aerospace

MDPI

Article

A Multi-Target Consensus-Based Auction Algorithm for Distributed Target Assignment in Cooperative Beyond-Visual-Range Air Combat

Weihua Li [1], Yongxi Lyu [1,2,*], Sifan Dai [1], Huakun Chen [1], Jingping Shi [1,2] and Yongfeng Li [1]

1 School of Automation, Northwestern Polytechnical University, Xi'an 710129, China
2 Shaanxi Province Key Laboratory of Flight Control and Simulation Technology, Xi'an 710129, China
* Correspondence: yongxilyu@nwpu.edu.cn; Tel.: +86-187-2954-2380

Abstract: With recent advances in airborne weapons, air combat tends to occur in the form of beyond-visual-range (BVR) combat and multi-aircraft cooperation. Target assignment is critical in multi-aircraft BVR air combat decision-making. Most previous research on target assignment for multi-aircraft cooperative BVR air combat has focused on centralized algorithms, which can be time-consuming and unreliable. This paper proposes an efficient distributed target assignment algorithm called the multi-target consensus-based auction algorithm (MTCBAA). First, by analyzing the main geometric aspects of BVR air combat, a target assignment model for cooperative BVR air combat was established. Next, based on a consensus-based auction algorithm (CBAA), the MTCBAA was developed to solve the target assignment problem by introducing a cooperative decision-making variable. Although the MTCBAA is based on a greedy mechanism, it can guarantee at least 50% global optimization performance, which was proven through a demonstration of the minimum optimization performance of a centralized target assignment algorithm, since the centralized algorithm is equivalent to the MTCBAA. Finally, experiments were conducted, including an experiment that illustrates the operation of the proposed algorithm, Monte Carlo comparisons with a centralized target assignment method based on the immune algorithm, and deployment experiments on a semi-physical simulation platform. Compared with the heuristic target assignment algorithm, the proposed algorithm significantly improved the target assignment efficiency. The practicality of the proposed algorithm was further verified through a distributed semi-physical simulation experiment.

Keywords: beyond-visual-range air combat; distributed target assignment; cooperative decision-making; consensus-based auction algorithm

Citation: Li, W.; Lyu, Y.; Dai, S.; Chen, H.; Shi, J.; Li, Y. A Multi-Target Consensus-Based Auction Algorithm for Distributed Target Assignment in Cooperative Beyond-Visual-Range Air Combat. *Aerospace* **2022**, *9*, 486. https://doi.org/10.3390/aerospace9090486

Academic Editor: Peng Wei

Received: 3 July 2022
Accepted: 29 August 2022
Published: 30 August 2022

1. Introduction

Due to recent advances in airborne weapons, air combat tends to be performed beyond the visual range [1]. In addition, the shortcomings of a single aircraft carrying limited payloads and having limited mission execution capacities and variety have become more prominent. As a potential solution for improving the probability of mission success, cooperative air combat [2], where multiple aircraft complement and coordinate with each other, has attracted great attention in recent years. In addition, target assignment [3,4], as one of the core elements in cooperative air combat decision-making, plays the role of coordinating the mission execution of each combat platform, thus maximizing the combat benefits. Therefore, target assignment in cooperative beyond-visual-range (BVR) air combat has been a research hotspot.

The earliest target assignment methods were developed in a centralized manner; namely, it was assumed that there was a central node in a fleet, which was referred to as a leader, while the other aircraft in the fleet were referred to as followers. The leader collected global situation information from the followers and ran an optimization algorithm to find global optimal or sub-optimal target assignment combinations for the fleet. Common target assignment methods based on the central node can be roughly divided into two categories: optimization methods [5] and heuristic methods [6,7]. With the development of artificial intelligence technology, centralized target assignment methods based on neural networks [8], reinforcement learning [9], and decision trees [10] have emerged in recent years. In the early stage, the centralized target assignment was the most widely associated target assignment approach but, to obtain global situational information for the central node, a large amount of communication is required. In addition, these optimization algorithms are generally time-consuming, which can lead to inadequate real-time assignment results in a rapidly changing battlefield. The heavy dependence of a fleet on the central node can also lead to a failure of the entire fleet when the central node is destroyed.

Aiming to address the shortcomings of the centralized target assignment algorithms, distributed target assignment algorithms have emerged. In the early distributed target assignment algorithms [11,12], the connected nodes sent their situation information to each other to achieve situational consistency [13–15]. Then, all nodes ran the centralized target assignment algorithm simultaneously to obtain their respective assignment solutions. This type of algorithm does not require a central node and, therefore, improves system robustness, but it cannot effectively solve the problems of large-scale communication and high time consumption in centralized target assignment, and the application of situation-consistency algorithms can significantly increase the target assignment time [16].

Another way to realize distributed target assignment is to use an auction mechanism. A typical algorithm of this type is the contract network-based auction algorithm [17]. In this algorithm, the target assignment process includes three stages, an inviting bid stage, a submitting bid stage, and a winner determination stage. At the beginning of each iteration, a central node is selected based on the predetermined mechanism. Then, each node in the fleet sends its bid for each task to the central node, and the central node determines the winner for each task according to the received bids. As only bidding information is exchanged in each iteration, the requirement for system communication bandwidth is reduced. However, due to the existence of the central node in each iteration, the communication topology in each iteration is limited.

In 2009, Choi et al. [18] proposed the consensus-based auction algorithm (CBAA) and the consensus-based bundle algorithm (CBBA to solve single-agent multi-target assignment problems. These algorithms were developed based on the idea of consistent assignment results. Both algorithms consist of two phases, an auction phase and a consensus phase. In the auction phase, each agent chooses the target with the largest gain for itself to join its own task bundle one by one using the greedy strategy. Then, in the consensus phase, each agent achieves conflict resolution based on the consistency rules. The CBAA and CBBA do not have any special requirements for network topology but cannot achieve accurate multi-agent single-target assignment. In 2010, Choi et al. [19] addressed the problem of accurate multi-agent single-target assignment through task decomposition, modifying the gain calculation method and task elimination mechanism, and proposed an enhanced algorithm based on the CBBA, which can achieve heterogeneous multi-agent collaboration to accomplish a common task. However, this algorithm can assign at most two agents to the same task at a time. Different task scenarios correspond to different constraints, and the above distributed task assignment methods still need to be modified when facing specific task scenarios. To address this limitation, derivative algorithms based on algorithms for specific application scenarios [20–24] have been developed. In [23,24], distributed algorithms for cooperative air combat target assignment were studied. However, these algorithms have difficulties in practical distributed deployment because they denote simple loops of the CBAA and need to handle global variables between the CBAA iterations. Furthermore, a few studies have been conducted on distributed target assignment algorithms in cooperative air combat.

In multi-aircraft cooperative air combat, where each aircraft has the capacity to attack multiple targets, there are both multi-aircraft cooperative attacks on single targets and single aircraft attacks on multiple targets. Moreover, the number of missiles each aircraft assigns to a target can be larger than one. The existing distributed target assignment algorithms cannot be directly used in such a scenario.

To address the target assignment problem in cooperative BVR air combat in a distributed manner, this paper proposes a multi-target consensus-based auction (MTCBAA) algorithm, which is a distributed target assignment algorithm for cooperative BVR air combat.

The main contributions of this work can be summarized as follows:

(1) A distributed target assignment algorithm for cooperative BVR air combat called the MTCBAA is proposed;
(2) The minimum performance of the MTCBAA is proven by demonstrating the minimum optimization performance of a centralized target assignment algorithm, since the centralized algorithm is equivalent to the MTCBAA;
(3) The minimum performance of the MTCBAA is also validated through comparison experiments with an immune algorithm-based centralized target assignment method;
(4) The proposed algorithm was deployed on a distributed semi-physical simulation platform to verify its practicality and real-time performance.

The remainder of this paper is organized as follows. Section 2 presents a target assignment model for cooperative BVR air combat. Section 3 describes the MTCBAA in detail and proves its minimum optimization performance. Section 4 presents and analyzes experiment results. Section 5 concludes the paper and discusses potential future research.

2. Target Assignment Model

The target assignment modeling in cooperative BVR air combat is based on the traditional non-parameter approach, which has been widely studied in the related literature [1,4]. In this study, the approach from [1] is adopted; i.e., the azimuth angle φ, entry angle q, and distance D between an aircraft and its target are considered to be the main modeling factors, as shown in Figure 1. No special advantage functions are used for the speed and altitude factors. The speed and altitude are taken into consideration when calculating the missile attack zone parameter and the non-escape zone parameter, which are involved in the distance advantage function. Thus, it can be considered that the speed and altitude are integrated into the distance advantage calculation.

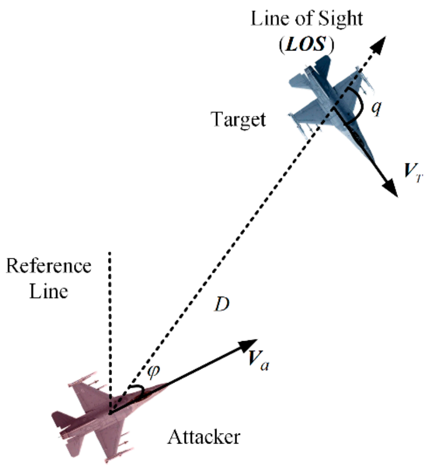

Figure 1. The relative situation of an attacker and a target.

As shown in Figure 1, the relative situation of an attacker and a target can be represented by the line of sight (LOS) information. The LOS information contains several parameters, such as the azimuth angle ϕ, entry angle q, and distance D. To better describe the parameters, the LOS vector is first defined here. A LOS vector is a vector that starts at the attacker's mass center and points to the target's mass center. The azimuth angle ϕ is defined as the angle between the attacker's velocity $\boldsymbol{V}_a$ and the LOS vector. When $\boldsymbol{V}_a$

rotates clockwise relative to $\boldsymbol{LOS}$, ϕ is positive; otherwise, ϕ is negative. The entry angle q is defined as the angle between the target's velocity $\boldsymbol{V}_T$ and the LOS vector. q is positive when $\boldsymbol{V}_T$ rotates clockwise relative to $\boldsymbol{LOS}$; otherwise, it is negative. D is the distance between the attacker's mass center and the target's mass center.

In the following, the attacking side—i.e., the side on which the target assignment algorithm is running—is regarded as a red side, while the other side—i.e., the side to which targets belong—is regarded as a blue side. The aircraft set of the red side is denoted by $\boldsymbol{R} = \{r_1, r_2, \ldots, r_N\}$, and the target set of the blue side is denoted by $\boldsymbol{B} = \{b_1, b_2, \ldots, b_M\}$, where r_i is the ith aircraft on the red side, b_j is the jth target on the blue side, N is the number of the red-side aircraft, and M is the number of the blue-side targets. Further, the azimuth angle advantage, entry angle advantage, and distance advantage are denoted by A_φ, A_q, and A_D, respectively. The three advantages are calculated as shown in Equations (1)–(3), which are derived from our previous work [1]:

$$A_\varphi = \begin{cases} 1 - \frac{|\varphi|}{5\varphi_{\max}} & 0 \le |\varphi| \le \varphi_{\max} \\ 0 & \varphi_{\max} \le |\varphi| \le 180 \end{cases}, \tag{1}$$

where $\phi_{\max}$ is the maximum allowable off-axis launch angle. Equation (1) indicates that, when ϕ is smaller than $\phi_{\max}$, the attacker can easily satisfy missile launching conditions, and thus the attacker has a larger azimuth angle advantage; otherwise the attacker cannot launch missiles.

$$A_q = e^{-\frac{(180-|q|)\pi}{180}}, -180 \le q \le 180, \tag{2}$$

This equation indicates that the attacker has the greatest entry angle advantage when q = 180°, while, when q = 0°, the attacker has the minimum entry angle advantage. The design of the equation is based on the characteristics of the missile attack zone.

$$A_D = \begin{cases} 1 & D < D_{NEZ\max} \\ 2^{-\frac{D - D_{NEZ\max}}{D_{WEZ\max} - D_{NEZ\max}}} & D_{NEZ\max} \le D \le D_{WEZ\max} \\ 0 & D > D_{WEZ\max} \end{cases}, \tag{3}$$

where $D_{WEZ\max}$ and $D_{NEZ\max}$ denote the far boundaries of the missile non-escape zone (NEZ) and weapon-engagement zone (WEZ), respectively. If the target is inside the attacker's NEZ, no matter how the target maneuvers, it will be hit by the attacker's missile, and the distance advantage equals 1 in such a situation. If the target is inside the WEZ and outside the NEZ, the distance advantage will decrease as D increases. Otherwise, if the target is outside the WEZ, the attacker cannot pose a threat to the target. For more details on the advantage functions, please refer to our previous work [1].

The weighted multiplication of the three advantages yields the integrated advantage, which is given by:

$$A = A_\varphi^{\gamma_1} A_q^{\gamma_2} A_D, \tag{4}$$

where γ_1 and γ_2 are weighted coefficients satisfying the condition $\gamma_1 + \gamma_2 = 1$.

Equation (4) can be used to calculate both the advantage of r_i with respect to b_j and the threat of b_j to r_i, which are denoted by A_{ij} and T_{ji}, respectively. As the value of A_{ij} is in the range of [0, 1], A_{ij} is considered the probability P_{ij} that r_i fires a missile at b_j and hits; i.e., it is assumed that, the greater the advantage of r_i with respect to b_j is, the higher the probability that it fires a missile that will hit b_j.

For the case where multiple missiles from the red side attack b_j cooperatively, the hit probability P_j can be calculated by:

$$P_j = 1 - \Pi_{i=1}^{N} \left(1 - P_{ij}\right)^{x_{ij}}, \tag{5}$$

where x_{ij} is the number of missiles assigned to b_j by r_i; all x_{ij} values form a two-dimensional assignment matrix, which is denoted by $\boldsymbol{X}$ in this study.

The total threat to the red side imposed by b_j can be calculated by:

$$T_j = \sum_{i=1}^{N} T_{ji}. \tag{6}$$

Then, for the red side, the profit from attacking b_j can be calculated by:

$$g_j = P_j T_j. \tag{7}$$

Ultimately, the target assignment model in cooperative BVR air combat can be defined as follows:

$$\begin{aligned} &\max \sum_{j=1}^{M} g_j \\ s.t.\ &\sum_{i=1}^{N} x_{ij} \leqslant \mathbb{M}_j(\boldsymbol{x}_{\cdot,j}, P_{\mathrm{TH}}), \\ &\sum_{j=1}^{M} x_{ij} \leqslant L_t, \\ &x_{ij} \in \{0, 1, 2, \cdots, L_t\}. \end{aligned} \tag{8}$$

Specifically, the ultimate goal of the target assignment is to maximize the total profit of the red side while satisfying the corresponding constraints. In Equation (8), the first constraint indicates that the number of missiles assigned to a target cannot exceed $\mathbb{M}_j$, where $\mathbb{M}_j$ indicates the maximum number of missiles that can be assigned to b_j; it is a function of vector $\boldsymbol{x}_{\cdot,j}$ composed of the elements in column j of matrix $\boldsymbol{X}$ and the hit probability threshold P_{TH}. If $\exists\ i \in \{1, 2, \cdots, N\}$, $x_{ij} > 0$ and $P_{ij} \geq P_{\mathrm{TH}}$, then $\mathbb{M}_j$ equals 1, otherwise $\mathbb{M}_j$ equals 2. The second constraint indicates that the number of missiles an aircraft can assign cannot exceed L_t, where L_t indicates the number of missiles each aircraft can carry.

3. Distributed Target Assignment Algorithm Design

In this section, the MTCBAA method is proposed to solve the target assignment model described in Section 2 in a distributed manner. The MTCBAA was developed based on the CBAA. Therefore, to describe the MTCBAA better, a brief introduction to the CBAA is provided in Section 3.1. The detailed design of the MTCBAA is described in Section 3.2. In Section 3.3, the minimum optimization performance of the MTCBAA is proven by demonstrating the minimum optimization performance of a centralized target assignment algorithm, since the centralized algorithm is equivalent to the MTCBAA.

3.1. Brief Introduction to CBAA

The CBAA represents a distributed target assignment algorithm that can solve one-to-one target assignment problems. Each iteration of the CBAA consists of two phases: the auction and consensus phases.

In the auction phase, based on the greedy principle, agents select targets with the greatest bidding value for themselves from the set of assignable targets as their bidding targets; an assignable target is a target that has not been assigned yet. The index and bidding value of the selected bidding targets are two variables integrated into an agent. Then, the index and bidding value determined in the auction phase are carried to the consensus phase by the agent.

In the consensus phase, agents exchange their estimated greatest bidding values with the other agents they are connected with. Then, based on the estimated greatest bidding values, bidding target index, and bidding value of the bidding target, agents can determine whether they win the bid for their selected targets. If an agent wins the bid, it is assigned to the selected target in the current iteration; otherwise, the agent proceeds to the next iteration to bid on other assignable targets. During the consensus phase, the estimated greatest bidding values need to be updated.

The two phases are iterated cyclically until the target assignment is completed. The specific operation of the CBAA is described in a previous study [18].

3.2. MTCBAA

The CBAA can solve only one-to-one target assignment problems, where each agent can select at most one target, and each target can be assigned to at most one agent. However, it cannot be applied to the multi-target attack problem of a single aircraft and the problem of multiple aircraft attacking the same target cooperatively in multi-aircraft cooperative BVR air combat, and thus is not consistent with the target assignment model defined by Equation (8).

In this section, an improved distributed target assignment algorithm for the multi-aircraft cooperative BVR air combat target assignment problem given by Equation (8), the MTCBAA, is introduced.

The MTCBAA is developed based on the CBAA, and its iterations also consist of the auction and consensus phases. In the MTCBAA, each agent r_i, $i \in \{1, 2, \ldots, N\}$, stores six local variables. The first variable is the bidding value vector $\boldsymbol{c}_i$, whose element c_{ij} denotes the bidding value of r_i for target b_j. The second variable is the assignment result vector $\boldsymbol{a}_i$, and its element a_{ij} denotes the number of missiles assigned to b_j by r_i; $\boldsymbol{a}_i$ is initialized as a zero vector. At the end of the consensus phase of each iteration, if r_i wins the bid on b_j, then $a_{ij} = a_{ij} + 1$; otherwise, a_{ij} remains the same. a_{ij} can be cumulative over multiple iterations, and its final value at the end of the assignment is not necessarily just 0 or 1. The third variable is the estimated greatest bidding value vector $\boldsymbol{y}_i$, where element y_{ij} represents the greatest bidding value estimated by r_i that all aircraft on the red side can provide for b_j. Here, the word "estimated" means $\boldsymbol{y}_i$ is a private variable of r_i, and r_i cannot directly know the greatest bidding value of its side for a given target but can only estimate it from the information obtained during the consensus phase $\boldsymbol{y}_i$ is used to determine whether r_i is eligible to bid on the target b_j. Only if r_i's bidding value for the target b_j c_{ij} is larger than y_{ij} is r_i eligible to bid on the target b_j, as shown in lines 8–9 of Algorithm 1. The fourth variable is the target validity vector, and its element h_{ij} satisfies the following expression:

$$h_{ij} = \Pi\left(c_{ij} > y_{ij}\right), \forall j \in \{1, 2, \cdots, M\}, \tag{9}$$

where $\Pi(\cdot)$ is the indicator function, whose value is 1 when the judgment condition in the parentheses is true; otherwise, it has a value of 0.

In this study, b_j is defined as a valid task for r_i when the bid value of r_i for b_j is greater than the estimated greatest bidding value. The fifth variable is the estimated missile assignment vector $\boldsymbol{m}_i$, where an element m_{ij} denotes the number of missiles assigned to b_j estimated by r_i. The definitions of $\boldsymbol{a}_i$ and $\boldsymbol{m}_i$ appear similar, but in reality, they are very different. An element in $\boldsymbol{a}_i$ indicates the number of missiles assigned by r_i to the corresponding target—for example, if a_{ij} equals 2, it indicates that r_i will assign two missiles to attack the target b_j. An element in $\boldsymbol{m}_i$ indicates the total number of missiles assigned to its corresponding target, and the number is estimated by r_i—for example, if m_{ij} equals 1, it indicates that, from the estimation of r_i, the red side assigns a total of one missile to attack the target b_j. $\boldsymbol{m}_i$ is also a private variable of r_i, and r_i cannot directly estimate the total number of missiles assigned to a given target but can only estimate it from the information obtained during the consensus phase, corresponding to line 5 of Algorithm 2.

The last variable is the cooperative decision-making vector $\boldsymbol{f}_i$, whose element f_{ij} is the hit probability for target b_j obtained by the current assignment scheme estimated by r_i. If the hit probability for target b_j does not reach the hit probability threshold P_{TH} and the number of missiles assigned to b_j does not reach two, then b_j will continue to participate in the assignment of the current iteration; otherwise, the assignment to b_j has been completed.

The auction phase of the MTCBAA is described in the following. The procedure of the auction phase of an agent r_i at an iteration t is shown in Algorithm 1.

Algorithm 1: The MTCBAA auction phase of an agent r_i at an iteration t

1: **procedure SELECT TARGET based on** $\boldsymbol{y}_i(t-1)$, $\boldsymbol{a}_i(t-1)$, $\boldsymbol{f}_i(t-1)$, $\boldsymbol{m}_i(t-1)$
2: $\boldsymbol{f}_i(t) = \boldsymbol{f}_i(t-1)$
3: $f'_{ij}(t) = 1 - \left(1 - f_{ij}(t)\right) \cdot \left(1 - P_{ij}\right), \forall j \in \{1, 2, \cdots, M\}$
4: Calculate $c_{ij}(t)$ using Equation (11)
5: $\boldsymbol{y}_i(t) = \boldsymbol{y}_i(t-1)$
6: $\boldsymbol{a}_i(t) = \boldsymbol{a}_i(t-1)$
7: $\boldsymbol{m}_i(t) = \boldsymbol{m}_i(t-1)$
8: $h_{ij} = \Pi\left(c_{ij} > y_{ij}\right), \forall j \in \{1, 2, \cdots, M\}$
9: **if** $\boldsymbol{h}_i(t) \neq 0$ **and** $\sum \boldsymbol{a}_i < L_t$ then
10: $J_i = \text{argmax}_j h_{ij} \cdot c_{ij}$
11: $f_{iJ_i}(t) = f'_{iJ_i}(t)$
12: $y_{iJ_i}(t) = c_{iJ_i}$
13: **end if**:
14: **end procedure**

As shown in step 1 of Algorithm 1, the input parameters of the auction phase of the MTCBAA are $\boldsymbol{y}_i(t-1)$, $\boldsymbol{a}_i(t-1)$, $\boldsymbol{f}_i(t-1)$, and $\boldsymbol{m}_i(t-1)$, which are updated in the previous iteration. Corresponding to steps 2 and 3, the auction phase begins with the calculation of the new hit probability when r_i assigns an additional missile to b_j based on Equation (10):

$$f'_{ij}(t) = 1 - \left(1 - f_{ij}(t)\right) \cdot \left(1 - P_{ij}\right), \forall j \in \{1, 2, \cdots, M\}. \tag{10}$$

Subsequently, corresponding to step 4, by combining the hit probability and Equations (6) and (7), the bidding value of r_i for b_j can be obtained by:

$$c_{ij}(t) = \begin{cases} f'_{ij}(t)T_j - f_{ij}(t)T_j, & \forall j \in \{j | f_{ij} < P_{\text{TH}} \ \& \ m_{ij} < 2\}, \\ 0, & \text{otherwise}. \end{cases}. \tag{11}$$

It can be seen that the bidding value c_{ij} represents an incremental profit for r_i with respect to b_j. In steps 5–7, the estimated greatest bidding value vector $\boldsymbol{y}_i$, the assignment result vector $\boldsymbol{a}_i$, and the estimated missile assignment vector $\boldsymbol{m}_i$ are updated.

In step 8 of the algorithm, the corresponding elements in $\boldsymbol{c}_i$ and $\boldsymbol{y}_i$ are compared to determine whether r_i is eligible to bid on each target. r_i is eligible to bid on a target b_j only if the value of r_i's bid on b_j, c_{ij}, is greater than the greatest bidding value on b_j estimated by r_i, at which point h_{ij} equals 1.

In line 9, whether r_i can still bid is first determined. r_i is considered to have the ability to bid only if the elements in h_i are not all zero; i.e., r_i is eligible to bid on at least one target, and r_i's L_t missiles have not all been assigned. Finally, corresponding to lines 10–12, r_i selects the valid task with the greatest bidding value as its target; meanwhile, the elements corresponding to the target in y_i and $\boldsymbol{f}_i$ should be updated. Then, r_i waits to proceed to the consensus phase.

The procedure of the consensus phase in the MTCBAA of an agent r_i at an iteration t is shown in Algorithm 2.

In the consensus phase of the MTCBAA, corresponding to lines 1 and 2, an agent r_i first exchanges the estimated greatest bidding vector and the cooperative decision-making vector with an agent r_k that satisfies the condition of g_{ik} = 1, where g_{ik} is an element of the adjacency matrix corresponding to the fleet's communication topology; g_{ik} = 1 indicates a communication connection between r_i and r_k.

After that, corresponding to lines 4 to 5, based on the received information and r_i's information, r_i updates the estimated missile assignment vector $\boldsymbol{m}_i$ by determining whether, for each target b_j, at least one red side aircraft has bid on it. If there is at least one aircraft involved in bidding on b_j—i.e., $\exists\ k \in \{1, 2, \ldots, N\}$—then $y_{kj} > 0$; accordingly, the σ_{yj}

obtained from the expression in line 4 is greater than zero, such that the expression in line 5 is equivalent to $m_{ij} = m_{ij} + 1$; otherwise, the expression in line 5 is equivalent to $m_{ij} = m_{ij} + 0$.

Algorithm 2: The MTCBAA consensus phase of an agent r_i at an iteration t

1: SEND $\boldsymbol{y}_i(t), \boldsymbol{f}_i(t)$ to r_k with $l_{ik}(t) = 1$
2: RECEIVE $\boldsymbol{y}_k(t), \boldsymbol{f}_k(t)$ from r_k with $l_{ik}(t) = 1$
3: **procedure UPDATE ASSIGNMENT SCHEME based on** $g_i(t), \boldsymbol{f}_k(t), \boldsymbol{y}_k(t)$, and J_i
4: $\sigma_{y_j} = \Sigma_{k:g_{ik}(t)=1} y_{kj}, \forall j \in \{1, 2, \cdots, M\}$
5: $m_{ij} = m_{ij} + \Pi\left(\sigma_{y_j} > 0\right), \forall j \in \{1, 2, \cdots, M\}$
6: $z_{ij} = \text{argmax}_k g_{ik}(t) \cdot y_{kj}(t), \forall j \in \{1, 2, \cdots, M\}$
7: **if** $z_{iJ_i} = i$ **then**
8: $a_{i,J_i}(t) = a_{i,J_i}(t) + 1$
9: **end if**
10: $f_{ij}(t) = f_{z_{ij},j}, \ \forall j \in \{1, 2, \cdots, M\}$
11: Update $y_{ij}(t)$ using Equation (13)
12: **end procedure**

Corresponding to lines 6–7, r_i can determine whether it wins the bid on the J_ith target; i.e., r_i determines whether the aircraft providing the global greatest bidding value on the J_ith target is r_i. From the expression in line 6, it can be obtained that the aircraft providing the global greatest bidding value for the target b_j is indexed by z_{ij}. Meanwhile, from the auction phase of r_i, it can be obtained that r_i is bidding on the J_ith target. Thus, in line 7, it is only necessary to determine whether $z_{i,Ji}$ is equal to i to know whether r_i wins the bid on the J_ith target. If this holds, in line 8, the corresponding assignment element a_{i,J_i} is increased by 1; otherwise, it remains the same.

In addition, the cooperative decision-making vector and the estimated greatest bidding vector need to be updated in lines 10 and 11, respectively. The cooperative decision-making vector is updated by

$$f_{ij}(t) = f_{z_{ij},j}, \ \forall j \in \{1, 2, \cdots, M\}, \tag{12}$$

where z_{ij} is the index of an aircraft that wins the bid on target b_j in the current iteration.

The estimated greatest bidding vector is updated by:

$$y_{ij}(t) = \begin{cases} g_{iz_{ij}}(t) \cdot \left[1 - \left(1 - f_{ij}(t)\right) \cdot \left(1 - P_{z_{ij}j}\right)\right] \cdot T_j - f_{ij}(t) \cdot T_j, \forall j \in \{j | f_{ij} < P_{\text{TH}} \text{ and } m_{ij} < 2\}, \\ 0, \quad \text{otherwise.} \end{cases}, \tag{13}$$

where, for target b_j, the update of its estimated greatest bidding value represents the prediction of the corresponding value in the next iteration. The prediction is undertaken by subtracting the profit corresponding to b_j at the completion in the current iteration from the profit corresponding to b_j when the aircraft that wins the bid on b_j in the current iteration assigns one more missile for b_j based on the current assignment scheme.

These two phases iterate between each other until the target assignment process is completed. Target assignment is terminated either when the hit probability threshold is reached for all targets or when there are no more missile resources left for further assignment.

3.3. Minimum Optimization Performance of MTCBAA

As shown in Algorithms 1 and 2, the MTCBAA is based on a greedy mechanism, so there is no guarantee that the algorithm can find an optimal assignment solution. However, the MTCBAA has a minimum optimization performance guarantee, which is demonstrated in this section. It should be noted that optimization performance here refers to the optimization of the assignment profit or assignment solution.

To prove the minimum optimization performance guarantee of the MTCBAA, an equivalent centralized target assignment algorithm, which is denoted as the multi-target sequential greedy algorithm (MTSGA), is proposed. The pseudo-code of the MTSGA is given in Algorithm 3. Subsequently, the proof of the minimum optimization guarantee of the MTSGA is provided. The proof of the equivalence of the MTCBAA and MTSGA corresponds to the proof of Theorem 1, and the proof of the minimum optimization performance guarantee of the MTCBAA and MTSGA corresponds to the proof of Theorem 2.

Algorithm 3: The MTSGA

1: $\boldsymbol{R}_1 = \boldsymbol{R}, \boldsymbol{B}_1 = \boldsymbol{B}$
2: $\eta_i = 0, \forall i \in \boldsymbol{R}$
3: $\boldsymbol{J}_i^{(0)} = \phi, \forall i \in \boldsymbol{R}$
4: $\boldsymbol{I}_j^{(0)} = \phi, \forall j \in \boldsymbol{B}$
5: $c_{ij}^{(1)} = c_{ij}\left(\boldsymbol{J}_i^{(0)}\right), \forall (i,j) \in \boldsymbol{R} \times \boldsymbol{B}$
6: $n = 1$
7: **while** assignment is not completed
8: $\quad (i_n^*, j_n^*) = \text{argmax}_{(i,j) \in \boldsymbol{R}_n \times \boldsymbol{B}_n} c_{ij}^{(n)}$
9: $\quad \eta_{i_n^*} = \eta_{i_n^*} + 1$
10: $\quad \boldsymbol{J}_{i_n^*}^{(n)} = \boldsymbol{J}_{i_n^*}^{(n-1)} \oplus_{end} \{j_n^*\}$
11: $\quad \boldsymbol{J}_i^{(n)} = \boldsymbol{J}_i^{(n-1)}, \forall i \neq i_n^*$
12: $\quad \boldsymbol{I}_{j_n^*}^{(n)} = \boldsymbol{I}_{j_n^*}^{(n-1)} \oplus_{end} \{i_n^*\}$
13: $\quad \boldsymbol{I}_j^{(n)} = \boldsymbol{I}_j^{(n-1)}, \forall j \neq j_n^*$
14: $\quad c_{ij}^{(n+1)} = c_{ij}\left(\boldsymbol{J}_i^{(n)}\right), \forall (i,j) \in \boldsymbol{R}_{n+1} \times \boldsymbol{B}_{n+1}$
15: $\quad$ **if** $\eta_{i_n^*} = L_t$, **then**
16: $\quad\quad \boldsymbol{R}_{n+1} = \boldsymbol{R}_n \backslash \{i_n^*\}$
17: $\quad\quad c_{i_n^*,j}^{(n+1)} = 0, \forall j \in \boldsymbol{B}$
18: $\quad$ **else**
19: $\quad\quad \boldsymbol{R}_{n+1} = \boldsymbol{R}_n$
20: $\quad$ **end if**
21: $\quad f_{j_n^*} = f\left(\boldsymbol{I}_{j_n^*}^{(n)}\right)$
22: $\quad$ **if** $f_{j_n^*} > P_{\text{TH}}$ **or** $\left|\boldsymbol{I}_{j_n^*}^{(n)}\right| >= 2$, **then**
23: $\quad\quad \boldsymbol{B}_{n+1} = \boldsymbol{B}_n \backslash \{j_n^*\}$
24: $\quad\quad c_{i,j_n^*}^{(n+1)} = 0, \forall i \in \boldsymbol{R}_{n+1}$
25: $\quad$ **else**
26: $\quad\quad \boldsymbol{B}_{n+1} = \boldsymbol{B}_n \backslash \{j_n^*\}$
27: $\quad$ **end if**
28: $\quad n = n + 1$
29: **end while**

In Algorithm 3, an aircraft set $\boldsymbol{R}_1$ and a target set $\boldsymbol{B}_1$ are assigned in the first iteration are initialized first in line 1. Then, the missile consumption η_i for an aircraft r_i, $\forall\, i \in \boldsymbol{R}_1$ is set to zero in line 2. An aircraft r_i is defined to have a target set $\boldsymbol{J}_i$; a target b_j is defined to have an aircraft set $\boldsymbol{I}_j$. The elements in the target set $\boldsymbol{J}_i$ denote indexes of targets assigned to r_i obtained in the MTSGA iterations sorted in the same order they have been added to the set. The elements in the aircraft set $\boldsymbol{I}_j$ denote indexes of aircraft that assign at least one missile to b_j; these elements are also obtained in the MTSGA iterations and sorted in the same order they have been added to the set. All target sets $\boldsymbol{J}_i$, $\forall\, i \in \boldsymbol{R}$ and aircraft sets $\boldsymbol{I}_j$, $\forall\, j \in \boldsymbol{B}$ are initialized as empty sets in lines 3 and 4. The bidding value of each aircraft to each target also needs to be initialized. The initialization work of the bidding values in Algorithm 3 is done in line 5, and the bidding values are initialized and updated by:

$$c_{ij}(\boldsymbol{J}_i) = \begin{cases} g_j\left(\boldsymbol{I}_j \oplus_{end} \{i\}\right) - g_j\left(\boldsymbol{I}_j\right), \ \forall j \in \left\{j \middle| f_j < P_{\text{TH}}\ \&\ \left|\boldsymbol{I}_j\right| < 2\right\} \\ 0, \qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases}, \tag{14}$$

where $\oplus_{end}$ is the operator that combines the sets before and after itself. Suppose there are two sets—namely, $\boldsymbol{A}$ = {1, 2, 3} and $\boldsymbol{B}$ = {4, 5, 6}; then, $\boldsymbol{A} \oplus_{end} \boldsymbol{B}$ = {1, 2, 3, 4, 5, 6}. $|\cdot|$ is the operator to count the number of elements in a set, and $g_j(\boldsymbol{I}_j)$ is the profit obtained by attacking b_j according to the assignment scheme $\boldsymbol{I}_j$, and it is calculated by:

$$g_j(\boldsymbol{I}_j) = \left[1 - \prod\nolimits_{k=1}^{|\boldsymbol{I}_j|} \left(1 - P_{\boldsymbol{I}_j(k),j}\right)\right] \cdot T_j, \tag{15}$$

where $\boldsymbol{I}_j(k)$ is the kth element of $\boldsymbol{I}_j$.

It can be easily noticed that the calculation expression of the profit of attacking b_j in Equation (15) is equivalent to Equation (7), and the calculation expression of the bidding value of r_i for b_j in Equation (14) is equivalent to Equation (11).

At the beginning of the next MTSGA iteration, based on the bidding value matrix in the current iteration, corresponding to line 8, the aircraft and target indexes i_n* and j_n* corresponding to the largest bidding value in the bidding value matrix are selected, and then the missile consumptions of the i_n*th aircraft (line 9), target sets (lines 10 and 11), aircraft sets (lines 12 and 13), and the bidding value matrix (line 14) are updated.

Lines 15–20 correspond to the update of the set $\boldsymbol{R}_{n+1}$, which is the set of aircraft entering the next assignment iteration. The first step is to determine in line 15 whether the missile consumption of the i_n*th aircraft has reached the upper limit of the number of missiles it carries, L_t. If the upper limit is reached, then $\boldsymbol{R}_n$ after removing i_n* should be assigned to the $\boldsymbol{R}_{n+1}$ according to line 16; otherwise, $\boldsymbol{R}_n$ can be assigned directly to the $\boldsymbol{R}_{n+1}$ according to line 19. In addition, note that if i_n* should be removed from the set, then the bid value of the i_n*th aircraft for each target should also be set to 0 according to line 17, so that the i_n*th aircraft will no longer participate in the subsequent bids.

The cooperative decision-making variable of the j_n*th target is updated in line 21.

Lines 22–27 correspond to the update of the set $\boldsymbol{B}_{n+1}$, which is the set of targets entering the next assignment iteration. The first step is to determine in line 22 whether the assignment to the j_n*th target has been completed. If it holds, then $\boldsymbol{B}_n$, after removing j_n*, should be assigned to $\boldsymbol{B}_{n+1}$, according to line 23; otherwise, $\boldsymbol{B}_n$ can be assigned directly to $\boldsymbol{B}_{n+1}$, according to line 26. In addition, note that if j_n* is removed from the set, then each aircraft's bid value for the j_n*th target should be set to 0 according to line 24, so that no aircraft will bid on the j_n*th target in the subsequent bids.

The iterations continue until the assignment is completed, as shown in line 7; the target assignment termination conditions are the same as for the MTCBAA; that is, target assignment is terminated either when the hit probability threshold is reached for all targets or when there are no more missile resources left for further assignment.

The equivalence theorem of the MTCBAA and MTSGA is given and proven in the following.

Theorem 1. *Under the same situational conditions, the MTCBAA and MTSGA generate the same target assignment scheme for the multi-aircraft cooperative BVR air combat target assignment problem given by Equation (8).*

Proof of Theorem 1. The proof of Theorem 1 is obtained using mathematical induction. Assume that the target set of r_i in the nth iteration is $\boldsymbol{J}_i^{(n)} = \left\{j_{k_1}^*, j_{k_2}^*, \cdots, j_{k_{Lin}}^*\right\}$, where $k_1 < k_2 < \cdots < k_{Lin}$ and $L_{in} = \left|\boldsymbol{J}_i^{(n)}\right|$. Then, the first element of $\boldsymbol{J}_i^{(n)}$ is determined by:

$$c_{i,j_{k_1}^*}^{(k_1)} = c_{i,j_{k_1}^*}(\phi) = \max_{j \in \boldsymbol{B}_{k_1}} c_{i,j}(\phi), \tag{16}$$

where $\boldsymbol{B}_{k_1}$ is the set of aircraft updated in the $(k_1 - 1)$th iteration of the MTSGA to enter the k_1th iteration of assignment, corresponding to rows 1 and 23 in Algorithm 3. In contrast, the MTCBAA assigns the first target to the aircraft r_i as follows:

$$J_i = \operatorname{argmax} c_{ij}(\phi) \times \Pi\left(c_{ij}(\phi) > y_{ij}\right). \tag{17}$$

Notice that $\forall j \notin \boldsymbol{B}_{k_1}$ and, because $\boldsymbol{J}_i$ is currently an empty set, target b_j has been assigned to aircraft other than r_i, so $c_{ij}(\phi) \leq y_{ij}, \forall j \notin \boldsymbol{B}_{k_1}$ holds. The problem of searching for the target corresponding to the maximum bidd ing value in set $\boldsymbol{B}$ for aircraft r_i, which is given by Equation (17), degenerates to the problem of searching for such a target in a smaller set $\boldsymbol{B}_{k_1}$. This matches the method of determining the target given by Equation (16); thus, $J_i = j^*_{k_1}$.

Next, assume that the MTSGA obtains the same first l targets for each aircraft r_i as the MTCBAA. Then, the (l + 1)th target of r_i in the MTSGA is determined by:

$$c^{(k_{l+1})}_{i,j^*_{k_{l+1}}} = c_{i,j^*_{k_{l+1}}}\left(\boldsymbol{J}_i^{(1:l)}\right) = \max_{j \in \boldsymbol{B}_{k_{l+1}}} c_{i,j}\left(\boldsymbol{J}_i^{(1:l)}\right), \tag{18}$$

where $\boldsymbol{J}_i^{(1:l)}$ represents the target set of r_i consisting of the first l targets.

For a target b_j, when its index j satisfies $j \notin \boldsymbol{B}_{k_{l+1}}$, corresponding to Equation (11) or (14), the new bidding value of r_i for b_j equals zero. Then, it holds that:

$$c_{ij}\left(\boldsymbol{J}_i^{(1:l)}\right) \times \Pi\left(c_{ij}\left(\boldsymbol{J}_i^{(1:l)}\right) > y_{ij}\right) = 0, \forall j \notin \boldsymbol{B}_{k_{l+1}}. \tag{19}$$

In contrast, the (l + 1)th target of r_i in the MTCBAA is determined by:

$$\max_{j \in \boldsymbol{B}} c_{ij}\left(\boldsymbol{J}_i^{(1:l)}\right) \times \Pi\left(c_{ij}\left(\boldsymbol{J}_i^{(1:l)}\right) > y_{ij}\right). \tag{20}$$

Substituting Equation (19) into Equation (20), Equation (20) can be further simplified as follows:

$$\begin{aligned} &\max_{j \in \boldsymbol{B}} c_{ij}\left(\boldsymbol{J}_i^{(1:l)}\right) \times \Pi\left(c_{ij}\left(\boldsymbol{J}_i^{(1:l)}\right) > y_{ij}\right) \\ &= \max_{j \in \boldsymbol{B}_{k_{l+1}}} c_{ij}\left(\boldsymbol{J}_i^{(1:l)}\right) \times \Pi\left(c_{ij}\left(\boldsymbol{J}_i^{(1:l)}\right) > y_{ij}\right) \\ &= \max_{j \in \boldsymbol{B}_{k_{l+1}}} c_{ij}\left(\boldsymbol{J}_i^{(1:l)}\right). \end{aligned} \tag{21}$$

The simplification result of Equation (21) is the same as that of Equation (18). Thus, it can be concluded that, if the first l targets of r_i obtained by the MTSGA and the MTCBAA are identical, then the (l + 1)th targets of r_i obtained by the two algorithms are also identical, which proves Theorem 1. □

Next, the minimum optimality guarantee theorem for the MTCBAA is given and proven.

Theorem 2. *The MTCBAA can guarantee at least 50% global optimality.*

Proof of Theorem 2. According to Theorem 1, the MTCBAA is equivalent to the MTSGA, so it is sufficient to prove that the MTSGA can guarantee at least 50% global optimality.

Suppose a target b_j has been assigned once, and the hit probability $P_{i^*_{k1},j}$ of b_j does not meet the hit probability threshold P_{TH}. When b_j has another opportunity to be assigned, because the remaining assignable resources at this time are fewer compared to the time when b_j was assigned for the first time, the maximum for the hit probability that each remaining assignable aircraft can provide for b_j, which is denoted as $P_{i^*_{k2},j}$, satisfies the inequality $P_{i^*_{k2},j} \leq P_{i^*_{k1},j}$; k_1 is the index of the iteration in which b_j was assigned for the first time, k_2 is the index of the iteration in which b_j is assigned for the second time, $i^*_{k_1}$ is the index of the aircraft that provides the maximum bidding value on b_j in the k_1th iteration of the MTSGA, and $i^*_{k_2}$ is the index of the aircraft that provides the maximum bidding value on b_j in the k_2th iteration of the MTSGA. Then, the maximum bidding value on b_j at its second assignment subtracted from the maximum bidding value on b_j at its first assignment satisfies Equation (22):

$$\begin{aligned} & c_{i^*_{k2},j} - c_{i^*_{k1},j} \\ = & \left[1 - \left(1 - P_{i^*_{k2},j}\right)\left(1 - P_{i^*_{k1},j}\right) - P_{i^*_{k1},j}\right] \cdot T_j - P_{i^*_{k1},j} \cdot T_j \\ = & \left(P_{i^*_{k2},j} - P_{i^*_{k2},j} \cdot P_{i^*_{k1},j} - P_{i^*_{k1},j}\right) \cdot T_j \\ \leq & \ 0 \end{aligned} . \tag{22}$$

In addition, from the operation of the MTSGA, it is evident that, for each of the other targets assigned after the first assignment of target b_j, regardless of whether this is its first or second assignment, its maximum bidding value is less than or equal to $c_{i^*_{k1},j}$.

For the convenience of presentation, in the following proof of Theorem 2, aircraft i^*_k is denoted as aircraft k, and target j^*_k is denoted as target k. In other words, the indexes in the MTSGA are denoted as follows:

$$i^*_k = k \quad j^*_k = k \quad \forall k \leq N_{\min}. \tag{23}$$

The subscripts for the rest of the variables need to be replaced accordingly. For instance, $c_{i^*_k,j^*_k}$ can be denoted as c_{kk}. Then, in the MTSGA, the following inequality holds:

$$c_{ii} \geq c_{jj}, \text{if } i < j. \tag{24}$$

As the aircraft and target pairs are selected based on the greedy principle in each iteration of the MTSGA, Equation (25) holds in parallel:

$$\begin{array}{l} c_{ii} \geq c_{ij}, \forall i, \forall j > i \\ c_{ii} \geq c_{jj} \geq c_{ji}, \forall i, \forall j > i \end{array} . \tag{25}$$

Assume the number of iterations of the MTSGA is denoted by $N_{\min}$. Then, $N_{\min}$ is between M and $2M$ when the missile resources are sufficient; i.e., the number of missiles is sufficient to complete the first assignment to all targets and the second assignment to targets that did not reach the hit probability threshold in the first assignment. In this study, M is the number of targets; $N_{\min} = M$ when the hit probability threshold is met by assigning each target only once; $N_{\min} = 2M$ when each target needs to be assigned twice; otherwise, $M < N_{\min} < 2M$. When the number of missiles does not satisfy the aforementioned conditions, then $N_{\min} = N \cdot L_t$, where N is the number of aircraft, and L_t is the number of missiles each aircraft loads. The value of $N_{\min}$ is determined by:

$$N_{\min} = \min\{N \cdot L_t, 2M\}. \tag{26}$$

That is, in the case of a sufficient number of missiles, it is assumed that the MTSGA is required to iterate $2M$ times, while in reality, the actual number of iterations required by the MTSGA may be less than $2M$. Assume that the actual number of iterations required by the MTSGA is denoted by N_{need}. For $(2M - N_{\min})$ additional iterations, the assignment results are generated as follows. Select $(2M - N_{\min})$ missiles that have not been assigned yet. Randomly arrange the indexes of their carrier aircraft and use them to form a new ordered set called the random aircraft set. Then, randomly arrange the indexes of targets that reach the hit probability threshold in the first assignment and form a new ordered set, called the random target set; the length of both sets is $(2M - N_{\min})$. Combine the elements in the two sets in order; the aircraft–target pairs are obtained in the additional $(2M - N_{\min})$ iterations of the MTSGA. As $(2M - N_{\min})$ targets have already reached the hit probability threshold condition in the first assignment, according to Equation (14), all the maximum bidding values corresponding to the additional $(2M - N_{\min})$ iterations of the MTSGA equal 0. Thus, the MTSGA with $2M$ iterations is equivalent to the MTSGA with N_{need} iterations and satisfies Equations (24) and (25). When $N_{\min} = N \cdot L_t$, there is no need to make a further assumption to change the number of iterations of the MTSGA. Hence, it can be concluded that the MTSGA with $N_{\min}$ iterations defined by Equation (26) is equivalent to the MTSGA that performs the required number of iterations for a particular problem.

Denote the optimized profit obtained by the MTCBAA or the MTSGA as OPT_{MTCBAA}, which can be calculated by:

$$OPT_{\text{MTCBAA}} = \sum_{i=1}^{N_{\min}} c_{ii}. \tag{27}$$

Next, assume that arbitrarily swapping the targets of aircraft i and j in the assignment scheme obtained by the MTSGA has no effect on the bidding value of an aircraft k on a target k, $\forall k \neq i, j$. In this study, this assumption is denoted as the target order-bidding value independent (TOBI) assumption. Further, assume that the assignment scheme obtained by the MTSGA is the worst assignment scheme; i.e., any swapping of targets of the two aircraft can improve the total assignment profit. For instance, for aircraft i and j, where $j > i$, because Equation (25) holds, after swapping their targets, the sum of profits of the two aircraft satisfies Equation (28):

$$c_{ij} + c_{ji} \leq c_{ii} + c_{ii} = 2c_{ii}. \tag{28}$$

Furthermore, when Equation (29) holds, the equal sign in Equation (28) also holds:

$$c_{ij} = c_{ji} = c_{ii}. \tag{29}$$

Therefore, if Equation (29) holds, aircraft i and j can achieve the maximum improvement in their profit sum by swapping their targets.

Next, assume that Equation (29) holds for all pairs of aircraft in the assignment scheme; i.e., the following equalities hold:

$$\begin{array}{l} c_{ii} = c_{ij}, \forall i, \forall j > i, \\ c_{ii} = c_{ji}, \forall i, \forall j > i. \end{array} \tag{30}$$

Then, for the current assignment scheme obtained by the MTSGA, the maximum improvement in the overall target assignment profit can be achieved by a series of target exchange operations between aircraft pairs. One of the ways to achieve the maximum improvement in the overall target assignment profit is given by Equation (31):

$$J_i^* = \begin{cases} N_{\min} - i + 1, & \text{if } i \in \{1, 2, \cdots, N_{\min}\} \\ 0, & \text{otherwise} \end{cases}. \tag{31}$$

Equation (31) indicates that an aircraft i exchanges its target with an aircraft $(N_{\min} - i + 1)$. In this way, the first $\lceil N_{\min}/2 \rceil$ aircraft can maintain the same profits as in the MTSGA, while the next $\lfloor N_{\min}/2 \rfloor$ can achieve the maximum profit improvements. In Equation (31), $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the upward and downward rounding operators, respectively.

Denote the global optimal target assignment profit when the TOBI assumption holds by OPT_{TOBI}; then, OPT_{TOBI} satisfies Equation (32):

$$\begin{aligned} OPT_{\text{TOBI}} &= \sum_{i=1}^{\lceil N_{\min}/2 \rceil} c_{ii} + \sum_{i=\lceil N_{\min}/2 \rceil + 1}^{N_{\min}} c_{(N_{\min}-i+1),(N_{\min}-i+1)} \\ &= 2 \times \sum_{i=1}^{\lfloor N_{\min}/2 \rfloor} c_{ii} + \sum_{i=\lfloor N_{\min}/2 \rfloor}^{\lceil N_{\min}/2 \rceil} c_{ii} \\ &\leq 2 \times \sum_{i=1}^{N_{\min}} c_{ii} = 2OPT_{\text{MTCBAA}} \end{aligned} \tag{32}$$

It should be noted that the TOBI assumption does not always hold for the target assignment problem defined by Equation (8). For a target b_j that needs to be assigned twice in the MTSGA, if its first assignment iteration is advanced by the target exchange shown in Equation (31)—i.e., the index of iteration corresponding to the first assignment of b_j in the MTSGA is k_1 and the index of iteration corresponding to the second assignment of b_j in MTSGA is k_2—after the target swapping corresponding to Equation (31), the index of iteration corresponding to the first assignment of b_j becomes $k'_1 = N_{\min} - k_2 + 1$, and $k'_1 < k_1$, which correspond to two cases. In the first case, the hit probability to b_j has met the hit probability threshold through the k'_1 iteration of the assignment, and $c_{i^*_{k_1},b_j}$ becomes 0 due to the constraints in Equation (14). In the second case, the hit probability for b_j still does not meet the hit probability threshold through the k'_1 assignment iteration, so the new bidding value on target b_j in the k_1th iteration is given by:

$$\begin{aligned} &c'_{i^*_{k_1},b_j} \\ &= \left[1 - \left(1 - P_{i^*_{k'_1},b_j}\right)\left(1 - P_{i^*_{k_1},b_j}\right)\right] \cdot T_{b_j} - P_{i^*_{k'_1},b_j} \cdot T_{b_j} \quad . \\ &= \left(P_{i^*_{k_1},b_j} - P_{i^*_{k_1},b_j} \cdot P_{i^*_{k'_1},b_j}\right) \cdot T_{b_j}. \end{aligned} \tag{33}$$

The original bidding value on target b_j in the k_1th iteration is given by:

$$c_{i^*_{k_1},b_j} = P_{i^*_{k_1},b_j} \cdot T_{b_j}. \tag{34}$$

Subtracting Equation (34) from Equation (33), we have:

$$c'_{i^*_{k_1},b_j} - c_{i^*_{k_1},b_j} = -P_{i^*_{k_1},b_j} \cdot P_{i^*_{k'_1},b_j} \cdot T_{b_j} \leq 0. \tag{35}$$

Thus, due to the introduction of the hit probability threshold constraint, the bidding value in the k_1th iteration is reduced in both cases compared to the TOBI case. This leads to the conclusion that the global optimal target assignment profit *MTOPT*, considering the hit probability threshold constraint, is upper-bounded by the global optimal target assignment profit when the TOBI assumption holds, so it holds that:

$$MTOPT \leq OPT_{\text{TOBI}} \leq 2OPT_{\text{MTCBAA}}. \tag{36}$$

Consequently, the MTCBAA can guarantee at least 50% global optimality, which proves Theorem 2 $\square$.

4. Experiments and Discussion

In Section 4.1, the operation of the MTCBAA is illustrated in an example. Section 4.2 describes the verification of the optimization efficiency and running time of the MTCBAA through a comparison experiment between the MTCBAA and the centralized target assignment algorithm based on the artificial immune algorithm. The comparison experiment was designed using the Monte Carlo method and Matlab parallel-computing toolbox (PCT). Finally, Section 4.3 describes the verification of the practicality and real-time performance of the MTCBAA through an experiment deploying the algorithm with a distributed semi-physical simulation platform.

Unless specified otherwise, all experiments used the air combat scenario with four red-side aircraft with multi-target attack capabilities cooperatively attacking four blue-side targets; i.e., N = 4 and M = 4. The number of missiles carried by each aircraft was set to L_t = 4; the maximum allowable off-axis launch angle of a missile was $\varphi_{\max}$ = 40°; the attack zone boundary distance of a missile was $D_{WEZ\max}$ = 80 km; the non-escape zone boundary distance of a missile was $D_{NEZ\max}$ = 60 km. In Equation (4), $\gamma_1 = 0.75$, $\gamma_2 = 0.25$, and the hit probability threshold was set to P_{TH} = 0.9.

The experimental configuration of experiments described in Sections 4.1 and 4.2 is presented in Table 1.

Table 1. Experimental configuration.

Parameter	Value
CPU	Intel(R) Core(TM) i7-10710U @ 1.10 GHz, 6 Cores
RAM	16 GB
Matlab	Matlab 2015b

4.1. Example Experiment

The situation information of the blue side relative to the red side was set in the example experiment as shown in Table 2.

Table 2. Situation information of the blue side relative to the red side.

Red Side	Blue Side	φ_{ij} (°)	q_{ij} (°)	D_{ij} (km)
r_1	b_1	−4	182	61
	b_2	6	179	72
	b_3	−6	175	69
	b_4	2	177	65
r_2	b_1	3	178	75
	b_2	−3	183	71
	b_3	4	184	62
	b_4	−6	177	64
r_3	b_1	−6	179	68
	b_2	−4	180	66
	b_3	5	183	71
	b_4	−1	181	76
r_4	b_1	−2	174	63
	b_2	4	178	79
	b_3	2	181	80
	b_4	6	179	70

In the situation shown in Table 2, the advantage matrix of the red side relative to the blue side was given by:

$$\boldsymbol{W} = \begin{bmatrix} 0.9431 & 0.6420 & 0.7001 & 0.8237 \\ 0.5828 & 0.6665 & 0.9031 & 0.8398 \\ 0.7375 & 0.8000 & 0.6615 & 0.5697 \\ 0.8714 & 0.5054 & 0.4941 & 0.6881 \end{bmatrix}, \tag{37}$$

where the elements in the ith row and jth column denote the advantage value of an aircraft r_i for a target b_j.

The threat matrix of the blue side to the red side was given by:

$$\boldsymbol{T} = \begin{bmatrix} 0.9421 & 0.5825 & 0.7355 & 0.8732 \\ 0.6403 & 0.6665 & 0.7982 & 0.5049 \\ 0.6997 & 0.9031 & 0.6607 & 0.4938 \\ 0.8242 & 0.8385 & 0.5697 & 0.6863 \end{bmatrix}, \tag{38}$$

where the elements in the jth row and ith column denote the threat values of a target b_j to an aircraft r_i.

The intermediate results obtained during the MTCBAA operation are shown in Table 3.

Table 3. Intermediate results of the MTCBAA.

Iteration Index	Bidding Value Matrix	Assignment Matrix	Hit Probability Vector
1	$\begin{bmatrix} 2.96 & 1.68 & 1.93 & 2.40 \\ 1.83 & 1.74 & 2.49 & 2.45 \\ 2.31 & 2.09 & 1.82 & 1.66 \\ 2.73 & 1.32 & 1.36 & 2.01 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.94 \\ 0 \\ 0.90 \\ 0 \end{bmatrix}$
2	$\begin{bmatrix} 0 & 1.68 & 0 & 2.40 \\ 0 & 1.74 & 0 & 2.45 \\ 0 & 2.09 & 0 & 1.66 \\ 0 & 1.32 & 0 & 2.01 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.94 \\ 0.80 \\ 0.90 \\ 0.84 \end{bmatrix}$
3	$\begin{bmatrix} 0 & 0.34 & 0 & 0.38 \\ 0 & 0.35 & 0 & 0.39 \\ 0 & 0.42 & 0 & 0.27 \\ 0 & 0.26 & 0 & 0.32 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.94 \\ 0.97 \\ 0.90 \\ 0.96 \end{bmatrix}$

In the MTCBAA, in the auction phase of the first iteration, r_1, r_2, r_3, and r_4 select b_1, b_3, b_1, and b_1, respectively, as targets based on the greedy principle. Then, in the consensus phase, as r_1 provided the maximum bidding value for b_1, r_1 won the bid for b_1; r_3 and r_4 were not assigned any target in the first iteration due to bid failure; r_2 was assigned to b_3 as desired due to separate participation in the auction of b_3. At the end of the first iteration, the hit probabilities for b_1 and b_3 met the hit probability threshold requirement, so these two targets were not involved in the assignment of the subsequent iterations. In the second iteration, r_1–r_4 participated in the assignment of b_2 and b_4; the specific process was similar to the first iteration, so it is not described here. After the completion of the second iteration, as the hit probabilities to b_2 and b_4 did not reach the hit probability threshold requirement and the numbers of missiles assigned to b_2 and b_4 were less than two, b_2 and b_4 entered the third assignment iteration of the MTCBAA. The process of the third assignment iteration was similar to that in the first two iterations. The bidding values calculated by Equation (11) denote the incremental profit. At the end of the third iteration, all the target assignments were completed, and the total profit was 10.79.

The target assignment results obtained by the MTSGA were the same as those obtained by the MTCBAA, but the running time of the MTSGA was 0.018 s, which was twice that of the MTCBAA, which was 0.01 s. This was because the MTSGA performed more iterations than the MTCBAA.

4.2. Monte Carlo Comparison Experiment

Next, a comparison experiment between the MTCBAA and a centralized target assignment algorithm based on the artificial immune algorithm [25] was conducted to verify the optimization efficiency and the running time of the MTCBAA.

As the MTCBAA is a distributed target assignment algorithm, it is challenging to compare it with a centralized target assignment algorithm in the same computing power environment. The distributed operation process of the MTCBAA was simulated for multiple aircraft using the PCT toolbox to deploy the MTCBAA to multiple workers, while the client simulated the communication network responsible for data exchange between multiple aircraft, as shown in Figure 2. Meanwhile, the centralized target assignment algorithm based on an artificial immune algorithm was deployed to run on a single core.

The comparison of the two algorithms was conducted using the Monte Carlo method. The azimuth angle φ_{ij} was in the range of $[-40^\circ, 40^\circ]$, the entry angle q_{ij} was in the range of $[140^\circ, 220^\circ]$, and the distance D_{ij} was in the range of [60, 80] km; their values were randomly set to compare the profits and running time of the two algorithms. The experiment included a total of 1000 iterations.

The profit comparison results of the two algorithms are shown in Figure 3.

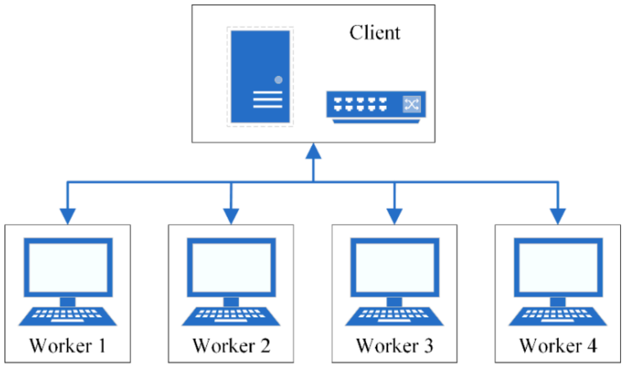

Figure 2. The PCT distributed simulation verification system architecture.

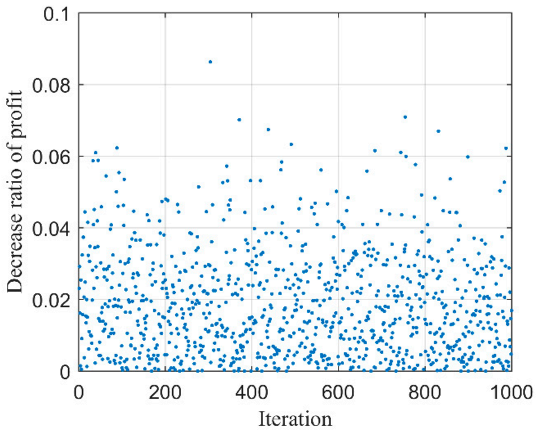

Figure 3. Profit comparison results of the MTCBAA and the artificial immune algorithm.

In Figure 3, the vertical coordinate represents the decrease ratio for the assignment profit obtained by the MTCBAA compared to the target assignment profit obtained by the centralized target assignment algorithm based on the artificial immune algorithm under the same simulation conditions. As shown in Figure 3, in each iteration of the comparison experiments, the decrease ratio for the target assignment profit obtained by the MTCBAA compared to that obtained by the centralized target assignment algorithm based on the artificial immune algorithm was within 10%, and in only 34 iterations the decrease ratio exceeded a value of 5%. In fact, the phenomenon of the decrease in the profit of the MTCBAA being basically controlled to 10% also held in two-on-two and three-on-three air combat scenarios. As the artificial immune algorithm is an optimization algorithm with excellent performance in finding a globally optimal solution, the target assignment profit of the centralized target assignment algorithm based on the artificial immune algorithm could be considered the global optimal target assignment profit. The results indicated that the decrease in the profit of the MTCBAA could be basically controlled to 10%, which represented a significant improvement compared to the 50% stated in Theorem 2. This was because it was basically impossible for any exchange of two aircraft-assigned targets, as is described in the proof of Theorem 2, to improve the target assignment profit.

The results also showed that the average running time taken by the MTCBAA for each iteration was 0.009 s, while that of the centralized target assignment algorithm based on the artificial immune algorithm was 2.45 s. Thus, the running time of the MTCBAA was basically negligible.

4.3. Deployment Experiment on Distributed Semi-Physical Simulation Platform

In this experiment, the MTCBAA was deployed to a distributed semi-physical simulation platform constructed in our lab to verify its practicality and real-time performance.

The architecture of the distributed semi-physical simulation platform is shown in Figure 4. The platform was designed based on the host–target mechanism. It included a host computer, which was used to generate code for each target computer based on the real-time workshop (RTW) and to control the operation of target computers, and four target computers based on an xPC target, which were used to provide a real-time verification environment for the algorithms. As there was no strict requirement for real-time communication between the host and target computers, their communication was performed via Ethernet. However, to ensure the real-time property of the simulation environment, the communication network between the target computers was required to have strict transmission determinacy and predictability, which was achieved using a reflective memory network. The reflective memory cards used in this experiment were PCI5565 cards. The Ethernet network had a star structure, while the reflective memory network had a ring structure. In addition, it was necessary to solve the problems relating to the start-up and time synchronizations for each target computer node, so the implementation method proposed in a previous study [26] was adopted. This method will not be described in this work. The method for the deployment of the MTCBAA to the distributed semi-physical simulation platform is described in the following.

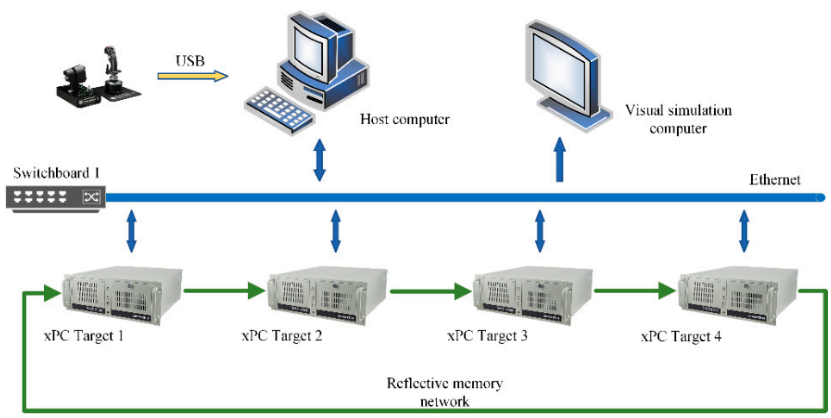

Figure 4. The architecture of the distributed simulation platform.

The operation process of the MTCBAA was described in detail in Section 3.2 and will not be repeated here. The focus of this section is on the realization of synchronization before each consensus phase between target computers for the MTCBAA.

The synchronization before each consensus phase between target computers was based on the double handshake mechanism, as shown in Figure 5. After the auction phase in each iteration, each target computer wrote "1" at the specified offset address in the reflected memory network and cyclically queried the other target computers for "1" at the corresponding offset address. If the desired result was reached within 2 ms, the first handshake terminated. Then, the second handshake was performed, where each target computer wrote "0" at the specified offset address in the reflected memory network and cyclically queried the other target computers for "0" at the corresponding offset address. If the desired result was reached within 500 μs, the second handshake terminated. Then, each target computer exchanged the data needed for the consensus phase with the other target computers and performed the consensus phase operations.

The MTCBAA was deployed to four target computers using the situation data in Section 4.1 as input, and the scheduling period of the MTCBAA was set to 20 ms. Then, the simulation process started. The xPC GUI of the target computer 1 is presented in Figure 6; the xPC GUIs of the other target computers were similar to that in Figure 6, so they are not presented in this section. In Figure 6, the vector in scope 1 corresponds to the assignment result vector a_1; the vector in scope 2 corresponds to the cooperative decision-making vector $\boldsymbol{f}_1$; the vector in scope 3 corresponds to the estimated missile assignment vector $\boldsymbol{m}_1$. As presented in Figure 6, the results were consistent with the results in Section 4.1. In addition, the TET values in Figure 6 show that the actual execution time of the MTCBAA in target computer 1 was 117 μs using the situation data in Section 4.1 as input, while the TET values of target computers 2–4 were 118, 118, and 116, respectively. The MTCBAA, as a synchronization algorithm, was synchronized by the double handshake mechanism, as described above. The total execution times of the four target computers were approximately equal.

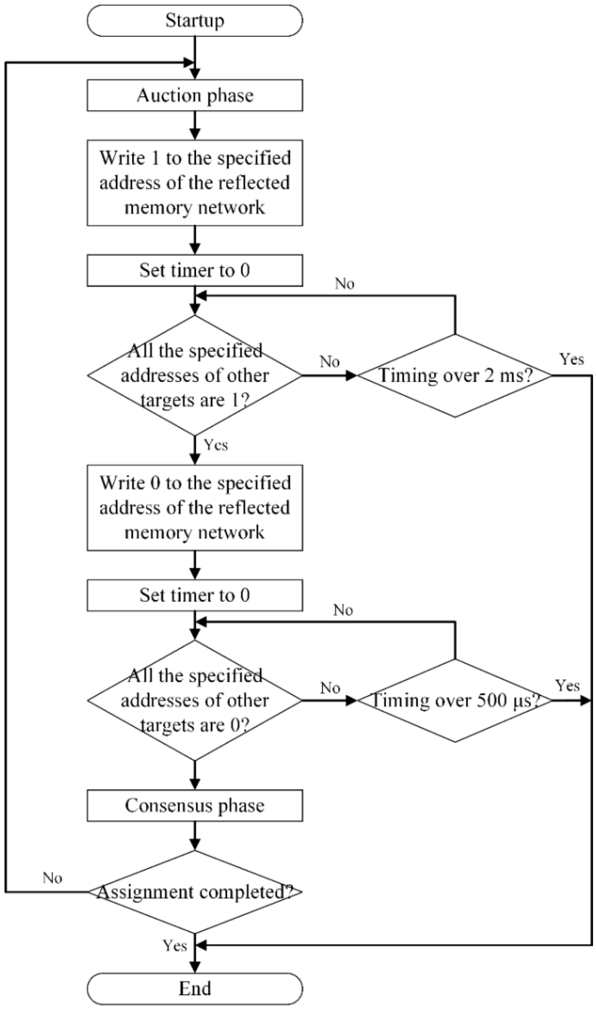

Figure 5. Flowchart of the double handshake mechanism for data communication.

Next, inputs were randomly generated using the conditions presented in Section 4.2, and 1000 iterations of distributed simulation verification experiments were conducted. The experiment results indicated that the MTCBAA had a maximum running time of 287 μs, a minimum running time of 38 μs, and an average running time of 172 μs when deployed to the target computers. Therefore, compared with the decision periods from previous studies on BVR air combat [27,28], the execution time of the MTCBAA was negligible.

Finally, two target computers were used as red aircraft, and the other two target computers were used as blue aircraft. Both sides were initially in a head-on situation. The red aircraft made maneuvering decisions using an expert system constructed in our lab and the proposed MTCBAA for target assignment, while the blue aircraft kept flying along straight lines. The scene of the engagement between the two sides is presented in Figure 7, where it can be seen that, when the blue aircraft entered the range of the missile attack zones of the red side, the two red aircraft assigned a missile to the corresponding target and fired. Then, each red aircraft switched to perform the single-side guide maneuver to reduce the approach rate with the enemy aircraft while providing medium guidance for its launched missiles.

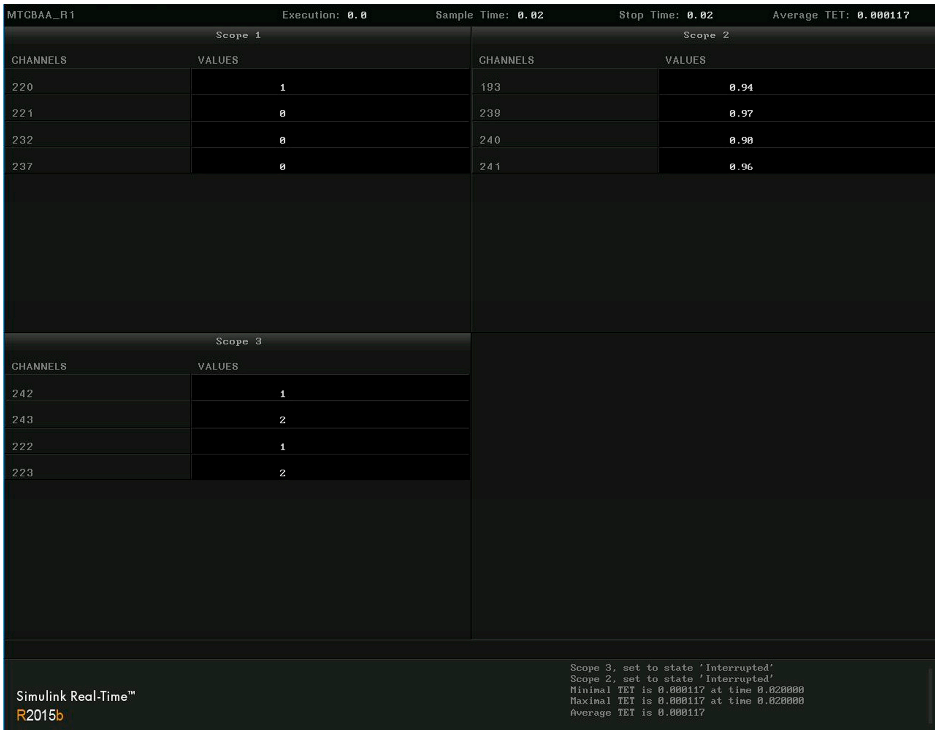

Figure 6. The xPC GUI of target computer 1.

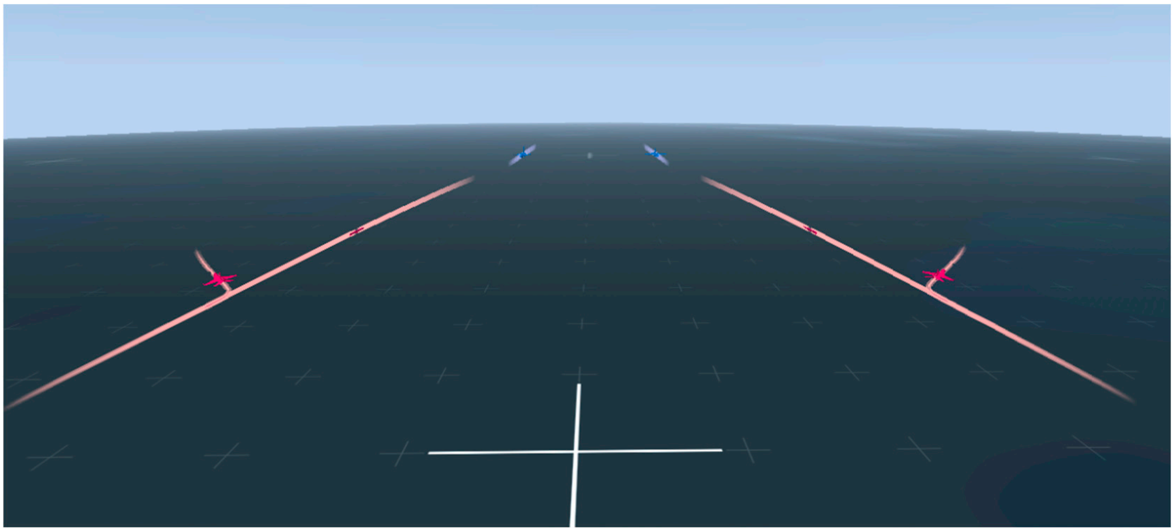

Figure 7. The two-on-two BVR air combat.

From the above experimental results, it can be seen that the MTCBAA proposed in this study shows obvious advantages in running time, and its distributed execution eliminates the dependence on the central node and improves the robustness of the system. Its practicability was demonstrated by deployment experiments on a distributed semi-physical simulation platform.

5. Conclusions

The existing centralized target assignment algorithms are inefficient and focus only on profit optimization, and the existence of the central node leads to low system robustness.

To address these problems, this paper proposes an efficient distributed target assignment algorithm for multi-aircraft cooperative BVR air combat.

As described in the methodology section, first, based on the non-parametric method, air combat situation modeling was performed on a target assignment basis for multi-aircraft cooperative BVR air combat. Next, based on the two-stage execution architecture of the CBAA and the introduction of the cooperative decision-making variable, a distributed target assignment algorithm for multi-aircraft cooperative BVR air combat, the MTCBAA, was designed. The MTCBAA can determine the assigned targets and missile numbers for all aircraft simultaneously. Finally, using the centralized target assignment algorithm—the MTSGA, which is equivalent to the MTCBAA—the fact that the MTCBAA can guarantee at least 50% optimization performance was proven.

In the experimental section, the operation process of the proposed MTCBAA was illustrated in an example. Using the Matlab parallel computing toolbox and the Monte Carlo method, the MTCBAA was compared with the immune algorithm-based centralized target assignment method in two-on-two, three-on-three, and four-on-four BVR air combat scenarios. The results showed that the proposed algorithm could keep profit reduction under 10% in the abovementioned scenarios, while reducing the average target assignment time from 2.45 s to 9 ms in the four-on-four scenarios. This proves the minimum optimization performance guarantee of the MTCBAA and verifies its efficiency advantage compared to the other algorithm. In addition, the results showed that. when the MTCBAA was deployed to a distributed semi-physical air combat simulation platform in four-on-four BVR air combat scenarios, it could achieve an average running time of 172 µs; thus, the efficiency and practicality of the proposed algorithm were further verified. Based on the experiment results, the proposed MTCBAA has excellent practicality and application prospects. In future work, other improvement mechanisms could be investigated to reduce the profit reduction further.

The work described here follows up on the work undertaken by our lab [1]. At the end of the cooperative occupation described in our previous work [1], the MTCBAA could be used for the target assignment of the first round of missile launches, thus forming an initial suppression posture against the enemy. When the first round of missile launches is over, the two combatants enter a more complex missile exchange phase, where target assignment is no longer applicable and more complex maneuvering and fire control decision policies are required. Therefore, we will investigate reinforcement learning-based multi-aircraft cooperative BVR air combat maneuvering and fire control decision-making in the future.

Author Contributions: Conceptualization, W.L. and Y.L. (Yongxi Lyu); methodology, W.L. and S.D.; software, S.D.; validation, S.D. and H.C.; formal analysis, W.L. and S.D.; investigation, Y.L. (Yongfeng Li); resources, J.S.; data curation, H.C.; writing—original draft preparation, W.L.; writing—review and editing, S.D.; visualization, Y.L. (Yongfeng Li); supervision, J.S.; project administration, J.S.; funding acquisition, Y.L. (Yongxi Lyu) and J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China, grant numbers 62173277 and 61573286; the Natural Science Foundation of Shaanxi Province, grant number 2022JM-011; the Aeronautical Science Foundation of China, grant number 201905053004; and the Shaanxi Province Key Laboratory of Flight Control and Simulation Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data generated or analyzed during this study are included in this article.

Acknowledgments: We thank LetPub (www.letpub.com) for their linguistic assistance during the preparation of this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

BVR	Beyond-visual-range
CBAA	Consensus-based auction algorithm
CBBA	Consensus-based bundle algorithm
SGA	Sequential greedy algorithm
MTCBAA	Multi-target consensus-based auction algorithm
MTSGA	Multi-target sequential greedy algorithm
PCT	Parallel computing toolbox

References

1. Li, W.-H.; Shi, J.-P.; Wu, Y.-Y.; Wang, Y.-P.; Lyu, Y.-X. A Multi-UCAV cooperative occupation method based on weapon engagement zones for beyond-visual-range air combat. *Def. Technol.* **2021**, *18*, 1006–1022. [CrossRef]
2. Li, S.; Chen, M.; Wang, Y.; Wu, Q. Air Combat Decision-Making of Multiple UCAVs Based on Constraint Strategy Games. *Def. Technol.* **2021**, *18*, 368–383. [CrossRef]
3. Zhang, J.; Xing, J. Cooperative task assignment of multi-UAV system. *Chin. J. Aeronaut.* **2020**, *33*, 2825–2827. [CrossRef]
4. Kline, A.; Ahner, D.; Hill, R. The Weapon-Target Assignment Problem. *Comput. Oper. Res.* **2019**, *105*, 226–236. [CrossRef]
5. Rasmussen, S.; Chandler, P.; Mitchell, J.; Schumacher, C.; Sparks, A. Optimal vs. Heuristic Assignment of Cooperative Autonomous Unmanned Air Vehicles. In Proceedings of the AIAA Guidance Navigation, and Control Conference and Exhibit, Austin, TX, USA, 11–14 August 2003. [CrossRef]
6. Wang, Z.; Liu, L.; Long, T.; Wen, Y. Multi-UAV reconnaissance task allocation for heterogeneous targets using an opposi-tion-based genetic algorithm with double-chromosome encoding. *Chin. J. Aeronaut.* **2018**, *31*, 339–350. [CrossRef]
7. Zhen, Z.; Zhu, P.; Xue, Y.; Ji, Y. Distributed intelligent self-organized mission planning of multi-UAV for dynamic targets cooperative search-attack. *Chin. J. Aeronaut.* **2019**, *32*, 2706–2716. [CrossRef]
8. Orhan, K.; Esra, K.; Ahmet, S. A multi-objective approach for dynamic missile allocation using artificial neural networks for time sensitive decisions. *Soft Comput.* **2021**, *25*, 10153–10166.
9. Zhu, J.; Zhao, C.; Li, X.; Bao, W. Multi-target Assignment and Intelligent Decision Based on Reinforcement Learning. *Acta Armamentarii* **2021**, *42*, 2040.
10. Zou, Z.; Chen, Q. Decision tree-based target assignment for the confrontation of multiple space vehicles. *Acta Aer-Onautica Astronaut. Sin.* **2022**. [CrossRef]
11. Mclain, T.W.; Beard, R.W. Coordination Variables, Coordination Functions, and Cooperative Timing Missions. *J. Guid. Control Dynam* **2005**, *28*, 150–161. [CrossRef]
12. Castanon, D.A.; Wu, C. Distributed algorithms for dynamic reassignment. In Proceedings of the 42nd IEEE International Conference on Decision and Control, Maui, HI, USA, 9–12 December 2003; Volume 1, pp. 13–18.
13. Fax, J.; Murray, R. Information Flow and Cooperative Control of Vehicle Formations. *IEEE Trans. Autom. Control* **2004**, *49*, 1465–1476. [CrossRef]
14. Ren, W.; Beard, R.W.; Kingston, D.B. Multi-agent Kalman consensus with relative uncertainty. In Proceedings of the American Control Conference, Portland, OR, USA, 8–10 June 2005; pp. 1865–1870.
15. Olfati-Saber, R.; Murray, R.M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **2004**, *49*, 1520–1533. [CrossRef]
16. Alighanbari, M.; How, J.P. Decentralized Task Assignment for Unmanned Aerial Vehicles. In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 15 December 2005; pp. 5668–5673.
17. Atkinson, M. Contract Nets for Control of Distributed Agents in Unmanned Air Vehicles. In Proceedings of the 2nd AIAA "Unmanned Unlimited" Conf. and Workshop & Exhibit, San Diego, CA, USA, 15–18 September 2003. [CrossRef]
18. Choi, H.L.; Brunet, L.; How, J.P. Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–926. [CrossRef]
19. Choi, H.L.; Whitten, A.K.; How, J.P. Decentralized task allocation for heterogeneous teams with cooperation constraints. In Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010; pp. 3057–3062.
20. Odonkor, P.; Ball, Z.; Chowdhury, S. Distributed operation of collaborating unmanned aerial vehicles for time-sensitive oil spill mapping. *Swarm Evol. Comput.* **2019**, *46*, 52–68. [CrossRef]
21. Fu, X.; Feng, P.; Gao, X. Swarm UAVs Task and Resource Dynamic Assignment Algorithm Based on Task Sequence Mechanism. *IEEE Access* **2019**, *7*, 41090–41100. [CrossRef]
22. Ye, F.; Chen, J.; Sun, Q.; Tian, Y.; Jiang, T. Decentralized task allocation for heterogeneous multi-UAV system with task coupling constraints. *J. Supercomput.* **2021**, *77*, 111–132. [CrossRef]
23. Chen, X.; Wei, X.; Xu, G. Multiple Unmanned Aerial Vehicle Decentralized Cooperative Air Combat Decision Making with Fuzzy Situation. *J. Shanghai Jiaotong Univ.* **2014**, *48*, 907–913.
24. Li, W.; Guo, L.; Shi, J. Research on Distributed Target Allocation in Multi-UAV Beyond Visual Range Cooperative Air Combat. In *Advances in Guidance, Navigation and Control*; Yan, L., Duan, H., Yu, X., Eds.; Springer: Singapore, 2021; pp. 1845–1855. [CrossRef]

25. Gao, Y.; Chen, S.; Yu, M.; Hai, J.; Fang, R. Target Allocation Method of Multi-Aircraft Cooperative Air Combat Based on Improved Artificial Immune Algorithm. *J. Northwestern Polytech. Univ.* **2019**, *37*, 354–360. [CrossRef]
26. Chen, H.; Liu, Y.; Sang, X. Study on Synchronization Design of Distributed Hardware-in-the-loop Simulation System for Spacecraft. *Bull. Sci. Technol.* **2017**, *33*, 61–66.
27. Piao, H.; Sun, Z.; Meng, G.; Chen, H.; Qu, B.; Lang, K.; Sun, Y.; Yang, S.; Peng, X. Beyond-Visual-Range Air Combat Tactics Auto-Generation by Reinforcement Learning. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
28. Sun, Z.; Piao, H.; Yang, Z.; Zhao, Y.; Zhan, G.; Zhou, D.; Meng, G.; Chen, H.; Chen, X.; Qu, B.; et al. Multi-agent hierarchical policy gradient for Air Combat Tactics emergence via self-play. *Eng. Appl. Artif. Intell.* **2021**, *98*, 104112. [CrossRef]