maintenance and support work of all missiles in the shortest possible time, and the other is to ensure that the maintenance tasks of some key missiles are given priority. However, due to the limitation of resources and

[1] *Corresponding author: Kainuo Chen, Affiliation :Naval Aviation University, Yantai 264001 , China
[1] Fuguang Zhang ,Affiliation: Naval Aviation University, Yantai 264001 , China
[1]Yantao Yin, Affiliation: Naval Aviation University, Yantai 264001 , China
[1]Guangchuan Du, Affiliation: Naval Aviation University, Yantai 264001 , China
[2]Han Zhang, Affiliation: Yantai Education Engineering Evaluation Institute , Yantai 264003 , China

management processes, the missile life extension repair scheduling needs to handle multiple tasks at the same time, and each task involves multiple operations. This leads to the project overtime of the missile life extension repair support time. Therefore, it is particularly urgent to manage and optimize the scheduling of missile life extension repair resources. Missile life extension repair scheduling management optimization is to solve the special flexible operation scheduling multi-objective problem of completing the maintenance and support work of all missiles in the shortest possible time and ensuring that the maintenance tasks of some key missiles are given priority.

Flexible job scheduling is a common problem in industrial manufacturing, production and support, and is an NP-hard problem in combinatorial optimization. This problem involves how to allocate resources to N job operations given M resources and determine the optimal resource usage order to improve scheduling performance and efficiency [4]. The main research direction to solve this problem is heuristic methods [5]. [6]et al. developed a genetic algorithm that initializes the population through global selection and local selection, adopts an improved chromosome representation, and uses specific crossover and mutation operations to minimize the completion time of the constrained scheduling problem. Shi et al. [8] designed a multi-objective differential evolution algorithm to minimize the maximum completion time and total outsourcing cost as the optimization goal to achieve the problem of undelayed delivery of customer orders through joint optimization of outsourcing and internal job scheduling in a job shop environment . Sun [7]used a variable domain search hybrid genetic algorithm to provide an effective solution to the completion time minimization problem of shop scheduling . Liu [9]et al. proposed to use a genetic algorithm model for optimization, construct a missile batch technical preparation scheduling optimization model, and solve the model using a genetic algorithm. Wei et al. **Error! Reference source not found.**proposed a binary hybrid improved genetic algorithm to solve the mixed-line production scheduling model of flexible workshops with equipment-energy consumption curves, and achieved good production guidance effects on multi-objective scheduling problems. Chen et [11]proposed a self-learning genetic algorithm (SLGA) based on reinforcement learning, and optimized the workshop operation resource scheduling by adjusting the parameters of the genetic algorithm. Guo et al. [12]established a maintenance support model for aircraft dispatch and solved it through a reinforcement learning algorithm, effectively improving the utilization rate of aircraft and saving maintenance costs . Zeng et al [13] A self-learning tabu search algorithm based on deep reinforcement learning (DSLTS) is proposed. The algorithm uses the tabu search algorithm as the basic optimization method and adopts a double-layer deep Q network to intelligently adjust the key parameters of the tabu search algorithm . Du et al. [14]proposed a study on the flexible job shop scheduling problem based on a hybrid multi-objective optimization algorithm. The algorithm combines the distribution estimation algorithm and the genetic algorithm to effectively deal with the scheduling problem constrained by electricity prices . Wu et al. [15]optimized the fixture loading and unloading time under dual resource constraints through an improved non-dominated sorting genetic algorithm, confirming the superiority of multi-resource scheduling scheme over single resource scheduling in production guidance .

This paper proposes an adaptive multi-objective differential evolution algorithm to solve the problem of missile life extension maintenance task scheduling. The article is structured as follows: Chapter 1 first describes the life extension maintenance task scheduling problem in detail, and establishes the corresponding mathematical model in Chapter 2. Chapter 3 discusses the theoretical basis of the algorithm in depth, including differential evolution algorithm, reinforcement learning and cost-effectiveness method. Chapter 4 introduces the adaptive multi-objective differential evolution algorithm in detail. Chapter 5 verifies the effectiveness of the algorithm through experiments and discusses the experimental results. Finally, the experimental results are summarized and the direction of future research is proposed .

II. Problem Description

The maintenance and support of a certain type of missile includes J operation stages, the theoretical completion time of each stage has been clearly defined, and the maintenance operations have been numbered according to the execution order. On this basis, the missile maintenance organization (grassroots level) received an equipment maintenance plan instruction, requiring the maintenance and support work of I missiles to be completed in the shortest possible time, while giving priority to ensuring that the maintenance work of F missiles is completed first. Assume that there are a total of R missile maintenance organizations with maintenance lines. Each line is equipped with the required operating equipment and operators, and to ensure the proficiency and reliability of the operators, the operators are bound to the maintenance operation room. Due to differences in missile batches and service years, the time it takes to process different missiles in the same maintenance operation room is also different . At the same time, due to the different proficiency of operators, the time required for the same missile in different maintenance

operation rooms will also be different . This paper conducts simulation analysis on the statistical data of previous maintenance guarantees to obtain the time required for different missiles to complete the maintenance operation stage in each optional maintenance operation room.

III. MATHEMATICAL MODEL

The mathematical model consists of three parts: variable definition, assumptions, and mathematical model. The variable definition is shown in Table 1:

Table 1Variable definitions

variable name	describe
r	Maintenance organization number, *r=1,2,. .. ,R*
i	Missile number, *i = 1, 2, . . . , I*
M_i	Missile numbered *i*
j	Maintenance operation number, *j=1,2,. .. ,J*
k	Maintenance organization operation room number , *k=1,2,. .. ,R*J*
R	The operation room numbered *k* , *k=1,2,. .. ,R*J*
$S_{i,j}$	The *jth* operation of the i - th missile
$ST_{i,,j}$	The start time of the *jth operation* of *the* ith missile
$FT_{i,,j}$	the jth *operation* of the i - th missile
$RT_{i,,j}$	The time required for the jth operation of *the* i - *th* missile
$SR_{n,i,j}$	In the nth maintenance *organization* , the *jth* operation of the *ith* missile
$ST_{n,i,,j}$	the *jth* operation on the ith *missile* in the *nth* maintenance organization
$FT_{n,i,,j}$	the *jth* operation on the ith *missile* in the *nth* maintenance organization
$RT_{n,i,,j}$	The time required for the jth operation on *the* ith missile *in* the nth *maintenance* organization
$Yi_{j,k}$	maintenance operation *j of* missile *i* is completed in maintenance operation room *k* . Yes ($Yi_{j,k}=1$), No ($Yi_{j,k}=0$)
$F_{i,j,e,f,k}$	Operation $Si_{,j}$ is processed before operation $Se_{,f}$ on operation room *k* . Yes ($Fi_{j,e,f,k}=1$), No ($Fi_{j,e,f,k}=0$)
FT	Time required to complete repair of the first *F missiles*
CI	first *F* missiles to be completed

Assumptions are as follows:

of a certain type of missile follows the following assumptions:

(1) All repair shops are available from the start.

(2) All missiles were ready for repair at the initial moment.

(3) During maintenance, all missiles have the same priority.

(4) At any given moment, each missile can only select one repair shop for repair.

(5) missile at a time .

(6) The maintenance operations need to be performed in order from the 1_{st} operation to the J_{th} operation .

(7) The maintenance process allows parallel execution of operations.

(8) It may be necessary to wait between different operations, which will affect the progress of maintenance.

(9) Once missile maintenance begins, it may not be suspended or terminated.

(10) Missiles can choose different maintenance lines for maintenance.

Based on the variable definitions and scheduling assumptions in Table 1, the mathematical model of the missile life extension maintenance multi-objective scheduling model is as follows:

$$\text{Objective function: } \begin{cases} \min f_1 = \sum_{i\in I}\sum_{j\in J} RT_{i,j} \\ \min f_2 = \sum_{i\in CI}\sum_{j\in J} C_{n,j} \end{cases} \tag{1}$$

Among them, the constraints are:

$$FT_{i,j+1} - RT_{i,j+1} \geq ST_{i,j}, \quad i=1,2,\ldots,I; \quad j=1,4 \tag{2}$$

$$FT_{i,3} - RT_{i,3} \geq ST_{i,1}, \quad i=1,2,\ldots,I \tag{3}$$

$$FT_{i,j+1} - RT_{i,j+1} \geq FT_{i,j}, i=1,2,\ldots,I; \quad j=3,6,7,9,10,11,\ldots,J \tag{4}$$

$$ST_{i,j} + Y_{i,j,k} \times RT_{i,j,k} \leq FT_{i,j}, i=1,2,\ldots,I; \quad j=1,2,\ldots,J; \quad k=1,2,\ldots,K \tag{5}$$

$$FT_{e,f} - Y_{e,f,k} \times RT_{e,f,k} + (1-F_{i,j,e,f,k}) \geq FT_{i,j}, \quad i,e=1,2,\ldots,I; \quad j,f=1,2,\ldots,J; k=1,2,\ldots,K \tag{6}$$

$$\sum_{k=0}^{K} Y_{i,j,k} = 1,\ \ i = 1,2,\ldots,I;\ \ \ j = 1,2,\ldots,J \tag{7}$$

$$ST_{i,j} \geq 0;\ \ FT_{i,j} \geq 0;\ \ T_{i,j} \geq 0;\ \ RT_{i,j,k} \geq 0 \tag{8}$$

$$R_{3*(j-1)+n} = SR_{n,i,j} \tag{9}$$

$$CT_i = \min_{1 \leq i \leq I \text{ and } i \notin CI_{i-1}} \{FT_{i,J}\}, CI_o = \varnothing;\ \ \ i = 1,2,\ldots,I \tag{10}$$

Formula (1) is the standard for minimizing the objective function of missile life extension maintenance scheduling, where f_1 and f_2 represent the two objectives of missile life extension maintenance scheduling, namely, completing the maintenance support work of I missiles in the shortest possible time and ensuring that the maintenance work of F missiles is completed first . Formula (2) to Formula (5) represent the order of the maintenance operation phases of this type of missile. Formula (6) indicates that only one missile can be processed in the same maintenance operation room at the same time. Formula (7) indicates that a missile can only choose one maintenance operation room during maintenance. Formula (8) ensures that the maintenance time is non-negative. Formula (9) represents the corresponding relationship between the maintenance operation room number and the optional operation room number in each maintenance operation phase. Formula (10) represents the order of missile completion and the completion time of each missile.

From formula (1), it can be seen that missile life extension maintenance scheduling is a multi-objective mathematical model. The objectives include completing the maintenance and support work of I missiles in the shortest possible time , while giving priority to completing the maintenance work of F missiles. In the field of flexible operation scheduling problems , multi-objective scheduling has been proven to be an NP-hard problem [16]. Missile life extension maintenance scheduling is a special problem of flexible operation scheduling, which is also an NP-hard problem.

IV. MULTI-OBJECTIVE DIFFERENTIAL EVOLUTION ALGORITHM FOR REINFORCEMENT LEARNING (RL-MODE)

It has been proven that reinforcement learning can be effectively applied to parameter learning of genetic algorithms. Differential evolution algorithms and genetic algorithms are both swarm intelligence algorithms. Compared with genetic algorithms, differential evolution algorithms have more advantages in global search of large-scale data[18]and can be more effectively applied to complex job scheduling problems. The differential strategy, scaling operation, and crossover operation of differential evolution algorithms will significantly affect the search performance. The differential strategy of formula (15) can improve the space for global optimization in a complex global search space with multiple local optimal solutions, and can improve the diversity of scheduling scheme search in the early stage of search. The differential strategy of formula (16) can converge to the global optimal solution faster when the global optimal solution is known or close, and can improve the timeliness of scheduling when scheduling large-scale jobs. The scaling operation is used to control the amplitude of the mutation step. Selecting an appropriate scaling factor helps the algorithm improve its performance on different problems. The crossover probability determines the degree to which the differential strategy is adopted when generating new individuals. An appropriate crossover probability can balance local search and global search. In this chapter , this paper designs an adaptive multi-objective differential evolution (RL-MODE) algorithm to intelligently adjust the differential strategy and scaling and crossover operation parameters. Fig.1 is the architecture diagram of the RL-MODE algorithm, including the multi-objective differential evolution (MODE) algorithm process and the reinforcement learning (RL) process. In each iteration, MODE provides state feedback to RL, and RL feeds back the scaling factor (F) and crossover probability (CR) to MODE through Q-learning calculation.

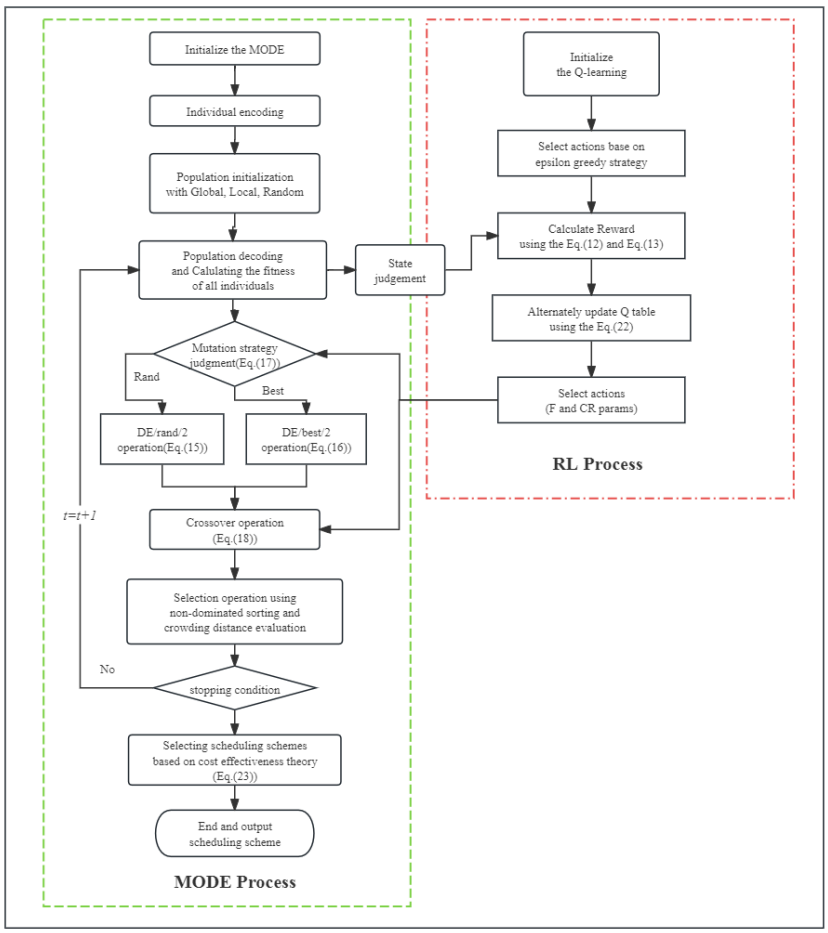

Fig. 1RL-MODE algorithm architecture diagram

A. Model Fusion

It has been proven that reinforcement learning can be effectively applied to parameter learning of genetic algorithms[19]. Differential evolution algorithms and genetic algorithms are both swarm intelligence algorithms. Compared with genetic algorithms, differential evolution algorithms have more advantages in global search of large-scale data[20]and can be more effectively applied to complex job scheduling problems. The differential strategy, scaling operation, and crossover operation of differential evolution algorithms will significantly affect the search performance. The differential strategy of formula (15) can improve the space for global optimization in a complex global search space with multiple local optimal solutions, and can improve the diversity of scheduling scheme search in the early stage of search. The differential strategy of formula (16) can converge to the global optimal solution faster when the global optimal solution is known or close, and can improve the timeliness of scheduling when scheduling large-scale jobs. The scaling operation is used to control the amplitude of the mutation step. Selecting an appropriate scaling factor helps the algorithm improve its performance on different problems. The crossover probability determines the degree to which the differential strategy is adopted when generating new individuals. An appropriate crossover probability can balance local search and global search. In this chapter , this paper designs an adaptive multi-objective differential evolution (RL-MODE) algorithm to intelligently adjust the differential strategy and scaling and crossover operation parameters. Fig.1 is the architecture diagram of the RL-MODE algorithm, including the multi-objective differential evolution (MODE) algorithm process and the reinforcement learning (RL) process. In each iteration, MODE provides state feedback to RL, and RL feeds back the scaling factor (F) and crossover probability (CR) to MODE through Q-learning calculation.

B. Model Setup

1) *Environmental status:*

Differential evolution algorithms and genetic algorithms are both swarm intelligence algorithms. The goal of missile life extension maintenance scheduling is shown in formula (1) , which is to find a set of equilibrium solutions that

minimize the time required to complete the maintenance of I missiles in the shortest possible time and prioritize the completion of the maintenance of F missiles first . This solution represents the Pareto frontier with the best value in the multi-objective differential evolution algorithm, that is, the optimal fitness function solution set of the algorithm. In the framework of the multi-objective differential evolution algorithm, the environmental state is defined as the fitness value of the population. This setting enables the algorithm to evaluate and optimize the fitness values of two key goals: one is to shorten the total time required to complete the maintenance of all missiles as much as possible, and the other is to ensure the priority maintenance of key missiles. The fitness values of these two goals are calculated through the fitness function (see formula (11) for details), which comprehensively considers time efficiency and task priority to guide the algorithm to evolve to the optimal solution.

$$f(x^t) = w_1 \times \frac{\max(f(x_i^t))}{\max(f(x_i^1))} + w_2 \times \frac{\sum_{i=1}^{N} | f(x_i^t) - \frac{1}{N}\sum_{i=1}^{N} f(x_i^t) |}{\sum_{j=1}^{N} | f(x_j^1) - \frac{1}{N}\sum_{j=1}^{N} f(x_j^1) |} \tag{11}$$

Among them, $f(x^t)$ represents the fitness of any individual in the $f(x^t)$ t^{th} generation, and the function is f_1a unified format of $f(x_i^1)$ and f_2 represents the fitness of the $f(x_j^1)$ i^{th} individual in the first generation, represents the j^{th} individual in the first generation, and the maximum $f(x_i^t)$refers to the fitness value of the $f(x^t)$ best individual in the t^{th} generation . It can reflect the quality state of the best individual and the overall quality state of the population in each iteration, where w_1 represents the weight factor of the state of the best individual, and w_2 reflects the state of the entire population. For the convenience of calculation, this paper normalizes by setting w_1 $+w_2$ =1. In multiple experiments, this paper obtains the best parameter combination of w_1 and w_2 values : when calculating the fitness of completing the $f(x^t)$maintenance and support work of $f_1 I$ missiles in the shortest possible time , the parameter values of w_1 and w_2 are set to 0.6 and 0.4 respectively . When calculating the fitness of giving priority to completing the maintenance work of *F* missiles first f_2, the parameter values of w_1 and w_2 are set to 0.7 and 0.3 respectively .

This paper divides the fitness values of the two objectives into 10 intervals respectively, and each interval represents the environment state of reinforcement learning. The first environment state $S_1 = [st_1, st_2, \ldots, st_{10}]$focuses on completing the maintenance and support work of *I* missiles in the shortest possible time. The second environment state $S_2 = [sc_1, sc_2, \ldots, sc_{10}]$focuses on ensuring that the maintenance work of *F* missiles is completed first. The environment state is mapped to the multi-objective differential evolution algorithm to realize the operation of the reinforcement algorithm to adaptively adjust the parameters of the multi-objective differential evolution algorithm.

2) *Action Set Status:*

The action set of reinforcement learning includes the scaling factor and crossover factor of the differential evolution algorithm. In each iteration, the agent will adopt different action strategies to select the action set of scaling factor and crossover probability. In the differential evolution algorithm, the scaling factor is usually set to 0.5 to 3, and the crossover probability is set to 0.3 to 0.9 [18]. In this paper, the scaling factor is divided into multiple regions with an interval value of 0.1, and the crossover probability is divided into multiple regions with an interval value of 0.05. During iteration, the corresponding value is obtained by random value in each region.

3) *Reward Function:*

The reward strategy of reinforcement learning is to provide rewards or punishments based on the individual's behavior to encourage the agent to learn the correct behavior. This paper designs a reward strategy based on the fitness value of the optimal individual and the average fitness value of the population . The definition of the reward strategy of the scaling factor and the crossover probability is as follows :

$$r_F = \frac{\max f(x_i^t) - \max f(x_i^{t-1})}{\max f(x_i^{t-1})} \tag{11}$$

$$r_m = \frac{\sum_{i=1}^{N} f(x_i^t) - \sum_{i=1}^{N} f(x_i^{t-1})}{\sum_{i=1}^{N} f(x_i^{t-1})} \tag{12}$$

Where $f(x_i^t)$is the fitness of the $f(x_i^{t-1})$$i^{th}$ individual in the t^{th} generation, and is the fitness of the i^{th} individual in the *t*-1generation. When the value of the best individual in the t^{th} generation is better than that in the *t*-1 generation,

it means that the individuals in the t^{th} generation have produced some evolutionary effects, which is helpful for solving the scheduling problem. Therefore, this paper gives a positive reward for the crossover probability. When the average fitness value of the population in the t-th generation is better than that in the *t*-1 generation, it means that the overall effect of the population in the t^{th} generation is better. Therefore, this paper dynamically adjusts the overall population to iterate in a better direction by giving a positive reward to the scaling factor.

4) *Action selection strategy:*

The Q-learning algorithm often uses the epsilon-greedy strategy in action selection. This strategy effectively balances exploration and utilization to adapt to complex decision-making environments. The specific implementation is as follows:

In the exploitation process, the action with the highest estimated *Q* value is selected with a probability of 1-ε. This approach ensures that the algorithm selects the best-performing action in most cases, with the goal of maximizing immediate rewards. In the exploration phase, the algorithm randomly selects any action with a probability of ε , ignoring its *Q* value, in order to explore unknown actions or states and discover possible potential advantages.

In practical applications, epsilon (ε) is generally set to a decimal between 0 and 1, and is set to 0.1 in this paper. The key to setting this parameter is the trade-off between exploration and utilization: a smaller ε value tends to strengthen utilization, which is conducive to rapid convergence to a better solution, while a larger ε value increases the breadth of exploration, helps avoid local optimality, and enhances the algorithm's global search ability. By adjusting the ε value, the Q-learning algorithm not only improves its adaptability to complex environments, but also ensures a dynamic balance between continuous learning and environmental adaptation, thereby achieving better performance in a changing environment.

C. Multi-objective differential evolution algorithm (MODE)

Objective Differential Evolution Algorithm Differential Evolution (MODE) is a global optimization metaheuristic algorithm based on population search, which is usually used to solve multi-objective optimization problems without explicit constraints [20]. The multi-objective differential evolution algorithm is particularly suitable for solving complex search problems such as scheduling due to its high efficiency in high-dimensional search space. Compared with other heuristic algorithms such as genetic algorithms, the mutation operation of the differential evolution algorithm helps the algorithm to escape from the local optimal solution and converge to the global optimal solution more quickly. This feature is widely used to solve multi-objective complex problems [21]. In the multi-objective differential evolution algorithm process, each potential scheduling solution is represented by an individual, and these individuals constitute the algorithm's population. The algorithm continuously generates new solutions through crossover and mutation operations, which introduce diversity in the solution space and explore new possibilities. Subsequently, through selection operations, the algorithm evaluates and retains the solutions with the best performance to ensure that the solution effectively approaches the Pareto optimal solution .

1) *Individual performance:*

In the multi-objective differential evolution algorithm, each target individual can be defined as $X = [x_1, x_2, \ldots, x_L, x_{L+1}, x_{L+2}, \ldots, x_{2L}]$a vector, and the range of each variable value in X is defined as $[-\delta, \delta]$. In the differential evolution algorithm, each individual represents a solution for a scheduling. This paper assumes that the total number of operations to be maintained in the scheduling environment is L , then the individual can be defined as $X = [x_1, x_2, \ldots, x_L, x_{L+1}, x_{L+2}, \ldots, x_{2L}]$ a vector. The vector of individual $[x_1, x_2, \ldots, x_L]$ X represents the encoding of the missile maintenance operation , and the vector $[x_{L+1}, x_{L+2}, \ldots, x_{2L}]$represents the operation-to-operation encoding of the missile maintenance (see Fig.1). The correspondence between (x_1, x_{L+1})operations and operation-to-operation is , (x_2, x_{L+2}),. .. , (x_L, x_{2L}), which respectively represent that the operation is x_1 processed by the operation-to-operation , $x_{L+1}$the operation is processed by the operation x_2-to -operation x_{L+2},..., and the operation is x_L processed by the operation-to-operation .x_{2L}

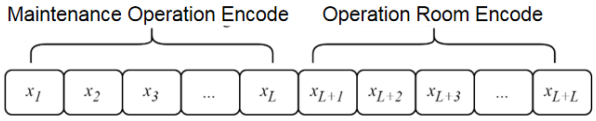

Fig. 2Individual coding method of multi-objective differential evolution algorithm

Since the individual encoding of the traditional multi-objective differential evolution algorithm is real number encoding, and the differential operator will produce decimals after calculation, while the operations and operation intervals in scheduling are represented by integers, the maximum position decoding rule effectively solves the

mapping problem between the individual real number encoding of the traditional differential algorithm and the integer representation in scheduling in solving the scheduling problem[22]. The maximum position decoding rule is as follows:

$$r_j = \text{round}\left(\frac{1}{2\delta}(l_j - 1)(x_j + \delta) + 1\right) \tag{13}$$

Formula (14) implements the mapping $x_j$of real values of variables in individuals $x_j \in [-\delta, \delta]$to integers $r_j \in [1, l_j]$, where $l_j$represents the serviceable operation interval of the operation l, δrepresents $x_j$the range of the variable (defined in this paper $\delta = 1$), $\text{round}(x)$and the function represents mapping the corresponding real parameter to the integer closest to it.

2) *Population initialization:*

In the multi-objective differential evolution algorithm, the traditional population initialization usually adopts a random method. Although this method is simple, it may lead to slow convergence of fitness values and poor solution quality. In order to improve the efficiency and quality of population initialization, this paper proposes three strategies: global initialization, local initialization and random initialization [22]. These strategies not only help to distribute the initial population more evenly, but also effectively balance the workload of the repair equipment and improve the utilization rate of the equipment.

The global initialization strategy is implemented through the following steps: First, initialize an array with the same length as the number of maintenance operation rooms, where the value of each element is initially set to zero, representing the time taken for the corresponding operation room. Then, randomly select a missile from the missile set, and starting from its first operation, gradually add the time taken for each operation of the selected missile to the corresponding position of the array. After each operation, select the operation room with the shortest time taken in the array for maintenance until all operations of the missile are assigned. Repeat this process until the maintenance operations of all missiles are assigned.

The local initialization strategy focuses on processing the maintenance operations of each missile one by one. Similar to the global initialization, this strategy also sets up an array of operation time consumption, but the initialization process is more orderly. Each missile is selected and assigned its maintenance operation in the order of the missile set. After the operation assignment of each missile is completed, the array is reset and continues to process the next missile.

The random initialization strategy retains randomness to increase the probability of exploring the solution space . In this strategy, after the time-consuming array between operations is initialized, the selection between each operation is completely random, which helps to cover a wide range of solution spaces, especially in exploring unknown areas in the initial stage.

3) *Differential Operation:*

Mutation operation is the most important step in the multi-objective differential evolution algorithm. It helps the algorithm to continuously improve individuals during the search process and find the global optimal solution [24]. In the differential evolution algorithm, the basic idea of the mutation operation is to generate new individuals by linearly combining existing individuals. These new individuals will be evaluated and selected in the subsequent evolution process. The mutation operation realizes individual mutation through differential strategy. The differential strategy can improve the performance of operation selection in the missile life extension maintenance scheduling process . Therefore, this paper proposes a variety of differential strategies for missile life extension maintenance scheduling. The differential strategy is expressed as follows:

a) DE/rand/1 algorithm

Five individuals $X_1, X_2, X_3, X_4, X_5$are randomly selected from the population . The $X_1 \sim X_5$represent five scheduling schemes respectively . The five scheduling schemes optimize the scheduling direction through formula (15) :

$$V_i = X_1 + F \times (X_2 - X_3) + F \times (X_4 - X_5) \tag{14}$$

b) DE/best/1 algorithm

The individual is defined X_{best} as the best individual in the population (i.e., the best scheduling solution), and is $X_1 \sim X_4$ randomly selected individuals in the population (i.e., representing any 4 scheduling solutions). The best individual and the random individual can optimize the scheduling direction through formula (16) :

$$V_i = X_{best} + F \times (X_1 - X_2) + F \times (X_3 - X_4) \tag{15}$$

In the above differential strategies (1) and (2), V_i is the new offspring generated by the mutation operation of the " $\times$ "i -th individual , $X_i$is the product operation, and F is the scaling factor, which is used to adjust the amplitude of the difference vector to affect the diversity and search ability of the new individuals generated by the mutation operation. The selection of the scaling factor is crucial to the performance of the differential evolution algorithm, which affects the convergence speed of the algorithm, the exploration ability of the search space, and the adaptability to different problems. Therefore, this paper uses the reinforcement algorithm to adaptively update the value of the scaling factor in the subsequent section.

Differential strategies (1) and (2) is that strategy (1) uses a completely random individual approach when selecting parent individuals, while strategy (2) uses a global optimal individual plus a random individual approach. Strategy (1) can improve the space for global optimization in a complex global search space with multiple local optimal solutions. Compared with the DE/rand/ 1 algorithm , it can improve the diversity of scheduling scheme searches in the early search . When the global optimal solution is known or close, strategy (2) can converge to the global optimal solution faster. Compared with the DE/best/ 1 algorithm , it can improve the timeliness of scheduling when scheduling large-scale jobs [24].

Therefore, this paper proposes an adaptive mutation operation to optimize the mutation operation of the scheduling job. The reinforcement algorithm dynamically adjusts the value of the scaling factor according to the job characteristics. At the same time, the mutation strategy is adaptively adjusted according to the population size and reinforcement learning state. The mutation strategy is adaptively adjusted through formula (17) :

$$differential\ strategy = \begin{cases} V_i = X_1 + F \times (X_2 - X_3), & N_i \le \min\left(\frac{N_s \times N_a}{2}, \sqrt[3]{max\_iter}\right) \\ V_i = X_{\text{best}} + F \times (X_2 - X_3), & N_i > \min\left(\frac{N_s \times N_a}{2}, \sqrt[3]{max\_iter}\right) \end{cases} \tag{16}$$

Wherein $N_i$represents the current number of iterations, $N_s$represents the total number of states, $\mathrm{N_a}$represents the total number of operations, *max_iter* represents the maximum number of iterations of the MODE algorithm, and $\min(x, y)$represents the minimum value of the variable x, y.

4) *Crossover Operation:*

Multi-objective differential evolution algorithm is a part used to generate new individuals. Together with the mutation operation, it constitutes the main step of the differential evolution algorithm. The purpose of the crossover operation is to combine the candidate solutions generated by the mutation operation with the original solution to generate the next generation of individuals . The scheduling scheme generated by the mutation operation combined with the original scheduling scheme can not only improve the diversity of the new scheduling scheme and avoid falling into the local optimal solution, but also retain some information of the original scheduling scheme in the new scheduling scheme, thereby facilitating local search. The crossover operation is completed by formula (18) :

$$U_{i,j} = \begin{cases} V_{i,j}, & \text{if rand}(0,1) \le CR \text{ or } j = j_{\text{rand}} \\ X_{i,j}, & \text{otherwise} \end{cases} \tag{17}$$

Where $U_{i,j}$ is the new offspring generated by the $V_{i,j}$crossover operation of the jth element of the i -th individual . is X_i the new individual generated by the mutation operation of formula (17). *CR* is the crossover probability. j_{rand} is a random value that ensures that at least one dimension of the experimental individual after crossover is provided by the mutant individual.

5) *Objective function*

multi-objective differential evolution algorithm is the mathematical model in the above Chapter 2 , that is, completing the maintenance of I missile in the shortest possible time and giving priority to completing the maintenance of F missiles . Completing the maintenance of I missile in the shortest possible time and giving priority to completing the maintenance of F missiles are the equilibrium solution sets calculated on the Pareto frontier by the differential evolution algorithm. The solutions on the Pareto frontier can optimize the scheduling of missile life extension repair resources . In the section 6) Select Operation section, this paper completes the evaluation calculation of the Pareto frontier solution through non-dominated sorting and congestion distance.

6) *Select Operation*

the maintenance of I missiles in the shortest possible time and give priority to the maintenance of F missiles. To effectively solve this goal, this paper adopts two strategies: non-dominated sorting and crowding distance. These strategies are widely used in the field of multi-objective optimization for solution selection and diversity maintenance[25]. Non-dominated sorting divides all individuals in the population into layers according to

dominance relationships. Individuals in each layer are not dominated by each other, thus forming multiple dominance fronts. This process ensures that the hierarchy and diversity of the space are understood, allowing the algorithm to explore multiple potential effective solutions at the same time. After completing the non-dominated sorting, this study further applies crowding distance calculation to evaluate the distribution density between individuals in the same front. Individuals with larger crowding distances indicate that their surroundings are sparse and more likely to be selected for generating offspring, which helps maintain the genetic diversity of the population. In the selection process, this paper gives priority to those individuals with high non-dominated levels and large crowding distances. This strategy not only improves the algorithm's ability to explore the solution space , but also ensures the breadth and depth of understanding. The specific selection steps include: first, all individuals are sorted by non-dominated order; second, within each level, they are sorted by crowding distance, and individuals with large distances are retained first; then, individuals are selected from each level through the roulette method to fill the next generation of populations; finally, the selected individuals will participate in subsequent crossover and mutation operations to form a new generation of solution sets.

D. Reinforcement Learning (RL)

Reinforcement Learning (RL) can help intelligent systems learn how to make decisions in different environments . Reinforcement learning systems can gradually improve their decision-making strategies based on environmental feedback to maximize the expected cumulative reward. Chen[12]used reinforcement learning to provide feedback on the genetic algorithm parameter environment and optimized the parameter set of the genetic algorithm during job scheduling. Su[4] automatically selected strategies to improve the candidate schedules of the self-organizing neural scheduler through reinforcement learning . The framework of the reinforcement learning model is shown in Fig.3 , which includes five basic components: agent , environment , state , action , and reward . At any given point in time , t , the state of the current environment observed by the agent S_t, and selects the corresponding action according to the established strategy A_t. Then, at time A_{t+1} , the agent performs this action, which is sent to the environment. The environment then reacts to this action , causing the agent to move from the current state S_tTransfer to the new state $S_{t+1}$and generate an immediate reward R_{t+1} . The agent updates its strategy accordingly and performs the next action A_{t+1} .

In this interactive learning process, the main goal of the agent is to learn a strategy to maximize the long-term cumulative reward. Specifically, the learned strategy is defined as a probability distribution mapping between states and actions, that is, the probability of taking each action in a specific state. In this way, the agent is able to continuously optimize its decision-making process to cope with the dynamic changes of the environment and achieve its goals. The learned strategy can be defined as: a mapping from the $s\epsilon S$ action probability distribution of each state $\pi(a|s)$and action $a\epsilon A$, that is, the probability of taking each action in the environment .S_t Probability of $A_t$an action :

$$\pi(a \mid s) = p(A_t = a \mid S_t = s),\ \exists t \tag{18}$$

the strategy π is a Markov process optimization problem. Given the initial state distribution $\rho_0$and strategy π, the probability of a T-step trajectory τoccurring in the Markov process is:

$$p(\tau \mid \pi) = p_0(S_0)\prod_{t=0}^{T-1} p(S_{t+1} \mid S_t, A_t)\pi(A_t \mid S_t) \tag{19}$$

Given a reward function R and all possible trajectories τ, the policy solution (π^*) can be expressed as:

$$\pi^* = \arg\max_{\pi}\left(\int_{\tau} p(\tau \mid \pi)R(\tau)\right) \tag{20}$$

Among them, $R(\tau)$represents the sum of rewards after time T, represents all possible trajectories, $argmax_{\pi}(.)$and represents the strategy of returning the maximum value of the expected reward function. Finally, the optimization problem of reinforcement learning can maximize the reward by optimizing the strategy.

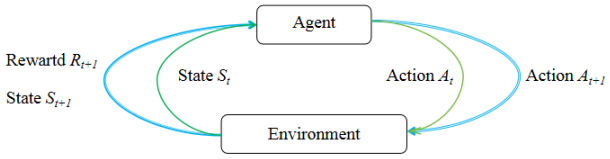

Fig. 3Strengthen the model framework

Q-learning is a model-free reinforcement learning algorithm that focuses on learning the optimal strategy directly from environmental interactions without modeling the dynamics of the environment. The update rule of Q-learning is shown in formula (22) :

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \arg\max_{A_{t+1}} Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) - Q(S_t, A_t)] \tag{21}$$

Where $S_t$represents the state at time t , represents $A_t$the action taken in $R_{t+1}$the current state , $S_t$and represents $S_t$the immediate reward obtained after S_{t+1}performing the action in the state. $A_t$is the result state, which is $A_{t+1}$the next best action that may be taken in αthe state . The learning rate $S_{t+1}$is usually set to a value between 0.1 and 1, and this paper takes a value of 0.5. R_{t+1}represents $S_t$the reward obtained after γperforming the action from the state. $A_t$is a discount factor, which is used to weigh the immediate reward and future reward. It is generally set to a positive number less than 1, and this paper takes a value of 0.5. represents the action $\underset{A_{t+1}}{\operatorname{argmax}} Q(S_{t+1}, A_{t+1})$ with the largest Q value $A_t$in the next state S_{t+1}.

The core goal of the learning algorithm is to approach the optimal strategy by continuously updating the Q value. After each state transition, the algorithm adjusts the value of the current action based on the received reward and the estimated maximum future reward, thereby maximizing the cumulative reward in long-term exploration.

In this paper, the Q-learning algorithm is applied to the learning of multi-objective differential evolution algorithm parameters to formulate strategies in the scheduling environment. Through this interactive learning process between the agent and the environment, Q-learning adjusts parameters and strategies to achieve the goal of completing the maintenance and support work of *I* missiles in the shortest possible time and giving priority to the maintenance work of *F* missiles, and maximizing the overall return.

E. Scheduling cost effectiveness evaluation

In missile life extension maintenance scheduling, the goal is to complete the maintenance and support work of missile *I* in the shortest possible time, while giving priority to ensuring that the maintenance work of missile *F* is completed first. These two goals reflect different optimization dimensions: one is overall efficiency, which reduces overall downtime and maintenance costs through rapid response and efficient execution; the other is strategic priority, which emphasizes handling key tasks in order of priority to maintain combat readiness effectiveness costs and strategic flexibility. Integrating these two goals, the scheduling strategy not only improves the economy and efficiency of maintenance operations, but also ensures the rapid recovery of key combat resources.

This paper proposes a method for evaluating the cost-effectiveness of scheduling . This method generates multiple groups of non-dominated solutions on the Pareto frontier through a multi-objective differential evolution algorithm. Each group of solutions represents a potential optimal scheduling solution, which shows different performances of the overall cost and the efficiency cost . The key lies in how to balance the weights of the overall cost and the efficiency cost . The difference in the calculation units between the two causes the complexity of the evaluation. To solve this problem, this study uses Z-score standardization to standardize the overall cost and the efficiency cost to a unified scale for fair comparison and evaluation. The specific calculation formula is as follows:

$$V_i = \alpha \times \frac{t_i - \mu_t}{\sigma_t} + (1-\alpha) \times \frac{c_i - \mu_G}{\sigma_G}, \quad i = 1, 2, \ldots, n \tag{22}$$

Where $V_i$is the scheduling benefit after the overall cost and the efficiency cost are balanced and unified , that is, the overall scheduling benefit of the i^{th} scheduling plan. The *i* is i^{th} Pareto frontier solution set in the scheduling solution set. The *n* is the number of Pareto frontier solution sets, is the t_i overall cost solution in the scheduling solution set (complete the maintenance and support work of *I* missiles in the shortest possible time) , is the c_i priority efficiency solution in the scheduling solution set (prioritize the maintenance work of *F* missiles to be completed first) , μ_t and μ_c represent the means of all t_i solution sets and solution sets respectively, σ_t and σ_c represent the standard deviations of all t_i solution sets and solution sets respectively, $t_i$and c_i represent the weight adjustment factors of the overall cost solution and the priority efficiency solution . In the multi-objective evolutionary algorithm, The V_i is used to measure the cost-effectiveness of multiple groups of non-dominated solutions in the Pareto frontier , where the lower the cost-effectiveness , the lower the scheduling cost and the higher the efficiency , which indicates that the scheduling plan is better . Through this method, the cost required in the scheduling plan can be quantified, providing a quantitative basis for decision-making in missile life extension maintenance scheduling.

V. EXPERIMENTAL RESULTS AND DISCUSSIO

A. Benchmark Setup and Evaluation

In order to evaluate the performance of the proposed RL-MODE algorithm, this paper uses a public missile maintenance scheduling benchmark dataset [1] for experimental comparison. The task of this dataset is to complete the maintenance scheduling of 36 missiles in the shortest possible time, ensuring that the maintenance work of 12 missiles is given priority. The maintenance process is divided into 16 stages, each with 3 optional operation rooms. Based on the assumptions and objective function constraints defined in Chapter 2 , the dataset is solved.

In terms of effect evaluation, this paper uses the following key indicators to comprehensively evaluate the performance of the algorithm:

(1) Hypervolume indicator [25]: It is used to measure the distribution of the multi-objective Pareto solution space . This indicator evaluates the quality of the solution by calculating the hypervolume space occupied by the Pareto frontier. A larger Hypervolume value indicates that the Pareto frontier solution is more widely distributed, thus providing more strategic options for missile maintenance scheduling.

(2) Missile maintenance scheduling efficiency: This mainly considers the overall maintenance completion time of the 36 missiles and the maintenance work time of the 12 missiles that are completed first. The optimization goal is to minimize the corresponding time at the same time to improve scheduling efficiency and response speed.

(3) Algorithm convergence iteration number: used to evaluate algorithm performance and efficiency, and reflects the convergence speed of the algorithm by recording the number of iterations required for the algorithm to reach an approximate optimal solution. Fewer iterations usually indicate that the algorithm has better optimization speed and efficiency.

In addition, to further verify the effectiveness of the RL-MODE algorithm, this paper **Error! Reference source not found.**algorithm and SLISA [28][27]algorithm. The performance comparison is mainly based on the above three evaluation indicators, which are described in detail in this Section.

B. Parameter settings

In order to comprehensively evaluate the performance of the RL-MODE algorithm, the algorithm code of this paper is written in the Python 3.11 environment, and the computer configuration is Intel Core i9-13900H CPU (2.6 GHz) and 32 GB of RAM. In order to ensure the reliability of the experimental results, this paper selects four algorithms, RL-MODE, AGA, HPEA and SLISA, and repeats them 10 times on the selected benchmark dataset. The evaluation indicators include Hypervolume, the overall maintenance completion time of 36 missiles, the maintenance work time of the 12 missiles that are completed first, and the number of algorithm convergence iterations. The average values of these indicators are obtained from 10 runs for analysis. In the experimental setting of the multi-objective differential evolution algorithm, the population size is 100 , the maximum number of iterations is 300 , the initial mutation probability is set to 0.3 , and the crossover probability is 0.5 .

C. Algorithm effect verification

First, this paper conducts experiments on benchmark data sets and verifies the effectiveness of the algorithm using visualization methods. The experiment demonstrates the convergence of the RL-MODE algorithm in dealing with multi-objective optimization problems, which is specifically reflected in the iteration trend of the overall maintenance completion time of 36 missiles and the maintenance work time of the 12 missiles that are completed first (as shown in Fig.4). In addition, this paper also shows the Gantt chart of the scheduling scheme obtained after 300 iterations, thus proving the effectiveness of this algorithm in solving the multi-objective problem of missile life extension maintenance (as shown in Fig.5 and Fig.6).

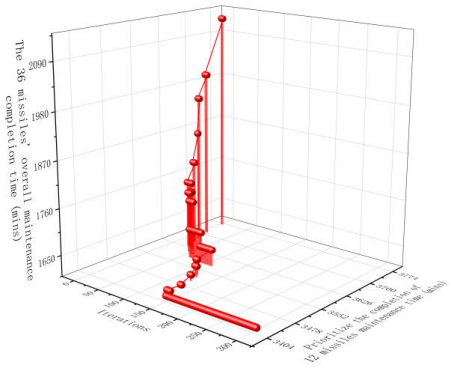

Fig. 4The iterative convergence trend of RL-MODE algorithm on benchmark datasets

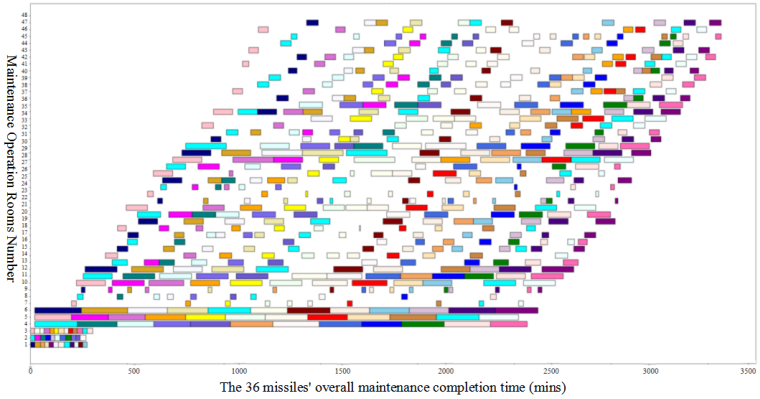

Fig. 5The RL-MODE algorithm completes the Gantt chart of the overall maintenance of 36 missiles.

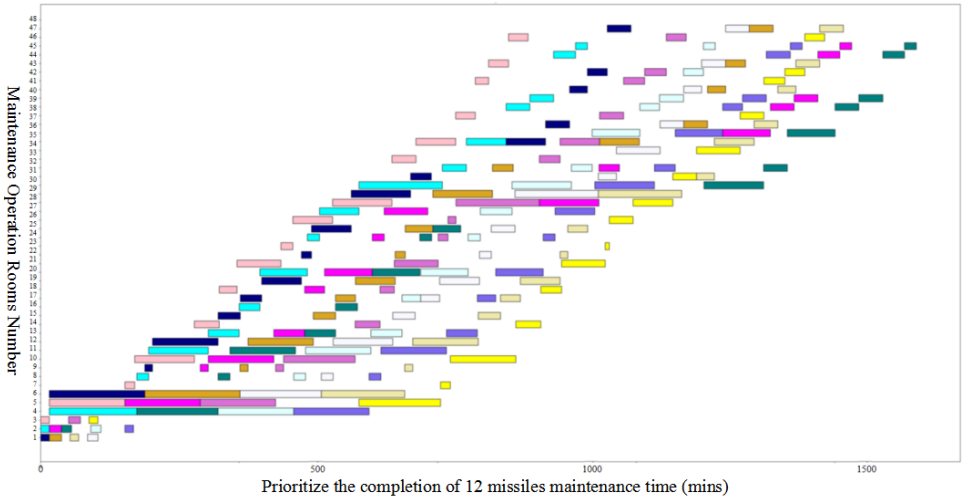

Fig. 6RL-MODE algorithm in the priority to complete the maintenance of 12 missiles Gantt chart

Fig.4 shows the convergence trend of the fitness value of the missile life extension maintenance scheduling target in the RL-MODE algorithm, where the X-axis represents the number of iterations, the Y-axis represents the overall maintenance completion time of 36 missiles, and the Z-axis represents the maintenance work time of the 12 missiles that are completed first. From the trend of change, it can be seen that the fitness values of the two targets are rapidly decreasing. From the X-axis, it can be seen that the algorithm basically converges at the 141st iteration, proving that the algorithm has rapid convergence.

Fig.5 and Fig.6 are Gantt charts of the overall maintenance of 36 missiles and the maintenance of 12 missiles that are completed as a priority . The ordinate represents the maintenance operation room number, and the abscissa represents the maintenance time. The blocks of different colors in the Gantt chart represent missiles with different numbers. The missile maintenance work in the Gantt chart is completed relatively concentratedly, without inefficiencies such as overlap and redundancy, which shows that the algorithm can reasonably allocate maintenance

resources. As shown in Fig.6, the algorithm effectively handles the multi-objective solution that includes priority tasks and overall maintenance work.

D. Algorithm effect comparison

In this section, we compare the experimental results of multiple algorithms in the literature to evaluate the performance of the RL-MODE algorithm on the missile maintenance scheduling problem. The algorithms in the literature include the improved genetic algorithm (AGA) **Error! Reference source not found.**, the multi-objective evolutionary algorithm HPEA[27], and SLISA [27]the overall maintenance completion time of 36 missiles (minutes), the maintenance time of 12 missiles with priority (minutes), and the number of convergence iterations.

Table 1The RL-MODE algorithm compares the effects of various algorithms.

Algorithm	Hypervolume	Overall maintenance of 36 missiles (minutes)	Prioritize the repair time of 12 missiles (minutes)	Convergence iterations
AGA	3745	3522	1702	220
HPEA	13168	3470	1654	207
SLISA	5743	3501	1692	178
RL-MODE	19287	3417	1605	141

As shown in Table 2 , compared with the AGA algorithm, the RL-MODE algorithm has increased the overall maintenance completion time of 36 missiles, the maintenance time of 12 missiles in priority, and the number of convergence iterations by 3%, 5.7%, and 36% respectively on average. This is because reinforcement learning can adaptively adjust the crossover and mutation process when optimizing the multi-objective differential evolution algorithm, and has more advantages in searching the solution space than the improved genetic algorithm , and converges faster. At the same time, the non-dominated sorting and crowding distance are used to select multi-objective solutions, which can sort the solution space and improve the quality and diversity of the solution. Compared with the HPEA algorithm, the overall maintenance completion time of 36 missiles, the maintenance time of 12 missiles in priority, and the number of convergence iterations are increased by 1.5%, 3%, and 32% on average, respectively. This is because reinforcement learning can adaptively adjust the crossover and mutation process according to the reward-penalty mechanism when optimizing the multi-objective differential evolution algorithm, and has more advantages in searching the solution space than the mixed integer linear programming , and converges faster. In the comparison of SLISA algorithm, SLISA algorithm solves the multi-objective problem by converting it into a single-objective problem with multiple objectives added together, and cannot balance the optimal value of each objective in the solution space . RL-MODE algorithm uses non-dominated sorting and congestion distance to evaluate the multi-objective solution when solving multi-objective problems, effectively solving the multi-objective optimization balance problem, and improving the overall maintenance completion time of 36 missiles, the maintenance time of 12 missiles with priority, and the number of convergence iterations by an average of 2.4%, 5.1%, and 21%, respectively. In addition, RL-MODE algorithm has more advantages than AGA, HPEA, and SLISA algorithms in the evaluation of the distribution of multi-objective solution space (Hypervolume value), which is improved by 4.2, 0.4, and 2.6 times, respectively. The Hypervolume value evaluation benchmarks for the overall maintenance completion time of 36 missiles and the maintenance time of 12 missiles with priority are set to 3550 and 1750 minutes.

4.5 Ablation experiment

of each operator in the RL-MODE algorithm , this paper designs three ablation algorithms to evaluate the performance of the operators. RL-MODE0 is the RL-MODE algorithm eliminating the mutation strategy operator of formula (15), MODE 0 is the RL-MODE algorithm eliminating reinforcement learning and the mutation strategy operator of formula (15), and MOGA is the RL-MODE algorithm eliminating reinforcement learning and using the GA algorithm.

Table 2Comparison of ablation effect of RL-MODE algorithm

Algorithm	Hypervolume	overall maintenance of 36 missiles (minutes)	Prioritize the repair time of 12 missiles (minutes)	Convergence iterations
RL-MODE	19287	3417	1605	141
RL-MODE0	13627	3446	1619	160

MODE 0	6788	3472	1663	187
MOGA	801	3530	1710	234

As shown in Table 3 , compared with the RL-MODE0 algorithm, the RL-MODE algorithm has increased the number of convergence iterations by 41%, and the overall maintenance completion time of 36 missiles and the maintenance time of 12 missiles with priority have increased by 0.9%. The reason is that both algorithms use reinforcement learning and multi-objective differential evolution optimization algorithms. In the comparison between the RL-MODE algorithm and the MODE algorithm, the overall maintenance completion time of 36 missiles and the maintenance time of 12 missiles with priority have increased by 1.6% and 3.5% respectively, because reinforcement learning optimizes the scaling factor and crossover probability of the multi-objective differential evolution algorithm, which can effectively avoid falling into the local optimum. In the comparison between the RL-MODE algorithm and the MOGA algorithm, the overall maintenance completion time of 36 missiles and the maintenance time of 12 missiles with priority have increased by 3.5% and 6.1%, and the number of convergence iterations has increased by 40%, indicating that the differential evolution algorithm optimized by reinforcement learning has more advantages than the genetic algorithm in the multi-task scheduling problem. In addition, the RL-MODE algorithm has more advantages than the above three algorithms in the evaluation of the spatial distribution of multi-objective solutions (Hypervolume value), which is improved by 0.4, 1.8 and 23 times respectively.

VI. Conclusion

This paper proposes the RL-MODE algorithm and establishes a mathematical scheduling model for missile life extension maintenance. By setting the optimization goals of minimizing the overall maintenance time and giving priority to completing key maintenance tasks, the RL-MODE algorithm successfully solves the multi-objective scheduling problem of missile life extension maintenance.

The algorithm integrates reinforcement learning and multi-objective differential evolution algorithms, and effectively improves scheduling efficiency by adaptively adjusting scaling factors and mutation probabilities . Specifically, the RL-MODE algorithm uses population size and current learning state information to adaptively adjust mutation strategies through a reinforcement learning framework. Subsequently , the Q-learning algorithm is used to achieve effective interaction between states and adaptive adjustment of parameters, further enhancing local and global search capabilities, optimizing the diversity of the early search stage and the response efficiency to large-scale scheduling tasks. Finally, the Pareto frontier target solution set is evaluated based on utility evaluation , and the scheduling solution with the highest utility is selected.

Experimental results on a missile maintenance benchmark dataset show that the RL-MODE algorithm outperforms existing comparison algorithms in multiple evaluation metrics, including Hypervolume, overall maintenance completion time, priority task completion time, and number of convergence iterations. These results highlight the significant advantages of the RL-MODE algorithm in dealing with complex scheduling problems and demonstrate its efficiency and practicality in practical applications. In addition, ablation experiments verify the effectiveness of the proposed optimization operator in improving scheduling performance.

Future work will further explore the potential of the RL-MODE algorithm to solve other types of scheduling problems, expand its scope of application, and continue to verify and optimize the applicability and effectiveness of the algorithm.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 62371465). We would like to thank Dr. Z and Dr. Y for their valuable contributions to this research. We would like to thank Mr. Z for his technical assistance during the experiments. We would also like to thank our families for their constant support and encouragement throughout this research.

References

[1] Huang Z, et al. Maintenance decision-making and management of submarine-launched ballistic missile equipment[M]. Ordnance Industry Press, 2020.

[2] Huang H, Han J, LIU X, et al. Research on Missile Equipment Health Management System for After-

Sales Support[J]. Modern Defense Technology, 2023, 51(1): 75.
[3] Chen K, Zhang F, Du G, et al. Study on the Key Problems of Tactical Missile Storage Life Extension Engineering Practice [J]. Ship Electronic Engineering, 2023, 43(04): 136-139+162.
[4] Wang H , Teng K, Lü W. Review on key technologies for missile storage and life-extension test [J]. Energetic Materials, 2019, 27(12): 1004-1016.
[5] Su J, Huang H, Li G, et al. Self-organizing neural scheduler for the flexible job shop problem with periodic maintenance and mandatory outsourcing constraints[J]. IEEE Transactions on Cybernetics, 2022.
[6] Xie J, Gao L, Peng K, et al. Review on flexible job shop scheduling[J]. IET collaborative intelligent manufacturing, 2019, 1(3): 67-77.
[7] Zhang G, Gao L, Shi Y. An effective genetic algorithm for the flexible job-shop scheduling problem[J]. Expert Systems with Applications, 2011, 38(4): 3563-3573.
[8] Shi S, Xiong H. Multi-objective differential evolution algorithm for no-tardiness job shop scheduling problem with outsourcing option[J]. Computer Integrated Manufacturing Systems, 1.
[9] Sun K, Zheng D, Song H, et al. Hybrid genetic algorithm with variable neighborhood search for flexible job shop scheduling problem in a machining system[J]. Expert Systems with Applications, 2023, 215: 119359.
[10] Liu Z, Song G, Zeng L, et al. Optimization model for multi-types surface-to-air missile batch technology preparation scheduling[J]. Ordnance Automation, 2021, 40(08): 47-51+60 .
[11] Wei X, Zhang Z , Tang D , et al. Energy-saving oriented multi-objective shop floor scheduling for mixed-line production of missile components[J]. Journal of Mechanical Engineering, 2018, 54(09): 45-54.
[12] Chen R, Yang B, Li S, et al. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem[J]. Computers & industrial engineering, 2020, 149: 106778.
[13] Guo R, Wang Y. Aircraft assignment method for optimal utilization of maintenance intervals[J]. Journal of System Simulation, 2023, 35(9): 1985.
[14] Zeng L , Ding L, Guan Z. Flexible job shop scheduling based on deep self-learning tabu search algorithm[J]. Computer Integrated Manufacturing Systems, 1.
[15] Du Y, Li J, Chen X, et al. Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2022.
[16] Wu X, Peng J, Xiao X, et al. An effective approach for the dual-resource flexible job shop scheduling problem considering loading and unloading[J]. Journal of Intelligent Manufacturing, 2021, 32: 707-728.
[17] Tkatek S, Bahti O, Lmzouari Y, et al. Artificial intelligence for improving the optimization of NP-hard problems: a review[J]. International Journal of Advanced Trends Computer Science and Applications, 2020, 9(5).
[18] Arulkumaran K, Cully A, Togelius J. Alphastar : An evolutionary computation perspective[C]//Proceedings of the genetic and evolutionary computation conference companion. 2019: 314-315.
[19] Al- Dabbagh RD, Neri F, Idris N, et al. Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy[J]. Swarm and Evolutionary Computation, 2018, 43: 284-311.
[20] Kachitvichyanukul V. Comparison of three evolutionary algorithms: GA, PSO, and DE[J]. Industrial Engineering and Management Systems, 2012, 11(3): 215-223.
[21] Wang G G , Gao D, Pedrycz W. Solving multiobjective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm[J]. IEEE Transactions on Industrial Informatics, 2022, 18(12): 8519-8528.
[22] Miyata H H , Nagano M S. The blocking flow shop scheduling problem: A comprehensive and conceptual review[J]. Expert Systems with Applications, 2019, 137: 130-156.
[23] Li Y, Huang W, Wu R, et al. An improved artificial bee colony algorithm for solving multi-objective low-carbon flexible job shop scheduling problem[J]. Applied Soft Computing, 2020, 95: 106544.
[24] Jiang D, Li L, Gong J, et al. Optimal trajectory searching based differential evolution[J]. International Journal of Wireless and Mobile Computing, 2015, 8(4): 384-393.
[25] Deng W, Zhang X, Zhou Y, et al. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems[J]. Information Sciences, 2022, 585: 441-453.
[26] Stanley KO, Clune J, Lehman J, et al. Designing neural networks through neuroevolution [J]. Nature Machine Intelligence, 2019, 1(1): 24-35.
[27] Li R, Gong W, Lu C. Self-adaptive multi-objective evolutionary algorithm for flexible job shop scheduling with fuzzy processing time[J]. Computers & Industrial Engineering, 2022, 168: 108099.
[28] Liu X, Liu L, Jiang T. A self-learning interior search algorithm based on reinforcement learning for energy-aware job shop scheduling problem with outsourcing option[J]. Journal of Intelligent & Fuzzy Systems, 2023 (Preprint): 1-16.