Toward Rapid and Optimal Strategy for Swarm Conflict: A Computational Game Approach

Tao Zhang, Dongying Ma, Chaoyong Li, Xiaodong Wang

Abstract—**The decision and control problem for swarm operations is crucial for autonomous management of military conflict. In this paper, we prove that the underlying decision and control problem can be treated as a noncooperative game problem, and then proceed to introduce a parallelized algorithm to seek the desired Nash equilibrium with the help of the maximum weight matching algorithm. The proposed scheme is proved to be optimal with a guaranteed ϵ-Nash solution and rapid computational speed. Simulation results verified the effectiveness of the proposed solutions.**

Index Terms—**Game theory, Nash equilibrium, swarm operation, search algorithm**

I. Introduction

In a military scenario, a large-scale, high-performance and multi-task decision making system has always been the central of swarm combat operations, wherein the maneuverability of combat units, the threat of different adversary and many other complex conditions must be taken into account. In general, the commander center of each side manages to seek feasible decisions to plan routes for combat units and allocate the limited resources to obtain a high performance solution. However, due to more complex dynamics and constraints, and the uncertainty of real conflict scenario, a desired optimal control solution is not always possible [1]. To this end, some novel algorithms, such as reinforcement learning [2], distributed learning [3] and convex optimization [4], have been studied.

Among many research methods, game theory is becoming a promising method in many fields for its effectiveness and superiority. [5] designed a guidance algorithm by seeking a Nash equilibrium and introduced a local observer to facilitate the local implementation of the proposed guidance protocol. In [6], the rendezvous problem between two spacecraft operating is investigated, and a sample data game strategy is introduced to alleviate the seeking process of the Nash equilibrium. Dong et al. [7] investigated the Nash equilibrium seeking problem among networked multi-player game over directed and switching topologies. One should note that the decision-making problem modeled by game theory in a military operation was first introduced by Cruz and Simaan [8–10], and based on that, some researchers have extended this method to more complex situations. [11] described a multistage influence diagram game for modeling the maneuvering decisions of pilots in one-on-one air combat. Zhang et al. [12] proposed an optimal task control scheme for military operations with dynamic game. In addition, multiple unmanned combat air vehicles attack-defense decision-making problem was studied in [13], which applied the dimensionality reduction based matrix game solving algorithm to solve large-scale matrix games.

On the other hand, a proper assignment of weapons to hostile targets, such as incoming missiles, with the objective of maximizing the destruction of the adversary or minimizing the loss of protected assets, has been widely studied. On the condition that all weapons engage a finite number of targets with known probabilities of kill, and no subsequent engagements are considered, some novel algorithms emerge, such as genetic algorithm [14] and particle swarm optimization [15]. In [16], an improved Hungarian method for linear weapon target assignment problems is proposed, but it requires more time in handling the many uncovered elements. But due to the flexibility and unpredictable actions of adversary and long time span in conflict operations, the above static target allocating scheme in the whole fast-changing battle procedure is unlikely to hold, which leads to the consideration that each engagement stage is assessed for the subsequent decisions. Therefore, it can better reflect responses to the dynamic combat environment and the unpredictable maneuvers of the adversary. Kurdi et al. [17] investigated bacteria forging behavior and proposed an adaptive task allocation for multi-UAV systems, which can maintain a steady runtime performance under different scales of swarm sizes. Leboucher et al. [18] developed an efficient method using a Hungarian method and a GA-PSO hybrid algorithm. Chen et al. [19] implemented three evolutionary decision-making algorithms, including a genetic algorithm and two memetic algorithms. Karasakal [20] addressed the issue of allocating the air defense missiles to incoming air targets in order to maximize the air defense effectiveness of a naval task group.

It should be pointed out that, albeit their effectiveness, the majority of the aforementioned results struggle to tackle either the task assignment problem, or the path planning problem on a relatively small scale, and may not take into account both of the two problems in a complex swarm combat scenario. To this end, some researchers seek to characterize more precisely the underlying decision control problem. In [21], a solution to the cooperative control problem represented by UAVs is presented, where the target assignment and coordinated intercept problem are addressed through a combination of the satisfying and social welfare paradigms, the Voronoi diagram and non-polynomial optimization method. Babel [22] studied

Tao and Chaoyong are with College of Electrical Engineering, Zhejiang University, Hangzhou, China, 310027. Corresponding to: Chaoyong@zju.edu.cn

Dongying and Xiaodong are with Beijing Institute of Electronic System Engineering, Beijing, 100854, China.

the motion planning and target assignment problem against static targets for a group of autonomous air vehicles in scenario where obstacle and threat avoidance is required. A model of a multi-UAV combat scenario, which uses noncooperative game to simulate the combat scenario and a variant of the Kuhn-Munkres algorithm to solve the target assignment in decision-making process is proposed in [12].

However, it should be pointed out that aforementioned schemes may struggle in large scale swarm operations, and their implementations dictate a central unit deploying algorithms and transmitting solutions to all units. In practice, distributed or parallel algorithm is preferred with full intention to harvest the computational power of all combat units. Notable contributions in this venue include, for instance, Hoepman [23] proposed a simple distributed greedy algorithm with a guaranteed $\frac{1}{2}$-approximation, that is, the estimated result can reach $\frac{1}{2}$ of the optimal ones; Thorsen [24] proposed a two-phased algorithm using atomic operations, that is *initial cardinality matching and short augmentations*. Note that, albeit its effectiveness, atomic operation could be lukewarm toward determining what should be done when the processor is unable to apply a short augmentation; Moreover, Azad [25] combined the maximum cardinality matching algorithm with the weight increasing cycles algorithm, such that the result becomes closer to the optimum, but the algorithm demands a fully connected graph, which could be troublesome to maintain in practical scenarios. In this paper, we propose a real-time decision and control scheme for the large scale swarm operations. In particular, we demonstrate that the underlying problem can be treated as a noncooperative matrix game problem. As a result, its optimal strategy can be captured by the Nash equilibrium (i.e., NE). Then, we propose a parallel computational framework to seek the NE with significantly reduced algorithmic complexity and $\epsilon-$Nash accuracy, leading to guaranteed performance in terms of computational speed and optimality.

II. Preliminaries and Problem Formulation

A. Preliminaries results on weighted bipartite graph

Before proceeding further, we introduce some preliminary results on graph theory. An undirected weighted bipartite graph $\mathcal{G} = (\mathcal{V}_L, \mathcal{V}_R, \mathcal{E}, \omega)$ can be described by the disjoint vertex set $\mathcal{V}_L = \{1, 2, ..., n_1\}$ and $\mathcal{V}_R = \{1, 2, ..., n_2\}, n_1 \leqslant n_2$, by the edge set $\mathcal{E} \subseteq \mathcal{V}_L \times \mathcal{V}_R$, and by the weight function $\omega : \mathcal{E} \to \Re^+$ which assigns a positive weight to each edge of $\mathcal{G}$. The neighbor set $\mathcal{E}_i \subset \mathcal{E}$ of a left vertex $i \in \mathcal{V}_L$ is the set of all edges associated with i. The weight ω_{ij} is the weight of edge between vertex $i \in \mathcal{V}_L$ and vertex $j \in \mathcal{V}_R$.

The set of all right vertices connected to the left vertex i is defined as $\mathcal{T}_i$, that is, $\mathcal{T}_i = \{j \in \mathcal{V}_R | (i, j) \in \mathcal{E}\}$. A matching $\mathcal{M}$ is a set of the weighted edges chosen in such a way that no two edges share an endpoint, and in what follows, if $\mathcal{M}$ contains all vertices of $\mathcal{V}_L$, we call it a *perfect matching* [26]. A *maximum weight matching* (MWM) is a matching $\mathcal{M}^*$ such that the sum of weights of all edges in $\mathcal{M}^*$ is the largest among all matchings, that is [27]

$$\mathcal{M}^* = \arg \max_{\mathcal{M} \subset \mathcal{E}} \sum_{e \in \mathcal{M}} \omega(e) \tag{1}$$

As is well known, albeit its effectiveness in finding optimal solutions, the MWM algorithm doesn't always come with computational efficiency described by running time. In fact, MWM algorithm struggles in practical implementations with large scale graph. Therefore, a trade-off between time efficiency and optimality of solution is desired in real world applications, which is commonly defined as the *approximation maximum weight matching* problem. In order to quantify the quality of the solution, we assume the quality of the solution of an approximation algorithm can be measured by a factor α, and hence the matching $w(\mathcal{M})$ is called a α-approximate matching if the weight is at least $w(\mathcal{M}) \geq \alpha \cdot w(\mathcal{M}^*)$.

Given a perfect matching $\mathcal{M}$, a k-Path $P_k = \{(i_1, m_{i_1}), (i_2, m_{i_2}), ..., (i_k, m_{i_k})\}$ is a path that the edges are alternatively matched and unmatched. The matched edge (i_1, m_{i_1}) is called the root. The merging process of P_k and edge $(i_{k+1}, m_{i_{k+1}})$ is called the growth of path P_k, that is

$$\begin{aligned} P_{k+1} &= P_k \cup (i_{k+1}, m_{i_{k+1}}) \\ &= \{(i_1, m_{i_1}), (i_2, m_{i_2}), ..., (i_k, m_{i_k}), (i_{k+1}, m_{i_{k+1}})\} \end{aligned} \tag{2}$$

A k-Path P is called a k-Circle $\mathcal{C}$ if $(i_k, m_{i_1}) \in \mathcal{E}$. The k-Path P and the k-Circle $\mathcal{C}$ are shown in Figure 1(a) and Figure 1(b). The symmetric difference between $\mathcal{C}$ and $\mathcal{M}$ is defined as

$$\mathcal{M} \oplus \mathcal{C} \triangleq (\mathcal{M} \backslash \mathcal{C}) \cup (\mathcal{C} \backslash \mathcal{M}) \tag{3}$$

Obviously, $\mathcal{M} \oplus \mathcal{C}$ is also a matching that has the same cardinality as the original matching $\mathcal{M}$. Thus, the gain $g(\mathcal{C})$ of a k-Circle $\mathcal{C}$ is defined as

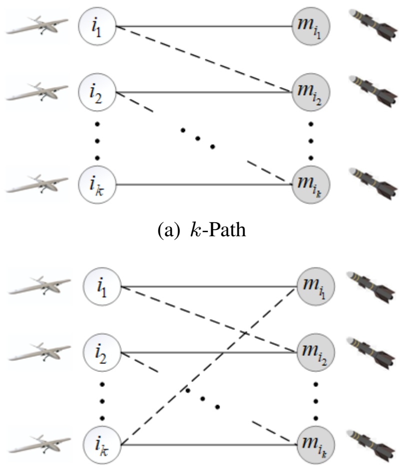
$$g(\mathcal{C}) \triangleq w(\mathcal{M} \oplus \mathcal{C}) - w(\mathcal{M}) \tag{4}$$

If $g(\mathcal{C}) > 0$, then $\mathcal{C}$ is called an augmenting circle, and without loss of any generality, we only consider augmenting circles of the short length, e.g., 2-circle.

In this paper, we propose to study a real-time task management problem among swarming combat units. Most specifically, we attempt to solve weapon target assignment problem with a approximation maximum weight matching algorithm, where the left vertices of the graph are assumed to be weapons and the right vertices are targets. In particular, special attention will be paid on how to convert the underlying problem into a game strategy, and how to establish a rapid computational scheme that allows game theory to scale up to complex swarming scenarios.

B. Swarm conflict and problem formulation

In a military conflict with swarming units, it is imperative to maintain the order of each group in order to ensure a maximal impact on the adversary. In other words, optimal task management among combat units is instrumental for the underlying problem, namely, how to assign each combat unit with specific adversarial target to yield a collectively optimal outcome, given the overall battlefield situation and self



(a) k-Path

(b) k-Circle

Fig. 1: k-Path and k-Circle. Matched edges are shown in solid lines, and non-$\mathcal{M}$ edges are shown in dashed lines.

awareness, and how to accomplish this in a rapid and optimal manner is desired for swarming operations. As previously established, the swarm conflict among networked combat units can be treated as a two player/group noncooperative game problem, with two player/group termed as "Blue" and "Red". Without loss of any generality, we assume that Blue is the offensive side and Red is the defensive side. Hence, the underlying swarm conflict management problem is effectively transformed to a Nash equilibrium (i.e., NE) seeking problem. In this paper, we propose a novel computational game strategy to tackle this problem, escalating game theory to large scale swarming operations and mitigating the otherwise exhaustive computation with a parallel structure, thus potentially offer real-time and optimal weapon target assignment in adversarial scenarios.

Suppose that Blue consists of N^B units and Red consists of N^R units, and the goal for each group is to collectively maximize the kill probability against its adversary, which, by definition, constitutes a noncooperative game problem for Blue and Red players/groups, and, as is well established [12], the Nash equilibrium of the underlying game captures the optimal solution for both players. Without loss of any generality, the motion of the combat unit i is depicted as in Figure 2, and its dynamical equations are formulated as follows

$$\left\{\begin{array}{l} \dot{x}_i = V_i \cos\theta_i \cos\varphi_i \\ \dot{y}_i = V_i \cos\theta_i \sin\varphi_i \\ \dot{z}_i = V_i \sin\theta_i \\ \dot{V}_i = -\frac{D_i}{m_i} - g\sin\theta_i \\ \dot{\theta}_i = -\left(a_{yi} + g\cos\theta_i\right)/V_i \\ \dot{\varphi}_i = \frac{a_{zi}}{V_i\cos\theta_i} \end{array}\right. \tag{5}$$

where $[x_i, y_i, z_i]$ is the position and V_i, θ_i, φ_i are velocity, pitch and azimuth angles, respectively, m_i is the mass, a_{yi} and a_{zi} are normal accelerations to be determined and we let $n_i = \sqrt{a_{yi}^2 + a_{zi}^2}$. D_i represents the drag force, and

$$D_i = \frac{1}{2} C_d \rho(z_i) S_i V_i^2 \tag{6}$$

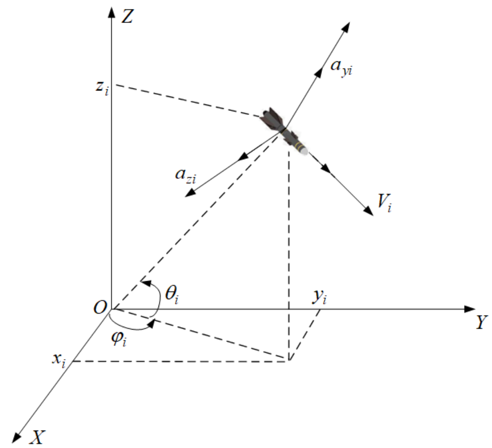


Fig. 2: Dynamics model of combat units

where C_d is atmospheric drag coefficient, $\rho(z_i)$ is the atmospheric density at the current altitude, and S_i is the reference area.

Let u_i be the input to the ith combat unit, and according to (5), u_i should consist of control and decision outcome, namely, motion control input (i.e., a_{yi} and a_{zi}) and target assignment outcome, that is,

$$\begin{array}{l} u_i = [a_{yi}, a_{zi}, c_i] \\ U^B = \left\{u_1^B, u_2^B, ..., u_{N^B}^B\right\} \\ U^R = \left\{u_1^R, u_2^R, ..., u_{N^R}^R\right\} \end{array} \tag{7}$$

where c_i is the attack vector of unit i, that is, $c_i = \left[c_{i1},\ ...,\ c_{i\hat{N}}\right]$ with c_{ij} be a Boolean value. $c_{ij} = 1$ indicates that weapon i views j as its target, $c_{ij} = 0$ denotes the opposite.

In this paper, we attempt to solve optimal control and decision problem for swarming operations, aiming to develop a rapid strategy to extract the desired U^B and U^R such that the overall implications for each combat group are unilaterally maximized. In other words, the desired U^B and U^R constitute a NE for the underlying noncooperative game problem, in this regard, the problem to be solved in this paper can be formulated as follows [28]

$$\begin{array}{rl} \textbf{P1:} & \textbf{find}\ \ U^{B*}, U^{R*} \\ \textbf{s.t.} & \left\{\begin{array}{l} J^B\left(U^{B*}, U^{R*}\right) \geq J^B\left(U^B, U^{R*}\right), \forall U^B \in \Delta^B \\ J^R\left(U^{B*}, U^{R*}\right) \geq J^R\left(U^{B*}, U^R\right), \forall U^R \in \Delta^R \end{array}\right. \end{array} \tag{8}$$

In problem **P1**, J^B and J^R are payoff function for the Blue and Red, respectively. Then, the desired optimal inputs U^{*B} and U^{*R} should satisfy the following conditions

$$\begin{array}{l} U^{*B} = \underset{U^B\in\Delta^B}{\arg\max}\, J^B(U^B, U^R) \\ U^{*R} = \underset{U^R\in\Delta^R}{\arg\max}\, J^R(U^B, U^R) \end{array} \tag{9}$$

where Δ_B and Δ_R are the sets of all admissible decisions, and $U = U^{mo} \times \mathcal{M}$ is the feasible decision set with U^{mo} defined as the motion control input and $\mathcal{M}$ the WTA outcome.

Under perfect condition, we define PK_{ij} as the perfect probability of kill should weapon i attack target j. However, effectiveness of kill in real combat scenario varies from weather condition to weapon deficiency, hence PK_{ij} should be recalibrated to reflect the instantaneous situations. In what follows, we consider weather conditions as an external influence while the maneuverability differences as an internal factors, and let K_{ij} be the actual probability of kill, and

$$K_{ij}\left(u_i,\hat{U}\right)=PK_{ij}\beta^w\beta_{ij}^c\left(n_j/n_i,\delta\right),j\in\mathcal{T}_i \tag{10}$$

where $\hat{U}$ is the adversary's decision set, and $\mathcal{T}_i$ represents the set of targets that can be observed by the sensors, while $0\le\beta^w\le 1$ is defined as the weather factor and $0\le\beta_{ij}^c\le 1$ the confidence factor that weapon i successfully attacks target j, $\beta_{ij}^c\left(n_j/n_i,\delta\right)$ is a monotonically decreasing function of the relative orientation and maneuverability differences between i and j, and δ is a critical factor depending on the motion ability. Note that $\beta_{ij}^c\to 1$ if $n_j/n_i\le\delta$, $\beta_{ij}^c\to 0$ if otherwise. That is, the effectiveness of a successful destruction will decrease rapidly when the adversary's maneuverability exceeds its expectation. Specifically, δ can be described by $n_j^{max}/n_i^{max}-\xi$, where ξ is a sufficiently small number, that is, β_{ij}^c decreases significantly as a_j/a_i approaches its maximum.

As previously established, a complete decision should consist of both motion control input and target assignment outcome. In this regard, we introduce $\gamma_{ij}\in[0,1]$ as a decision factor for motion control, which is a monotonically decreasing function, and

$$\begin{array}{l}\gamma_{ij}\left(u_i,\hat{U}\right)=\gamma_{ij}\left(\left|R_{ij}^X\right|,\dot{q}_{ij}^X\right),j\in T_i\\ X=\left\{\begin{array}{l}1,\text{if } i \text{ belongs to Blue}\\ -1,\text{if } i \text{ belongs to Red}\end{array}\right.\end{array} \tag{11}$$

where $\dot{q}_{ij}\left(u_i,\hat{U}\right)$ is line-of-sight (i.e., LOS) rate and the relative distance $R_{ij}\left(u_i,\hat{U}\right)$ between units i and its intent target j are

$$R_{ij}\left(u_i,\hat{U}\right)=\left[x_i-x_j,y_i-y_j,z_i-z_j\right] \tag{12}$$

Before proceeding further, the following assumptions are made for the underlying problem:

Assumption 1. *At most one target can be assigned to any specific weapon (i.e., combat unit), that is,*

$$\left\{\begin{array}{l}{c_i}^Tc_j=0,\ for\ each\ unit\ i\neq j\ of\ the\ same\ side\\ \sum\limits_{i=1}^{N}c_{ij}=1,j=1,2,...,\hat{N}\end{array}\right. \tag{13}$$

Assumption 2. *The individual probability of perfect kill PK_{ij} and the individual destructive value W_j for target j is known to all combat units of each group, and $\sum\limits_{j=1}^{\hat{N}}W_j=1$.*

By *Assumption* 1, the overall payoff functions for the Blue and Red player become

$$\begin{array}{l}J^B(U^B,U^R)=\sum\limits_{i=1}^{N^R}J_{ij}^B=\\ \quad\sum\limits_{i=1}^{N^B}W_j^RK_{ij}^R\left(u_i^B,u_j^R\right)\gamma_{ij}^R\left(u_i^B,u_j^R\right),j\in T_i\\ J^R(U^B,U^R)=\sum\limits_{i=1}^{N^R}J_{ij}^R=\\ \quad\sum\limits_{i=1}^{N^R}W_j^BK_{ij}^B\left(u_j^B,u_i^R\right)\gamma_{ij}^B\left(u_j^B,u_i^R\right),j\in T_i\end{array} \tag{14}$$

However, it should be pointed out that solving (14) for NE (i.e., u_i^{*B} and u_i^{*R}) is essentially a matrix game problem, and its computation, if possible, is proved to be time consuming and exhaustive [12]. Hence, a tradeoff between solution accuracy and computational solvency must be sought in order to ensure a timely decision for large scale swarming operations, and such solution, albeit sub-optimal, is deemed practical and salient. In this paper, we propose a parallelized computation scheme to seek a ϵ-Nash solution for the underlying problem and prove that the solution accuracy, captured by ϵ, is uniformly bounded. In this regard, the problem **P1** becomes

$$\begin{array}{l}\textbf{P2: find}\ \ \hat{U}^B,\hat{U}^R\\ \quad\textbf{s.t.}\left\{\begin{array}{l}J^B\left(\hat{U}^B,\hat{U}^R\right)\ge J^B\left(U^B,\hat{U}^R\right)-\epsilon,\forall U^B\in\Delta^B\\ J^R\left(\hat{U}^B,\hat{U}^R\right)\ge J^R\left(\hat{U}^B,U^R\right)-\epsilon,\forall U^R\in\Delta^R\end{array}\right.\end{array} \tag{15}$$

Hence, with problem **P2**, the underlying decision and control problem is successfully converted to a noncooperative game problem in general, and a computational NE seeking problem in particular given the fact that all combat units exhibit nonlinear behaviors governed by (5). In addition, as revealed in (14), the combat unit will first decide which target yield the best collective kill probability, and then move accordingly to maximize J^B or J^R. The decision are expected in real time with updated situational information (i.e. distance, LOS rate). Hence, how to construct a rapid computation scheme to find the desired NE with acceptable accuracy (i.e., ϵ) is instrumental in swarming conflict management, and will be addressed in the next section.

III. Main results

In this section, we proceed to solve problem **P2** using the parallel heavy weight matching (i.e., PHWM) algorithm taken from the graph matching theory. As previously established, analytical solution for a pure NE is impossible in this case due to the intricacy of nonlinear dynamics and the number of combat units, and a computational strategy is thus in order and a rapid search algorithm becomes instrumental for the underlying problem. To be more exact, the difficulty of solving the matrix game problem grows exponentially with the escalation of operation scale, as specified in [8, 9, 12]. In what follows, we will attempt to integrate PHWM algorithm to the Action-Reaction Search (i.e., ARS) scheme, whose effectiveness has already been verified in the same venue [12, 29], and actively seeking the NE through *optimal response strategy* that alternates between two players, as depicted in

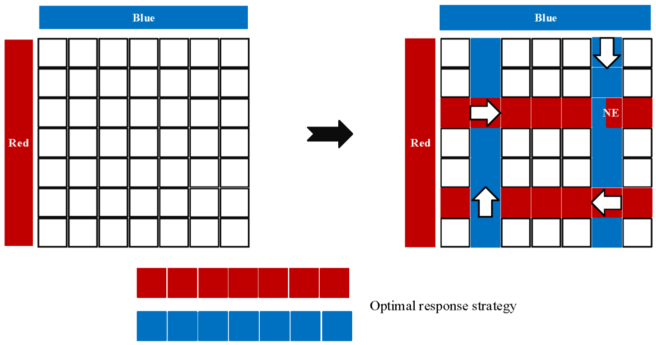


Fig. 3: Action-reaction search

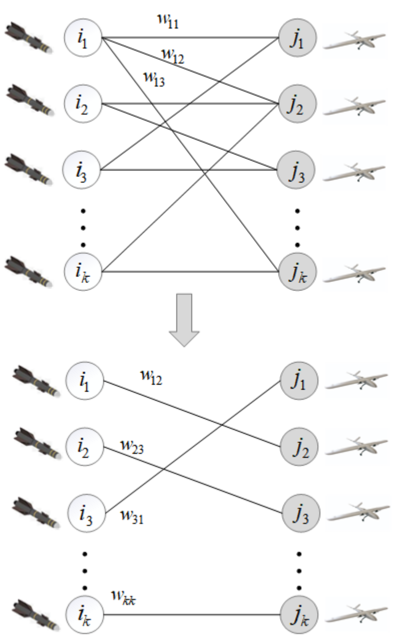


Fig. 4: Decision making procedure with bipartite graph

Figure 3. The search process starts with an empty matrix, in other words, a matrix where each element J_{ij} is not calculated yet. Then, beginning with an arbitrary decision of Red or Blue, its opponent makes an optimal response to its decision, and the procedure repeats until a NE (if any) is found.

In particular, two-player noncooperative game problem, or swarming conflict management problem to an extent, can be conducted over a bipartite graph. In this case, the combat units can be treated as two sets of respective and exclusive vertices, and if target $j \in \mathcal{T}_i$, we say that unit i has an *edge* e_{ij} connected with target j, and its weight w_{ij} is

$$w_{ij} \in \{J_{ij}\left(u_i^{mo}, u_j\right) | u_i^{mo} \in \Delta_i^{mo}\}, j \in \mathcal{T}_i \tag{16}$$

Moreover, as indicated in (16), w_{ij} could be chosen of any value of J_{ij} that is attributed by the admissible set of the ith unit, and its cardinality depends on the size of the admissible set Δ_i^{mo} of i, as implied in Figure 4. However, the combination of the edge set will explode during each procedure in ARS simple because of the ubiquitous choices of w_{ij}, leading to excessive computational burden. For instance, suppose that $N^B = N^R = 10$ and the size of motion control input of each unit is $card\left(\Delta_i^{mo}\right) = 100$, then we have 100^{10} $\mathcal{G}$ combinations and solution to target assignment will be extremely difficult, if even possible, in a single iteration. In this regard, we select w_{ij} to be the maximal J_{ij}, that is,

$$w_{ij} = w_{ij}^* = \max_{\substack{a_{yi} \in \Delta_i^{ay} \\ a_{zi} \in \Delta_i^{az}}} J_{ij}(a_{yi}, a_{zi}, u_j), j \in T_i \tag{17}$$

where $\Delta_i^{ay} \times \Delta_i^{az} = \Delta_i^{mo}$. Thus, $\mathcal{G}$ will be reduced to $\mathcal{G}^*$ with all other options pruned off.

Let $\bar{w}_{ij}$ and J_{es} be the weight and payoff associated with the optimal search scheme, respectively, commonly conducted with a comprehensive and exhaustive computation scheme, and let w_{ij}^* and J_{pr} be the weight and payoff associated with the proposed pruning strategy, respectively, that is

$$\begin{aligned} J_{es}(U_{es}, \hat{U}) &= \sum_{i=1}^{N} \bar{w}_{ij} \\ J_{pr}(U_{pr}, \hat{U}) &= \sum_{i=1}^{N} w_{ij}^* \end{aligned} \tag{18}$$

where $\hat{U}$ is the decision set of the adversary. The following lemma demonstrates that the pruning principle does not evolve at the expense of the optimality of the target assignment decisions.

Lemma 1. *For problem **P2**, suppose graph weight specified in* (16) *with payoffs J_{es} and J_{pr} defined in* (18)*. Then, if $\hat{U}$ is chosen to be the same for J_{es} and J_{pr}, we have $J_{pr} = J_{es}$.*

Proof. Before moving further, we introduce the weight $\bar{w}_{ij}$ as defined in (16) but not necessarily the maximum. Then, we have $\bar{w}_{ij} \le w_{ij}^*$. Let $\omega^*\left(\mathcal{M}\right)$ be the sum of all w_{ij}^*, and $\bar{\omega}\left(\mathcal{M}\right)$ be the sum of all $\bar{w}_{ij}$, we obtain

$$\begin{aligned} \omega^*\left(\mathcal{M}\right) &= J_{pr}(U_{pr}, \hat{U}) = \sum_{e_{ij} \in \mathcal{M}} w_{ij}^* \\ \bar{\omega}\left(\mathcal{M}\right) &= J_{es}(U_{es}, \hat{U}) = \sum_{e_{ij} \in \mathcal{M}} \bar{w}_{ij} \end{aligned} \tag{19}$$

It is apparent that, for any matching $\mathcal{M}$, we have

$$\bar{\omega}\left(\mathcal{M}\right) \le \omega^*\left(\mathcal{M}\right) \tag{20}$$

Let $\mathcal{M}_{pr}^*$ and $\mathcal{M}_{es}^*$ be the assignment outcomes associated with the pruning scheme and the exhaustive search scheme, respectively. Without loss of any generality, we assume both schemes are conducted with the PHWM algorithm, and due to the comprehensive nature of the exhaustive search strategy, its matching set contains that of the pruning scheme, that is

$$\omega^*\left(\mathcal{M}_{pr}^*\right) \le \bar{\omega}\left(\mathcal{M}_{es}^*\right) \tag{21}$$

In case of $\mathcal{M}_{pr}^* = \mathcal{M}_{es}^*$, it follows from (20) that

$$\bar{\omega}\left(\mathcal{M}_{es}^*\right) \le \omega^*\left(\mathcal{M}_{pr}^*\right) \tag{22}$$

In case of $\mathcal{M}_{pr}^* \neq \mathcal{M}_{es}^*$, it is obvious that $\mathcal{M}_{es}^*$ is an alternative solution in terms of $\mathcal{G}^*$, that is

$$\omega^*\left(\mathcal{M}_{es}^*\right) \le \omega^*\left(\mathcal{M}_{pr}^*\right) \tag{23}$$

Note that $\bar{\omega}\left(\mathcal{M}_{es}^{*}\right) \leq \omega^{*}\left(\mathcal{M}_{es}^{*}\right)$, then we have

$$\bar{\omega}\left(\mathcal{M}_{es}^{*}\right) \leq \omega^{*}\left(\mathcal{M}_{pr}^{*}\right) \tag{24}$$

Then, it follows from (21), (22) and (24) that $\bar{\omega}\left(\mathcal{M}_{es}^{*}\right) = \omega^{*}\left(\mathcal{M}_{pr}^{*}\right)$. On the other hand, (16) shows that the motion control is attached to an assignment $\mathcal{M}$, then the payoff function J in (14) can be written as the weight of $\mathcal{M}$, which is

$$\begin{aligned} J_{pr} &= \omega^{*}\left(\mathcal{M}_{pr}^{*}\right) \\ J_{es} &= \bar{\omega}\left(\mathcal{M}_{es}^{*}\right) \end{aligned} \tag{25}$$

Thus, we have

$$J_{pr} = J_{es} \tag{26}$$

and that completes the proof. □

By invoking *Lemma 1*, it is clear that the pruning scheme does not mitigate the NE of the underlying game. Instead, it will significantly speed up the seeking process with the help of the Kuhn-Munkres (i.e., KM) algorithm, as proved in [12]. However, it should be noted that, albeit its effectiveness, execution of the KM algorithm dictates a centralized agent/node and, hence, makes the computation resources of the overall combat group redundant. In light of the discovery and in order for a real time decision, we propose a parallelized scheme to solve **P2** with the help of the PHWM algorithm, and prove that a ϵ-NE can be rapidly ensured with significantly reduced algebraic complexity.

Should *Assumption 1* be satisfied, a perfect matching $\mathcal{M}$ can be expected. Then, the PDFM algorithm can be readily applied for each unit, its procedures are summarized in Algorithm 1. It can be easily concluded that, in the PDFM algorithm, we propose for units with fewer connected edges (i.e., degree) to have a higher priority to possess an edge. Moreover, in the case that the degree of a unit decreases to 1, then this edge will be associated with this particular unit permanently, which again ensures *Assumption 1*. Since every unit prefers an edge with the maximal weight, the algorithm will eventually produce an optimal outcome.

As implied in *Algorithm 1*, for the ith unit, the PDFM algorithm starts with an empty matching $\mathcal{M} = \emptyset$ and an initial degree d_i being the number of associated edges of unit i, that is, $d_i = |\mathcal{T}_i|$. Then, information exchange is operated via an all-to-all communication among units within the group, during which unit i attempts to claim its largest connected edge by broadcasting it to the group and receives information from its neighbors, and a winner will be claimed afterward for the unit with the smallest degree, or the unit with a smaller ID should two units have identical degrees. If unit i wins, it simply does nothing and gets ready for the next iteration, otherwise it gives up the edge and decreases its degree by 1. As such, after at most $|\mathcal{T}_i|$ iterations, a perfect matching can be established over all units. Then, we proceed to optimize the weight of the matching with augmenting circles. Intuitively speaking, all augmenting circles of all lengths should be optimized in order to achieve the best overall outcome, but this would be extremely time consuming and counterproductive. Hence, in this paper, we propose a parallel local optimization (i.e., PLO) scheme that dedicates to find a tradeoff between accuracy and efficiency. In particular, the proposed PLO algorithm, as illustrated in *Algorithm 2*, takes a k-circles augmentation approach that only optimizes short-length circles quantified by k, which is chosen to capture the said tradeoff, and as summarized in *Lemma 2*, the proposed PLO algorithm ensures a $\frac{2}{3}$-approximation if $k = 2$ is selected.

Algorithm 1 Parallel Degree-First Matching (PDFM)

OUTPUT: An initial perfect match $\mathcal{M}$
BEGIN:
for each unit $i \in \mathcal{V}_L$ **do**
  Sort the edges in $\mathcal{E}_i$ in descending order by weight.
  Let d_i be the number of connected edges, i.e., its initial degree.
  Let $\mathcal{O}_i \leftarrow \mathcal{T}_i$, $\mathcal{M} \leftarrow \emptyset$.
  Let $success_i \leftarrow false$.
  while $\exists j \in \mathcal{V}_L$ such that $success_j = false$ **do**
    Let $m_i \in \mathcal{O}_i$ be the target ID that has maximum weight.
    Broadcast d_i and m_i to all the units.
    Receive d_j and m_j from all other unit $j \in \mathcal{V}_L$.
    Let $winner \leftarrow i$.
    for each m_j **do**
      if $m_j = m_i$ **then**
        # We only compare units that has the same expected target as unit i #
        if $(d_j < d_i)$ **or** $(d_j = d_i$ **and** $j < i)$ **then**
          # An unit with smaller degree wins the target. #
          Let $winner \leftarrow j$.
          Break **for** loop.
        end if
      end if
    end for
    if $winner = i$ **then**
      # Win. #
      $success_i \leftarrow true$
    else
      # Lose. #
      $d_i \leftarrow d_i - 1$, $\mathcal{O}_i \leftarrow \mathcal{O}_i \backslash m_i$.
      $success_i \leftarrow false$
    end if
    Broadcast $success_i$ to all the units.
  end while
  $\mathcal{M} \leftarrow \mathcal{M} \cup (i, m_i)$
end for
END

Lemma 2. *Given a bipartite graph $\mathcal{G}$ with edge matching defined in* (1)*. Then, under Assumption 1 and Algorithms 1 and 2, the PHWM algorithm has an expected running time of* $O\left(m \log \frac{1}{\varepsilon} + deg(v)\right)$*, where m is the number of edges in $\mathcal{G}$.*

Proof. It is well established that, for a k-circle $\mathcal{C}$, the expected gain satisfies [30]

$$\mathbb{E}(g(\mathcal{C})) \geqslant \frac{1}{n}\left(w^{*} - \omega(\mathcal{M})\right) \tag{27}$$

Algorithm 2 Parallel Local Optimization (PLO)

INPUT: A perfect matching $\mathcal{M}$ from Algorithm 1
OUTPUT: A heavy weight perfect matching $\mathcal{M}$

1: Set maximum iteration steps K
2: Let $it \leftarrow 0$
3: **for** unit $i \in \mathcal{V}_L$ **do**
4:   **while** $it < K$ **do**
5:     Find an augmenting circle $\mathcal{C}_i$ using depth-first search.
6:     Broadcast $\mathcal{C}_i$ to all other units.
7:     Receive all $\mathcal{C}_j, j \in \mathcal{V}_L$ from others.
8:     **if** $\forall j \in \mathcal{V}_L$, $\mathcal{C}_j$ is empty **then**
9:       **# No augmenting circle be found. End iteration. #**
10:       GO TO END.
11:     **else**
12:       **for each** $\mathcal{C}_j$ in descending order by $g(\mathcal{C}_i)$ **do**
13:         **if** $\nexists k \in \mathcal{C}_j$ is visited **then**
14:           $\mathcal{M} \leftarrow \mathcal{M} \oplus \mathcal{C}_j$.
15:           Mark all $k \in \mathcal{C}_j$ as visited.
16:         **end if**
17:       **end for**
18:       $it \leftarrow it + 1$.
19:     **end if**
20:   **end while**
21: **end for**
END

Let $\mathcal{M}_0$ be the initial matching constructed from Algorithm 1 and $\mathcal{M}_i$ represent the matching after augmenting i k-circles, then

$$\mathbb{E}(\omega(\mathcal{M}_i)) \geqslant \left(1 - e^{-i/n}\right)\omega^* \tag{28}$$

If we set $i = n \ln \frac{1}{\varepsilon}$, then the following inequality can be obtained:

$$\mathbb{E}(\omega(\mathcal{M}_i)) \geqslant (1 - \varepsilon)\,\omega^* \tag{29}$$

which indicates that if we find $n \ln \frac{1}{\varepsilon}$ short circles and use them for augmenting, we can obtain a $\frac{k}{k+1}$-approximation. Since each left vertex (unit) works in parallel, n/k at most and 1 at least augmenting circle can be used in one iteration.

On the other hand, the average degree of left vertices is m/n, where m refers the number of edges in $\mathcal{G}$. Thus the best case for iteration time is

$$\frac{n \ln \frac{1}{\varepsilon}}{n/k} \times \frac{m}{n} = \frac{km}{n} \ln \frac{1}{\varepsilon} \tag{30}$$

and the worst case is $m \ln \frac{1}{\varepsilon}$. Since k is small and $k < n$, the running time complexity of the PLO is $O\left(m \log \frac{1}{\varepsilon}\right)$. The expected running time of the PDFM algorithm is $O\left(deg(v)\right)$ where $deg(v)$ indicates the maximum degree of all left vertices, so the total running time of the PHWM becomes

$$O\left(m \log \frac{1}{\varepsilon} + deg(v)\right) \tag{31}$$

which completes the proof. □

Theorem 1. *Consider a military conflict with swarming units and the dynamics in* (5). *If Assumption 1 and 2 are satisfied and the ARS and the proposed PHWM algorithm are applied in the search of NE, then a $\frac{1}{3}$-NE in **P2** can be found by choosing $k = 2$.*

Proof. Lemma 1 has proved that the pruning method guarantees the invariance of the results of the decision making process, so we only need to prove the sub-optimality of the PHWM algorithm.

For any matching $\mathcal{M}$, there exists a set A of independent k-circles such that[**?**]

$$\omega(\mathcal{M} \oplus A) \geqslant \frac{k+1}{2k+1} \times \frac{k}{k+1}\omega(\mathcal{M}^*) + \frac{k}{2k+1}\omega(\mathcal{M}) \tag{32}$$

Let $\omega^* = \frac{k}{k+1}\omega(\mathcal{M}^*)$, and let $\omega(\mathcal{M}_n)$ be the weight after n iterations of augmenting, thus we have

$$\omega(\mathcal{M}_{n+1}) \geqslant \frac{k+1}{2k+1}\omega^* + \frac{k}{2k+1}\omega(\mathcal{M}_n) \tag{33}$$

and

$$\begin{aligned} \omega(\mathcal{M}_{n+1}) \geqslant \tfrac{k+1}{2k+1}\left(1 + \tfrac{k}{2k+1} + \cdots + \left(\tfrac{k}{2k+1}\right)^{n-1}\right)\omega^* \\ + \left(\tfrac{k}{2k+1}\right)^{n}\omega(\mathcal{M}_0) \end{aligned} \tag{34}$$

As n increases sufficiently large, the resulting weight is

$$\omega(\hat{\mathcal{M}}) \geqslant \omega^* = \frac{k}{k+1}\omega(\mathcal{M}^*) \tag{35}$$

If we choose $n > 1$, the expected weight is $\frac{2}{3}$-approximation if $k = 2$, which completes the first part of proposition.

As mentioned before, $J = \omega(\mathcal{M})$, so we have a sub-optimal group payoff that satisfies

$$J(\hat{U}) \geq \frac{2}{3} J(U^*), 0 < \epsilon < 1 \tag{36}$$

with $\epsilon = \frac{1}{3}$. Since the individual J is in the range of $[0, 1]$, the above equation turns into

$$J(\hat{U}) \geq J(U^*) - \frac{1}{3} \tag{37}$$

And the NE in **P2** becomes

$$\begin{cases} J^B\left(\hat{U}^B, \hat{U}^R\right) \geq J^B\left(U^B, \hat{U}^R\right) - \frac{1}{3}, \forall U^B \in \Delta^B \\ J^R\left(\hat{U}^B, \hat{U}^R\right) \geq J^R\left(\hat{U}^B, U^R\right) - \frac{1}{3}, \forall U^R \in \Delta^R \end{cases} \tag{38}$$

That completes the proof. □

The flowchart of our overall decision-making procedure is given in Figure 5.

IV. Simulation Results and Discussions

In this section, effectiveness of the proposed PHWM algorithm is verified with a comprehensive swarm conflict scenario. Note that all simulations are carried out on a personal desktop computer with a 3.6GHz AMD Ryzen 3500X CPU, and we use OpenMP for multithreading to parallelize all **for unit** statements in Line 1 of *Algorithm 1* and Line 3 of *Algorithm 2*. Without loss of any generality, performance of the proposed

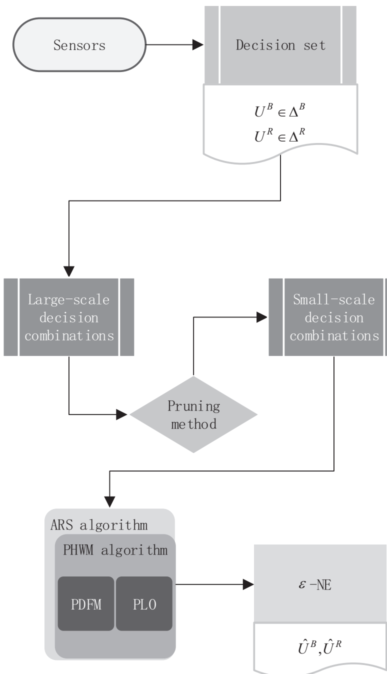


Fig. 5: The flowchart of our overall decision-making procedure

algorithm will be evaluated by the average running time of all threads. In what follows, the cardinality of combat units is assumed to be identical for both forces, and the Blue force is assumed to be on the offensive side containing N^B missiles that are targeting a fixed base guarded by the the Red force, which, in retaliation, launches N^R missiles to intercept its Blue counterparts. In our next experiments, we assume that Blue and Red have the same number of combat units, and we use N to denote the total unit number, that is $N = N^B + N^R$. The weather condition factor $\beta^w = 1$ and $\delta = 2$ in (10). The motion factor γ_{ij}^R of the Red force is defined as follows:

$$\gamma_{ij}^R \left(u_i, \hat{U}\right) = \exp\left(-\left(\left|\bar{R}_{ij}\right|^2 + |\bar{\dot{q}}|^2\right)\right), j \in \mathcal{T}_i \tag{39}$$

where $\bar{R}_{ij}, \bar{\dot{q}} \in [0, 1]$ are normalized distance and LOS rate.

Analogously, γ_{ij}^B of the Blue force is chosen as

$$\gamma_{ij}^B \left(u_i, \hat{U}\right) = \exp\left(\left|\bar{R}_{ij}\right|^2 + |\bar{\dot{q}}|^2\right) + \exp\left(-\left|\bar{R}_{i0}\right|^2\right), j \in \mathcal{T}_i \tag{40}$$

where $\bar{R}_{i0}$ is the distance between unit i and the fixed base of Red. The initial conditions and probability of kill for each unit are summarized in Table I.

We begin simulation with a simple example to verify the effectiveness of the proposed algorithm in large scale swarm combat missions. In this case, we assume both forces have 50 missiles and thus the decision sets are denoted as $U^B = \{u_1^B, u_2^B, ..., u_{50}^B\}$ and $U^R = \{u_1^R, u_2^R, ..., u_{50}^R\}$, respectively. The probability of kill PK_{ij} and destructive value W_j are randomly generated within the range of 0 to 1 and $\sum_{j=1}^{50} W_j = 1$. In addition, without loss of any generality, we assume target assignment task will only be performed during the midcourse guidance phase, the weapon-to-target designations will be fixed during the homing/terminal guidance simply due to the fact that the engagement is relatively short in the homing phase. For a fair comparison, performances of the game-based swarm conflict simulation using the Kuhn-Munkres algorithm (i.e., KM) algorithm and the proposed PHWM algorithm are rigorously studied, all simulations are conducted with the same initial conditions.

TABLE I: Initial conditions

Parameters	Red	Blue
Maximal normal acceleration (g)	7	6
Maximal speed (Mach)	1.5	1
Mass (kg)	143	143
Reference area (m^2)	0.02	0.02
Initial speed (Mach)	1.5	1
Initial position (km)	(0,0,0)	(20,20,5)
Midcourse guidance sampling time (sec)	0.5	0.5
Homing guidance sampling time (sec)	0.05	0.05
Homing guidance initiated distance (km)	10	10

Figure 6 demonstrates the fundamental concept of the proposed strategy about the calculations of the optimal trajectory. Namely, each combat unit will produce a cluster of candidate trajectories copacetic to the size of the decision set, and then an optimal trajectory will be selected rapidly using the proposed PHWM algorithm. Moreover, in order to examine the resiliency of the proposed algorithm against large scale swarm operations, simulations with varying number of combat units are performed. As summarized in Table I, the sampling/simulation step for the midcourse and homing guidance is assumed to be of 0.5/0.05 second. Noted that the CPU time of KM and our proposed PHWM algorithms refers to the time spent in calculating a single sampling step decision, a comparison of the single step CPU time is plotted in Figure 7, which demonstrates that decision time for the KM algorithm is acceptable in the case that N is relatively small, i.e., $N < 20$, but it drastically slows down if otherwise, and fails to meet the criterion once N reaches above 50. However, the decision time for our proposed parallel algorithm remains steadily rapid as N increases, and could still offer real time decision in the homing guidance phase even when N approaches 100.

Figure 8(a) and Figure 8(b) provide a comparison of the normal acceleration and group payoff value, it is obvious that the normal accelerations generated by both schemes are almost identical, while the discrepancies between scaled payoff are less than 0.04 in value, or in other words, the accuracy of the proposed PHWM algorithm is above 95% of that of the KM algorithm, which, as previous established, is regarded as the optimal solution to the underlying problem.

In the meantime, the outcomes of air combat scenarios are also examined to validate the effectiveness of the proposed method. In this experiment, we only demonstrate the performance of Red interceptors. Without loss of any generality, Blue missiles in the above four examples use game-based

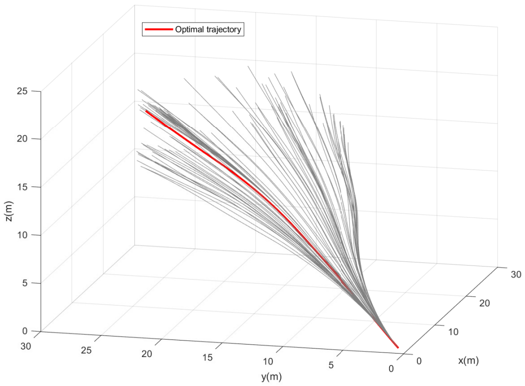


Fig. 6: Candidate trajectories corresponding to the decision set of a Red missile

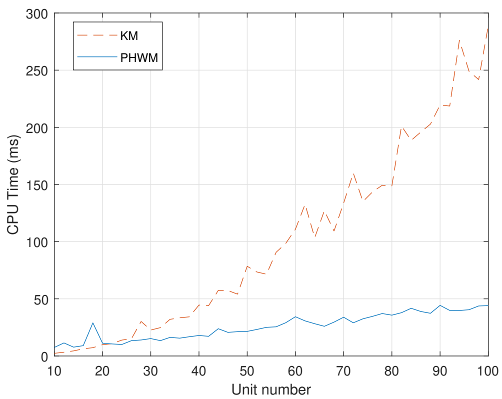
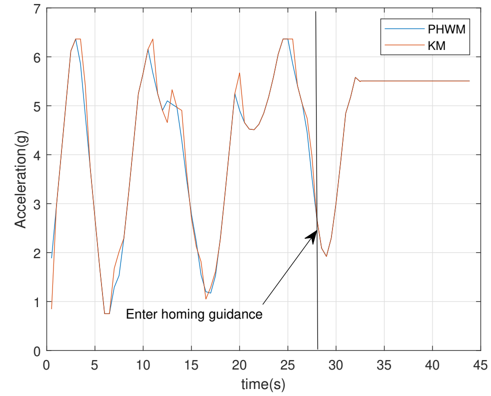
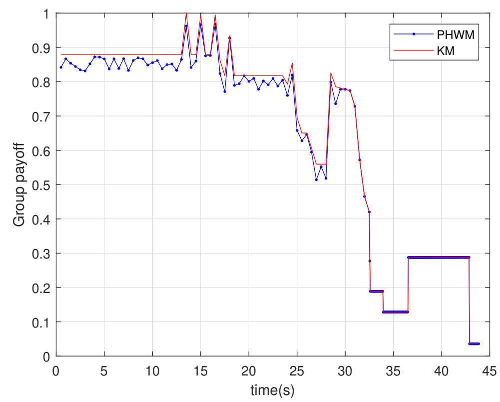


Fig. 7: CPU time



(a) Normal acceleration of a Red missile



(b) Payoff

Fig. 8: Comparison between the KM and PHWM algorithms

decisions to evade from interceptors. The miss distances of Red interceptors and Blue targets are measured by 3 miss distance levels (i.e., MDL). Figure 9 demonstrates the outcome of air combat scenarios with different number of units N, implying that no matter how N varies, our algorithm can steadily provide a very good interception performance. More specifically, our method can achieve a 100% interception when N is relatively small, and can guarantee at least 70% success rate even in a complex combat scenario. It is worth mentioning that simulations using the KM algorithm with the same experimental conditions have identical results with the PHWM algorithm.

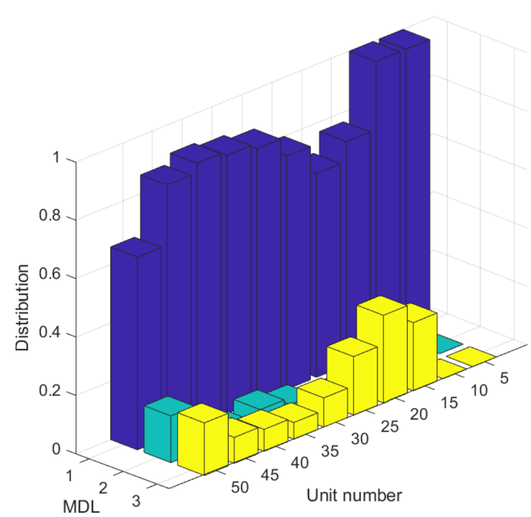


Fig. 9: Distribution of miss distance of Red missile[1]

V. Conclusions

This paper studied decision and control problems for swarm conflict management, we demonstrate that motion control and target assignment among large scale swarm operations can be treated as a noncooperative matrix game problem, and we then propose a parallel computation framework to ensure each unit seek the Nash equilibrium of the underlying game. We prove rigorously that the proposed conjecture could ensure a ϵ-Nash solutions and its algorithmic complexity is

[1]Miss distance level 1: $< 5m$; level 2: $5 - 10m$; level 3: $> 10m$

significantly reduced with graph maximum weight matching principle. Simulation results verified that the proposed scheme offers great performance, in terms of solution accuracy, with a rapid decision time, paving the way for potential applications in real-time decision making.

Acknowledgments

This work is supported, in part, by the NSFC 62088101 Autonomous Intelligent Unmanned Systems, by the Zhejiang Provincial Natural Science Foundation of China (Grant No. LR20F030003), and by the National Key Basic Research Strengthen Foundation of China (Grant No. 2021-JCJQ-JJ-1183 and 2020-JCJQ-JJ-176).

References

[1] P. Bagul, K. W. Jenkins, and L. Könözsy, "Computational engineering analysis of external geometrical modifications on MQ-1 unmanned combat aerial vehicle," *Chinese Journal of Aeronautics*, vol. 33, no. 4, pp. 1154–1165, 2020.

[2] H. Liu, F. Peng, H. Modares, and B. Kiumarsi, "Heterogeneous formation control of multiple rotorcrafts with unknown dynamics by reinforcement learning," *Information Sciences*, vol. 558, pp. 194–207, 2021.

[3] C.-X. Shi and G.-H. Yang, "Distributed learning over networks: Effect of using historical observations," *IEEE Transactions on Automatic Control*, vol. 65, no. 12, pp. 5503–5509, 2020.

[4] X. Yan and L. He, "Unpowered approach and landing trajectory planning using second-order cone programming," *Aerospace Science and Technology*, vol. 101, p. 105841, 2020.

[5] S. Chen, Y. Yang, D. Ma, X. Wang, K. Li, and C. Li, "Cooperative guidance law with impact angle coordination: A Nash approach," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2021.

[6] Y. Liu, X. Chen, Y. Zhang, and C. Li, "Sample data game strategy for active rendezvous with disturbance rejection," *Aerospace Science and Technology*, vol. 121, p. 107358, 2022.

[7] H. Dong, J. Li, C. Li, and D. Qi, "Continuous-time distributed Nash strategy over switching topologies with gain adaptation," *Systems & Control Letters*, vol. 153, p. 104955, 2021.

[8] J. B. Cruz, M. A. Simaan, G. Aca, H. Jiang, B. Letellier, M. Li, and Y. Liu, "Game-theoretic modeling and control of a military air operation," *IEEE Transactions on Aerospace & Electronic Systems*, vol. 37, no. 4, pp. 1393–1405, 2001.

[9] J. Cruz, J. B., M. A. Simaan, A. Gacic, and Y. Liu, "Moving horizon Nash strategies for a military air operation," *IEEE Transactions on Aerospace & Electronic Systems*, vol. 38, no. 3, pp. 989–999, 2002.

[10] Y. Liu, M. A. Simaan, and J. B. Cruz, "An application of dynamic Nash task assignment strategies to multi-team military air operations," *Automatica*, vol. 39, no. 8, pp. 1469–1478, 2003.

[11] K. Virtanen, J. Karelahti, and T. Raivio, "Modeling air combat by a moving horizon influence diagram game," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1080–1091, 2006.

[12] T. Zhang, C. Li, D. Ma, X. Wang, and C. Li, "An optimal task management and control scheme for military operations with dynamic game strategy," *Aerospace Science and Technology*, vol. 115, p. 106815, 2021.

[13] S. Li, M. Chen, Y. Wang, and Q. Wu, "A fast algorithm to solve large-scale matrix games based on dimensionality reduction and its application in multiple unmanned combat air vehicles attack-defense decision-making," *Information Sciences*, vol. 594, pp. 305–321, 2022.

[14] H. Lu, H. Zhang, X. Zhang, and R. Han, "An improved genetic algorithm for target assignment, optimization of naval fleet air defense," in *6th World Congress on Intelligent Control and Automation*, vol. 1, 2006, pp. 3401–3405.

[15] X. Zeng, Y. Zhu, L. Nan, K. Hu, B. Niu, and X. He, "Solving weapon-target assignment problem using discrete particle swarm optimization," in *6th World Congress on Intelligent Control and Automation*, vol. 1, 2006, pp. 3562–3565.

[16] E. Munapo, "Development of an accelerating hungarian method for assignment problems," *Eastern-European Journal of Enterprise Technologies*, vol. 4, no. 4-106, pp. 6–13, 2020.

[17] H. Kurdi, M. F. AlDaood, S. Al-Megren, E. Aloboud, A. S. Aldawood, and K. Youcef-Toumi, "Adaptive task allocation for multi-UAV systems based on bacteria foraging behaviour," *Applied Soft Computing*, vol. 83, p. 105643, 2019.

[18] C. Leboucher, H.-S. Shin, P. Siarry, R. Chelouah, S. L. Menec, and A. Tsourdos, "A two-step optimisation method for dynamic weapon target assignment problem," in *Recent Advances on Meta-Heuristics and Their Application to Real Scenarios*, Rijeka, 2013, ch. 5.

[19] J. Chen, B. Xin, Z. Peng, L. Dou, and J. Zhang, "Evolutionary decision-makings for the dynamic weapon-target assignment problem," *Science in China Series F: Information Sciences*, vol. 52, no. 11, pp. 2006–2018, 2009.

[20] O. Karasakal, "Air defense missile-target allocation models for a naval task group," *Computers Operations Research*, vol. 35, no. 6, pp. 1759–1770, 2008.

[21] R. Beard, T. McLain, M. Goodrich, and E. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 911–922, 2002.

[22] L. Babel, "Coordinated target assignment and UAV path planning with timing constraints," *Journal of Intelligent & Robotic Systems*, vol. 94, no. 3, pp. 857–869, 2019.

[23] J.-H. Hoepman, "Simple distributed weighted matchings," *arXiv preprint cs/0410047*, 2004.

[24] A. Thorsen, P. Merkey, and F. Manne, "A simple parallel approximation algorithm for maximum weight matching," in *Proceedings of the Third Conference on Parti-*

tioned Global Address Space Programing Models, 2009, pp. 1–6.

[25] A. Azad, A. Buluç, X. S. Li, X. Wang, and J. Langguth, "A distributed-memory algorithm for computing a heavy-weight perfect matching on bipartite graphs," *SIAM Journal on Scientific Computing*, vol. 42, no. 4, pp. C143–C168, 2020.

[26] A. Gibbons, *Algorithmic Graph Theory*. Cambridge University Press, 1985.

[27] G. Chartrand, *Introductory Graph Theory*. Courier Corporation, 1977.

[28] W. Lin, C. Li, Z. Qu, and M. A. Simaan, "Distributed formation control with open-loop Nash strategy," *Automatica*, vol. 106, pp. 266–273, 2019.

[29] D. G. Galati, "Game theoretic target assignment strategies in competitive multi-team systems," Ph.D. dissertation, University of Pittsburgh, 2005.

[30] S. Pettie and P. Sanders, "A simpler linear time 2/3-ε approximation for maximum weight matching," *Information Processing Letters*, vol. 91, no. 6, pp. 271–276, 2004.