IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 October 2023, accepted 8 November 2023, date of publication 13 November 2023,
date of current version 22 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332513

RESEARCH ARTICLE

A Time-Driven Dynamic Weapon Target Assignment Method

CHANG LIU[1,2], JIANG LI[1], YE WANG[1,2], YANG YU[1], LIHONG GUO[1], YUAN GAO[1,2], YANG CHEN[1], AND FENG ZHANG[3]

[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]School of Aviation Operations and Services, Aviation University of Air Force, Changchun 130022, China

Corresponding author: Jiang Li (cclijiang@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61977059.

ABSTRACT The traditional dynamic weapon target assignment model is a combination of multiple static weapon target assignment stages. The assignment of the next stage is carried out after the result of the previous static weapon target assignment is settled. However, between two static weapon target assignment stages, the threat ranking of multiple targets may change with time, and the traditional dynamic weapon target assignment model does not take this time issue into consideration. This paper proposes a ''time sampling dynamic weapon assignment model''. This model divides the decision-making stage by setting the time interval of data collection, and it can capture the real-time changes in the target threat degree and make timely decisions. With this model, this study designed a dynamic weapon target assignment method based on the reinforcement learning algorithm. Additionally, according to this method, a comparative experiment with different sampling time divisions was designed, and a better sampling time division method was obtained. Finally, a comparative experiment between the reinforcement learning algorithm and the traditional heuristic algorithm was designed in this study. The simulation results show that, compared with the traditional heuristic algorithm, the proposed assignment model and the reinforcement learning algorithm are better in terms of decision-making timeliness and global considerations.

INDEX TERMS Dynamic weapon target assignment, simulation model, reinforcement learning, heuristic algorithm, PPO algorithm.

I. INTRODUCTION

Weapon target assignment (WTA) is one of the most important research issues for command and control (C2), which mainly involves the study on how to effectively reduce the damage caused by the attacker and how to realize the effective assignment of defensive resources in the case of limited defensive resources. The WTA problem belongs to the NP-complete problem [1]. The main characteristic of this problem is that with the increase in weapons and targets, the dimension of the solution space increases exponentially, and the solution can only be obtained through an approximate solution algorithm. Therefore, the task of building a reasonable model and adopting a fast and accurate convergence algorithm has become an important research direction for weapon target assignment.

The problem of weapon target assignment is divided into the static weapon target assignment (SWTA, i.e., static WTA) problem and the dynamic weapon target assignment (DWTA, i.e., dynamic WTA) problem. The SWTA problem was first modeled by Manne in 1958 [2]. The SWTA problem defines a scenario in which the defender can observe a known number of missiles (targets) and a finite number of interceptors (weapons), know the probability that the weapon successfully destroys the target (kill probability), and can assign targets and weapons in a single scene. In the SWTA problem, changes in the subsequent battle situation are not considered; thus, the time factor is not included. Due to increases in defenders' demand for timeliness, the research on the SWTA problem can no longer meet the needs of actual

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Oliva.

combat. Therefore, the DWTA problem has gradually become a hot spot in current research. The dynamic weapon target assignment problem takes dynamic factors into account. The number of missiles and interceptors changes dynamically. Therefore, ''weapon-target'' not only needs to be optimized in a single scenario, but it also considers whether the effectiveness of other scenarios can meet the demand. In the early days, Eckler, Burr and others conducted research on the DWTA problem, and they proposed and discussed the solutions for it [3].

According to Murphey [4], DWTA has two model types. The first model assumes that all targets are known from the beginning, but in a different order of attack. The second model assumes some or all of the targets are random, and that not all are known at the beginning. For the first model, Eckler and Burr [3] initially proposed the shoot-look-shoot scheme in the 1970s. This meant that weapons were assigned to the target in the first engagement, and that subsequent engagements allowed the remaining weapons to be assigned to any surviving target. Leboucher et al. [5] considered targets that survived engagement–those that could be re-engaged in subsequent iterations. Xin et al. [6] pointed out that if the defender observes that the target is destroyed in stage t, it must re-evaluate the target in stage t+1, and the weapons pre-assigned to the destroyed target must be reassigned. Zhengrong et al. [7] proposed a multi-stage attack planning method, established a multi-stage attack target function model, and applied the multi-stage attack method to gradually weaken the target defense, thus ultimately achieving better combat effectiveness.

For the second model, Burr et al. [8] assume that the total number and distribution of attackers are not known, and that the defender must first set a strategy. Murphey [9] proposed a specific model of the multi-stage problem, in which weapons have the same probability of killing the targets of equal value. A known number of targets are reached in the first stage, and only the probability distribution is known for the targets reached in the remaining stages. Ahner and Parson [10] considers the uncertainty of the second stage and assigned the available weapons by including the expected value of the second stage in the first stage target. Silav [11] considers that the incoming air targets are random, and defines the objective function as the maximization of the probability of no-leaker in the engagement sequence of the weapon system. Based on the above model, various algorithms have been proposed to solve the DWTA problem since the 1970s.

First, there are mathematical programming methods using for DWTA. Examples of DWTA combined with algorithms include: using large-scale neighborhood search algorithms (VLSN) [12], geometric methods [13], dynamic programming [14], [15], etc. These methods are fast and accurate when dealing with low-dimensional small problems, but the programming is cumbersome, and it is difficult to deal with problems with large variable dimensions. Modern combat situations are often complex and changeable, and traditional mathematical planning methods cannot deal with such problems.

Second, due to the computational complexity of WTA, many papers have applied heuristic algorithms. These algorithms are developed by simulating certain natural phenomena or processes, and are focused on providing real-time solutions rather than guaranteeing optimal solutions. Various heuristic algorithms have been proposed in the past like particle swarm optimization (PSO) [16], evolutionary algorithm (EA) [17], artificial bee colony (ABC) [18] and many more. In recent years, heuristic algorithms have been developed and improved in terms of accuracy and algorithm stability, including monarch butterfly optimization(MBO) [19], butterfly optimization algorithm (BOA) [20], harris hawks optimization (HHO) [21], etc. Some of the above heuristic algorithms are applied in the WTA field. Examples of algorithm applications include using the precise and heuristic combination algorithm [22], the improved monarch butterfly optimization(MBO) [23], the immune-based ant colony optimization (ACO) algorithm [24], the particle swarm optimization (PSO) method [25], [26], the artificial bee colony algorithm (ABC) [27], [28], the parallel simulated algorithm (PSA) [29], evolutionary algorithm (EA) [30], etc.

The models and algorithms of DWTA are summarized above, and the deficiencies in them are detailed below.

The common feature of the two traditional dynamic weapon target assignment models is that they are both a combination of static weapon target assignment stages. A static weapon target assignment stage is the process from collecting the combat situation to giving the assignment decision, and there is a strike stage between the adjacent static weapon target assignment stages. Except for the first static weapon target assignment stage, which is assigned according to the initial combat situation, the rest of the assignment stages are all made after the previous strike. However, between the two static weapon target assignment stages, and before the end of the previous strike stage, the threat ranking of targets may change with time and may require the defender to adjust the strategy in time. The traditional dynamic weapon target assignment model does not respond to this change in time.

As for the current DWTA algorithm, there are the following problems: First, the real time performance of the algorithm is insufficient. As the decision calculation time increases, the decision stages overlap with each other. The time of the latter decision stage is occupied by the previous decision stage, which may cause the defender to miss some intervals of missile strike moments, thus resulting in expired decisions (Figure 1 shows the real-time insufficient example). Second, the optimization strategy of the algorithm is ''short-sighted''. It can only make judgments on the current situation, and cannot make overall plans for the future situation on the basis of the current situation.

Due to the problem that the traditional DWTA model cannot change the strategy in time according to the change

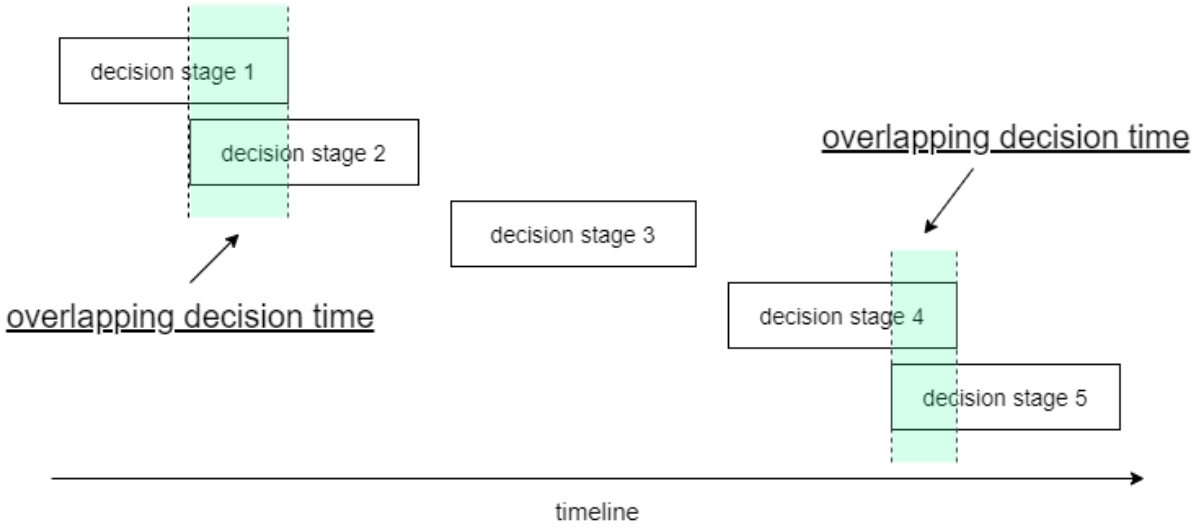

FIGURE 1. A real-time insufficient example. The time of decision stage 2 is occupied by decision stage 1.

in the target threat degree, a new DWTA model is proposed in this paper. This model samples the states of every time interval and makes decisions after sampling. It can solve the problem where the strategy cannot be changed in time when the target threat changes.

In recent years, with the widespread application of reinforcement learning, some scholars have applied reinforcement learning to the WTA field and achieved good results [31], [32]. After the learning stage is completed, the algorithm of reinforcement learning can quickly provide strategies. It can effectively solve the problem of producing outdated solutions. Moreover, the reward of reinforcement learning is the accumulation of overall benefits, and considering the possibility of changes in the situation after the current time, which can solve the problem of strategic shortsightedness.

Reinforcement learning is divided into value function method and policy gradient method. Compared with the value function method, the policy gradient method has smoother convergence and is robust to disturbances. The actor-critic method is a commonly used policy gradient method. It has the advantage of being able to perform a single-step update, which is faster than the traditional PG round update. Commonly used actor-critics include DDPG, A3C, TRPO, PPO, etc. Among them, TRPO [33] has the advantages of stable iteration step size and no violent fluctuation of the optimization curve. As the improved algorithm of TRPO, PPO [34] not only inherits the advantages of TRPO, but also improves the proxy objective, thus simplifying the calculation difficulty. This makes PPO the mainstream among actor-critic algorithms. For the new model in this paper, we employ PPO algorithm to solve the DWTA problem.

The main contribution of this paper can be summarized as follows.

1. This paper proposes a ''time-sampling dynamic weapon assignment model''(abbreviated as TS-DWTA), which is used to solve the problem that the traditional DWTA model cannot change the strategy in time with the change of the target threat level.

2. In this paper, the TS-DWTA model is optimized with reinforcement learning and heuristic algorithms, and their effects are compared. Then the study verifies which algorithm is more suitable for TS-DWTA model.

3. This study conducts separate experiments on different sampling intervals of the TS-DWTA model to demonstrate the influence of sampling intervals on the model results.

The structure of this paper is as follows: In Section I, we introduce the research status of weapon target assignments and discuss the current DWTA model and algorithm. In Section II, we present a model of ''temporal sampling dynamic weapon target assignment'', and then apply reinforcement learning and traditional heuristic algorithms to the model. In Section III, several typical simulation examples are designed to compare the computational speed and defense effect of the heuristic algorithm and reinforcement learning method that are used in this model. Furthermore, we compared the defense capability of different time sampling interval models. In Section IV, we summarize the full paper.

II. RESEARCH METHODS

In section I we discussed the background of weapon target assignment. In section II, we will discuss the model of weapon target assignment and the algorithms used.

A. RESEARCH MODEL

The traditional dynamic weapon target assignment model is a combination of multiple static weapon target assignment stages, and there is a strike stage between the adjacent static weapon target assignment stages. The assignment of the next stage will be carried out after the strike stage, resulting in the neglect of the moments between static weapon target assignment stages. Therefore, this paper proposes a ''time-sampling dynamic weapon assignment model''(abbreviated

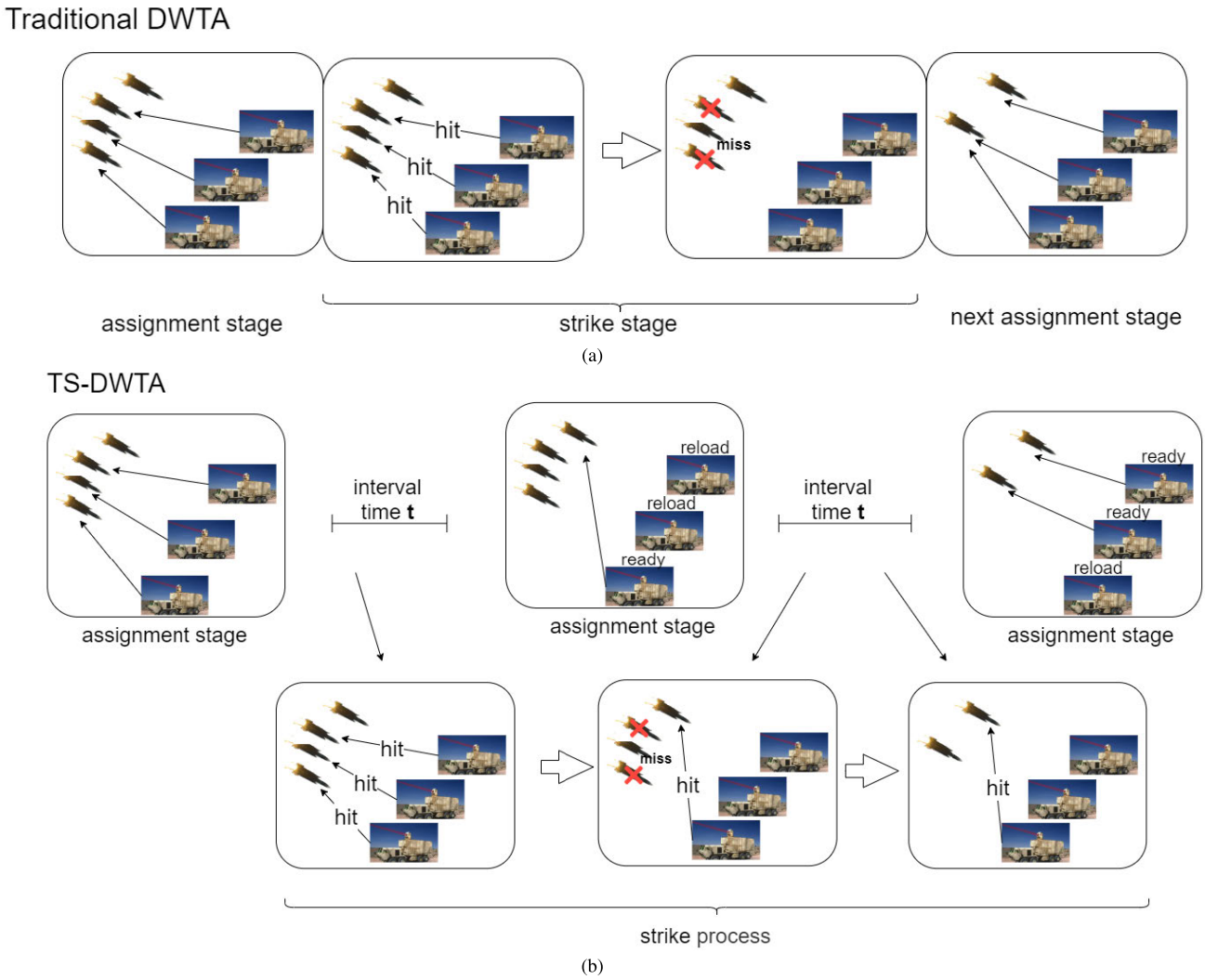

FIGURE 2. (a) The weapon target assignment timings for the traditional DWTA. (b) The weapon target assignment timings for the TS-DWTA model. And we assume weapons have different reload times. The TS-DWTA model can assign targets to ready weapons while the previous strike is not complete.

as TS-DWTA), which divides the weapon decision-making steps into time intervals. The defender samples the per time unit to obtain environmental state data and to give a decision-making plan. In each step, the model is updated regardless of whether the strike is completed or not. Moreover, this model can flexibly set the sampling time to avoid an out-sync between the received information and the strategy given. In addition, for weapons such as anti-aircraft guns, electro-optical jamming, and blinding equipment, which continuously exert influence on the target, it is difficult to clearly divide the strike step, and the ''TS-DWTA'' is more suitable for them.

Compared with the traditional model, the assignment idea of the ''TS-DWTA'' in the first assignment stage is the same. Strategies are all about distributing weapons to discovered targets. However, the ''TS-DWTA'' enters the next decision-making stage in the next unit time. There is no need to wait all the assigned weapon strikes are over, i.e., there is no need to divide the attack stage. In the ''TS-DWTA'', the weapons as-signed in the previous stage may not end the strike and cannot participate in the assignment, but the weapons that did not participate in the strike and the weapons that partially completed the strike can be assigned. This is shown in Figure2.

The model in this paper refers to the idea of the second model that was detailed in Section I. The number of incoming targets in each batch is unknown, and the time of attack is unknown, but only the time of attack and the upper and lower limits of the targets are known. This model assumes the following:

(1) There is only one set of assets;

(2) The confrontation target is set as a missile flying at a constant speed, and the distance from the asset is the same when it is discovered by the defender. Thus, the time from being discovered by the defense to striking the asset is the same. The upper limit of the time from when the missile is discovered by the defender to when it hits the ''asset'' is t_m;

(3) The missile hitting the ''asset'' will cause certain damage to the asset. The damage intensity of the missile i to the ''asset'' is set to D_i, and it is assumed that D is divided

TABLE 1. The notations employed in this paper.

t_m	The upper limit of the time from when the missile is discovered by the defender to when it hits the "asset"
D_i	The damage intensity of the missile i to the "asset". After a missile with a damage intensity of D_i hits an "asset", the damage suffered by the "asset" is D_i
D_{sum}	the sum of the "asset" damage D in each episode
$E(D_{sum})$	the average D_{sum}
$t_i \in [t_{iL}, t_{iH}]$	t_i is the missile attack time interval. t_iL,t_iH are the upper and lower bounds of the set missile attack interval
f_i	the missile attack frequency
$b_n \in [b_{nL}, b_{nH}]$	b_n is the number of missiles in n round. b_nL,b_nH are the upper and lower bounds of the number of missiles per round
a	the number of weapons
a_w	It means the "w"th weapon
p	The probability of a single weapon destroying a missile per unit time
$p_{0.5}$	The probability of a single weapon destroying a missile per 0.5 s
p_1	The probability of a single weapon destroying a missile per 1 s
s	The missile detection distance
v	Missile speed
th	the missile threat degree
S	State of reinforcement learning
dim	The size of the action space in reinforcement learning
r	reward of reinforcement learning
t_{step}	the sampling time, i.e., the step time interval
$t_{compute}$	the time taken by different algorithms to provide a decision in the decision-making step.

into nine levels. After a missile with a damage intensity of D_i hits an "asset", the damage suffered by the "asset" is D_i;

(4) There are multiple rounds of incoming missiles, the missile attack time interval $t_i \in [t_{iL}, t_{iH}]$ (t_{iL}, t_{iH} are the upper and lower bounds of the set missile attack interval), the number of missiles in each round $b_n \in [b_{nL}, b_{nH}]$ (b_nL,b_nH are the upper and lower bounds of the number of missiles per round), and t_i and b_n are not constant;

(5) The weapons are set as follows: the number of weapons is a, it is a continuous strike weapon, there is no firepower transfer time, and the probability of a successful strike in the same time period remains unchanged. The probability of a single weapon destroying a missile per unit time is p, the performance of each weapon is the same, and the probability of multiple weapons successfully hitting the same missile is independent and identically distributed.

For clarity, the notations employed in this paper are also listed in Table 1.

The confrontation process is as follows: 1. The first batch of missiles arrives, and the number is $b_n \in [b_{nL}, b_{nH}]$. Defenders strike when they spot a missile. The missile detection distance is s;

2. The defender defends against the missile. If there is a spare weapon, the weapon can strike the missile at any time;

3. After the time interval $t_i \in [t_{iL}, t_{iH}]$, the next round of missiles arrives. Go to Step 1;

4. If the preset missile round is reached or the "asset" is destroyed, the simulation episode ends (here episode refers to a simulation process).

When a missile hits an "asset" or the missile is destroyed, the missile is removed from the missile sequence. Since the missiles are constantly coming and the number is uncertain, if the method of assigning a number to each missile is adopted, then the missiles will have too many numbers and storage will be difficult. Therefore, we adopt a cyclic numbering method in this paper. When the storage space is full, the earliest batch of missiles in the storage space are deleted, and the free space and corresponding numbers to the new batch of missiles was saved. However, this requires the earliest batch of missiles in the storage space to be discovered for more than t_m (t_m is the maximum time between the defender discovering the missile and hitting the asset).

The storage method is as follows: the upper limit of the number of missiles in each round is b_{nH}, the lower limit of the missile attack interval is t_{iL}, and the upper limit of time t_m can be estimated (in fact, the number of missiles in each round will not be too large, and the frequency of missile attacks will not be too frequent). We reserve b_{nH} storage

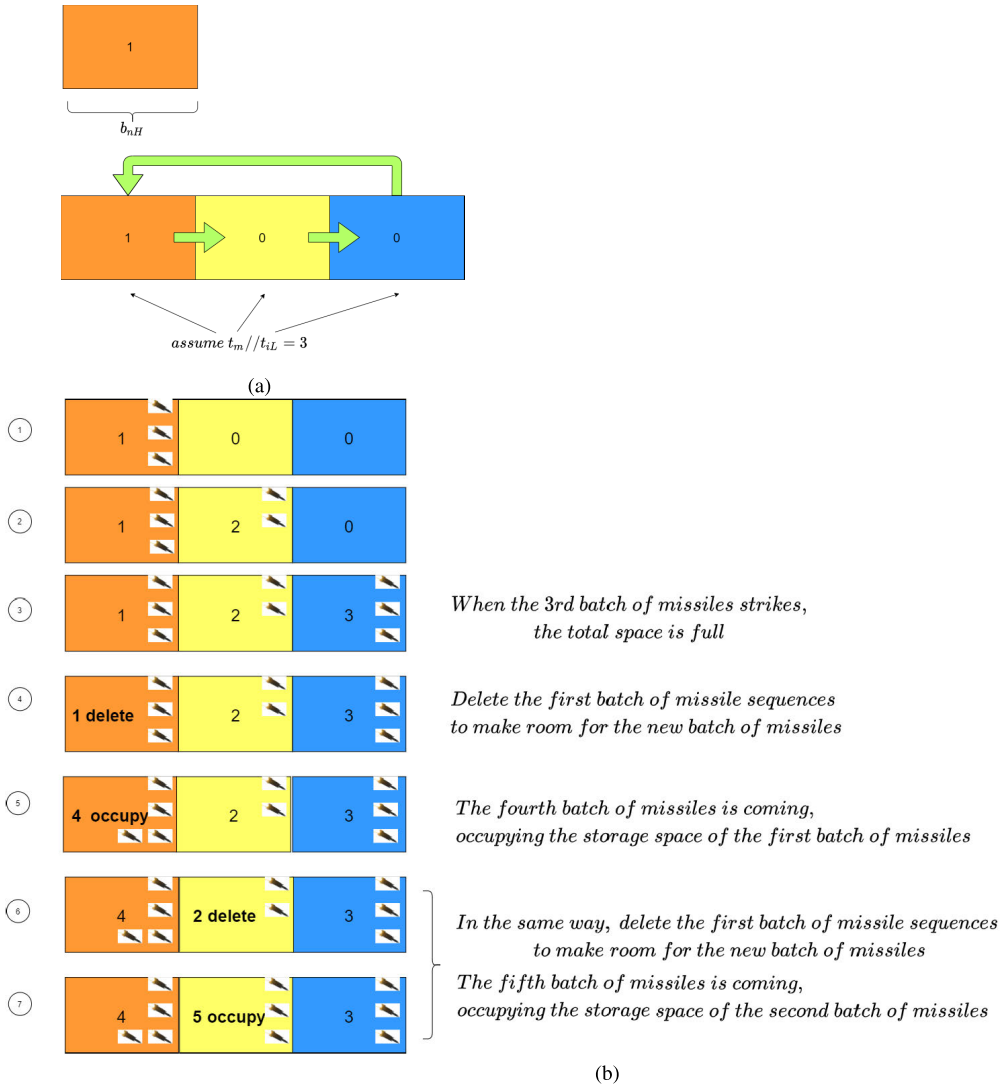

FIGURE 3. (a) Schematic diagram of the storage space. (b) Schematic diagram of the missile storage sequence.

space for each round of missiles. The total storage space size is $b_{nH} * (t_m//t_{iL})$, and the total batch size is $t_m//t_{iL}$ ("//" refers to integer division). If the storage is carried out in the above way, the detection time of a batch of secondary missiles must be greater than or equal to the time upper limit t_m, which is when the space is full. By deleting this batch of missiles, there is now room for new incoming missiles. This is shown in Figure 3.

Compared with the storage method in which each batch of missiles is given new numbers, the storage method in this paper has the following advantages:

1. Save storage space. It is not necessary to assign new storage space for each new batch of incoming missile information.

2. It can handle the storage problem of multiple batches of missiles. If the battle situation lasts for too long, the assigned storage space required by this method remains unchanged.

3. Convenient algorithm calculation. Using the method in this paper to number the missiles, the information length of the missiles is fixed. Then the length of the information inputting to the algorithm is fixed, which is convenient for the algorithm to process the information.

Based on the above model assumptions and confrontation process, we give the following objective function and constraints.

The objective function is related to the threat value of the target and the probability of target destruction. The purpose of the objective function is to maximize the threat eliminated by the weapon. The objective function expression is as Equation 1.

$$R = \max(\sum_{i=1}^{N} f(th_i, p_i)). \tag{1}$$

R is the objective function and N is the total number of targets observable in the air. th_i is the threat degree of the i-th target, and p is the probability of i-th target destruction. f is a function related to th and p. The specific objective function will be given in section B and section C combined with the specific algorithm.

Define the decision vector $[a_1, a_2, \cdots, a_{wn}]$, a_i represents the target of the weapon attack. wn represents the number of weapons.

Constraints include the following:

1. The target of the weapon can only be selected in the missile storage list. i.e. $a_i \in [0, M]$, M is the upper limit of missile storage.

2. A weapon can only hit one target at a time. As Equation 2 shows.

$$\begin{cases} a = i, i \in [0, M], & \text{if } a \text{ attacks target } i \\ a = 0, & \text{if } a \text{ does not attack} \end{cases} \tag{2}$$

B. COMBINATION OF THE "TS-DWTA MODEL" AND REINFORCEMENT LEARNING ALGORITHM

In section A, we introduces the TS-DWTA model. In the introduction section, the advantages of reinforcement learning are introduced. In this section, we combine reinforcement learning with the TS-DWTA model to illustrate.

Reinforcement learning is learning what to do–how to map situations to actions–so as to maximize a numerical reward signal [35]. Reinforcement learning regards learning as a trial and evaluation process. An agent chooses an action for the environment and the environment changes state after accepting the action, which, at the same time, generates a reinforcement signal (reward or punishment) to feed back to the agent. Agent chooses the next action according to the reinforcement signal and the current state of the environment. The selection principle is to increase the probability of receiving positive reinforcement (reward). The agent of reinforcement learning learns a set of behavior strategies after training, rather than just obtaining a solution to a problem. Through the obtained behavioral strategies, reinforcement learning can solve problems quickly and efficiently.

The reinforcement learning algorithm applied in this paper is the proximal policy optimization algorithm (PPO) [34]. In the past, the actor-critic algorithm was particularly sensitive to the step size, and it was difficult to choose an appropriate step size. In the training process, if the difference between the old and new policies is too large, it is not conducive to learning. The PPO algorithm inherits the TRPO algorithm to improve this and simplify it on the proxy target. The critic network in the PPO algorithm is the same as the traditional actor-critic algorithm and is used to estimate the return value. In the actor network–in order to reduce the large gradient and risky changes in the policy π_θ, which was caused by the update–the pruned objective function was used to replace this update. The idea is to limit the ratio of old and new strategies to a certain range $[1-\epsilon, 1+\epsilon]$. Through this method, the excessive difference between the strategies before and after the update is avoided, so that the update step size is within a reasonable range. It enables mini-batch updates over multiple training steps, which solves the problem where the step size is difficult to determine in the policy gradient algorithm.

The proxy objectives of the PPO algorithm $L^{clip}(\theta)$ are as Equation 3.

$$L^{clip}(\theta) = E[\min(l_t(\theta)\hat{A}_t, clip(l_t\ (\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]. \tag{3}$$

where $l_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ represents the likelihood ratio, and $\hat{A}_t$ is the generalized advantage estimate. $clip()$ means given an interval, and values outside the interval are clipped to the interval edges.

The advantage estimate $\hat{A}_t$ is expressed by generalized advantage estimator (GAE), and the GAE expression is as Equation 4.

$$\hat{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V. \tag{4}$$

where δ and λ are the proportional parameters, $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t).V(s_t)$ is the state value estimated by the critic network. The V in δ_{t+1}^V also means this.

This article divides the steps by time intervals. Each step updates the ''environment''. The ''environment'' model in each step handles the following operations:

1. Change the target of the weapon according to the input action;

2. Determine whether the missile is removed from the missile queue due to being destroyed or hitting an ''asset'';

3. Update the missile status according to the time;

4. Give the action reward for the step.

In this way, the DWTA model combined with reinforcement learning is a model with time continuity.

The reinforcement learning algorithm training process is shown in Algorithm 1.

Next, the weapon target assignment problem was described from the state, action, and reward functions of reinforcement learning.

In this article, the action of a single weapon is the number of the missile selected by the weapon: 0 means no weapon is used, 1, 2…n means strike missile 1, strike missile 2…strike missile n, a total of n+1 dimensions are set. For example, $a_w = 3$ means that the ''w''th weapon hits missile 3. When a missile is destroyed or hits the ''asset'', consider ''strike this missile'' as an unavailable action. Non-selectable actions need to be masked to block these actions.

If the model adopts a single-agent network structure, all weapons will be regarded as the same agent and output joint actions, i.e., the combination of all weapon actions is output as a vector action group $[a_1, a_2, \cdots, a_{wn}]$ by the actor network. Furthermore, a_n is a weapon with n action; the size of its action space is dim^{wn}, where dim represents the number of missiles that can be selected by a single weapon, and wn is the number of weapons. As the action space and weapons increase, the dimension of the action space will explode, thereby making calculation difficult.

The actor network in this paper adopts a multi-agent network structure. Usually, the agent of the multi-agent neural network can only observe part of the state, and the observed states are also different. However, this paper assumes that

Algorithm 1 The Reinforcement Learning Algorithm Training Process

Parameter initialization (learning rate, learning batch size, etc.).
State initialization (initialize the state of missiles and weapons).
Loop n episodes:
  Loop episode length or until done is True
    Agent receives state parameter. Next actions for all weapons are given by multiple actor neural networks.
    Execute the step function according to the action. Give the state, reward r and whether it is done.
    Actions, state, and rewards are stored for updates.
  After each episode, the reward is used to calculate the return, and the Agent uses the overall reward to update the critic network parameters.
  Compute the advantage function using the critic network output and rewards. Combined with the action probability of each agent, the actor network parameters are updated.
When the preset algebra is reached, the algorithm ends.

all agents can observe the known global state, because the defenders share information and obtain the same state. In this model, each weapon is assigned an actor network, and the actor network outputs the action selected by a single weapon. The size of the action space is dim, which is much smaller than dim^{wn}. This simplifies the computation of the action space. Moreover, single-agent reinforcement learning can only give rewards to the action groups of multiple weapon combinations, while multi-agent reinforcement learning can give different rewards to the actions of each weapon. In this model, multi-agent reinforcement learning can give each agent a clearer reinforcement direction.

The multi-agent decision-making model is shown in Figure 4.

The state is the threat degree of the missile, which is divided into two aspects: the distance from the missile to the ''asset'' and the damage intensity of the missile. State S is expressed as Equation 5.

$$S = [s, D]. \tag{5}$$

where s is the distance from the missile to the ''asset'' and D is the damage intensity of the missile.

The reward r reflects the pros and cons of the actions in each step (in this paper, each step is a unit of time). There are two methods to select r: One is to directly guide the agent to choose which missiles to choose, i.e., to provide the agent with a high re-ward for when the weapon chooses an assignment plan that the decision maker thinks is excellent. The advantage of this approach is that the agent can obtain knowledge that matches the artificial strategy. The disadvantage is that if the artificial strategy is not good, the agent cannot provide a good strategy. The second method is to use the ''environment'' value (including the parameters of entities such as weapons, missiles, and assets in the confrontation) as r. The advantage of this approach is that the agent can determines the strategy directly according to the quality of the ''environment''. If the ''environment'' value is selected well, then the agent's behavior can make the strategy develop in a positive direction quickly. The disadvantage is that if the agent were to select an ''environment'' value that will not change after multiple steps, it may become challenging for the agent to learn an effective strategy. For example, the primary goal of TS-DWTA is to reduce the number of missiles that hit ''assets''. If the number of missiles that hit ''assets'' is directly used as the measure of the reward r, then the re-ward will remain unchanged for multiple steps.

This article uses the second method. The ''environmental value'' is selected as the reward.

Let the missile threat degree th be:

$$th = \frac{D}{s}. \tag{6}$$

where s is the distance from the missile to the ''asset'', and D is the damage intensity of the missile. It means that the closer the missile is to the asset and the greater the damage intensity, then the greater its threat.

For any agent, i.e., any weapon, the sum of the negative numbers of all missile threats is used as the first reward r_1:

$$r_1 = -\sum_{1}^{i} th_i. \tag{7}$$

where i represents the number of missiles and r_1 represents a holistic reward and a threat to the overall environment. The optimization goal is to minimize the overall threat.

The threat degree of the missile selected by weapon w is used as the second reward r_2.

$$r_2 = th_w * p. \tag{8}$$

where th represents the missile threat degree. w represents the selected missile number by the weapon. p is the probability that the weapon will destroy the missile. This is an individual reward, which means that the weapon hits a missile with a high threat and that the reward is high.

When applying only the reward mentioned above, the weapon will frequently change the chosen target. This would be a waste of resources, as shown in Figure 5.

Therefore, when the action of a single weapon is different from the action selected in the previous step, the agent will be given a negative reward, which is set to a constant C.

Then, the reward for a single weapon is as equation 9.

$$r = r_1 + r_2 + C = -\sum_{1}^{i} th_i + th_w * p + C. \tag{9}$$

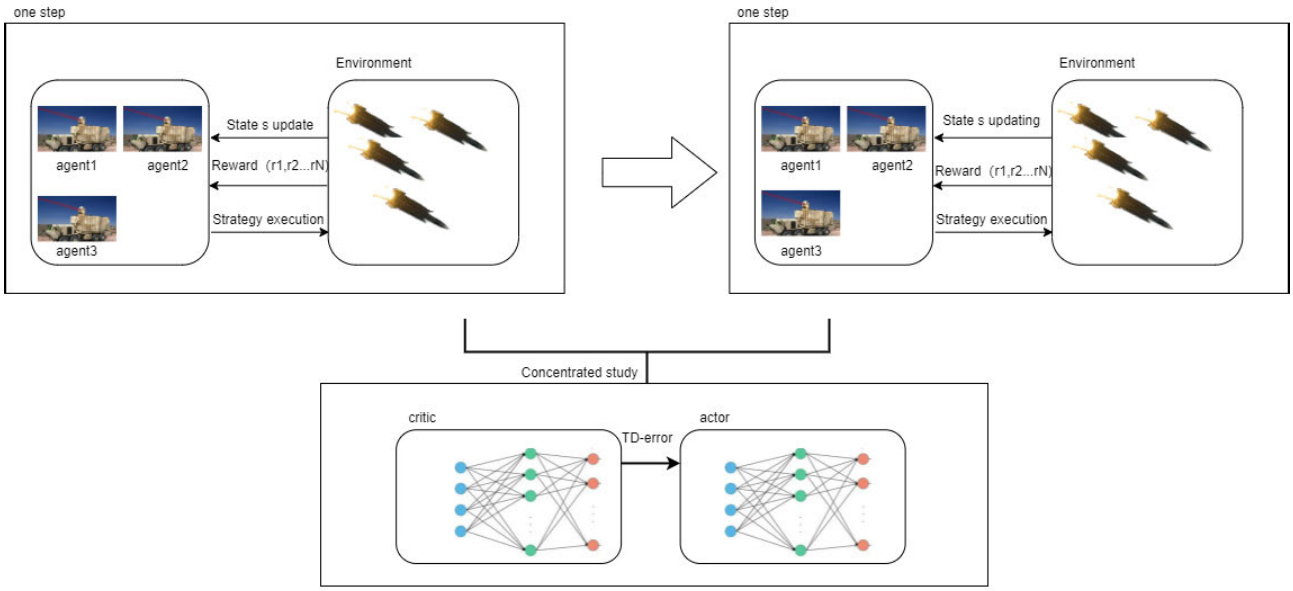

FIGURE 4. Schematic diagram of the multi-agent decision-making model.

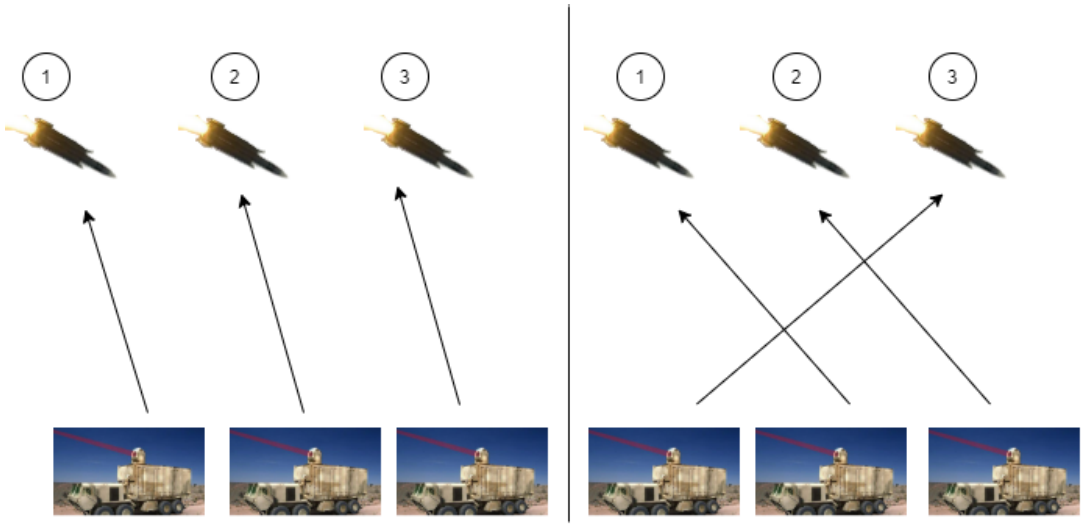

FIGURE 5. In the picture, the strike effect on the left and right sides is the same, but the corresponding weapons are different. If the assignment method on the left side is changed to the right one, the reward re-mains unchanged but strike resources are wasted.

C. COMBINATION OF THE "TS-DWTA MODEL" AND HEURISTIC ALGORITHM

In section B, we introduce the combination of reinforcement learning and the TS-DWTA model. In this section, we will introduce the combination of heuristic algorithm and TS-DWTA model. And compare it with reinforcement learning in terms of process.

The heuristic algorithm is a commonly used traditional DWTA algorithm. The heuristic algorithms used in this paper for comparison are the particle swarm optimization (PSO) algorithm, artificial bee colony algorithm (ABC) and butterfly optimization algorithm (BOA).

The PSO algorithm is a random search algorithm based on the group cooperation that is developed by simulating the foraging behavior of birds. It is generally considered to be a type of swarm intelligence [16]. PSO is initialized as a group of random solutions, and it then iteratively finds the optimal solution. In each iteration, the particle updates itself by tracking two "extreme values" *pbest* and *gbest* (*pbest* is the best position found by itself so far, and *gbest* is the best position found by all particles in the entire population). After finding these two optimal values, the particle updates its speed and position through the following formula Equation 10 and Equation 11.

$$v_i = \omega * v_i + c_1 * rand() * (pbest_i - x_i) + c_2 * rand() * (gbest_i - x_i). \quad (10)$$

$$x_i = x_i + v_i. \quad (11)$$

In the above formula, $i = 1,2,\ldots,N$, where N is the total number of particles in the group. v_i is the particle velocity, ω

is the inertia weight, rand() is a zero-to-one random number. x_i is the current position of the particle. c_1 and c_2 are learning factors, and ω is called the inertia factor (its value is non-negative).

The artificial bee colony algorithm (ABC) is inspired by the intelligent foraging behavior of bee colonies [18]. The ABC algorithm divides the artificial bee colony into three categories by simulating the honey-gathering mechanism of actual bees: honey-gathering bees, observation bees, and scout bees. The goal of the entire bee colony is to find a nectar source with the largest amount of nectar. In the ABC algorithm, the bees use the previous nectar source information to find new nectar sources and they share the nectar source information with the observer bees. The observer bees wait in the hive and look for new nectar sources based on the information shared by the bees. The task of scout bees is to find a new source of valuable nectar, and they will randomly search for nectar sources near the hive.

The butterfly optimization algorithm (BOA) is a heuristic algorithm proposed by Arora and Singh in 2019 [20]. The algorithm is inspired by the foraging and mating behavior of butterflies, which receive/sense and analyze scents in the air to determine potential directions to food sources/mating partners. Butterflies can identify different scents and perceive their intensity. When a butterfly moves from one location to another, its fitness changes accordingly. When the butterfly senses that another butterfly is emitting more scent in the area, it moves closer–a phase known as the global search. In another case, when the butterfly cannot perceive a scent larger than itself, it will move randomly–this stage is called the local search.

In the ''TS-DWTA'' model, the heuristic algorithm makes a decision on the assignment scheme every time interval, i.e., the heuristic swarm is used to iterate the current optimal solution at each time interval.

The specific decision-making process is:

1. When the sampling time is reached, missile parameters are obtained from the environment (number of missiles, threat degree, distance from ''assets'').

2. According to the missile parameters and the target optimization function, through the heuristic algorithm, the optimal solution is iterated. The weapon assignment decision is obtained according to the optimal solution.

3. Weapons and missiles interact with the environment according to the assignment decision until the next sampling time.

The heuristic algorithm optimizes the current stage, but the algorithm can only calculate the optimal solution of the current stage. The objective function of the heuristic algorithm r_3 is as Equation 12.

$$r_3 = -\frac{D}{s} * p. \tag{12}$$

where s is the distance of the missile from the ''asset'', D is the damage intensity of the missile, and p is the probability of the missile being destroyed.

This means that the greater the probability of destroying a missile with a high threat, the higher the reward.

In addition, when the action of a single weapon is different from the action selected in the previous step, a negative reward will be given to the agent, which is set to a constant C (the same as in Section II-B).

The total rewards r is as Equation 13.

$$r = r_3 + C. \tag{13}$$

The DWTA algorithm flow based on a heuristic algorithm is shown in Algorithm 2.

The decision-making flow chart of the combination of reinforcement learning and heuristic algorithm with TS-DWTA is given in Figure 6.

III. RESULTS

In section II, we introduced the TS-DWTA model, the combination of reinforcement learning and the TS-DWTA model, the combination of the comparison algorithm (heuristic algorithm) and the TS-DWTA model. The verification results of the calculation speed and protection ability of the algorithm and model are below, and the ''TS-DWTA'' model of different time intervals is compared. The simulation runs on a PC side, 8G memory, 4 cores 2.5GHz, and a CPU i5-7300HQ.

A. ALGORITHM PARAMETER AND MODEL PARAMETERS SETTING

For the reinforcement learning PPO, the learning rate of the action network was set as $A_{LR} = 0.001$, the learning rate of the evaluation network was $C_{LR} = 0.001$, the discount factor $\gamma = 0.995$, and the generalization advantage estimate $\lambda = 0.95$. For each iteration, 600 samples were collected, and training was performed 20 times in mini-batches of 300 samples. The model parameter is the value of a certain scene of end defense, and is shown in Table 2.

B. SIMULATION RESULT ANALYSIS

Take the sampling time, i.e., the step time interval $t_{step} =$ 1s, and assume that the unit time $t_{step} =$ 1s the success rate of destroying the missile is 0.3; then, use the PPO algorithm model for simulation. The reward curve during the PPO learning step is shown in Figure 7. The reward in the figure is the overall reward of the three agents in one episode (the simulation round). It can be seen that the reward r was significantly improved. This shows that reinforcement learning improves the strike efficiency of weapons.

1) COMPARISON EXPERIMENT OF UNIT DECISION TIME OF DIFFERENT ALGORITHMS

Experiment 1) tests the time taken by different algorithms to provide a decision in the decision-making step, which is expressed by $t_{compute}$. Experiment 1 gives the $t_{compute}$ of the PPO algorithm and heuristic algorithms. The parameters of the PPO algorithm are shown in Section III-A. Heuristic algorithms use certain different algorithm parameters, and the

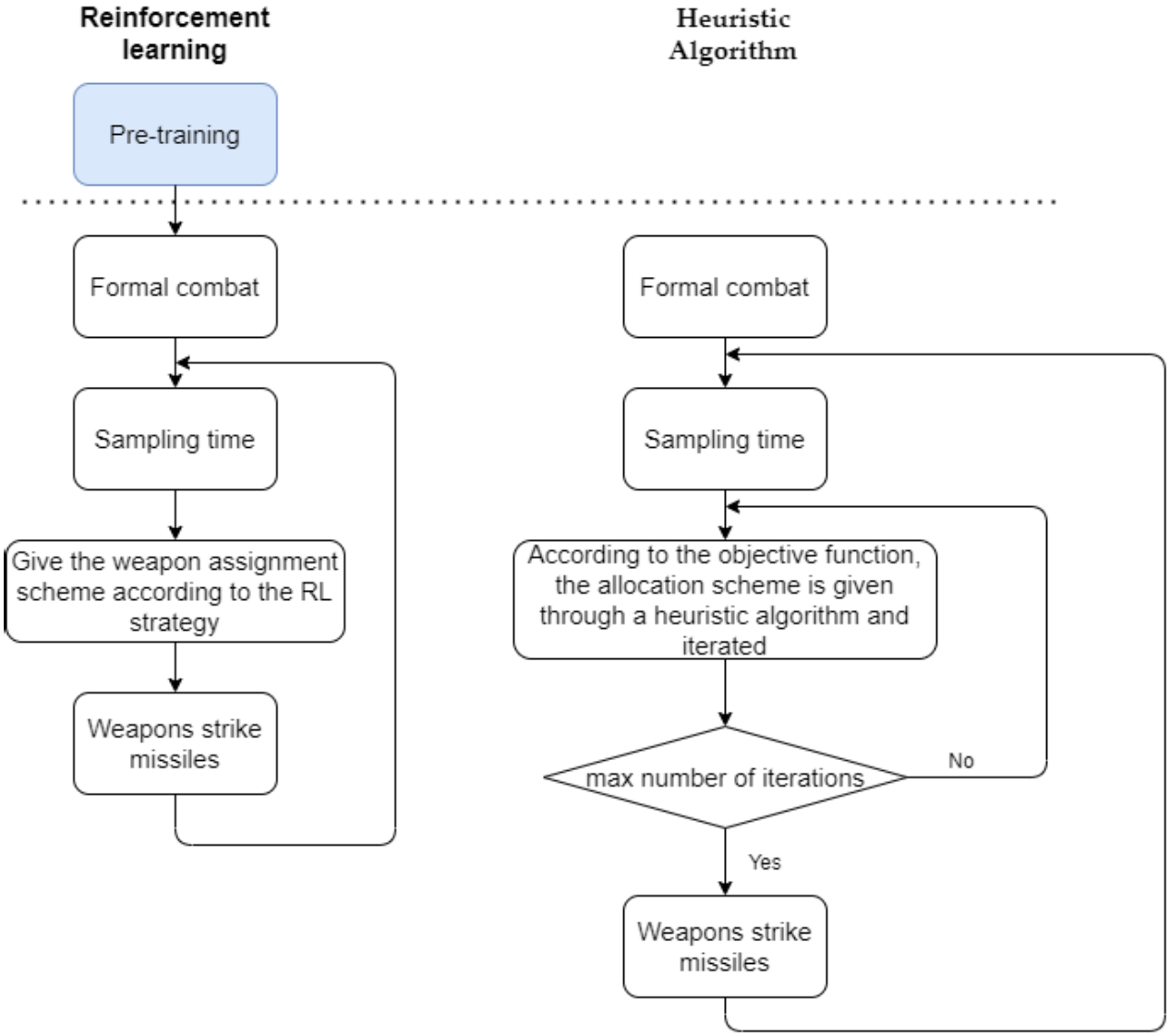

FIGURE 6. On the left is reinforcement learning decision flowchart. On the left is heuristic algorithm decision flowchart.

TABLE 2. Combat simulation parameters.

Missile first detection distance	10 km
Missile speed v	600 m/s
Number of weapons a	3
Missile attack interval t_i	[3 s, 4 s]
Number of missiles per round b_n	[4, 6]
Total missile rounds	5
The damage intensity of the missile	[1,9]

number of populations and the number of iterations in the parameters are different.

We simulated the model by applying the PPO algorithm and heuristic algorithms with different parameters, and tested the time taken by different algorithms to make one assignment decision. The experiment was repeated 100 times, and the simulation results are detailed in Table 3.

$t_{compute}$ represents a delay, which refers to the time from when the battle situation information is input to the agent to when the agent makes one assignment decision. If $t_{compute}$ is too long, it will cause the decision to miss its applicable battle situation and become invalid. Taking the terminal defense distance of 10 km and the missile flight speed of 600 m/s as an example, if the decision-making time delay reaches 2 s, the

Algorithm 2 The Reinforcement Learning Algorithm Training Process

```
Parameter initialization (learning rate, learning batch size, etc.).
State initialization (initialize missiles and weapons state, initial best action).
Loop n episodes:
    Loop episode length or until done is True
        Execute the step function according to the best action. Gives the next step states.
        Calculate the new best action using the heuristic algorithm by the new states.
When the preset algebra is reached, the algorithm ends.
```

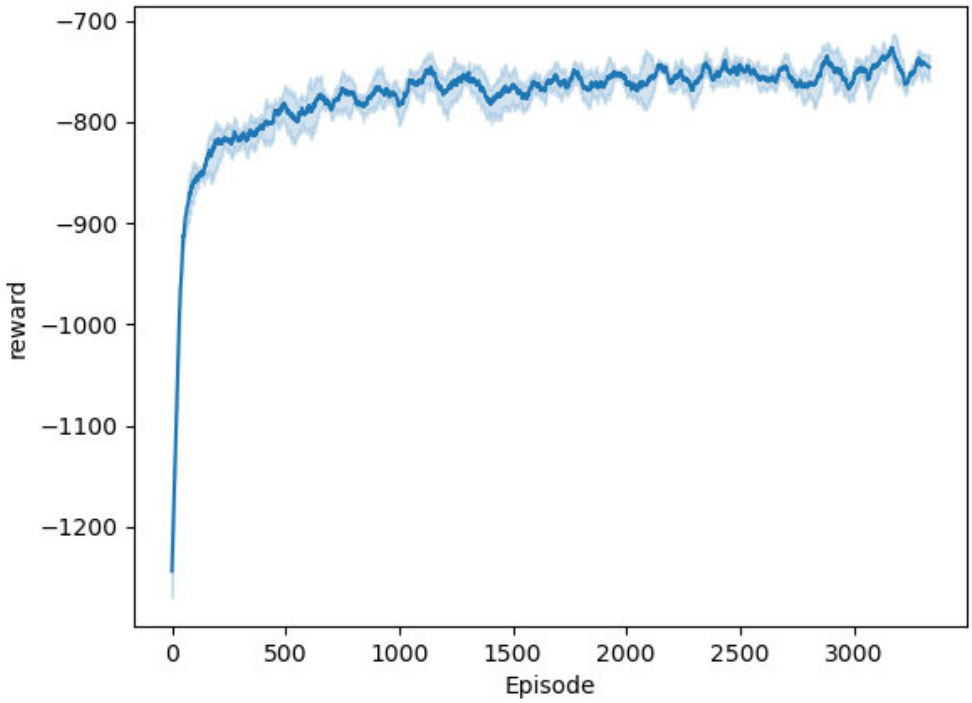

FIGURE 7. Weapon target assignment reward curve. The curve in the figure is the comprehensive effect of 5 repeated experiments, and the blue range is the maximum and minimum value range of 5 experiments. The solid line curve is the average curve of 5 experiments.

difference between the theoretical decision-making position and the actual assignment position will be 1200 m; thus, the error would be too large. However, it is acceptable when the delay is within 1 s. Therefore, the $t_{compute}$ of the simulation experiment in this paper is less than 1 s.

By comparing the PPO and several heuristic algorithms, it can be seen that the speed of reinforcement learning PPO processing problems is better than that of heuristic algorithms. This is because after reinforcement learning is trained, PPO's neural network has ''stored'' the learning experience. In formal combat, the stored experience can be used to directly draw results. It is not necessary to iteratively calculate every time a new target is being attacked as the heuristic algorithm requires. As for the defensive ability of the algorithm, it will be compared in the next experiment.

2) COMPARISON EXPERIMENT OF ''ASSET'' DAMAGE DEGREE OF DIFFERENT ALGORITHMS

Experiment 2) takes the sampling time, i.e., the step time interval $t_{step} = 1$ s, and assumes that when $t_{step} = 1$ s, the success rate of a single weapon destroying a missile per unit time is 0.3.

In Experiment 2), the experimental algorithm is PPO, and several heuristic algorithms with parameters of max $t_{compute}$<1 s are used as comparison algorithms. The corresponding parameters are detailed in Table 4.

t_{step} is the time interval for the defender to obtain battle information and $t_{compute}$ is the time it takes for the agent to make decisions after the defender obtains the battle information. Generally, $t_{compute} < t_{step}$; otherwise, the assignment of a battle situation has not been processed, and the next battle situation will be input to the agent.

The number of missile rounds in each episode is 5. After each episode, set the sum of the ''asset'' damage D in each episode as D_{sum}(after a missile with a damage intensity of D_i hits an ''asset'', the damage suffered by the ''asset'' is D_i), and use it as a reference for algorithm comparison.

Take 200 episodes results for comparison and take the average: $E(D_{sum}) = D_{sum}/200$.

The results of $E(D_{sum})$ are shown in Table 5.

TABLE 3. Average $t_{compute}$ and max $t_{compute}$ of the PPO and heuristic algorithms.

	Iterations	Number of Populations	Average $t_{compute}$	Max $t_{compute}$
PSO	100	50	0.408 s	0.606 s
	200	50	0.798 s	1.158 s
	300	50	1.147 s	1.639 s
	100	100	0.772 s	0.915 s
	200	100	1.431 s	2.071 s
	300	100	1.977 s	3.061 s
ABC	100	50	0.754 s	0.965 s
	200	50	1.986 s	2.677 s
	300	50	3.332 s	4.114 s
	100	100	2.192 s	2.635 s
	200	100	4.698 s	6.191 s
	300	100	7.559 s	9.116 s
BOA	100	50	0.803 s	0.963 s
	200	50	1.742 s	2.017 s
	300	50	2.589 s	3.016 s
	100	100	1.805 s	2.091 s
	200	100	3.189 s	4.083 s
	300	100	5.983 s	6.760 s
PPO	-	-	0.009 s	0.013 s

TABLE 4. Average $t_{compute}$ and max $t_{compute}$ of the heuristic algorithms in Experiment 2).

	Iterations	Number of Populations	Average $t_{compute}$	Max $t_{compute}$
PSO	100	50	0.408 s	0.606 s
	100	100	0.772 s	0.915 s
ABC	100	50	0.754 s	0.965 s
BOA	100	50	0.803 s	0.963 s

TABLE 5. D_{sum} of the PPO and heuristic algorithms.

Algorithm		$E(D_{sum})$
PPO		5.436
PSO	Iteration:100 population:50	9.453
	Iteration:100 population:100	6.421
ABC	Iteration:100 population:50	6.635
BOA	Iteration:100 population:50	7.192

Histogram of the number of each algorithm's D_{sum} exceeding a certain value in 200 episodes, as is shown in Figure 8.

It can be seen from Table 5 that $E(D_{sum})$ of the PPO is smaller than that of heuristic algorithms. This proves that the PPO algorithm is more effective in dealing with TS-DWTA.

TABLE 6. D_{sum} of the PPO algorithm with different t_{step}.

t_{step}	0.5 s	1 s	1.5 s
$E(D_{sum})$	3.875	5.705	9.195

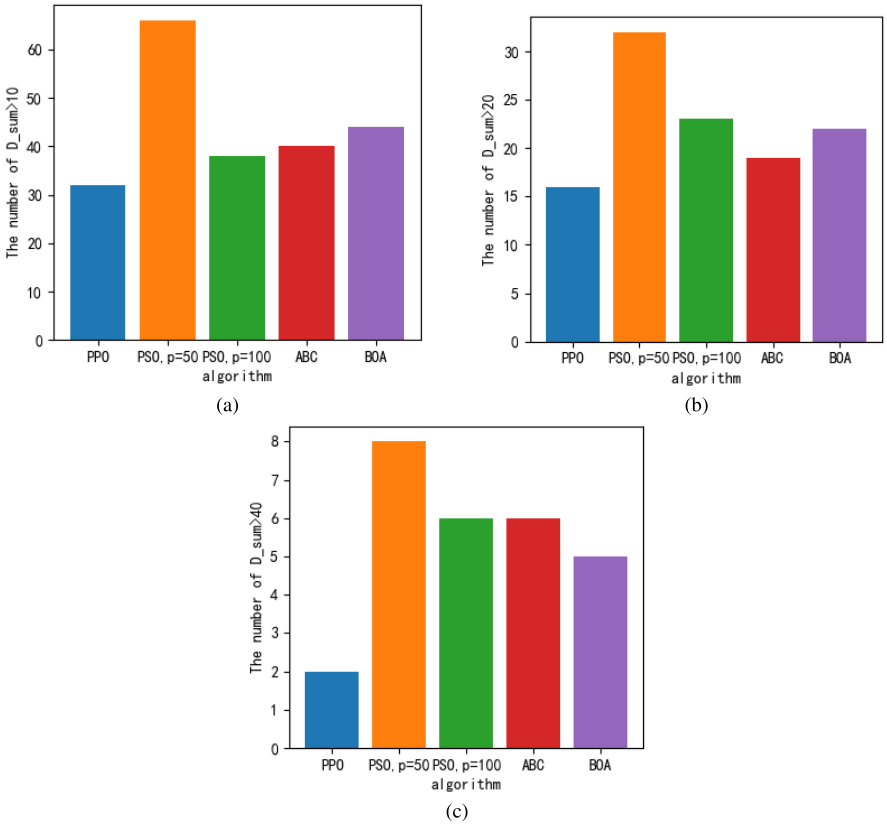

FIGURE 8. Histogram of the number of each algorithm's D_{sum} exceeding a certain value of 200 episodes. The p in PSO means population. (a) Histogram of the number of each algorithm's D_{sum} exceeding 10. (b) Histogram of the number of each algorithm's D_{sum} exceeding 20. (c) Histogram of the number of each algorithm's D_{sum} exceeding 40.

In Figure 8, $D_{sum} = 10$ is the limit of slight damage to "asset", $D_{sum} = 20$ is the limit of medium damage to "asset", and $D_{sum} = 40$ is the limit of serious damage to "asset". The heuristic algorithm is not as good as PPO in terms of whether the D_{sum} value is too large in the model. It shows that the model with the heuristic algorithm suffers more serious damage after multiple rounds of simulation. This is because heuristic algorithms focus on the level of the current missile threat value, while ignoring the consideration of the missile threat value in the future. This results in the defender not retaining a part of the defensive force to deal with the incoming targets whose current threat level is low but will increase rapidly in the future. The PPO reinforcement learning algorithm comprehensively considers the time, focusing on reducing the threat level in the overall time.

3) TS-DWTA COMPARISON EXPERIMENT OF "ASSETS" DAMAGE DEGREE AT DIFFERENT SAMPLING TIME

Experiment 3) compares the PPO with different sampling time.

Take three sampling times of $t_{step} = 0.5$ s, $t_{step} = 1$ s and $t_{step} = 1.5$ s to test the impact of different sampling times on the defense effect.

Take 0.5 s as the basic sampling interval, $t_{step} = 0.5$ s is one decision for one sampling interval, $t_{step} = 1$ s is one decision for two sampling intervals, and $t_{step} = 1.5$ s is one decision for three sampling intervals.

Section II-A assumes that the probability of successful strikes in the same time period remains unchanged. In order to be consistent with the success rate, $p_1 = 0.3$ for a single weapon destroying a missile when $t_{step} = 1$ s in Experiment 2). Furthermore, it is assumed that when $t_{step} = 0.5$ s, the success rate of a single weapon destroying a missile is $p_{0.5} = 0.163$. Then, there is

$$p_1 = 1 - (1 - p_0.5)(1 - p_0.5) = 0.3. \tag{14}$$

Use D_{sum} as a reference for algorithm comparison. Take 200 episodes results for comparison. The results of $E(D_sum)$ are shown in Table 6.

The D_{sum} scatter plot of 200 episodes is as shown in Figure 9.

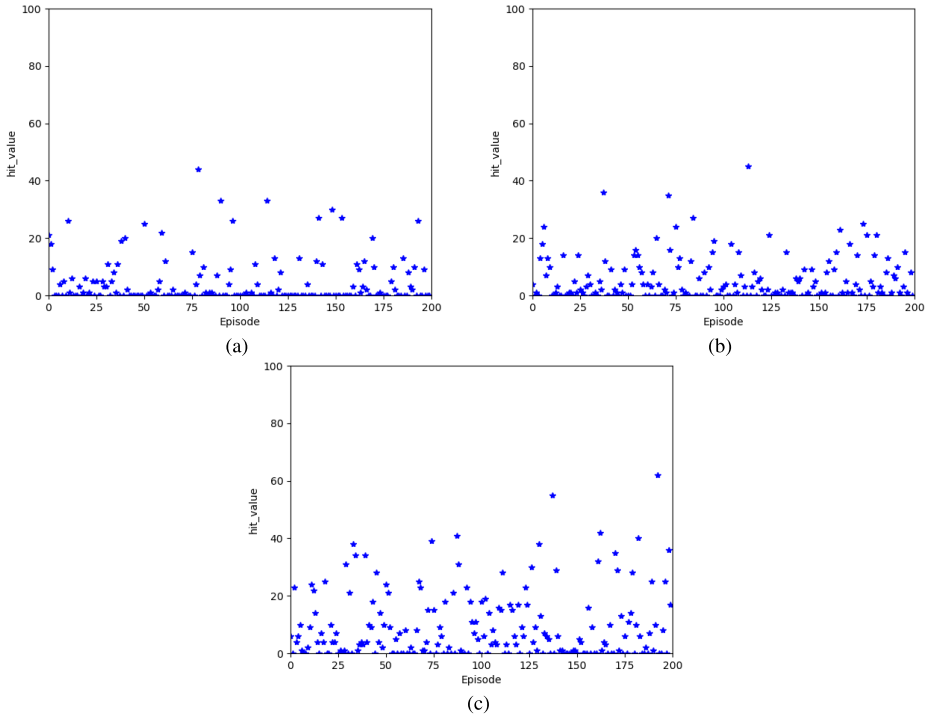

FIGURE 9. The D_{sum} scatter plot of 200 episodes with different t_{step} for PPO. (a) $t_{step} = 0.5$. (b) $t_{step} = 1$. (c) $t_{step} = 1.5$.

It can be seen from Table 6 that the D_{sum}m of $t_{step} = 0.5$s is lower than that of $t_{step} = 1$ s and $t_{step} = 1.5$ s, i.e., the damage to the assets is lesser.

In Figure 9, The D_{sum} scatter of 9(a) is generally lower than that of 9(b), and the D_{sum} scatter of 9(b) is generally lower than that of 9(c). It also shows that the D_{sum} of $t_{step} = 0.5$ s is lower than that of $t_{step} = 1$ s and $t_{step} = 1.5$ s.This shows that increasing the sampling frequency will make the algorithm application effect better. The results of this experiment show the influence of the frequency of information interaction on the outcome. It corresponds to the decision frequency. In actual decision-making, if conditions permit, increasing the frequency of decision-making can correct the shortcomings in decision-making in time and improve decision-making efficiency.

At the same time, the actual sampling interval capability should also be considered in detail. Limited by the ability to acquire information on the actual battlefield, the sampling time will not be reduced indefinitely. The model sampling time should be greater than the information acquisition time interval that can be provided by the actual battlefield. Under the premise of ensuring that the t_{step} is greater than the actual sampling interval, then the smaller t_{step} is, the better the defense effect will be.

4) THE EFFECT OF COMBAT SIMULATION PARAMETERS ON THE ALGORITHM

In the study, the missile attack frequency f_i (corresponds to missile attack interval t_i) and the number of missiles per round b_n are affected by the storage space. According to the storage space model in Section II-A, if f_i is higher than the minimum f_i, the numbers of the old and new missile batches will overlap. b_n exceeding maximum b_n will cause a single batch of missile numbers to overflow the storage space. This can cause errors in the calculations of algorithms (including PPO and various heuristics). To solve this problem, it is necessary to leave a certain margin for max f_i and max b_n. That is, the actual f_i and b_n are smaller than max f_i and max b_n. On the premise that the missile number does not overflow, changing f_i and b_n, the format of the input value of the algorithm remains unchanged, and the strike capability of the algorithm decision is still maintained.

Theoretically, when the f_i and b_n of the actual battle are large enough, max f_i and max b_n can also be very large, which will not have a great impact on the decision-making effect of the algorithm. However, max f_i and max b_n should not be too far from the actual f_i and b_n of the battle. Excessive parameters will make the input have multiple empty inputs (for example, only 5 valid storage objects are stored in 20 storage locations), i.e., the parameters are sparse. This affects training and computation performance. The best max f_i and max b_n are close to the actual parameters without overflow.

The wave number has no effect on the algorithm. Because the missile number is stored cyclically, it will not affect the input format of the algorithm, so the algorithm performance will not be affected.

IV. CONCLUSION

This paper proposes a DWTA model divided into stages by time: " TS-DWTA". Compared with the method in which

the traditional DWTA model divides multiple static weapon target assignment stages, the model in this paper divides the entire combat process into time intervals; in addition, it can divide the process when the target threat level changes between the traditional DWTA stages. It makes up for the untimely decision-making problem caused by the limitation of stage divisions in the traditional DWTA model.

The time sampling of TS-DWTA can be set flexibly; thus, this work studied the influence of different sampling times on the defense effect under the condition of satisfying the objective conditions. Through the simulation comparison of different sampling times, it is concluded that the shorter the sampling time, the better the defense effect.

At the same time, this study applied the reinforcement learning PPO algorithm to the DWTA problem. With the help of rapid calculation after the training is completed, the timeliness of the calculation of the DWTA problem was solved. And with the cumulative feature of time dimension rewards in reinforcement learning, the issues in DWTA where the current decision-making step cannot consider the subsequent steps are solved. By comparing with the heuristic algorithm, it can be seen that the calculation speed of reinforcement learning is several orders of magnitude higher than that of the heuristic algorithm, and the ability to take into account the future decision-making stage is also better.

Due to the use of the new model, the research method in this paper cannot be directly compared with other methods using the traditional model. But there is the indirect comparison method. The traditional DWTA model can be regarded as the TS-DWTA model in which the sampling interval is the duration of the strike step. The TS-DWTA model proposed in this paper effectively expands the traditional DWTA model by flexibly setting the sampling time and combining the reinforcement learning method with the model. The simulation results verify the effectiveness of the method proposed in this paper.

At the same time, the combination of reinforcement learning and TS-DWTA as a new algorithm and model combination also has certain application difficulties:

1. How to define the reward function. The reward function (corresponding to the objective function of the heuristic algorithm) has a great influence on the effect of reinforcement learning. However, there is no uniform standard for setting the reward function. Therefore, for the DWTA problem, how to set up the most effective reward function still needs to be explored.

2. How to define the state space. There are a lot of state information of weapons and targets in the DWTA problem. Extracting effective state information can effectively improve the training results of reinforcement learning.

3. How to improve the generalization ability. The generalization ability of reinforcement learning is an important issue in reinforcement learning. Reinforcement learning must ensure that the training results are applicable to practical applications. For the DWTA problem, reinforcement learning needs to learn enough battlefield situations to perform well in practice.

Solving these difficulties will be the focus of the next step in the research of TS-DWTA with reinforcement learning.

REFERENCES

[1] S. P. Lloyd and H. S. Witsenhausen, "Weapon allocation is NP-complete," in *Proc. Summer Comput. Simul. Conf.*, 1986, pp. 1054–1058.
[2] A. S. Manne, "A target-assignment problem," *Oper. Res.*, vol. 6, no. 3, pp. 346–351, Jun. 1958.
[3] A. R. Eckler and S. A. Burr, *Mathematical Models of Target Coverage and Missile Allocation*. Alexandria, VA, USA: Military Operations Research Society, 1972.
[4] R. A. Murphey, "Target-based weapon target assignment problems," *Nonlinear Assignment Problems: Algorithms and Applications*. Berlin, Germany: Springer, pp. 39–53, 2000.
[5] C. Leboucher, H.-S. Shin, P. Siarry, R. Chelouah, S. Le Ménec, and A. Tsourdos, "A two-step optimisation method for dynamic weapon target assignment problem," in *Recent Advances on Meta-Heuristics and Their Application to Real Scenarios*. Rijeka, Croatia: InTech, 2013, pp. 109–129.
[6] B. Xin, J. Chen, Z. Peng, L. Dou, and J. Zhang, "An efficient rule-based constructive heuristic to solve dynamic weapon-target assignment problem," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 3, pp. 598–606, May 2011.
[7] J. Zhengrong, L. Faxing, and W. Hangyu, "Multi-stage attack weapon target allocation method based on defense area analysis," *J. Syst. Eng. Electron.*, vol. 31, no. 3, pp. 539–550, Jun. 2020.
[8] S. A. Burr, J. E. Falk, and A. F. Karr, "Integer prim-read solutions to a class of target defense problems," *Oper. Res.*, vol. 33, no. 4, pp. 726–745, Aug. 1985.
[9] R. A. Murphey, "An approximate algorithm for a weapon target assignment stochastic program," in *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*. Boston, MA, USA: Springer, 2000, pp. 406–421.
[10] D. K. Ahner and C. R. Parson, "Optimal multi-stage allocation of weapons to targets using adaptive dynamic programming," *Optim. Lett.*, vol. 9, no. 8, pp. 1689–1701, Dec. 2015.
[11] A. Silav, E. Karasakal, and O. Karasakal, "Bi-objective dynamic weapon-target assignment problem with stability measure," *Ann. Oper. Res.*, vol. 311, no. 2, pp. 1229–1247, Apr. 2022.
[12] M.-Z. Lee, "Constrained weapon-target assignment: Enhanced very large scale neighborhood search algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 40, no. 1, pp. 198–204, Jan. 2010.
[13] C. Leboucher, H.-S. Shin, S. L. Ménec, A. Tsourdos, and A. Kotenkoff, "Optimal weapon target assignment based on an geometric approach," *IFAC Proc. Volumes*, vol. 46, no. 19, pp. 341–346, 2013.
[14] M. T. Davis, M. J. Robbins, and B. J. Lunday, "Approximate dynamic programming for missile defense interceptor fire control," *Eur. J. Oper. Res.*, vol. 259, no. 3, pp. 873–886, Jun. 2017.
[15] M. F. Hocaoğlu, "Weapon target assignment optimization for land based multi-air defense systems: A goal programming approach," *Comput. Ind. Eng.*, vol. 128, pp. 681–689, Feb. 2019.
[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Aug. 1995, pp. 1942–1948.
[17] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms Appl.* London, U.K.: Psychology Press, 1985, pp. 93–100.
[18] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Eng. Fac. Comput. Eng. Dept., Erciyes Univ., Kayseri, Türkiye, Tech. Rep. TR06, 2005.
[19] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, pp. 1995–2014, May 2015.
[20] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Feb. 2019.
[21] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
[22] R. K. Ahuja, A. Kumar, K. C. Jha, and J. B. Orlin, "Exact and heuristic algorithms for the weapon-target assignment problem," *Oper. Res.*, vol. 55, no. 6, pp. 1136–1146, Dec. 2007.

[23] Y. Jishuai, G. Hongwu, L. Xiaoma, and P. Chi, "Improved monarch butterfly optimization for multi-to-multi weapon-target assignment problems," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 1391–1396.
[24] Z.-J. Lee, C.-Y. Lee, and S.-F. Su, "An immunity-based ant colony optimization algorithm for solving weapon-target assignment problem," *Appl. Soft Comput.*, vol. 2, no. 1, pp. 39–47, Aug. 2002.
[25] H. Zhai, W. Wang, Q. Li, and W. Zhang, "Weapon-target assignment based on improved PSO algorithm," in *Proc. 33rd Chin. Control Decis. Conf. (CCDC)*, May 2021, pp. 6320–6325.
[26] G. Peng, Y. Fang, S. Chen, W. Peng, and D. Yang, "A hybrid multiobjective discrete particle swarm optimization algorithm for cooperative air combat DWTA," *J. Optim.*, vol. 2017, Apr. 2017, Art. no. 8063767.
[27] T. Chang, D. Kong, N. Hao, K. Xu, and G. Yang, "Solving the dynamic weapon target assignment problem by an improved artificial bee colony algorithm with heuristic factor initialization," *Appl. Soft Comput.*, vol. 70, pp. 845–863, Sep. 2018.
[28] D. Guo, Z. Liang, P. Jiang, X. Dong, Q. Li, and Z. Ren, "Weapon-target assignment for multi-to-multi interception with grouping constraint," *IEEE Access*, vol. 7, pp. 34838–34849, 2019.
[29] E. Sonuc, B. Sen, and S. Bayir, "A parallel simulated annealing algorithm for weapon-target assignment problem," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 4, pp. 87–92, 2017.
[30] X. Wu, C. Chen, and S. Ding, "A modified MOEA/D algorithm for solving bi-objective multi-stage weapon-target assignment problem," *IEEE Access*, vol. 9, pp. 71832–71848, 2021.
[31] F. Meng, K. Tian, and C. Wu, "Deep reinforcement learning-based radar network target assignment," *IEEE Sensors J.*, vol. 21, no. 14, pp. 16315–16327, Jul. 2021.
[32] Q. Cheng, D. Chen, and J. Gong, "Weapon-target assignment of ballistic missiles based on Q-learning and genetic algorithm," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Oct. 2021, pp. 908–912.
[33] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.
[34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[35] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

CHANG LIU received the B.S. degree in automation from the Beijing Institute of Technology, in 2015. He is currently pursuing the Ph.D. degree in mechatronics engineering with the Changchun Institute of Optics, Fine Mechanics and Physics (CIOMP), Chinese Academy of Sciences, Changchun, China. His current research interests include resource allocation and reinforcement learning.

JIANG LI received the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 2014. He is currently a Research Fellow with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His current research interests include intelligent simulation and electro-optical system design.

YE WANG is currently pursuing the Ph.D. degree with the Chinese Academy of Sciences, Changchun, China. She is an Assistant Researcher with the Changchun Institute of Optics, Fine Mechanics and Physics (CIOMP). Her current research interests include intelligent decision-making and multi-agent deep reinforcement learning.

YANG YU received the M.S. and Ph.D. degrees from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 2008 and 2012, respectively. He is currently a Research Fellow with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His current research interests include decision-making AI, intelligent computing, and applications.

LIHONG GUO received the M.S. and Ph.D. degrees from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 1999 and 2003, respectively. She is currently a Research Fellow with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. Her current research interests include intelligent computing and its applications.

YUAN GAO received the B.S. degree in mechatronics engineering from the North University of China (NUC), in 2019. He is currently pursuing the Ph.D. degree in mechatronics engineering with the Changchun Institute of Optics, Fine Mechanics and Physics (CIOMP), Chinese Academy of Sciences, Changchun, China. His current research interests include intelligent decision-making and multi-agent deep reinforcement learning.

YANG CHEN received the Ph.D. degree from the University of Chinese Academy of Sciences. He is currently an Associate Research Fellow with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His current research interests include missile guidance and control, computer simulation technology, and reinforcement learning.

FENG ZHANG received the Ph.D. degree from the University of the Chinese Academy of Sciences, in 2022. He is currently with the Aviation University of Air Force. His current research interests include reinforcement learning, simulation experiments, and intelligent planning.