

Research Article

Decentralized Algorithms for Weapon-Target Assignment in Swarming Combat System

Peng Zhao ¹, Jianzhong Wang ^{1,2} and Lingren Kong¹

¹School of Mechatronic Engineering, Beijing Institute of Technology, Beijing 100081, China

²State Key Laboratory of Explosion Science and Technology, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Jianzhong Wang; cwjzwang@bit.edu.cn

Received 8 April 2019; Revised 14 July 2019; Accepted 5 August 2019; Published 27 August 2019

Academic Editor: Francisco J. Montáns

Copyright © 2019 Peng Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Swarming small unmanned aerial or ground vehicles (UAVs or UGVs) have attracted the attention of worldwide military powers as weapons, and the weapon-target assignment (WTA) problem is extremely significant for swarming combat. The problem involves assigning weapons to targets in a decentralized manner such that the total damage effect of targets is maximized while considering the nonlinear cumulative damage effect. Two improved optimization algorithms are presented in the study. One is the redesigned auction-based algorithm in which the bidding rules are properly modified such that the auction-based algorithm is applied for the first time to solve a nonlinear WTA problem. The other one is the improved task swap algorithm that eliminates the restriction in which the weights of the edges on graph G must be positive. Computational results for up to 120 weapons and 110 targets indicate that the redesigned auction-based algorithm yields an average improvement of 37% over the conventional auction-based algorithm in terms of solution quality while the additional running time is negligible. The improved task swap algorithm and the other two popular task swap algorithms almost achieve the same optimal value, while the average time-savings of the proposed algorithm correspond to 53% and 74% when compared to the other two popular task swap algorithms. Furthermore, the hybrid algorithm that combines the above two improved algorithms is examined. Simulations indicate that the hybrid algorithm exhibits superiority in terms of solution quality and time consumption over separately implementing the aforementioned two improved algorithms.

1. Introduction

WTA plays an increasing role given the growing requirement for the collaborative engagement of several weapons relative to several targets in modern battlefields [1]. Various extant studies focused on solutions for WTA that are mainly applied in the traditional high-value weapon platforms such as air defense WTA [2–4] or ship target strikes [5, 6]. Compared to expensive equipment, swarming UAVs or UGVs attract the attention of military powers worldwide because of advantages including strong autonomy, low cost, high flexibility, and fast upgrades [7]. For example, the United States of America uses the advantages of a large quantity of low-cost UAVs to synergistically destroy a few key enemy targets [8, 9]. Similarly, WTA is of great significance in the swarming combat system. Given the

large-scale and decentralized autonomous framework of the swarming combat system, it is necessary to develop more efficient WTA algorithms to maximize the total damage value of enemy targets in a decentralized manner.

Several centralized algorithms to solve WTA problems are proposed based on the research results of operations research and example include the heuristic algorithm [10], memetic algorithm [11], genetic algorithm [12, 13], particle swarm optimization algorithm [14], bee colony algorithm [15], and game theoretic strategy [16]. A centralized algorithm efficiently obtains an optimal solution if the number of weapons and targets is low. However, in a large system, a centralized algorithm can result in heavy workload with respect to computation/communication because the central controller is typically responsible for collecting, processing, and disseminating information for the entire system.

Additionally, the heavy workload can result in low efficiency for the entire system. Furthermore, the failure of the central controller results in the failure of the entire system [17].

In order to cope with the large-scale system and enhance reliability, several distributed algorithms are presented in the research areas of multirobot task assignment [18, 19]. However, damage probabilities of weapons to targets are not additive. The expression for the WTA problem does not typically correspond to a linear sum equation, and thus, it is difficult to immediately apply the distributed algorithms in the areas of multirobot task assignment. In this paper, we introduce two new methods derived from the auction-based algorithm and task swap-based algorithm to optimize the assignment of weapons to targets, given the cumulative effect.

The auction method is explained in terms of economic competition concepts in human's economic activities [20]. In [21–23], the conflict-free assignment solution is achieved by introducing auxiliary variable termed as task price, and making robots bid a price for tasks via an iterative auction mechanism. The algorithms presented in [24, 25] adopt the auction-based framework in a manner similar to [21–23] and extend the algorithms to problems with more constraints on the tasks, such as adding task deadline constraints or group constraints. However, the extension of the auction-based algorithm to solve the nonlinear WTA problem has rarely been studied. The WTA problem in [1] is similar to our version of the problem where the cumulative damage effect is considered. However, the method in [1] includes the prepared assignment guidance table, which needs a central auctioneer and is not suitable for solving the distributed assignment problem introduced in this paper. This paper further develops the auction-based algorithm, adds a judgment criterion for bidding, and uses the maximum consensus technique rationally. The results show that the solution quality of the redesigned auction method significantly improves while the extra running time is negligible.

This paper uses the task swap method to refine the assignment solution after the auction method. The task swap method is applied well to solve the one-to-one mapping assignment problem where each weapon is assigned with a unique target, and an optimal assignment solution is obtained when it runs to completion. Additionally, the task swap method can start from any initial assignment such that it is easy to refine the suboptimal assignment computed via other methods [26]. Significant attention has been focused on the task swap method such as exchanging tasks between pairs of robots [27] or among two or more robots [28]. Recently, some studies [29, 30] proposed a task swap-based method that combined the duality theory with decentralized task allocation approaches. However, despite its decentralized form, the method assumes that global communication exists at each stage when the algorithm is running. Based on the study in [29], a fully decentralized task swap-based method that combines duality theory with Dijkstra's algorithm in graph theory was proposed [17], and the method is suitable for scenarios in which robots communicate only with neighbors in their vicinity. However, in [17], the search for a task swap requires satisfying the condition that the

weight value of the edge on the graph G is positive, which decreases the searching scope and extends the searching time. In this paper, we retrofit the method to search for a task swap loop. The weight value of the edge and loop in graph G has no restrictions, which accelerates the optimization process. The simulation results show that the convergence rate of the improved task swap method exceeds that of other task swap-based methods.

Finally, we analyze a hybrid algorithm that combines the redesigned auction-based algorithm and improved task swap algorithm. That is, the redesigned auction-based algorithm is applied to obtain an assignment solution where a reasonable number of suitable weapons are assigned to targets, and the improved task swap algorithm is then applied to refine the assignment solution after the auction-based algorithm. The simulation results indicate that when compared with implementing one of the algorithms separately, the hybrid algorithm exhibits better performance in solution quality, and the convergence rate is not lower than that of the redesigned auction-based algorithm or improved task swap algorithm.

This paper is organized as follows: In Section 2, we describe the background of this paper and model the WTA problem. In Sections 3 and 4, we present a redesigned auction-based algorithm and an improved task swap-based algorithm, respectively. In Section 5, we present the simulation and analyze the results. In Section 6, we summarize the conclusion and outline future research studies.

2. Weapon-Target Assignment Model

The WTA problem in this paper involves determining a proper assignment of weapons to targets in a decentralized manner with the objective of maximizing damage effect of enemy targets. A simple scenario of the weapons used against the targets is shown in Figure 1. The scenario exists in applications of intelligent minefield, i.e., intelligent mines with limited ability of reconnaissance, communication, and movement lurk in the battlefield, and they cooperatively attack the enemy targets in the area. We assume that the weapons in this paper are small unmanned ground vehicles (UGVs) carrying a gun or explosive device, and the targets are soldiers or military vehicles.

We assume that there are M weapons $\mathbf{W} = \{W_1, \dots, W_M\}$ and N targets $\mathbf{T} = \{T_1, \dots, T_N\}$. The solution of the WTA problem is based on the following assumptions.

Assumption 1. The capability of the weapons is limited, and thus, the destruction of a target typically requires more than one weapon. The number of the weapons exceeds that of the targets, i.e., $M > N$. Each weapon is assigned to exactly one target, and each target is attacked by one or more weapons.

Assumption 2. The problem in this paper is related to static weapon-target assignment (SWTA), i.e., all the weapons attack the targets in a single stage.

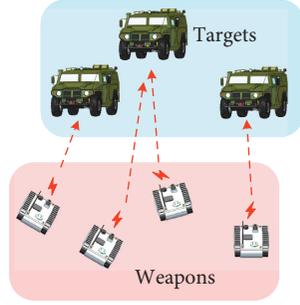


FIGURE 1: Illustration of WTA.

Assumption 3. Each weapon W_i knows a few basic attributes of the targets, and W_i maintains local information about the damage probability P_{ij} of each target T_j attacked by W_i . P_{ij} reflects the likelihood that target T_j is destroyed by weapon W_i . Given a certain type of target, the damage probability corresponds to a function of relative distance or relative angle between it and a weapon.

Assumption 4. Each weapon is homogenous, and the value val_j of each target is identical. Without loss of generality, we assume that $\text{val}_j = 1$.

The objective is to assign all weapons to the targets such that the total damage value of the targets is maximized, and this is formally stated as follows:

$$\max f(x) = \sum_{j=1}^N \text{val}_j \left[1 - \prod_{i=1}^M (1 - P_{ij})^{x_{ij}} \right], \quad (1)$$

subject to

$$\sum_{j=1}^N x_{ij} = 1, \quad i = 1, 2, \dots, M, \quad (2)$$

$$\sum_{i=1}^M x_{ij} \geq 1, \quad j = 1, 2, \dots, N, \quad (3)$$

$$x_{ij} = \{0, 1\}, \quad \forall i, j, \quad (4)$$

where x_{ij} is a binary variable that indicates whether W_i is assigned to T_j . If W_i is assigned to T_j , then $x_{ij} = 1$, otherwise, $x_{ij} = 0$. The constraint (2) guarantees that a weapon is assigned to exactly one target at a time. The constraint (3) states that a target is attacked by at least one weapon. It should be noted that the damage probability in objective function (1) is not additive. Additionally, conventional decentralized algorithms, such as an auction-based algorithm or task swap algorithm, are used to solve the linear sum assignment problem. It is necessary to develop new decentralized algorithms to solve the WTA problem.

3. Auction-Based Algorithm Design

In this section, the redesigned auction-based algorithm is proposed to solve the WTA problem in a distributed manner. The auction-based algorithm operates in a manner

similar to a real auction where individuals bid for items by raising their prices via competitive bidding. The total payoffs are almost maximized by individuals iteratively maximizing their own payoffs.

3.1. Conventional Auction-Based Algorithm. The conventional auction-based algorithm is introduced to solve the classic linear sum assignment problem with the one-to-one assignment constraint in this subsection. We assume that the price for target T_j at iteration τ is $p_j(\tau) \geq 0$, and this is identical to the auxiliary variables termed as task price in [21, 25]. The payoff for weapon W_i relative to target T_j is u_{ij} . We assume that u_{ij} does not change during the iterations because the basic attributes of the weapons and targets change slightly during the assignment. The revenue for weapon W_i used against target T_j at iteration τ is $v_j^i(\tau) = u_{ij} - p_j(\tau)$. Each weapon bids price $p_j(\tau)$ for the target with the highest revenue. With respect to weapon W_i , if another weapon has bid higher price than that of weapon W_i 's previous bid for target $T_{k^*(\tau-1)}$, then the revenue of $T_{k^*(\tau-1)}$ changes and is potentially not the highest, and the new desired target $T_{k^*(\tau)}$ with the highest revenue should be selected. Subsequently, weapon W_i makes a new bid for target $T_{k^*(\tau)}$ and increases the price of $T_{k^*(\tau)}$. Therefore, the iterative biddings gradually maximize total payoff $\sum u_{ij}$ while the one-to-one assignment constraint is satisfied.

The assignment result obtained by the auction-based algorithm is equivalent to an economic equilibrium result, i.e., each individual is satisfied with the goods obtained and no longer possesses an incentive to bid for other goods.

3.2. Redesigned Auction-Based Algorithm. The conventional auction-based algorithm provides an almost optimal solution for the linear sum assignment problem. However, the WTA problem is always formulated as the nonlinear programming model as shown in equation (1). Thus, the algorithm does not work well in the WTA problem. Furthermore, the conventional auction-based algorithm is designed for the problem wherein each target is exclusively assigned to a unique weapon. The method of adding virtual targets or weapons can be used to solve the assignment problem with unequal number of weapons and targets. In contrast to the popular method, we propose a method of adding a criterion to verify whether the damage value is improved. The solution is near optimal for the WTA problem while the constraint of the number of weapons on each target is satisfied.

3.2.1. Price Update Rules. The objective of the WTA problem corresponds to the total damage of the targets, $\sum_{j=1}^N \text{val}_j \times P_j$ as shown in equation (1), where $P_j = 1 - \prod_{i=1}^M (1 - P_{ij})^{x_{ij}}$. We assume that $\text{val}_j = 1$ in assumption 4. If the targets are attacked with the higher damage probability by the weapons, then the total damage of targets $\sum_{j=1}^N P_j$ should be maximized.

We set the damage probability P_{ij} to be the payoff u_{ij} , i.e., $u_{ij} = P_{ij}$. In the auction-based algorithm, each weapon is

an economic agent acting in its own best interest. With respect to weapon W_i , let k^* denote the index of the target with the highest revenue:

$$k^* = \arg \max v_j^i = \arg \max (u_{ij} - p_j), \quad \forall T_j \in \mathbf{T}_k, \quad (5)$$

where \mathbf{T}_k denotes the set of the candidate targets. The price p_j is an auxiliary variable and is used to resolve the WTA problem to satisfy the constraint (3). When the algorithm terminates, the quality of assignment solution does not depend on p_j , i.e., the output assignment solution is almost optimal in terms of payoff u_{ij} instead of the revenue $v_j^i = u_{ij} - p_j$ [24].

Let k_Δ^* denote the index of the target with the second highest revenue and is expressed as follows:

$$k_\Delta^* = \arg \max_{j \in \mathbf{T}_k \setminus \mathbf{T}_{k^*}} (u_{ij} - p_j). \quad (6)$$

In the auction-based algorithm, weapon W_i increases the price and bids for its desired target. By bidding iteratively, the auction-based algorithm gradually ensures that every weapon is assigned to its desired target with the almost highest payoff u_{ij} . At iteration τ , if required, the price is updated based on a certain rule as follows:

$$\begin{aligned} p_{k^*}(\tau) &= p_{k^*}(\tau - 1) + \left[(u_{ik^*} - p_{k^*}(\tau - 1)) \right. \\ &\quad \left. - (u_{ik_\Delta^*} - p_{k_\Delta^*}(\tau - 1)) \right] + \varepsilon \\ &= p_{k^*}(\tau - 1) + (v_{k^*}(\tau - 1) - v_{k_\Delta^*}(\tau - 1)) + \varepsilon. \end{aligned} \quad (7)$$

In equation (7), the new price of the target increases by at least ε . Based on the original idea in an auction-based algorithm, the parameter is constant, and this avoids an endless bidding cycle when two or more weapons repeatedly bid for the same target. Instead of avoiding the algorithm from cycling, we use the parameter to control the rate of convergence because related studies indicate that the higher the ε , the faster the auction-based algorithm converges. In this paper, we define $\varepsilon = 0.5 \times u_{ik^*}$.

3.2.2. Bidding Criterion. The WTA and linear sum assignment problem have different objective functions. When the sum of the payoffs is maximized, the cumulative damage probability of the targets is potentially not maximized. Thus, extending from the conventional auction-based algorithm, we integrate the additional criterion to verify whether the total damage value is improved. In terms of W_i , when some other weapon W_l exhibits a higher price for W_i 's previous desired target, then this implies that weapon W_l is more suitable to attack the target. At this time, weapon W_i must select a new target to bid. In the implementation, (1) we select the target with the highest revenue based on equation (5) from the candidate target set \mathbf{T}_k . (2) We calculate the new value of the objective function and verify whether the new value is improved when compared with the previous value, if true, i.e.,

$$f_i(\tau - 1) < f_i(\tau), \quad (8)$$

then the new desired target is selected, wherein f_i denotes the value calculated by using equation (1). Otherwise, we repeat procedures (1) and (2) to continue to determine a new target from the remaining targets until the criterion is true or the candidate target set is empty. The implementation is shown in lines 8 to 23 in Algorithm 1.

3.2.3. Maximum Consensus Technique. Specifically, each weapon must know the current price for each target and assignment information on each weapon-to-target pair during the bidding procedure. In other words, the assignment information among all weapons must be consensus. A maximum consensus technique [23] is used to satisfy the aforesaid requirement. The idea is that at each iteration τ , each weapon W_i locally maintains and updates a list of current highest price $p_j^i(\tau)$ for each target by communicating with its neighbors, $W_l \in \mathcal{N}_{W_i} \cup \{W_i\}$, and uses the highest price among its neighbors as the local price of the targets. This is expressed as follows:

$$p_j^i(\tau) = \max_{W_l \in \mathcal{N}_{W_i} \cup \{W_i\}} p_j^l(\tau - 1). \quad (9)$$

The reason for obtaining each other's assignment information is to calculate the value of the objective function such that the weapon can use the value to verify whether the solution is improved. Similarly, the same technique is applied such that the assignment information is consensus among all weapons.

Although the objective function value should be identical, different weapons have different values due to the paucity of each other's recent assignment information in the distributed implementation. Let $f_m(\tau)$ be the symbol of the local objective function value calculated by each weapon W_m at current iteration τ . For further explanation of the symbol $f_m^i(\tau)$, the superscript denotes that weapon W_i receives the message $f_m(\tau)$ from weapon W_m . With respect to weapon W_i , $f_m^i(\tau)$, $\forall m \in [1, M]$ is maintained and updated by communicating with its neighbors, and it uses the highest objective function value among its neighbors as the local value. This is expressed as follows:

$$f_m^i(\tau) = \max_{W_l \in \mathcal{N}_{W_i} \cup \{W_i\}} f_m^l(\tau - 1), \quad (10)$$

let $\alpha_m(\tau)$ denote the index of the target attacked by weapon W_m at iteration τ . Each weapon W_i locally maintains and updates the lists of the desired targets' index, $\alpha_m^i(\tau)$, and the corresponding payoff $u_{m\alpha_m^i(\tau)}^i$ for all the weapons. The superscript i denotes that the lists are maintained by weapon W_i .

For the sake of explanation, we define a message for bidding in the following format: $\text{Message}^i(\tau) = \{i, \mathbf{P}, \mathbf{F}, \mathbf{a}, \mathbf{u}\}$, where $\mathbf{P} = [p_1^i(\tau), \dots, p_N^i(\tau)]$, $\mathbf{F} = [f_1^i(\tau), \dots, f_M^i(\tau)]$, $\mathbf{a} = [\alpha_1^i(\tau), \dots, \alpha_M^i(\tau)]$, and $\mathbf{u} = [u_{1\alpha_1^i(\tau)}^i, \dots, u_{M\alpha_M^i(\tau)}^i]$. Each weapon W_i locally maintains and updates the value of $\alpha_m^i(\tau)$ and $u_{m\alpha_m^i(\tau)}^i$ from messages with the highest objective function value among its neighbors, i.e.,

$$\{\alpha_m^i(\tau), u_{m\alpha_m^i(\tau)}^i\} = \arg \max_{W_l \in \mathcal{N}_{W_i} \cup \{W_i\}} f_m^l(\tau - 1). \quad (11)$$

Input: $u_{ij}, p_j^i(\tau-1), \forall j \in [1, N]; k^*(\tau-1); f_m^i(\tau-1), \alpha_m^i(\tau-1), u_{m\alpha_m(\tau-1)}, \forall m \in [1, M];$
Output: $p_j^i(\tau), k^*(\tau), f_m^i(\tau), \alpha_m^i(\tau), u_{m\alpha_m(\tau)}$
(1) //Update the local assignment information among its neighbors \mathcal{N}_{W_i} based on the maximum consensus technique.
(2) $p_j^i(\tau) = \max_{W_l \in \mathcal{N}_{W_i} \cup \{W_i\}} p_j^l(\tau-1);$
(3) $f_m^i(\tau) = \max_{W_l \in \mathcal{N}_{W_i} \cup \{W_i\}} f_m^l(\tau-1);$
(4) $\{\alpha_m^i(\tau-1), u_{m\alpha_m(\tau-1)}\} = \arg \max_{W_l \in \mathcal{N}_{W_i} \cup \{W_i\}} f_m^l(\tau-1);$ //The value of α_m and $u_{m\alpha_m}$ denotes each weapon-to-target assignment pair $\langle W_m, T_{\alpha_m} \rangle$ and damage probability of the assignment, respectively. The information is used to calculate the value of the objective function in Step 14.
(5) $\alpha_m^i(\tau) = \alpha_m^i(\tau-1);$
(6) $u_{m\alpha_m(\tau)} = u_{m\alpha_m(\tau-1)};$
(7) //Weapon W_i selects a new desired target and updates the price if another weapon exhibits a bid higher than W_i 's previous price.
(8) **If** $p_{k^*}^i(\tau-1) < p_{k^*}^i(\tau)$ **or** $f_m^i(\tau-1) = 0$ **then**
(9) Initialize the desired targets set $\mathbf{T}_k = \{T_1, \dots, T_N\};$
(10) $v_j^i = u_{ij} - p_j^i(\tau);$
(11) $k^*(\tau) = \arg \max v_j^i, \forall j \in \mathbf{T}_k;$
(12) $k_{\Delta}^* = \arg \max_{j \in \mathbf{T}_k / T_{k^*(\tau)}} v_j^i;$
(13) $\varepsilon = 0.5 \times u_{ik^*};$
(14) The new value of the objective function, $f_i^i(\tau)$, is obtained by using equation (1);
(15) **If** $f_i^i(\tau-1) < f_i^i(\tau)$ **or** $p_{k^*}^i(\tau) = 0$ **then**
(16) $p_{k^*}^i(\tau) = p_{k^*}^i(\tau) + v_{k^*}^i(\tau) - v_{k_{\Delta}^*}^i(\tau) + \varepsilon;$ //Update price
(17) Assign new value to the variables, i.e., $f_i^i(\tau) = f_i^i(\tau), \alpha_i^i(\tau) = k^*(\tau), u_{i\alpha_i(\tau)} = u_{ik^*(\tau)};$
(18) **Else**
(19) $\mathbf{T}_k = \mathbf{T}_k / T_{k^*(\tau)};$
(20) Retain the previous price and assignment information;
(21) **If** $\mathbf{T}_k \neq \emptyset$ **then** go to step 11;
(22) **End if**
(23) **End if**

ALGORITHM 1: The redesigned auction-based algorithm AB1.

Although the price and assignment information may not reach all weapons immediately, accurate information is eventually propagated to all weapons in the connected network.

A single iteration of the redesigned auction-based algorithm for weapon W_i is described in Algorithm 1. In fact, the weapons can run Algorithm 1 simultaneously or asynchronously. Algorithm 1 can be summarized as follows:

- (1) Weapon W_i receives bidding Message^l from its neighbors and updates its own local information such as $\mathbf{P}, \mathbf{F}, \boldsymbol{\alpha}, \mathbf{u}$ (see lines 1 to 6).
- (2) Whether the price $p_{k^*}^i(\tau-1)$ of the previous assigned target has been increased by other weapons is verified. If true, then the weapon W_i should select a new desired target based on the additional criterion as previously mentioned. If there exists a new desired target, then its price and the corresponding assignment information are updated, and a new bidding message is then broadcast (see lines 7 to 23).

Algorithm 1 terminates when all weapons are satisfied for their desired targets, i.e., the price \mathbf{P} does not change for $\Delta \leq M$ iterations, where Δ is the weapons' network diameter that indicates the maximal hop of the network.

3.3. Performance Analysis. Algorithm 1 for all weapons terminates in a finite number of iterations with an assignment that satisfies the constraint wherein each target is

attacked by at least one weapon. The time complexity of the redesigned auction-based algorithm (Algorithm 1) is $O(\Delta(N^3 + MN)((\max\{u_{ij}\} - \min\{u_{ij}\})/\varepsilon))$, where $O((N^2 + M))$ denotes the running time of Algorithm 1 for each weapon and $N((\max\{u_{ij}\} - \min\{u_{ij}\})/\varepsilon)$ denotes the maximum number of iterations for all weapons to run Algorithm 1.

In practice, the results show that the redesigned auction-based algorithm generally works better than the conventional auction-based algorithm. The main reasons are as follows.

In order to solve the WTA problem via the conventional auction-based algorithm, the objective function as shown in equation (1) is simplified as equation (12) because the conventional auction-based algorithm is aimed at solving the linear sum assignment problem.

$$\max \sum_{j=1}^N \sum_{i=1}^M x_{ij} P_{ij}, \quad (12)$$

where x_{ij} is subject to the constraints (2)–(4). For ease of exposition, we define $P_{sij} = 1 - P_{ij}$ as the survival probability of the target T_j attacked by weapon W_i . Equation (12) is equivalent to the below equation:

$$\min \sum_{j=1}^N \sum_{i=1}^M x_{ij} P_{sij}. \quad (13)$$

Given the nonlinear cumulative effect, we propose the redesigned auction-based algorithm to solve the

nonlinear WTA problem. The objective function (1) is equivalent to minimize the survival probability of the targets as follows:

$$\max \sum_{j=1}^N \left[1 - \prod_{i=1}^M (1 - P_{ij})^{x_{ij}} \right] = \min \sum_{j=1}^N \prod_{i=1}^M P_{sij}^{x_{ij}} \quad (14)$$

Based on the arithmetic mean-geometric mean (AM-GM) inequality, for all nonnegative real numbers a_1, a_2, \dots, a_n , we have $((a_1 + a_2 + \dots + a_n)/n) \geq \sqrt[n]{a_1 a_2 \dots a_n}$, equality holds if and only if $a_1 = a_2 = \dots = a_n$. Extending from the AM-GM inequality, we assume that there exists a set of nonnegative real numbers b_1, b_2, \dots, b_n , and $\sum_{k=1}^n a_k = \sum_{k=1}^n b_k = \text{const}$, and if the variance of the numbers $\{b_k\}$ is higher than that of $\{a_k\}$, i.e., $\text{var}(b_k) > \text{var}(a_k)$, then $\prod_{k=1}^n b_k < \prod_{k=1}^n a_k$.

Similarly, we observe equations (13) and (14) and assume that there exists an optimal solution $x_{ij}^* \in \{0, 1\}$, $\forall i, j$ for the linear summation (13); however, the solution $x_{ij}^* \in \{0, 1\}$, $\forall i, j$ is not optimal for the objective function (14), namely, $\sum_{i=1}^M x_{ij}^* P_{sij} \leq \sum_{i=1}^M x_{ij} P_{sij}$, but $\prod_{i=1}^M P_{sij}^{x_{ij}^*} \not\leq \prod_{i=1}^M P_{sij}^{x_{ij}}$, where $x_{ij} \in \{0, 1\}$, $\forall i, j$ is any feasible solution. There are potentially more suitable solutions for the nonlinear WTA problem.

In order to find more suitable solutions, the redesigned auction-based algorithm improves by adding a judgment condition to verify whether the solution for the nonlinear WTA problem improves (see lines 15 to 22 in Algorithm 1). Thus, the redesigned auction-based algorithm generally works better than the conventional auction-based algorithm. Furthermore, the simulation results also demonstrate it.

However, it is hard to provide mathematical proof that the solution for the WTA problem is optimal or almost optimal. The main reason lies in that the additional criterion evaluates the new assignments in a ‘‘greedy’’ manner and may impede some potential good solutions.

4. Task Swap-Based Algorithm Design

The weapons start moving and launching an attack on the targets when the assignment is completed by the auction-based algorithm. As the weapons move towards the targets, there is still time to optimize the assignment. The task swap algorithm can be started and interrupted at any time, and thus it is used to further optimize the assignment after the redesigned auction-based algorithm. The main idea of the task swap algorithm is that two or more weapons refine their assignment via exchanging tasks between themselves. Specifically, the term of ‘‘perform a task’’ implies that the target is attacked by the weapon. For further details, the weapons that need to adjust the tasks form a group that is termed as the swap loop. In order to adjust the assignment, each weapon on the swap loop substitutes its task with its successor’s along the closed loop. In [17], the search for a swap loop is used by the primal-dual theory and single source shortest path algorithm such as the Dijkstra algorithm. The process of searching a swap loop involves determining a valid loop on graph $G = (V, E)$ (detailed explanation in Section 4.3). In this paper, a new algorithm is proposed, where we remove the restriction that the weights of the edges on a searching loop must be

nonnegative. The improved algorithm decreases the searching length on graph G and searches for multiple swap loops concurrently, thereby improving the efficiency of the solution.

4.1. Task Swap Method. The scene of weapons relative to targets is convenient to describe the concept of the task swap method, as shown in Figure 2(a). Figure 2(b) shows a matrix of the survival probability of the targets attacked by the weapons. It should be noted that the number of weapons is equal to the number of targets in the example. Specifically, after the assignment by the auction-based algorithm, each weapon is assigned to each target such that every weapon possesses exactly one task to perform, and this case is considered as the one-to-one mapping problem.

To decrease the sum of the survival probability of the targets, the weapons exchange their tasks under certain rules, as shown in Figure 3. Let α be the assignment vector, i.e., $\alpha = [\alpha(1), \dots, \alpha(M)]$, where $\alpha(k)$ denotes the task assigned to weapon W_k . The swap loop is denoted as $\alpha(1) \rightarrow \alpha(2) \rightarrow \dots \rightarrow \alpha(k) \rightarrow \alpha(1)$, where $k \geq 2$ denotes the length of the swap loop. The arrows in Figure 3(c) denote that the task of the previous weapon (e.g., weapon W_1) is passed to the next weapon (e.g., weapon W_2). We simplify the description of the swap loop as $(12, \dots, k)$, for example, the swap loop in Figure 3 is expressed as (123) .

4.2. Primal-Dual Problem. The objective of the WTA problem is to maximize the damage probability of the targets, as shown in equation (1). The sum of the damage probability and the survival probability is 1, and thus the objective turns into minimizing the survival probability of the targets, and this is expressed as follows:

$$\min g = \sum_{j=1}^N \left[\text{val}_j \prod_{i=1}^M (1 - P_{ij})^{x_{ij}} \right]. \quad (15)$$

The task swap algorithm is used to solve the linear sum assignment problem. As previously mentioned, we integrate an additional criterion to verify whether the solution of the WTA problem improves. For the sake of explanation, a linear sum model is used here without considering the cumulative damage effect as follows:

$$\min \sum x_{ij} P_{sij}, \quad (16)$$

$$\begin{aligned} \sum_{j=1}^N x_{ij} &= 1, \quad \forall i \in [1, M], \\ \sum_{i=1}^M x_{ij} &= 1, \quad \forall j \in [1, N], \\ x_{ij} &= \{0, 1\}, \quad \forall i, j, \end{aligned} \quad (17)$$

where $P_{sij} = 1 - P_{ij}$.

Definition 1. Equations (16) and (17) are denoted as the primal problem in the dual theory.

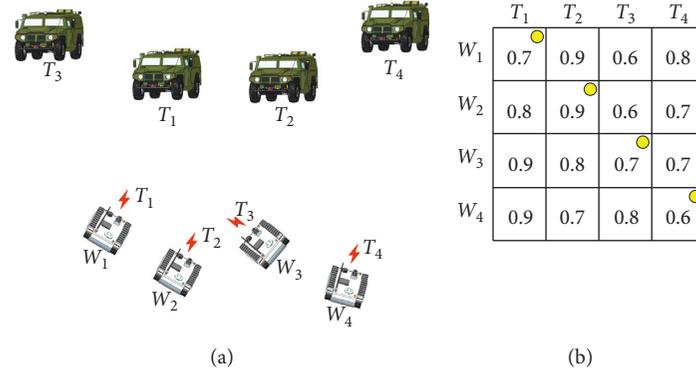


FIGURE 2: Illustration for the concept of the task swap method. (a) Scene of weapons relative to targets. (b) The entries in the matrix correspond to the survival probabilities when targets are attacked by weapons. The circular marks denote the assignment of weapon-target pair $\langle W_i, T_j \rangle$.

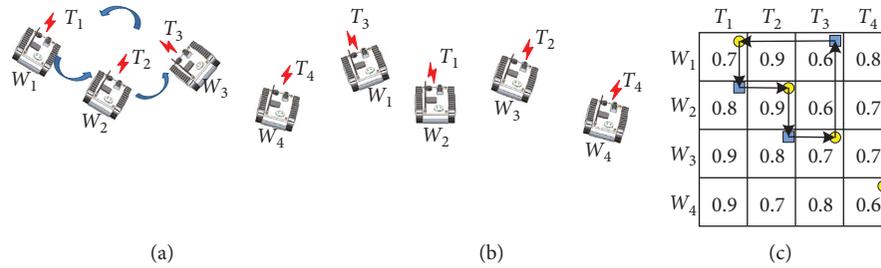


FIGURE 3: Illustration for the process of a task swap. (a) The weapons on the loop pass their tasks to their successors along the closed cycle. (b) New task assignment result. (c) In the survival probability matrix, the connecting arrows form a swap loop. The circular marks denote the old assignment, and the square marks denote the new assignment. The sum of the survival probability of the new assignment is decreased, and this implies that the damage efficiency is improved.

Definition 2. There are corresponding dual variables u_i and v_j , wherein the dual problem is expressed as follows:

$$\max h = \sum_i u_i + \sum_j v_j, \quad (18)$$

$$\text{s.t. } u_i + v_j \leq P_{sij}, \quad \forall (i, j), \quad (19)$$

where the symbols u_i and v_j are used as dual variables in the rest of this paper and should not be confused with the symbol of the payoff u_{ij} and revenue v_j^i mentioned earlier.

Definition 3. Reduced costs are defined as follows:

$$\bar{c}_{ij} = P_{sij} - u_i - v_j, \quad \forall (i, j). \quad (20)$$

Reduced cost is an auxiliary variable used to imply whether the weapon-target assignment $\langle W_i, T_j \rangle$ is feasible. Based on equation (20), the weapon-target pair $\langle W_i, T_j \rangle$ is feasible only if $\bar{c}_{ij} \geq 0$.

4.3. Dijkstra's Algorithm. Let us define graph $G = (V, E)$, as shown in Figure 4. Graph G is built from the weapons communication network topology. The node on graph G is composed of a weapon W_m and its current assigned target $\alpha(m)$, i.e., $V_m = (W_m, \alpha(m)) \in V$. The directed edge from V_m to V_n denotes a new assignment $(W_n, \alpha(m))$, i.e.,

weapon W_m passes its task $\alpha(m)$ to weapon W_n . The weight of the edge is $w(V_m, V_n) = \bar{c}_{n\alpha(m)}$, which is equal to the element of the n th row and $\alpha(m)$ th column in the reduced cost matrix (Figure 5).

In [17], a novel algorithm is proposed to search for a task swap loop, where the problem is transformed into searching for a loop on graph G . Figure 4 shows the process of searching for a loop for a task swap loop by using the Dijkstra algorithm. The Dijkstra algorithm is explained in [31] in detail. Based on Definition 3, the edge is feasible only if $w(V_m, V_n) = \bar{c}_{n\alpha(m)} \geq 0$. It should be noted that the non-negative weight of the edge also corresponds to a constraint for running the Dijkstra algorithm. Thus, while searching for a task swap loop, each node selects the next candidate node only from outgoing edges with nonnegative weights to join the swap chain. Without loss of generality, let V_1 denote the source node, and the other nodes are termed as leaf nodes. A valid task swap loop is formed when a candidate node V_m on the swap chain exhibits an outgoing edge to the source node with the weight $w(V_m, V_1) < 0$, and the total cycle weight $c_L = c_P + w(V_m, V_1) < 0$. After tasks exchange, the objective as shown in equation (16) decreases by $|c_L|$ (see [17] for the proof in detail).

4.4. Improved Task Swap Algorithm. Based on the related work, we conclude that the assignment result is improved after tasks exchanging if the total weight of the edges along

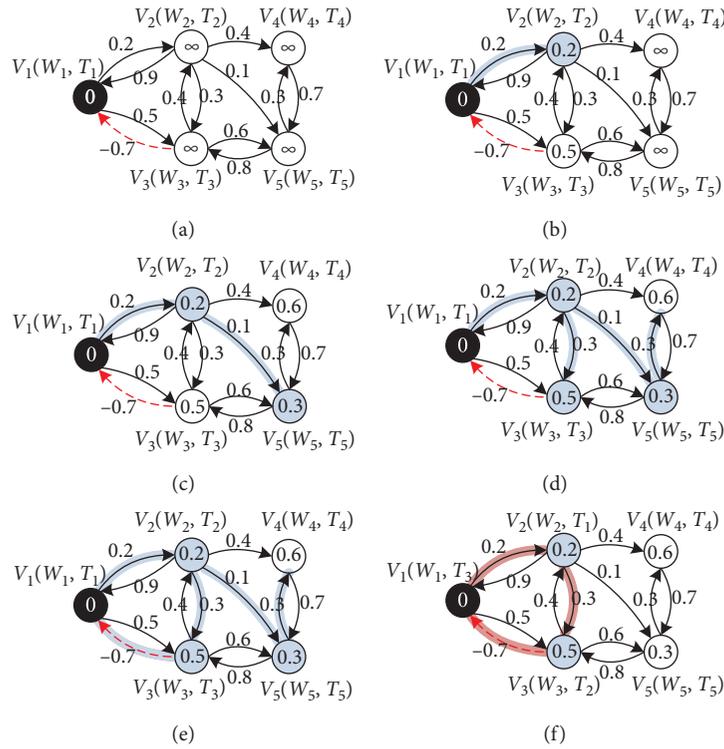


FIGURE 4: A swap loop is found on graph $G = (V, E)$. The dark node denotes the source node, and the shaded nodes denote the nodes on the loop (possibly temporarily); the shaded edges denote the spanning tree of searching for a swap loop. The solid edges denote its weight $w \geq 0$ (see the value beside the edge), and the dashed edges denote $w < 0$. The value in the node is the shortest distance from the source. (a)–(e) show the process of searching for a swap loop on graph G by using the Dijkstra algorithm. (e) The edge from V_3 to V_1 with $w = -0.7$ closes the loop, and the total weights on the loop is $c_L = 0.5 - 0.7 = -0.2$. (f) The new assignment after exchanging tasks.

	T_1	T_2	T_3	T_4	T_5
W_1	0	0.9	-0.7	—	—
W_2	0.2	0	0.4	—	—
W_3	0.5	0.3	0	—	0.8
W_4	—	0.4	—	0	0.3
W_5	—	0.1	0.6	0.7	0

FIGURE 5: A reduced cost matrix. The entries in the matrix correspond to the weights of the edges in Figure 4. The shaded loop in the matrix corresponds to the swap loop in Figure 4(f).

the swap loop is less than zero. Thus, the search for a swap loop turns into determining a loop with total weight $c_L < 0$ on graph G .

As required by the Dijkstra algorithm, the weight of the edges on graph G is nonnegative. Otherwise, the leaf nodes cannot obtain the results about the shortest distance and path between the source node and themselves by running the Dijkstra algorithm. Specifically, when there is a loop with negative weight, the distance between the leaf node and source node tends to $-\infty$, i.e., there are no shortest distance

and path between the leaf node and source node. The Bellman–Ford algorithm is used to solve the single source shortest path problem and determine whether the graph exhibits a loop with a negative weight. However, the Bellman–Ford algorithm repeatedly requires $|V| - 1$ relaxation operations on each edge, and thus it exhibits a long running time and is not suitable for scenes with large-scale nodes.

This paper further improves over the Dijkstra-based searching process for finding a task swap loop, and this allows edges and loops with negative weight to appear on graph G . During the process of searching for a task swap loop, if the shortest distance of a leaf node V_m on the swap chain (such as the shaded nodes in Figure 4) is less than the previous value, then the node V_m may be included in the loop with a negative weight. Then node V_m stops the current search for a swap loop and is used as the new source node, and a new search commences again (see lines 15 to 22 in Algorithm 2). If there exists a loop with $c_L < 0$ between the source node V_{m0} and leaf node, then a task swap loop that reduces equation (16) is determined. However, for the nonlinear WTA problem, the new assignment after exchanging the tasks along the task swap loop is potentially not improved. The weapons should calculate the new value of the objective function based on equation (1) and verify whether the new value improves when compared with the previous value. If true, then a valid task swap loop is formed (see lines

```

(1) Initialize the dual variables  $u_m = P_{sm\alpha(m)}, v_n = 0, \forall m, n$ , the reduced cost  $\bar{c}_{mn} = P_{smn} - u_m - v_n$ ;
(2) Let  $S = G \cdot V$  be the set of candidate source nodes;
(3) While  $S \neq \emptyset$ 
(4)  $V_{m0}: (m0, n0) = \operatorname{argmin}_{V_m \in S} \bar{c}_{mn}$  is voted among neighbors as the source node;  $S = S/V_{m0}$ ; if  $\bar{c}_{m0n0} \geq 0$ , the assignment cannot be improved, then Algorithm 2 terminates;
(5) Let  $Q = G \cdot V$  be the set of the candidate nodes to be added to the swap chain, let  $\bar{Q} = \{V_{m0}\}$  be the set of the nodes that joined the swap chain (leaf nodes);
(6) Let  $V_m \cdot d$  be the distance from node  $V_m$  to the source node, let  $V_m \cdot \pi$  be  $V_m$ 's predecessor node; initialize  $V_m \cdot d = \infty, V_m \cdot \pi = \text{NIL}$ , and  $V_{m0} \cdot d = 0$ ;
(7) While  $Q \neq \emptyset$ 
(8)  $V_{\text{pop}} = \min_{V_m \in Q} \{V_m \cdot d\}$ ;
(9)  $Q = Q/V_{\text{pop}}; \bar{Q} = \bar{Q} \cup V_{\text{pop}}$ ;
(10) For each node  $V_m$  on outgoing edge of  $V_{\text{pop}}$  do
(11) If  $V_m \cdot d > V_{\text{pop}} \cdot d + w(V_{\text{pop}}, V_m)$  then
(12) //This step is termed as the relaxation operation.
(13)  $V_m \cdot d = V_{\text{pop}} \cdot d + w(V_{\text{pop}}, V_m)$ ; set  $V_m \cdot \pi = V_{\text{pop}}$ ;
(14) End if
(15) If  $V_m \in \bar{Q}$  and the distance  $V_m \cdot d$  is smaller than its previous value then
(16) If  $V_m \in S$  then
(17) Change the source node:  $V_{m0} = V_m; S = S/V_m$ ;
(18) Go to step 5;
(19) Else
(20) Go to step 3;
(21) End if
(22) End if
(23) If there exists a loop with  $c_L < 0$  between the source node  $V_{m0}$  and a leaf node then
(24) If  $f(\tau - 1) < f(\tau)$  then
(25) The valid task swap loop is found, and Algorithm 2 terminates.
(26) End if
(27) End if
(28) End for
(29) End while
(30) End while

```

ALGORITHM 2: The improved task swap algorithm DWLCC.

23 to 27). The improved task swap algorithm is described in Algorithm 2.

After exchanging tasks along the task swap loop, any weapon W_m on the loop adjusts the dual variables as $u_m = P_{sm\alpha'(m)}, v_n = 0, \forall m, n$, where subscript $\alpha'(m)$ denotes the index of the new assigned target $T_{\alpha'(m)}$ to W_m , and resets the reduced cost $\bar{c}_{mn} = P_{smn} - u_m - v_n$. It is observed that there is no dependency on the dual variables between the previous and next processes of searching swap loops. Furthermore, only the local weapons on the task swap loop need to adjust the dual variables, and other weapons do not need to adjust the dual variables. The features facilitate asynchronous searching for multiple task swap loops concurrently, and this speeds up the process of optimizing the WTA solution.

The following example is presented to further illustrate the flow of Algorithm 2. Figure 6(a) shows a survival probability matrix of each target attacked by each weapon. Figure 6(b) shows a reduced cost matrix obtained by the survival probability matrix (see line 1, Algorithm 2). Figure 7 shows the process of running Algorithm 2 to search for a task swap loop. In Figure 7(c),

the distance from node V_2 on the swap chain (the shaded edge) to the source node is $V_2 \cdot d = 0$, and this is smaller than its previous value of 0.1 as shown in Figure 7(b). Then, we conclude that node V_2 is potentially contained on a loop with a negative weight. Based on lines 15–22, Algorithm 2, node V_2 becomes the new source node and continues to search for a task swap loop. It should be noted that, at this moment, a loop containing node V_2 cannot be considered as a valid swap loop that improves the assignment. This is because a negative weight of an edge on the loop may also result in a reduction of $V_2 \cdot d$, while the total weight of the loop is positive and the loop does not improve the assignment. Figure 7(e) shows the shortest distance of the source node is $V_2 \cdot d = -0.1 < 0$, and the total weight of the loop is $c_L = -0.1 + 0 < 0$, i.e., a swap loop that improves the assignment is formed. As shown in the survival matrix (Figure 6(a)), it decreases from 1.7 to 1.6 for the sum of the survival probabilities of weapon-target assignment pairs after exchanging the tasks along the task swap loop. Moreover, the amount of reduction 0.1 is identical to the absolute value of the total weight of the loop $|c_L|$.

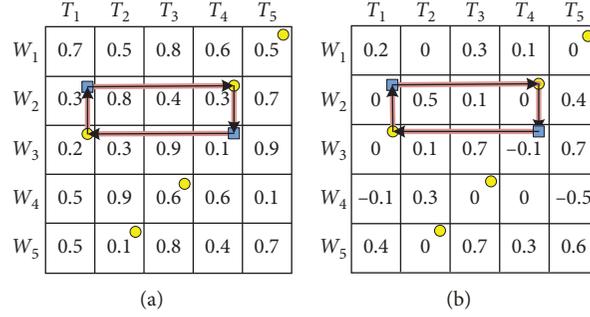


FIGURE 6: Illustration of a swap loop. (a) A survival probability matrix of each weapon-target pair; the circular marks and the square marks denote the old and new assignments, respectively. (b) A reduced cost matrix of each weapon-target pair.

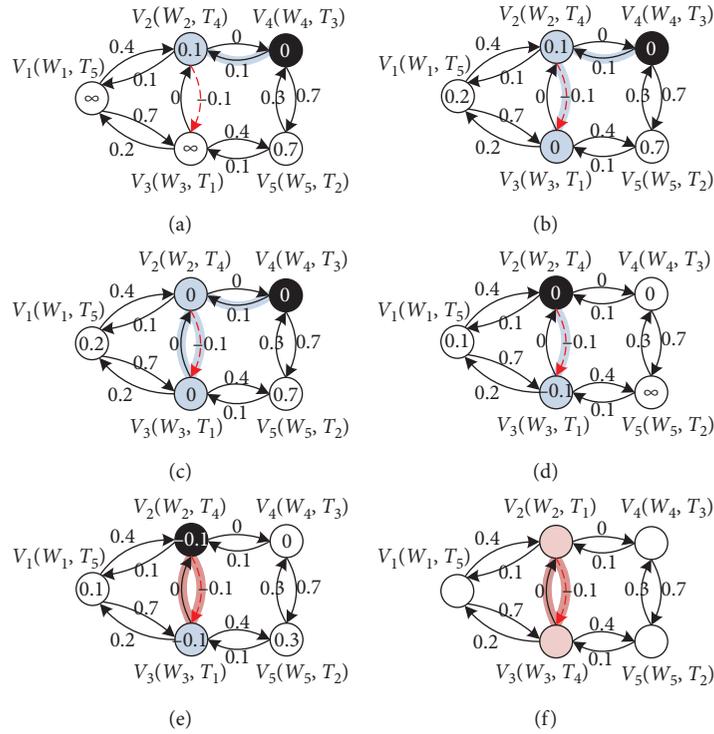


FIGURE 7: Illustration of the flow of Algorithm 2. The symbols have the same meaning as in Figure 4. (a)–(e) show how a task swap loop is found. (f) The new assignment after exchanging tasks.

4.5. Performance Analysis. The time complexity of the improved task swap algorithm to search for a single task swap loop is $O(|V|^2)$, where $|V| = M$ in this paper. The analysis is analogous to Dijkstra's algorithm. In order to decrease the total survival probability of the targets, the weapons run Algorithm 2 repeatedly until no further improvements are possible. The number of times Algorithm 2 is run depends on the initial assignment. When compared with the task swap algorithm in [17], the proposed algorithm is more likely to detect a swap loop with $c_L < 0$, and this converges faster. This is because Algorithm 2 allows edges and loops with negative weight to appear on graph G , and the expansion of the search space contributes to more easy detection of a swap loop with $c_L < 0$. The following simulation results demonstrate the superiority of the proposed algorithm.

Proposition 1. We assume a case where $M = N$. Let $w(V_m, V_n) \in \mathbb{R}$ denotes the weight of the edge on task swap loop, and this also corresponds to the value of the new assignment entries in the reduced cost matrix (see the square entries in Figure 6(b)). Define c_L as $c_L = \sum_L w(V_m, V_n)$, then c_L denotes the difference between the new and old solutions, $c_L = g_{\text{new}} - g_{\text{old}}$, where g is obtained in equation (15).

Proof. Initialize $u_m = P_{s\alpha(m)}$, $v_n = 0$, $\forall m, n$, and then we obtain $w(V_m, V_n) = \bar{c}_{n\alpha(m)} = P_{s\alpha(m)} - u_n - v_{\alpha(m)} = P_{s\alpha(m)} - P_{s\alpha(n)}$.

Node V_m is the predecessor of V_n , i.e., weapon W_m passes its task $\alpha(m)$ to weapon W_n , and the value $P_{s\alpha(n)}$ of weapon W_n becomes $P_{s\alpha(m)}$. Let $g_{\text{old}} = \sum_{i \notin L} P_{s\alpha(i)} + \sum_{n \in L} P_{s\alpha(n)}$ and $g_{\text{new}} = \sum_{i \notin L} P_{s\alpha(i)} + \sum_{n \in L, V_m = V_n} P_{s\alpha(m)}$. We have

$$\begin{aligned}
\mathcal{G}_{\text{new}} - \mathcal{G}_{\text{old}} &= \sum_{n \in L, V_m = V_n \cdot \pi} P_{sna(m)} - \sum_{n \in L} P_{sna(n)} \\
&= \sum_{n \in L, V_m = V_n \cdot \pi} \bar{c}_{na(m)} = \sum_L w(V_m, V_n) = c_L.
\end{aligned} \tag{21}$$

Thus, Proposition 1 is concluded. \square

Based on Proposition 1, if the numbers of weapons and targets are identical, then the solution is improved by c_L every time after exchanging tasks along the task swap loop. Finally, the solution of the objective (15) with an equal number of weapons and targets converges to the optimal solution. Unfortunately, for the same reason given in Section 3.3, it is hard to provide mathematical proof that Algorithm 2 is optimal or almost optimal for the WTA problem with the unequal numbers of weapons and targets.

5. Experiments

In this section, we verify the good performance of the redesigned auction-based algorithm, improved task swap algorithm, and hybrid algorithm of the aforementioned two algorithms. We implemented the algorithms for different numbers of weapons and tasks. We assume that the weapons network topology is fixed and complete. We randomly generated the damage probabilities P_{ij} from a uniform distribution in (0,1). All algorithms are implemented in MATLAB and run on an Intel Core i7 @ 3.60 GHz processor with 16 GB RAM.

Furthermore, the example videos demonstrate the validity of our algorithms implementation via semi-physical experiments and can be found at <https://doi.org/10.6084/m9.figshare.7506248>. The videos indicate that the algorithms can be performed feasibly in a distributed manner.

5.1. Simulation Results of Auction-Based Algorithm. For ease of description, we name this paper's redesigned auction-based algorithm AB1 and the conventional auction-based algorithm AB2. The conventional auction-based algorithm implies that the auction-based algorithm is combined with the method of adding virtual targets in [22] such that the WTA problem is reduced to the linear sum problem without considering the cumulative effect. The performances of AB1 and AB2 are compared in terms of solution quality and time consumption. The solution quality is expressed via the mean of the optimized damage value (equation (1)) of 100 simulations. The time consumption is the mean of the runtime of 100 simulations.

The results are shown in Table 1. Table 1 shows that when compared with AB2, AB1 has a higher damage value, and its advantages become more significant when the problem size increases. For up to 120 weapons and 110 targets, the damage value obtained by AB1 is improved by 37% compared to the value obtained by AB2. Although the runtime of AB1 slightly exceeds that of AB2, they are both processed

within 0.5 second for up to 120 weapons and 110 targets. The time requirement for WTA is satisfied because ground targets generally move slowly.

To present and analyze the performances of AB1 and AB2 in a more visual manner, the damage value and runtime in Table 1 are plotted in Figures 8 and 9, respectively. Furthermore, the figures reflect the robustness of the algorithms measured via the standard deviation of the damage value and runtime. As shown in Figure 8, the mean and standard deviation of the optimized damage value of AB1 both exceed that of AB2, and the difference between AB1 and AB2 increases when the problem size increases. As shown in Figure 9, AB1 performs slightly worse than AB2 in terms of time consumption.

Figure 10 further shows the performance of algorithms relative to time. Without loss of generality, only Case 4 is shown. We observe that AB1 achieves the optimal value calculated by AB2 earlier than AB2 does. From this perspective, the computational efficiency of AB1 is not worse than that of AB2.

In summary, the proposed redesigned auction-based algorithm significantly improves the solution quality of the WTA problem in a distributed manner while the time consumption is acceptable. Additionally, the proposed algorithm exhibits good performance to solve the large scale of the WTA problem with unequal number of weapons and targets.

5.2. Simulation Results of Task Swap-Based Algorithm.

Three task swap-based algorithms are compared: the proposed algorithm in this paper termed as the Dijkstra with less constrained condition (DWLCC) algorithm; Dijkstra algorithm in the literature [17]; and breath-first-search (BFS) algorithm, which searches for the swap loop in a manner analogous to a BFS approach in the literature [29].

The solution quality and time consumption of the three algorithms are shown in Figures 11 and 12, respectively. All statistics are the average values of 100 simulations. The initial assignments for the algorithms are arbitrary: weapons are randomly assigned to targets. With respect to different numbers of weapons and targets, there are four cases as previously mentioned in Table 1.

As shown in Figure 11, the three algorithms almost achieve the same optimal value by the time that the loop with $c_L < 0$ cannot be found anymore [17]. However, the Dijkstra algorithm has the constraint wherein the task swap loops must be searched from the edges with nonnegative weight, and a few potential task swap loops from the edges with negative weight cannot be found. Thus, under the same conditions, the solution quality of the Dijkstra algorithm is slightly worse than that of the proposed DWLCC algorithm and the BFS algorithm. As shown in Figure 12, DWLCC performs better, in terms of time consumption, than the other two algorithms. For up to 120 weapons and 110 targets, the time consumption of DWLCC is 53% less than that of the Dijkstra algorithm, and 74% less than that of the BFS algorithm.

In order to observe in a more visual manner, the performance of algorithms over time is plotted as shown in

TABLE 1: Solution quality and time consumption of AB1 and AB2.

Case	Number of weapons M	Number of targets N	AB1		AB2	
			Damage value	Runtime (sec.)	Damage value	Runtime (sec.)
Case 1	30	20	19.46	0.034	17.29	0.025
Case 2	60	50	48.91	0.079	38.32	0.050
Case 3	90	80	78.55	0.189	58.71	0.119
Case 4	120	110	108.26	0.369	79.08	0.224

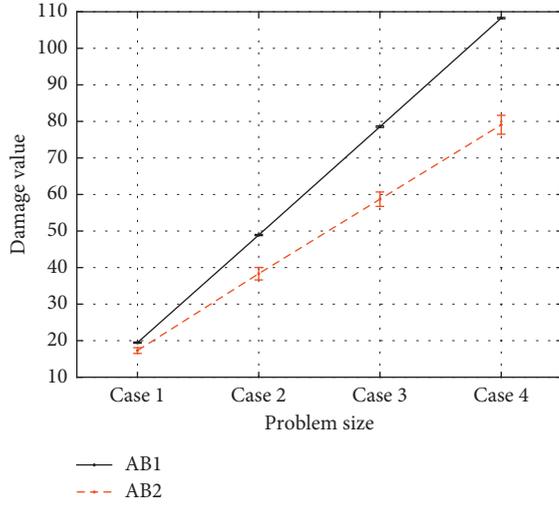


FIGURE 8: Damage value of AB1 and AB2.

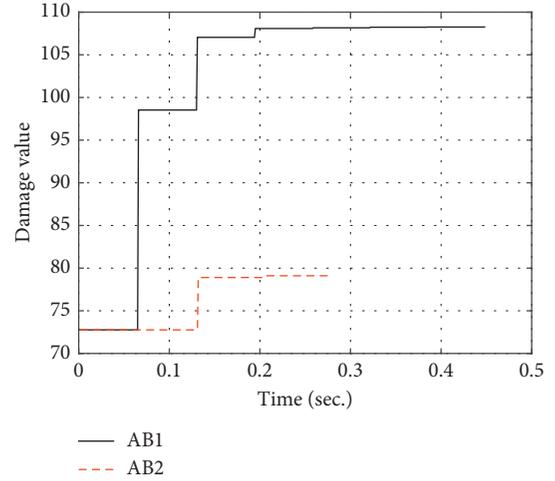


FIGURE 10: Damage value over time.

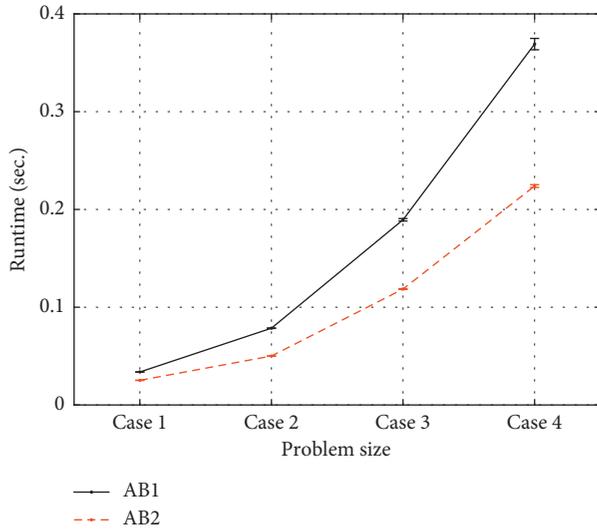


FIGURE 9: Runtime of AB1 and AB2.

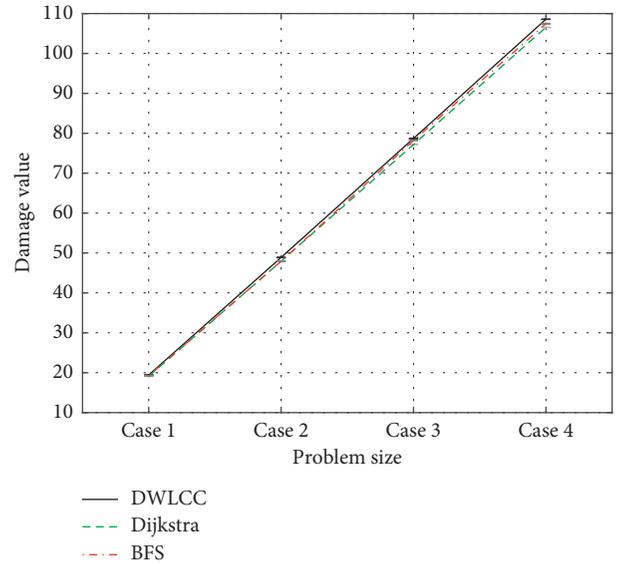


FIGURE 11: Damage value of DWLCC, Dijkstra, and BFS.

Figure 13. Without loss of generality, only Case 4 is shown. Figure 13 shows clearly that the proposed DWLCC algorithm has the best solution quality and the fastest convergence speed among the three algorithms. The proposed DWLCC algorithm can also be adapted well to a distributed scenario.

5.3. *Simulation Results of the Hybrid Algorithm.* The hybrid algorithm proposed in this paper is divided into two stages.

- (1) First, the AB1 algorithm is implemented. With respect to each weapon W_i , the initialization inputs of the hybrid algorithm are set as the same as that of the AB1 algorithm. The outputs of the hybrid algorithm in this stage are the initial assignments, $\langle W_i, T_{\alpha_i} \rangle, \forall i$, which satisfy the constraint conditions (2)–(4). The AB1 algorithm terminates when the price P of all targets does not change for Δ iterations.

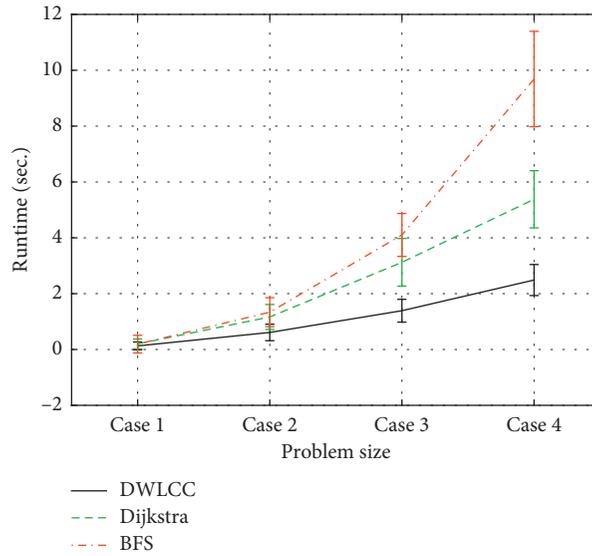


FIGURE 12: Runtime of DWLCC, Dijkstra, and BFS.

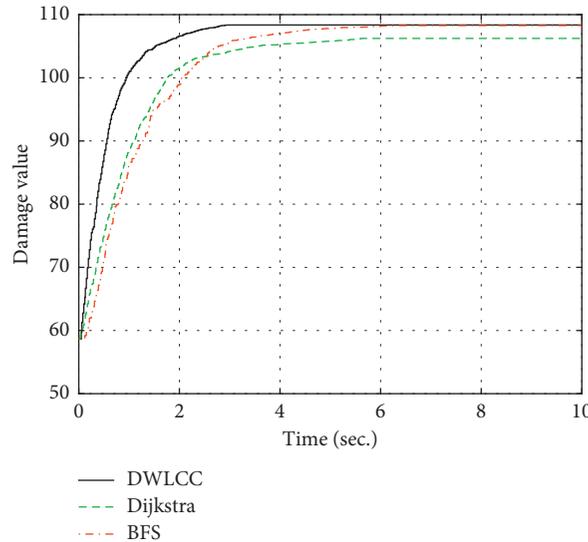


FIGURE 13: Damage value over time.

(2) Second, after the AB1 algorithm, the DWLCC algorithm is implemented to refine the assignment obtained by AB1. The initialization input of the DWLCC algorithm is the assignment obtained by AB1, and the output of the DWLCC algorithm is the refined assignment.

Particularly, the number of weapons is greater than the number of targets, but the swap-based algorithm still addresses the problem in this scenario. This is because each weapon holds a “task” after the AB1 algorithm, although the “tasks” performed by different weapons may share the same target. So long as each weapon can hold a task at any time, the weapons can exchange their tasks to improve the assignment. The non-one-to-one mapping scenario (weapon-to-target) becomes one-to-one mapping scenario (weapon-to-task).

The results of the hybrid algorithm, AB1, and DWLCC are shown in Table 2. For different numbers of weapons and targets, there are four cases as previously mentioned in Table 1. All statistics are the average values of 100 simulations.

Table 2 shows that the hybrid algorithm has best solution quality. The hybrid algorithm and the DWLCC algorithm do not converge to the same optimal solution. This makes sense because the assignment result obtained by the first stage AB1 algorithm in the hybrid algorithm optimizes the initial input of the second stage DWLCC algorithm.

Table 2 shows that the time consumption of the hybrid algorithm slightly exceeds that of AB1, but it is evidently less than that of DWLCC. The weapons start moving and attacking the targets when the first stage (the AB1 algorithm) is completed. Thus, to obtain the same solution

TABLE 2: Solution quality and time consumption of hybrid, AB1, and DWLCC.

Case	Hybrid		AB1		DWLCC	
	Damage value	Runtime (sec.)	Damage value	Runtime (sec.)	Damage value	Runtime (sec.)
Case 1	19.54	0.255	19.45	0.043	19.43	2.105
Case 2	49.05	0.314	48.87	0.089	48.89	2.639
Case 3	78.83	0.545	78.56	0.241	78.70	2.987
Case 4	108.66	0.780	108.27	0.401	108.57	3.215

quality, the hybrid algorithm is completed as fast as the AB1 algorithm.

Thus, the hybrid algorithm is proposed to fully utilize the advantages of both AB1 and DWLCC. Compared to implementing AB1 and DWLCC separately, the hybrid algorithm exhibits better solution quality with no degradation with respect to the time consumption.

6. Conclusions

In order to solve the WTA problem in a decentralized manner, we propose a redesigned auction-based algorithm, an improved task swap algorithm, and a hybrid algorithm. First, given that the cumulative effect is not additive, we use the maximum consensus technique twice and add a judging criterion to solve the WTA problem. Simulations indicate that, when compared to the conventional auction-based algorithm, the proposed redesigned auction-based algorithm (AB1) represents significant improvements in solution quality (37% on average) while the additional running time is negligible. Second, to refine the task swap algorithm, we eliminate the restriction that the weights of the edges on graph G must be positive, and the optimization process is accelerated. Simulation results indicate that, in the same conditions, the improved task swap algorithm (DWLCC) converges significantly faster to the optimal solution. The average time-savings over the Dijkstra algorithm and BFS algorithm correspond to 53% and 74%, respectively. Third, the hybrid algorithm is analyzed to fully utilize the advantages of AB1 and DWLCC. Simulations and analyses indicate that the hybrid algorithm is superior to implementing AB1 and DWLCC separately in terms of solution quality and time consumption.

6.1. Future Work. Unfortunately, all the weapons are exhausted when the assignment is completed. This scene is meaningful because small unmanned weapons are typically low cost. However, it is necessary to investigate further maximization of the total damage value while minimizing the weapon consumption in a distributed manner.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Defense Industrial Technology Development Program of China (JCKY2017602C016).

Supplementary Materials

An example video demonstrates the validity of our algorithms implementation and can be found at . (*Supplementary Materials*)

References

- [1] Z. R. Bogdanowicz, A. Tolano, K. Patel, and N. P. Coleman, "Optimization of weapon-target pairings based on kill probabilities," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1835–1844, 2013.
- [2] X. Hu, P. Luo, X. Zhang, and J. Wang, "Improved ant colony optimization for weapon-target assignment," *Mathematical Problems in Engineering*, vol. 2018, Article ID 6481635, 14 pages, 2018.
- [3] H. Xu, Q. Xing, and Z. Tian, "MOQPSO-D/S for air and missile defense WTA problem under uncertainty," *Mathematical Problems in Engineering*, vol. 2017, Article ID 9897153, 13 pages, 2017.
- [4] M. T. Davis, M. J. Robbins, and B. J. Lunday, "Approximate dynamic programming for missile defense interceptor fire control," *European Journal of Operational Research*, vol. 259, no. 3, pp. 873–886, 2017.
- [5] O. Karasakal, N. E. Özdemirel, and L. Kandiller, "Anti-ship missile defense for a naval task group," *Naval Research Logistics (NRL)*, vol. 58, no. 3, pp. 304–321, 2011.
- [6] Z.-J. Lee, S.-F. Su, and C.-Y. Lee, "Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 1, pp. 113–121, 2003.
- [7] N. Jia, Z. Yang, and K. Yang, "Operational effectiveness evaluation of the swarming UAVs combat system based on a system dynamics model," *IEEE Access*, vol. 7, pp. 25209–25224, 2019.
- [8] G. Jia and Z. Hou, "The analysis and enlightenment about the UAV swarming project of the United States Military," *National Defense Science & Technology*, vol. 38, no. 4, pp. 53–55, 2017.
- [9] O. Ilaya, C. Bil, and M. Evans, "Control design for unmanned aerial vehicle swarming," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 222, no. 4, pp. 549–567, 2008.
- [10] 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 Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 41, no. 3, pp. 598–606, 2011.

- [11] J. Wang, P. Luo, and J. Zhou, "A memetic algorithm for constrained weapon target assignment problems," in *Proceedings of the 2017 2nd IEEE International Conference on Computational Intelligence and Applications*, pp. 182–188, Beijing, China, September 2017.
- [12] C. Wang, G. Fu, D. Zhang, H. Wang, and J. Zhao, "Genetic algorithm-based variable value control method for solving the ground target attacking weapon-target allocation problem," *Mathematical Problems in Engineering*, vol. 2019, Article ID 6761073, 9 pages, 2019.
- [13] Y. Li, Y. Kou, and Z. Li, "An improved nondominated sorting genetic algorithm III method for solving multiobjective weapon-target assignment Part I: the value of fighter combat," *International Journal of Aerospace Engineering*, vol. 2018, Article ID 8302324, 23 pages, 2018.
- [14] H. Chen, Z. Liu, Y. Sun, and Y. Li, "Particle swarm optimization based on genetic operators for sensor-weapon-target assignment," in *Proceedings of the 2012 Fifth International Symposium on Computational Intelligence and Design*, pp. 170–173, Hangzhou, China, October 2012.
- [15] 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," *Applied Soft Computing*, vol. 70, pp. 845–863, 2018.
- [16] D. J. Bachmann, R. J. Evans, and B. Moran, "Game theoretic analysis of adaptive radar jamming," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1081–1100, 2011.
- [17] L. Liu, N. Michael, and D. A. Shell, "Communication constrained task allocation with optimized local task swaps," *Autonomous Robots*, vol. 39, no. 3, pp. 429–444, 2015.
- [18] S. Öztürk and A. E. Kuzucuoğlu, "A multi-robot coordination approach for autonomous runway Foreign Object Debris (FOD) clearance," *Robotics and Autonomous Systems*, vol. 75, pp. 244–259, 2016.
- [19] A. Farinelli, L. Iocchi, and D. Nardi, "Distributed on-line dynamic task assignment for multi-robot patrolling," *Autonomous Robots*, vol. 41, no. 6, pp. 1321–1345, 2017.
- [20] R. G. Smith, "The contract net protocol: high-level communication and control in a distributed problem solver," *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, 1980.
- [21] D. P. Bertsekas, "The auction-based algorithm: a distributed relaxation method for the assignment problem," *Annals of Operations Research*, vol. 14, no. 1, pp. 105–123, 1988.
- [22] D. P. Bertsekas, "Auction algorithms for network flow problems: a tutorial introduction," *Computational Optimization and Applications*, vol. 1, no. 1, pp. 7–66, 1992.
- [23] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction-based algorithm for the assignment problem," in *Proceedings of the 2008 47th IEEE Conference on Decision and Control*, pp. 1212–1217, Cancun, Mexico, December 2008.
- [24] L. Luo, N. Chakraborty, and K. Sycara, "Distributed algorithms for multirobot task assignment with task deadline constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 876–888, 2015.
- [25] L. Luo, N. Chakraborty, and K. Sycara, "Provably-good distributed algorithm for constrained multi-robot task assignment for grouped tasks," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 19–30, 2015.
- [26] L. Liu and D. A. Shell, "An anytime assignment algorithm: from local task swapping to global optimality," *Autonomous Robots*, vol. 35, no. 4, pp. 271–286, 2013.
- [27] T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-UAV context," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pp. 3622–3627, New Orleans, LA, USA, April 2004.
- [28] X. Zheng and S. Koenig, "K-swaps: cooperative negotiation for solving task-allocation problems," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 373–378, Pasadena, CA, USA, July 2009.
- [29] L. Liu and D. A. Shell, "A distributable and computation flexible assignment algorithm: from local task swapping to global optimality," in *Proceedings of Robotics: Science and Systems*, Cambridge, MA, USA, July 2012.
- [30] Z. Ma, L. Liu, and G. S. Sukhatme, "An adaptive K-opt method for solving traveling salesman problem," in *Proceedings of the 2016 IEEE 55th Conference on Decision and Control*, pp. 6537–6543, Las Vegas, NV, USA, December 2016.
- [31] H. C. Thomas and E. L. Charles, *Introduction to Algorithms*, Massachusetts Institute of Technology, Boston, MA, USA, 3rd edition, 2009.

