Defence Science Journal, Vol. 74, No. 5, September 2024, pp. 662-670, DOI : 10.14429/dsj.74.19632

Static Weapon-Target Assignment Based on Battle Probabilities and Time-Discounted Reward

Nam Eung H.* and Hyung Jun K.

Department of Land Systems, Hanwha Systems Co., Seongnam-si - 13524, South Korea

**E-mail: skadnd144@hanwha.com*

ABSTRACT

Target-based weapon-target assignment (WTA) aims to minimize the total value of enemies. It means that maximizing the total reduced value of the enemies is the objective of the target-based WTA. The reward of an assignment is typically set as the reduction in the enemy's value when an ally and an enemy have combat, and the value is calculated by multiplying the current value of the enemy by the probability of the enemy's survival after the combat. However, allies may be assigned to enemies who are far away if the reward is calculated similarly. Additionally, a method of calculating battle probability that reflects the characteristics and deployment of enemies and allies is needed in order to apply it in the defense industry. In this paper, we propose a target-based static weapon-target assignment to solve these problems. First, we propose a method to calculate battle probabilities for one-to-one, one-to-many, many-to-one, and many-to-many combat. The probabilities are composed of 4 cases; ally-survival-enemy-survival, ally-survival-enemy-destroyed, ally-destroyed-enemy-survival, and ally-destroyed-enemy-destroyed. Then a time-discounted reward for assignment based on the battle probabilities is calculated to consider the time it takes to have combat. Finally, the tank combat simulation results are discussed. The performance of the proposed WTA algorithm is highlighted through an analysis of assignment results and a comparison of outcomes based on the application of time-discounted rewards.

Keywords: Weapon-target assignment; Target allocation; Battle probabilities; Combat simulation

1. INTRODUCTION

Nowadays, combatants are changing from humans to unmanned robots to minimize troop loss. The soldiers fight by controlling unmanned robots remotely from a safe area instead of engaging in direct combat. Before the unmanned robots became autonomous, soldiers had to remotely control the robots individually. As autonomous technologies advance, unmanned robots can operate semi-autonomously. If the order of missions and the path to reach each mission point are given to each robot, they will perform the missions autonomously. A soldier only controls a robot directly when remote control is necessary, allowing the soldier to manage several robots simultaneously.

The Weapon-Target Assignment (WTA) is an essential technique for a soldier to operate multiple robots. It assigns weapons or allies to targets or enemies in order to engage in combat effectively by reflecting the battlefield situation and combatant information. The target-based WTA is a type of WTA that aims to destroy as many enemies as possible[1]. Hence, it aims to minimize the total value of targets (enemies). It is suitable for determining the target of each ally when the information about combatants and the battlefield situation is known. The static WTA makes a single assignment considering only the current situation. It means that the information about combatants, battlefield situation, and assignments based on them must be updated iteratively to be used it in real-time.

The target-based static WTA algorithms have been developed in various ways inspired by nature[2-4]. Some WTA algorithms are used Particle Swarm Optimization (PSO) method, which is based on swarm behavior in nature[5-8]. It uses some solution candidates called particles to get an optimized solution in WTA. They move randomly towards local solution (optimal position of each particle) so far, and global solution (optimal position of all particles) so far. An optimal solution of WTA problem can be obtained after iterations if the number of particles and iterations is enough. The WTA has also developed based on behavior of water wave[9].

It uses several waves to get an optimized solution, and they move according to the 3 actions; propagation, refraction, and breaking. The Firefly Algorithm (FA) which is based on the behavior of firefly is also applied to WTA[10-11]. It assumes that attraction of firefly is proportional to its brightness. A firefly that does not have the best brightness will move affected by other fireflies that are brighter than the one. The firefly that has the best brightness is not affected by other fireflies, so it will move randomly to get the better solution. Other meta-heuristic algorithms such as ant colony algorithm[12-13] and genetic algorithm[14-16] are also used to solve WTA problems.

The reinforcement learning based approaches also can be applied to solve WTA problems[17]. The key point of reinforcement learning is to find solution by exploration and

Received : 22 November 2023, Revised : 30 March 2024
Accepted : 30 April 2024, Online published : 19 September 2024

exploitation, so it is powerful method if the formula for solving problems is hard to find. However, it requires a lot of time and CPU/GPUs to derives the optimal solution if the solution space is extremely large. Hence, other various approaches to get optimal solutions of WTA problems quickly are still being developed[18].

The reward of an assignment in above WTAs is set to the amount of reducing value of enemy which is calculated by multiplying the current value of the enemy by the probability of survival of the enemy after combat. The assignment reward has 2 problems. One is that the method to calculate battle probabilities, main factor in determining WTA performance, did not addressed. The probabilities must reflect the characteristics and deployment of enemies and allies in order for effective assignment. The other is that the reward did not consider the time it takes to have combat. Hence, inefficient case often occur in which allies are assigned to enemies at long distance.

In this paper, we propose a target-based static weapon-target assignment to solve these problems. We assume that each ally (or weapon) should be assigned only one enemy (or target) and several allies can be assigned same enemy. First, we introduce the formulation of general weapon-target assignment problem. The method for calculating battle probabilities, one of key factor affecting the performance of assignment, is proposed. The battle probabilities when the ally and enemy have one-to-one combat are calculated considering several hitting points and their area of each combatant. Then they are used to calculate probabilities for one-to-many, many-to-one, and many-to-many combat to calculate reward. Rewarding function of the algorithm is designed to decay over time to give more rewards to assignments that have less time it takes to have combat. To verify the proposed algorithm, some tank combat simulation is performed. The tank data from war-gaming website is used for realistic review.

The rest of this paper is organized as follows. Section 2 provides the formulation of general weapon-target assignment problems. Section 3 describes the proposed WTA algorithm including battle probability calculation and time-discounted rewarding schemes. Section 4 shows some combat simulation results using the proposed algorithm. The simulation results are analyzed whether they come out as intended. Finally, Section 5 provides some concluding remarks.

2. PROBLEM FORMULATION

In this paper, WTA algorithms for static target-based case is proposed. The purpose of target-based WTA is to minimize total value of survived enemies. The value of survived enemy can be calculated as multiplication of initial value of the enemy and survival probability of the enemy. The survival probability can be calculated by the product of each survival probability after combats with an ally who is assigned the enemy. So, the formulation of target-based WTA problem will be as follows,

$$min \sum_{b=1}^{N_e} \left(V_b \prod_{a=1}^{N_a} q_{ab}^{x_{ab}} \right) \tag{1}$$

where, N_e and N_a are number of enemies and allies respectively and V_b is the initial value of enemy b. The value of the enemy should be proportional to the power of the enemy to meet the purpose of the target-based WTA. q_{ab} is a survival probability of enemy b when ally a is assigned enemy b. It can be treated as maximization of total reduced value of the enemies as below.

$$max \sum_{b=1}^{N_e} \left(V_b \left(1 - \prod_{a=1}^{N_a} q_{ab}^{x_{ab}} \right) \right) \tag{2}$$

We assume that an ally should be assigned only one enemy. It can be formularized as below,

$$\sum_{b=1}^{N_e} x_{ab} = 1 \; \forall a \in \boldsymbol{A}, x_{ab} \in \{0, 1\} \tag{3}$$

where, x_{ab} is assignment flag that is 1 if ally a is assigned enemy b and 0 otherwise. A is a set of allies who are participated in the combat. In other words, it is a candidate set of allies for assignment.

3. PROPOSED ALGORITHM

In this Section, the proposed weapon-target assignment algorithm is described. As seen in Eqn (2), the survival probability is one of key factors for assignment. So, we first propose the battle probability calculation scheme which includes the survival probability.

3.1 Battle Probability Calculation Scheme

3.1.1 One-To-One Combat

To calculate the battle probability for one-to-one combat, a partial hit probability and partial destruction probability table is used. The partial hit probability shows the probability for hitting each hit point of the target entity and for not hitting all hit points of the target entity when an entity attacks the target entity. The probabilities are determined by the size of the target entity and the attack dispersion of the attacker. Note that the combatant, such as tanks or armored vehicles, has multiple hit points because protection capability for each part of the combatant is different.

The partial destruction probability shows the probability of destroying the target when an entity hits each hit point of the target entity. The probabilities are determined by attacker's attack power and armor penetration and target entity's armor of hitting point and health. Note that the sum of each row of partial hit probability should be 1. Also the partial destruction probability will be 0 if the entity does not hit the target. The information of all allies can be obtained by tele-communication and the information of all enemies can be obtained through tactical networks or surveillance and reconnaissance. Hence, the partial hit and partial destruction probabilities for allies and enemies can be made as in Table 1.

The battle probability for one-to-one combat is consisted as 4 case; ally-survived-enemy-survived, ally-survived-enemy-destroyed, ally-destroyed-enemy-survival, and ally-destroyed-enemy-destroyed. The target destruction probability for an ally or enemy can be calculated as the sum of the products of the partial hit probability and the partial destruction probability for each hit point as follows,

$$w_a^e = \sum_{n=1}^{N_{p,e}} p_a^{e,n} d_a^{e,n} \; \forall a \in \boldsymbol{A}, \forall e \in \boldsymbol{E} \tag{4}$$

Table 1. The table of partial hit and partial destruction probabilities

Target \ Hit point	1	2	⋯	Miss
A01	$p_{E01}^{A01,1}$	$p_{E01}^{A01,2}$	⋯	$p_{E01}^{A01,N}$
	$d_{E01}^{A01,1}$	$d_{E01}^{A01,2}$		0
A02	$p_{E01}^{A02,1}$	$p_{E01}^{A02,2}$	⋯	$p_{E01}^{A02,N}$
	$d_{E01}^{A02,1}$	$d_{E01}^{A02,2}$		0
A03	$p_{E01}^{A03,1}$	$p_{E01}^{A03,2}$	⋯	$p_{E01}^{A03,N}$
	$d_{E01}^{A03,1}$	$d_{E01}^{A03,2}$		0
⋮	⋮	⋮	⋮	⋮
E01	$p_{A01}^{E01,1}$	$p_{A01}^{E01,2}$	⋯	$p_{A01}^{E01,N}$
	$d_{A01}^{E01,1}$	$d_{A01}^{E01,2}$		0
E02	$p_{A01}^{E02,1}$	$p_{A01}^{E02,2}$	⋯	$p_{A01}^{E02,N}$
	$d_{A01}^{E02,1}$	$d_{A01}^{E02,2}$		0
E03	$p_{A01}^{E03,1}$	$p_{A01}^{E03,2}$	⋯	$p_{A01}^{E03,N}$
	$d_{A01}^{E03,1}$	$d_{A01}^{E03,2}$		0
⋮	⋮	⋮	⋮	⋮

$$w_e^a = \sum_{n=1}^{N_{p,a}} p_e^{a,n} d_e^{a,n} \ \forall a \in \boldsymbol{A}, \forall e \in \boldsymbol{E} \tag{5}$$

where, w_a^e is the target destruction probability when an ally a attacks an enemy e and w_e^a means vice versa. p and d are the partial hit and destruction probabilities as described in Table 1. N_p means the number of hit points including miss. Then, the 4 battle probabilities for one-to-one combat can be calculated as in Table 2. ASES, ASED, ADES, and ADED in Table 2 mean Ally-Survived-Enemy-Survived, Ally-Survived-Enemy-Destroyed, Ally-Destroyed-Enemy-Survived, and Ally-Destroyed-Enemy-Destroyed respectively. Note that the enemy used in Table 2 means the target enemy.

Table 2. Battle probabilities for 4 cases

Case	Probability
ASES	$(1-w_a^e)\times(1-w_e^a)$
ASED	$w_a^e \times (1-w_e^a)$
ADES	$(1-w_a^e)\times w_e^a$
ADED	$w_a^e \times w_e^a$

3.1.2 One-to-many combat

The one-to-many combat occurs if there are several enemies that can attack only an ally who aims a target enemy. In calculation of battle probability for one-to-one combat, probabilities of ally's and enemy's destruction are just w_e^a and w_a^e respectively since they only can be destroyed by each other. In the one-to-many combat, however, several enemies can destroy the ally while the target enemy can be destroyed only by the ally. So, the probabilities of ally's destruction should be changed as below,

$$w_{e_g}^a = w_{e_1}^a + (1-w_{e_1}^a)w_{e_2}^a + (1-w_{e_1}^a)(1-w_{e_2}^a)w_{e_3}^a + \cdots$$
$$= \sum_{m=1}^{N_{e_g}} \left(w_{e_m}^a \prod_{n=1}^{m-1} (1-w_{e_n}^a) \right) \tag{6}$$

where, N_{e_g} is the number of enemies who can destroy the ally and $\boldsymbol{e_g} = \{e_1, e_2, \cdots, e_{N_{e_g}}\}$ is a set of enemies who can attack the ally a. Then, 4 battle probabilities in one-to-many combat can be calculated as in Table 2 if, w_e^a is changed to $w_{e_g}^a$ in Eqn. (6).

3.1.3 Many-To-One Combat

The many-to-one combat occurs if an ally attacks a target enemy and there are several allies that a target enemy can attack. In calculation of battle probability for one-to-one combat, probabilities of ally's destruction are just w_e^a since the target enemy can attack only the ally. In the many-to-one combat, however, the target enemy can attack several allies because the allies are in attack range of the target enemy. So, the probabilities of ally's destruction should be reduced by $1/M$ times where, M means the number of allies that the target enemy can attack. Then, 4 battle probabilities in many-to-one combat can be calculated as in Table 2 if w_e^a is changed to w_e^a/M.

3.1.4 Many-To-Many Combat

The many-to-many combat occurs if there are several enemies that can attack not only an ally who is attacking a target enemy but also other allies. In one-to-many combat, there are several enemies that can only attack the ally who aims the target enemy. In many-to-many combat, however, the several enemies can attack other allies who are in the attack range. It means that the probability that they attack an ally who is attacking the target enemy will be reduced by $1/M$ times. The meaning of M extends to the number of allies that enemy who can attack the target-aimed ally can attack. Note that whether the enemies can attack other allies is determined based only on the current deployment. So the probabilities of ally's destruction should be changed as follows,

$$w_{e_g}^a = \frac{1}{M_1} w_{e_1}^a + \frac{1}{M_2}(1-w_{e_1}^a)w_{e_2}^a$$
$$+ \frac{1}{M_3}(1-w_{e_1}^a)(1-w_{e_2}^a)w_{e_3}^a + \cdots$$
$$= \sum_{m=1}^{N_{e_g}} \left(\frac{1}{M_m} w_{e_m}^a \prod_{n=1}^{m-1} (1-w_{e_n}^a) \right) \tag{7}$$

where, M_m is the number of allies that enemy m who can attack the target-aimed ally can attack. The probability of the target enemy's destruction is same as one-to-many case, so 4 battle probabilities in many-to-many combat can be calculated as in Table 2 if w_e^a is changed to $w_{e_g}^a$ in Eqn. (7).

3.2 Rewarding Scheme

To draw out the reward for sequential assignment, Eqn. 2 should be expended further as follows,

$$
\begin{aligned}
&max \sum_{b=1}^{N_e}\left(V_b\left(1-\prod_{a=1}^{N_a} q_{ab}^{x_{ab}}\right)\right) = V_1\left(1-q_{11}^{x_{11}}q_{21}^{x_{21}}\cdots\right)\\
&+V_2\left(1-q_{12}^{x_{12}}q_{22}^{x_{22}}\cdots\right)+\cdots\\
&=V_1\left(1-q_{11}^{x_{11}}\right)+V_1q_{11}^{x_{11}}\left(1-q_{21}^{x_{21}}\right)\ +\cdots\\
&\ +V_2\left(1-q_{12}^{x_{12}}\right)+V_2q_{12}^{x_{12}}\left(1-q_{22}^{x_{22}}\right)\ +\cdots
\end{aligned}
\tag{8}
$$

where q_{ab} can be calculated using the battle probabilities in Sections 3.1.1 to 3.1.3. V_b should be proportional to the power of the enemy to meet the purpose of the target-based WTA, so we set initial V_b as the multiplication of attack and health. Then, the reward can be set as follows,

$$
\left(D_bH_b\prod_{n=1}^{N_a} q_{nb}^{x_{nb}}\right)(1-q_{ab}),b\in \boldsymbol{E} \tag{9}
$$

where, D_b and H_b is the attack and health of the enemy b respectively. The reward also should be set to decay over time in order to take into account the time it takes to have combat. Hence, the reward for assignment is calculated as below,

$$
\left(D_bH_b\gamma^{t/\sigma}\prod_{n=1}^{N_a} q_{nb}^{x_{nb}}\right)(1-q_{ab}),b\in \boldsymbol{E} \tag{10}
$$

where, γ is attenuation rate which is $0<\gamma<1$ and t is the time it takes for ally a and enemy b to have combat in seconds. σ is time-scaling factor for effective assignment.

Algorithm 1. Sequential assignment algorithm.

```
calculate 1:1 battle probabilities
calculate hit-time of allies
identify combat types for allies
while (the assignment did not finished)
   renew values of enemies
   for m=1: N_a
      if find (x_(m,all) = = 1) = = none
         for n = 1:N_b
            calculate battle probabilities
            with respect to combat type
            r^(i)_(m,n) = reward as in Eqn. (10)
         end for
      else
         r^(i)_(m,all) = 0
      end if
   end for
   (a*, b*) = argmax_(m,n) r^(i)_(m,n),
   i is the number of iteration
   x(a*,b*) = 1
end while
```

3.3 Sequential Assignment Algorithm

To assign allies to enemies using proposed rewarding scheme in Eqn. (10), the sequential assignment algorithm should be used until all allies are assigned. Note that the values of enemies should be renewed before each assignment. The combat probabilities are calculated according to the combat type, and the probabilities are used to calculate rewards. The rewards for ally who is assigned already should be set to 0. After calculation of reward table, (ally, enemy) pair that has maximum reward is found and the ally in the pair is assigned enemy in the pair. The pseudocode for sequential assignment algorithm is described in Algorithm 1.

4. RESULT AND DISCUSSION

To verify the proposed WTA, tank simulation results for 5 cases are discussed in this section. The simulations use 4 tanks: T71 CMCD (light tank), Indien-Panzer (medium tank), M-IV-Y (heavy tank), and Tortoise (tank destroyer). The specifications of them include attack power, armor penetration, dispersion, maximum health power, armor of hull and turret, and etc., and they are brought from one of famous war-game sites[19].

In the simulations, six assumptions are applied for calculation of the partial hit and destruction probabilities.

Assumption 1. Tanks can only attack the front, rear, right and left sides of a tank because there is no armor information for the top and bottom sides of the tank.

Assumption 2. The aiming square is used instead of aiming circle to calculate partial hit probabilities easily. The length of a side of the square is set to twice the dispersion.

Assumption 3. The probability of a bullet being fired to any point within the aiming square is same for easy calculation.

Assumption 4. All tanks automatically aim at the turret or hull of the target tank. It will aim to the point with the highest probability of partial hit for each case.

Assumption 5. All tanks have a maximum range to attack, so they can attack only if the distance to the target tank is within the range.

Assumption 6. All tanks have shape described in Figure 1 where l, h, and w mean the total length, height, width of the tank respectively.

Then, the number of hit points for a tank are 7; front, side, rear of hull and turret of the tank and miss. The partial hit probabilities of front and rear sides of the tank are same because the area of front side is same with the area of rear side. The partial hit probabilities of left and right sides of the tank are also same for the same reason. Hence, the partial hit probability for front /rear and left/right sides of the tank can be calculated according to the area of aiming square and the area of the hull/turret of tank in the aiming square as described in Fig. 2-3.

To calculate rewards, the attenuation rate γ and the time-scaling factor σ in Eqn. (10) are set to 0.9 and 10, respectively. Also, the maximum range to attack is set to 500 m for all tanks.

4.1 Simulation Case 1

The simulation case 1 is for checking whether an ally is

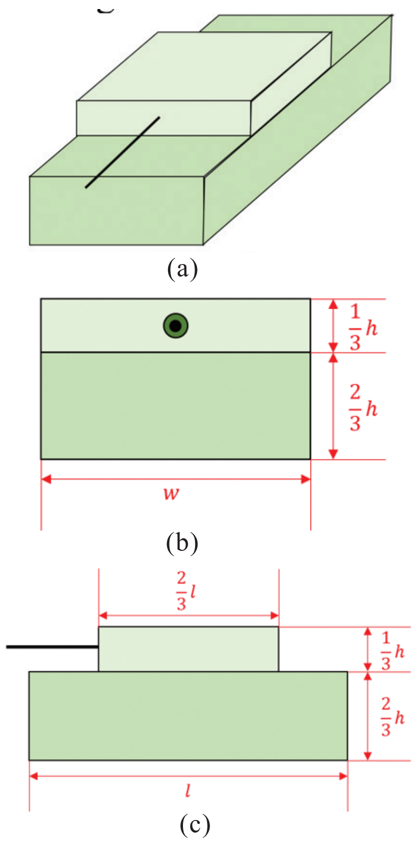


Figure 1. Shape of tank, (a) 3d view; (b) front view; and (c) side view.

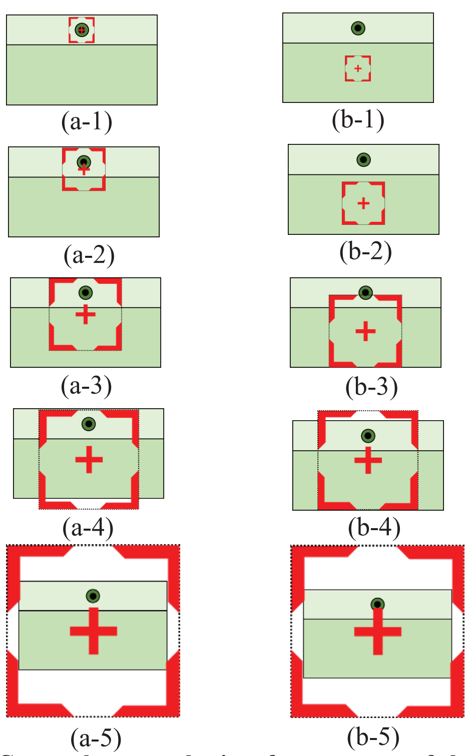


Figure 2. Cases when a tank aims front or rear of the other: (a) Aiming hull; and (b) Aiming turret.

assigned the enemy that has lower value when there are two enemy tanks with different power at the same distance from the ally. The enemies are set to a light tank and a tank destroyer which is extremely powerful than light tank, respectively. Then, the ally should be assigned a light tank enemy. The health power of the ally and the enemies are all set to maximum. The distance between the ally and each enemy is set to 1500 m and the distance between the enemies is too far to help each other.

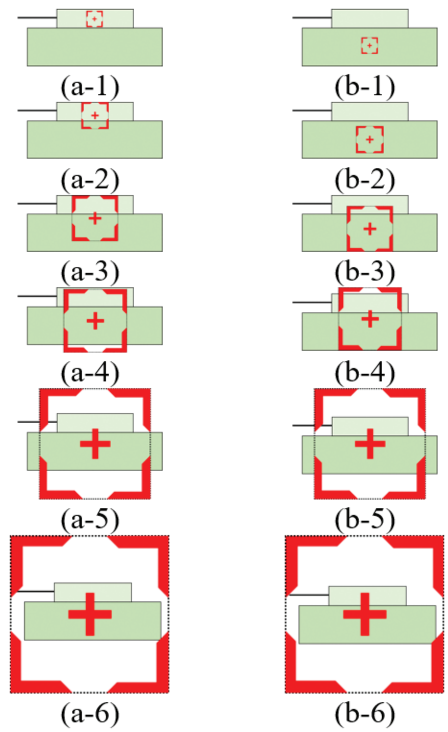


Figure 3. Cases when a tank aims left or right of the other: (a) Aiming hull; (b) Aiming turret.

4.2 Simulation Case 2

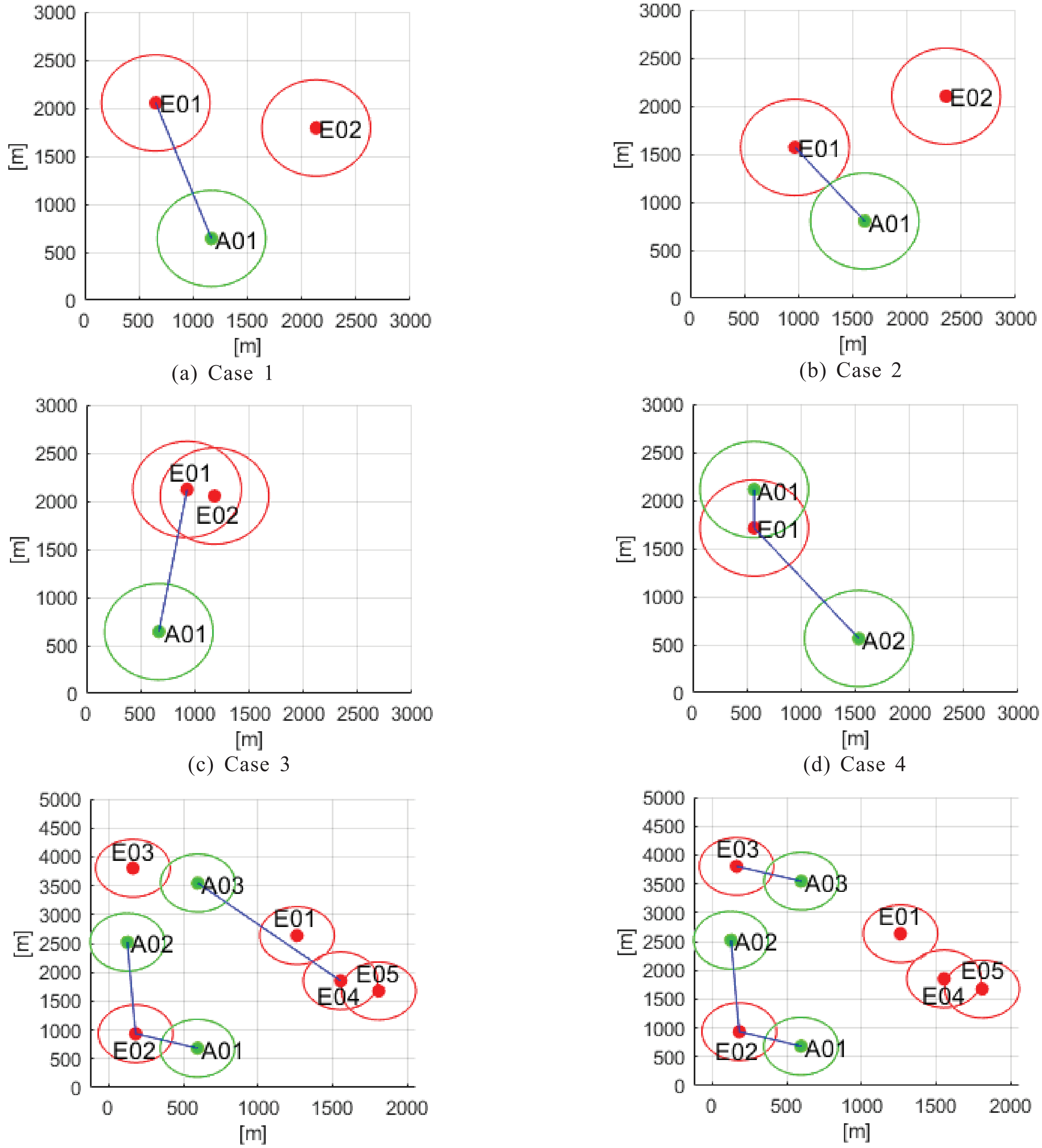
The simulation case 2 is for checking whether an ally is assigned the enemy that is closer to the ally when there are several enemies whose types of tank are same. The tank types are set to medium tank all, and the health power of the ally and the enemies are all set to maximum. The distance between the ally and each enemy is set to 1000 m and 1500 m respectively and the distance between the enemies is too far to help each other.

4.3 Simulation Case 3

The simulation case 3 is for checking whether the probability of survival from combat for an ally becomes lower as the number of enemies who can attack the ally increases. The environment of simulation is set as in simulation case 1 except the distance between each enemy. The distance between the enemies is set enough to help each other. The health power of the ally and the enemies are all set to maximum.

4.4 Simulation Case 4

The simulation case 4 is for checking whether the probability of destruction from combat for an ally becomes lower as the number of allies who can be attacked from the enemy increases. The enemy is set to medium tank and the



(a) Case 1 (b) Case 2

(c) Case 3 (d) Case 4

(e) Case 5-1 (f) Case 5-2

Figure 4. The assignment results for simulation cases.

allies are set to light and medium tank respectively. The light tank of the ally is placed in attack range of the enemy. The distance between the enemy and the light tank of the ally and the distance between the enemy and the medium tank of the ally are set to 400 and 1500 m, respectively. The health power of the ally and the enemies are all set to maximum.

4.5 Simulation Case 5

The simulation case 5 is for checking the impact of the proposed time-diminishing reward on assignment. So the assignment result derived from the proposed algorithm is compared with the result from sequential greedy algorithm where the reward does not decrease over time. The number of enemy and ally are set to 5 and 3 respectively and the type of them are set sequentially by repeating the order: light tank, medium tank, heavy tank, and tank destroyer. The position of them are set randomly and the health power of them are all set to maximum.

4.6 Simulation Results

The assignment results are depicted in Fig. 4. Figure 4(a) shows the assignment result for case 1. The ally is assigned E01, the light tank, as expected. Figure 4(b) shows the assignment result for case 2. The ally is assigned E01 who is closer than the other ally as expected.

Table 3. Battle probabilities for case 3

Attack \ Battle result	Win-Win	Win-Lose	Lose-Win	Lose-Lose
A01→ E01 (Case 1)	0.8497	0.0721	0.0721	0.0061
A01→ E02 (Case 1)	0.7579	0.0087	0.2307	0.0026
A01→ E01 (Case 3)	0.6514	0.0553	0.2704	0.0229
A01→ E02 (Case 3)	0.6986	0.0080	0.2900	0.0033

Table 4. Battle probabilities for case 4

Attack \ Battle result	Win-Win	Win-Lose	Lose-Win	Lose-Lose
A01→ E01 (1:1 Prob.)	0.7260	0.0669	0.1894	0.0175
A02→ E01 (1:1 Prob.)	0.7733	0.1061	0.1061	0.0146
A01→ E01 (2:1 Prob.)	0.8209	0.0757	0.0947	0.0087
A02→ E01 (2:1 Prob.)	0.8263	0.1134	0.0530	0.0073

Table 5. Enemies' total values for each case

Case \ Assign	Before	After
Case 1	656600.000	652422.691
Case 2	572000.000	547557.100
Case 3	656600.000	652422.691
Case 4	286000.000	247349.891
Case 5-1	1447200.000	1411259.021
Case 5-2	1447200.000	1401550.404

Figure 4(c) shows the assignment result for case 3 and the battle probabilities for case 3 are described in Table 3. The probabilities for Win-Win and Win-Lose when A01 is assigned E01 are 0.8497 and 0.0721 respectively in case 1 while 0.6514 and 0.0553 in case 3. Also, the probabilities for Win-Win and Win-Lose when A01 is assigned E02 are 0.7579 and 0.0087 respectively in case 1 while 0.6986 and 0.0080 in case 3. The table 3 shows that survival probabilities of the ally became lower and the destruction probabilities for the ally became higher for case 3 compared to case 1 as expected. Figure 4(d) shows the assignment result for case 4 and the battle probabilities for case 4 are described in Table 4.

The probabilities for Win-Win and Win-Lose when A01 is assigned E01 are 0.7260 and 0.0669 respectively if A01 and E01 have one-to-one combat while 0.8209 and 0.0757 in case 4. Also, the probabilities for Win-Win and Win-Lose when A02 is assigned E01 are 0.7733 and 0.1061 respectively if A01 and E01 have one-to-one combat while 0.8263 and 0.1134 in Case 4. Table 4 shows that survival probabilities of the ally became higher and the destruction probabilities for the ally became lower for case 4 compared to one-to-one combat probabilities as expected.

Figure 4(e) and Fig. 4(f) shows the assignment result for case 5. The difference between Fig. 4(e) and 4(f) is who is assigned to A03. In Fig. 4(e), A03 is assigned E04 because the reward of E04 is higher than E03 for A03 without considering the time it takes to have combat. However, the distance between A03 and E04 is too long so the result is inefficient. In Fig. 4(f), A03 is assigned E03 because the time it takes to have combat with A03 and E04 is too late. So the reward for E04 is diminished, and the reward for E03 becomes higher than E04. The assignment results of the proposed algorithm in Fig. 4 are quite reasonable and realistic.

The total values of enemies for each case are shown in Table 5. The enemies' total values are lowered after assignment in all cases as expected. Especially, it can be seen that the proposed algorithm that considers time reduces the total value of the enemies more than general sequential greedy that does not consider time algorithm in case 5.

5. CONCLUSIONS

In this paper, we propose a target-based static weapon-target assignment algorithm to solve the problems with existing WTA algorithms. One problem is that the method to calculate battle probabilities, main factor in determining WTA performance, did not addressed. To solve that, the partial hit and destruction probabilities are calculated based on the information of each combatant. Then, the battle probabilities when the ally and enemy have one-to-one combat are calculated based on the calculated partial hit and destruction probabilities. The battle probabilities are expanded to the probabilities for one-to-many, many-to-one, and many-to-many combat.

The other is that the reward did not consider the time it takes to have combat. So, the rewarding function for assignment is designed to decay over time to give more rewards to assignments that have less time it takes to have combat.

To verify the proposed algorithm, some tank combat simulation is performed. The tank data from war-gaming website is used for realistic review. The simulations are performed for 5 cases: (a) There are one ally and two enemies and all enemies have the same distance to the ally with different initial value. It is to verify that the ally is not assigned the extremely powerful enemy if the proposed algorithm is used. (b) There are one ally and two enemies and all enemies have same value with different distance to the ally. It is to verify that the ally is assigned to the closest enemy when the values of enemies are same if the proposed algorithm is used. (c) The combat probabilities for one-ally-two-enemy are compared to the probabilities for one-ally-one-enemy. It is to verify that the destruction probabilities of the enemies are decreased and the survival probabilities of the enemies are increased as the number of enemies who can attack the ally increases if the proposed algorithm is used. (d) The combat probabilities for two-ally-one-enemy are compared to the probabilities for one-ally-one-enemy. It is to verify that the destruction probabilities of the enemies are increased and the survival probabilities of the enemies are decreased as the number of allies who can be attacked by the target enemy if the proposed algorithm is used. (e) The assignment result derived by the proposed algorithm is compared with the result derived by sequential algorithm that the time-diminishing reward is not applied. It is to verify that the proposed algorithm considers the time it takes to have combat reasonably. The simulation results are analyzed in detail, and the performance of the proposed algorithm is verified.

There are some limitations in the proposed algorithm. Especially, the orientation angles of the allies and enemies are not considered when calculating the partial hit probability, so the partial hit probabilities for front and rear sides of each combatant are same even if two combatants face each other. It affects the battle probability calculation, so the assignment results may not be realistic. Hence, the learning-based object value evaluation and battle probability calculation considering the orientation angles of the allies, the aiming point, and etc. for assignment results more suitable to reality is our next goal of research.

REFERENCES

1. Murphey, R.A. Target-based weapon target assignment problems. *In* Nonlinear assignment problems, edited by P.M. Pardalos & L.S. Pitsoulis. Springer, Boston, WA, USA, 2000. pp. 39–53. doi:10.1007/978-1-4757-3155-2_3
2. Yang, Xin-She. Nature-inspired metaheuristic algorithms. Elsevier, United Kingdom, 2014. 258 p.
3. Ma, F.; Ni, M.; Gao, B. & Yu, Z. An efficient algorithm for the weapon target assignment problem. *In* Proceedings of IEEE International Conference on Information and Automation, IEEE, Lijiang, China, 2015. doi:10.1109/ICInfA.2015.7279633
4. Altinoz, O.T. Evolving model for synchronous weapon target assignment problem. *In* Proceedings of International Conference on Innovations in Intelligent Systems and Applications, IEEE, Kocaeli, Turkey, 2021. doi: 10.1109/INISTA52262.2021.9548606
5. Kong, L.; Wang, J. & Zhao, P. Solving dynamic weapon target assignment problem by an improved multiobjective particle swarm optimization algorithm. *Appl. Sci.,* 2021, **11**(19), 9254. doi: 10.3390/app11199254
6. Fu, G.; Wang, C.; Zhang, D.; Zhao, J. & Wang, H. A multiobjective particle swarm optimization algorithm based on multipopulation coevolution for weapon-target assignment. *Math. Probl. Eng.,* 2019, 1424590. doi: 10.1155/2019/1424590
7. Yang, L.; Zhai, Z.; Li, Y. & Huang, Y. A multi-information particle swarm optimization algorithm for weapon target assignment of multiple kill vehicle. *In* Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, IEEE, Auckland, New Zealand, 2018. doi: 10.1109/AIM.2018.8452418
8. Haoran, Z.; Weihong, W.; Qingze, L. & Wei, Z. Weapon-target assignment based on improved PSO algorithm. *In* Proceedings of 33rd Chinese Control and Decision Conference, IEEE, Kunming, China, 2021. doi:10.1109/CCDC52312.2021.9601736
9. Zheng, Y.J. Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* 2015, **55**, 1-11. doi: 10.1016/j.cor.2014.10.008
10. Yang, X.S. Firefly algorithms for multimodal optimization. *In* Proceedings of 5th International Symposium on Stochastic Algorithms: Foundations and Applications, Springer, Heidelberg, Berlin, 2009. doi: 10.1007/978-3-642-04944-6_14
11. Yang, X.S. & He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* 2013, **1**(1), 36-50. doi: 10.1504/IJSI.2013.055801
12. Hu, X.; Luo, P.; Zhang, X. & Wang, J. Improved ant colony optimization for weapon-target assignment. *Math. Probl. Eng.* 2018, **2018**, 6481635. doi: 10.1155/2018/6481635
13. Yanxia, W.; Longjun, Q.; Zhi, G. & Lifeng, M. Weapon target assignment problem satisfying expected damage probabilities based on ant colony algorithm. *J. Syst. Eng. Electron.* 2008, **19**(5), 939-44. doi: 10.1016/S1004-4132(08)60179-6
14. Lee, Z.J. & Lee, W.L. A Hybrid search algorithm of ant colony optimization and genetic algorithm applied to weapon-target assignment problems. *In* Proceedings of 4th International Conference on Intelligent Data Engineering and Automated Learning, Springer, Hong Kong, China, 2003. doi: 10.1007/978-3-540-45080-1_37
15. Zhang, K.; Zhou, D.; Yang, Z.; Pan, Q. & Kong, W. Constrained multi-object weapon target assignment for area targets by efficient evolutionary algorithm. *IEEE Access*, 2019, **7**, 176339-60. doi: 10.1109/ACCESS.2019.2955482
16. Shang, G.; Zaiyue, Z.; Xiaoru, Z. & Cungen, C. Immune Genetic Algorithm for Weapon-Target Assignment Problem. *In* Proceedings of Workshop on Intelligent Information Technology Application, IEEE, Zhangjiajie, China, 2007. doi: 10.1109/IITA.2007.24
17. Shuai, L.; Xiaoyuan, H.; Xiao, X.; Tan, Z.; Chenye, S. & Jiabao, L. Weapon-target assignment strategy in joint combat decision-making based on multi-head deep reinforcement learning. *IEEE Access*, 2023, **11**, 113740-51. doi: 10.1109/ACCESS.2023.3324193
18. Xiaojian, Y.; Huiyang, Y. & Tao, X. Solving multi-objective weapon-target assignment considering reliability by improved MOEA/D-AM2M. *Neurocomputing*, 2024, **563**(1), 126906. doi:10.1016/j.neucom.2023.126906
19. World of Tanks. Available online: https://worldoftanks.com/tankopedia (Accessed on 27 October 2023).

ACKNOWLEDGEMENTS

This work was supported by Korea Research Institute for defense Technology planning and advancement (KRIT) grant funded by the Korea government (DAPA (Defense Acquisition Program Administration)) (No. KRIT-CT-22-068, Intelligent Tactical C2 for MUM-T, 2022).

CONTRIBUTORS

Mr Nam Eung H. obtained the M.S. degree in the Electrical and Electronic Engineering from Yonsei University, Seoul, Korea and currently, he is working at Hanwha systems Co., Seongnam, Korea. His research interests include: Nonlinear control, fault tolerant control, satellite navigation system, and MUM-T (Manned-Unmanned Teaming).

Contribution to the current study: He conceptualised technical idea, developed mathematical model, built software, carried out the results of simulations, and prepared manuscript.

Mr Hyung Jun K. obtained the M.S. degree in the Information and Electronic Engineering from Ajou University, Suwon, Korea and working at Hanwha systems Co., Seongnam, Korea. His research interests include: AI techniques for MUM-T (Manned-Unmanned Teaming).

Contribution to the current study: He provided feedback for the applicability of the weapon-target assignment technique to ground control systems. He also helped in revising the manuscript.