A Game Theoretic Approach to Mission Planning
For Multiple Aerial Platforms

Dan Shen*, Jose B. Cruz, Jr.,†, Genshe Chen‡, Chiman Kwan§ and Alan Vannevel**

Coordinated mission planning is one of the core steps to effectively exploit the capabilities of cooperative control of multiple UAVs. In this paper, we develop an effective team composition and tasking mechanism and an optimal team dynamics and tactics algorithm for mission planning under a hierarchical game theoretic framework. Our knowledge/experience based on static non-cooperative and non-zero Nash games are used for team composition and tasking to schedule tasks at the mission level and allocate resources associated with these tasks. Our event based dynamic non-cooperative (Nash) game is used for team dynamics and tactics to assign targets and decide the optimal salvo size for each aerial platform to achieve the minimum remaining platforms of red and the maximum remaining platforms of Blue at the end of a battle. A cooperative jamming deployment method has been developed to maximize the total probability of survival of Blue aerial platforms. A simulation software package has been developed with connectivity to the Boeing OEP (Open Experimental Platform) to demonstrate the performance of our proposed algorithms. Simulations have verified that our proposed algorithms are scalable, stable, and satisfactory in performance.

I. Introduction

In recent years a significant shift of focus has occurred in the field of autonomous UAVs as researchers have begun to investigate problems involving multiple rather than single UAV. As a result of this focus on multiple UAV system, multiple UAV coordination has received significant attention. As in natural systems, cooperation may assume a hierarchical form and the control processes may be distributed or decentralized. Due to the dynamic nature of individuals and interaction between them, the problems associated with cooperative systems usually include many uncertainties. Moreover, in many cases cooperative systems are required to operate in an adversary environment.

Planning is an essential requisite to fulfill (complex) military missions successfully. The purpose of planning is to generate and maintain efficient plans in order to guide a mission. Important planning activities are, among others, determination of an efficient route (usually called path planning) for unmanned aerial platforms such as UAVs and allocation of a set of platforms to a number of targets (usually called mission planning). Planning may need to take into account many factors such as the platform’s maximum turn radius, signature information, communication limitation, environment (terrain, weather), and target’s distribution. Desired optimization criteria are based on the engagement rules and tactics. It could be formulated by either a single objective function or multiple objective functions. The engagement rules and the weapon systems govern other features and constraints of the problem, such as platform’s maneuvering capabilities.

Cooperative mission planning for autonomous vehicle teams is of great interest. A significant amount of current research activities focus on cooperative control of UAVs and some possible research directions in this field are unified in [3] and [7]. Cooperative real-time search and task allocation algorithms are presented in [8]. A genetic algorithm for task allocation is proposed in [10]. Another mission planning approach is described in [9]. In Solutions to general UAV cooperative control problems in adversarial environments can be obtained by solving game

* Ph. D. Student, Department of Electrical and Computer Engineering, 2015 Neil Ave, Columbus, OH, 43210
† Professor, Department of Electrical and Computer Engineering, 2015 Neil Ave, Columbus, OH, 43210.
‡ Senior Research Scientist, Intelligent Automation, Inc., AIAA Senior Member. E-mail: gchen@i-a-i.com.
§ Vice President, Intelligent Automation, Inc., 15400 Calhoun Dr., Suite 400, Rockville, MD 20855
** Office of Navy Research.
problems introduced in [4] and implemented in [1]. Additional game-based works focusing on target assignment of a
group of UAVs are in [5] and [6].

The contribution of this paper is as follows: First, we present an improved game theoretic framework of mission
planning. Second, we incorporate the idea of expert/knowledge systems. Third, we implement our approach in
software with connectivity to the OEP (Open Experimental Platform) [2] from Boeing. The overall architecture is
described in Section II, in which the upper level non-cooperative and non-zero game based TCT is reported with
details first. Then we present our lower level TDT. Simulation results are reported in section III. Finally, conclusions
are in section IV.

II. Game Theoretic Mission Planning

The main goal of coordinated mission planning is to develop and provide an effective team composition and tasking
(TCT) mechanism and an optimal team dynamics and tactics (TDT) algorithm to destroy the opposing force combat
capabilities. In order to accomplish this, a hierarchical game theoretic framework is developed here. Our
Coordinated Mission Planning approach is developed and implemented as a software package, which connects to the
MICA (Mixed Initiative Control of Automa-Teams) OEP (Open Experimental Platform).

The main purpose of our knowledge/experience, based on a static non-cooperative and non-zero sum Nash
game, is to develop and provide an algorithm to schedule tasks at the mission level and allocate resources associated
with these tasks. First, the Rules of Engagements are included here as constraints. Second, several criteria are
considered in order to develop our object (cost) function. Third, the game approach addresses the uncertainty of the
presence of an adversarial battle environment.

The upper level game provides an algorithm for non-homogenous resource allocation based on the following
information: the number of Red Areas in a scenario, the number of targets in each Red Area, the air defensibility of
each target, the target status, classified as potential, known or unknown, and information ability of each target, such
as sensor signatures and communication capability, and the Blue force information as well. Our game framework at
this level will estimate the loss of both sides by a probability inner battle field model, which is based the information
of both sides. The team composition and tasking information architecture is developed and used at the lower level
game.

The event based lower level non-cooperative (Nash) game is used to assign targets and decide the salvo size for
each aerial platform. Furthermore, the lower game will find an optimal deployment of decoys and avoid collateral
damage. Here we assume that the Red units are also optimizing and coordinating their targeting strategy against the
Blue units and as a result determine the target selection strategy based on a game theoretic approach. At the same
time, the lower level game determines an “estimate” of the Red team’s salvo size strategy.

The major accomplishments at this level are as follows: A non-zero-sum non-cooperative game theoretical
algorithm has been developed to determine the optimal salvo size to achieve the minimum remaining platforms of
red and the maximum remaining platforms of blue at the end of a battle; a de-centralized target assignment
algorithm is developed to find optimal aerial platform – Red target pairs; a cooperative decoy deployment method
has been developed to maximize the total probability of survival of Blue aerial platforms.

Fig. 1 shows the relationship between TCT and TDT. They will be described in the following subsections
A. Upper Level Team Composition and Tasking

Suppose there are $N^{BP}$ Blue Aerial Platforms, $N^{BF}$ blue force concentration areas and $N^{RA}$ red areas with red forces in a typical scenario. The strategic objectives of blue forces during the fixed time period $[0, T]$ include: to protect the blue concentrations from attack by red SSMs, armour and troops; to neutralize the Integrated Air Defense System (IADS) and eliminate red SAM sites in order to provide safe operations for Blue Army fixed and rotary aircraft; to eliminate the EW Radars, SAMs and C2 Facilities authorized; to avoid destroying civilians/non-combatants and cities/cultural landmarks (for example, the location of Red command and control facilities may be in or near schools, churches and hospitals).

Let $B^{SSN}, B^{SWN}, B^{LSN}, B^{LWN}, B^{CON}$ denote the number of Small Sensors, Small Weapons, Large Sensors, Large Weapons and Combination Blue Aerial Platforms involved in the battle respectively and $BP$ be the type number of the Blue Platform. Obviously, here $P^B = N^{SS} + N^{SW} + N^{LS} + N^{LW} + N^{CO} = N^{BP}$. All the Blue Aerial Platforms are equipped with Warning Sensors which detect SAM Tracking Radars. However, the Weapon Aerial Platforms only have ESM, no GMTI & Imaging radars. Let $N^{RLSAM}, N^{RMSAM}, N^{RT}, N^{ESM}, N^{EW}, N^{RTR}, N^{RR}$ denote the total number of the Red Long SAM sites, Red Medium SAM sites, Red Troops, ESMs, EW radars, Red Tracking Radars, Red Search Radars and $RP$ be the type number of the Red Platform.

At this level, the objective of each side is to allocate their forces into $N^{RA} + N^{RF}$ areas to obtain the highest performance score. The score is based on the unit values of the destroyed units. Table 1 and Table 2 give an example of the target values for the Blue Force and the Red Force respectively.

**Table 1 Target Values for Blue Force**

<table>
<thead>
<tr>
<th>Unit Type</th>
<th>Unit Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESM sensor, or EW radar</td>
<td>10</td>
</tr>
<tr>
<td>Medium SAM</td>
<td>20</td>
</tr>
<tr>
<td>Long SAM</td>
<td>30</td>
</tr>
<tr>
<td>Red Troop (Mobile HQ, Personnel Carrier)</td>
<td>15</td>
</tr>
<tr>
<td>Red Search Radar, Red Tracking Radar</td>
<td>12</td>
</tr>
</tbody>
</table>

**Table 2 Target Values for Red Force**

<table>
<thead>
<tr>
<th>Unit Type</th>
<th>Unit Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Sensor Blue Aerial Platform (SS)</td>
<td>10</td>
</tr>
<tr>
<td>Large Sensor Blue Aerial Platform (LS)</td>
<td>20</td>
</tr>
</tbody>
</table>
Small Weapon Blue Aerial Platform (SW) 15
Large Weapon Blue Aerial Platform (LW) 30
Combination Blue Aerial Platform (CO) 20

The control commands for the Blue Force and the Red Force are denoted by $U^B$ and $U^R$ respectively.

$$U^B = \left(n^B_{i,j}\right)_{i^{\text{blue platforms}}, j^{\text{Red units}}}$$

where, $n^B_{i,j} \in \{1^{\text{Red units}} \cup \{0\}\}$ and $\sum_{i=1}^{N_{\text{blue platforms}}} n^B_{i,1} = N^{\text{RSS}}$, $\sum_{i=1}^{N_{\text{blue platforms}}} n^B_{i,2} = N^{\text{RSAM}}$, $\sum_{i=1}^{N_{\text{blue platforms}}} n^B_{i,3} = N^{\text{RT}}$, $\sum_{i=1}^{N_{\text{blue platforms}}} n^B_{i,4} = N^{\text{ESM}}$ and $\sum_{i=1}^{N_{\text{blue platforms}}} n^B_{i,5} = N^{\text{BLW}}$.

$$U^R = \left[n^R_{i,j}\right]_{i^{\text{Red units}}, j^{\text{blue platforms}}}$$

where, $n^R_{i,j} \in \{1^{\text{blue platforms}} \cup \{0\}\}$ and $\sum_{i=1}^{N_{\text{red targets}}} n^R_{i,1} = N^{\text{RSAM}}$, $\sum_{i=1}^{N_{\text{red targets}}} n^R_{i,2} = N^{\text{RT}}$, $\sum_{i=1}^{N_{\text{red targets}}} n^R_{i,3} = N^{\text{ESM}}$ and $\sum_{i=1}^{N_{\text{red targets}}} n^R_{i,4} = N^{\text{BLW}}$ and $\sum_{i=1}^{N_{\text{red targets}}} n^R_{i,5} = N^{\text{CO}}$.

Then the objective functions are defined as follows,

$$J^B(u^B, u^R) = \sum_{i=1}^{N_{\text{blue platforms}}} \sum_{j=1}^{N_{\text{red targets}}} f_{RB}(u^B, u^R, i, j; T)V^R_j$$

$$J^R(u^B, u^R) = \sum_{i=1}^{N_{\text{red targets}}} \sum_{j=1}^{N_{\text{blue platforms}}} f_{RB}(u^B, u^R, i, j; T)V^B_j$$

where,

$f_{RB}(u^B, u^R, i, j; T)$ is the experience/knowledge based probability of the event that a type $j$ Red Force will be killed by the Blue Force at Area $i$.

$f_{RB}(u^B, u^R, i, j; T)$ is the experience/knowledge based probability of the event that a type $j$ Blue Force will be killed by the Red Force at Area $i$.

$V^R_j$ and $V^B_j$ are the platform values of type $j$ of the Red force and the Blue force respectively.

After each stage of the battle, the real results will be used as feedback information to adjust the two experience/knowledge based probability functions $f_{RB}(u^B, u^R, i, j; T)$ and $f_{RB}(u^B, u^R, i, j; T)$. At this stage, our upper level knowledge/experience based static non-cooperative and non-zero Nash game is closed loop and self adaptive.

**B. Lower Level Team Dynamics and Tactics**

At this level, there are two main issues: how to assign the red targets to each blue platform, and how to optimally terminate the assigned Red Targets.

After getting the updated battle field information from our search approach, all blue platforms in a specific area or team will be assigned or reassigned to red targets. Here we present the de-centralized target assignment algorithm.

Suppose there is a scenario: $N_{\text{red targets}}$ red targets in an area with two most-likely-type probabilities $P^1_{i}\text{ and } P^2_{i}$ of $P^{\text{1st}}_{i}$ and $P^{\text{2nd}}_{i}$ for each red unit. The platform type of the $i^{\text{th}}$ blue unit is denoted by $P^{\text{th}}_{i}$. For each blue platform, define the performance function as
\[
\Phi_n = \frac{1}{V_{p,\alpha}^n} \left\{ \sum_{i=1}^n w_i \left[ p_i^{1\text{st}} \left( V_{p,\alpha}^R(P_{i,\text{1st}}^{\alpha}, P_{i,\text{1st}}^{\beta}) - V_{p,\alpha}^B(P_{i,\text{1st}}^{\alpha}, P_{i,\text{1st}}^{\beta}) - w_{i,a} \frac{d_{i,a}}{s_n} \right) \right] - w_r^d \delta_n S_n \right\} 
\]

where,
\[ w_i^r, w_{i,a}, w_r^d \in [0,1] \] are relative weights;

\[ \nu_Y^R \text{ and } \nu_Y^B \] are the unit values of the Y type Red unit and Blue unit, respectively;

\[ P^{RB}(X,Y) \] is the probability of the event that an X type Red unit will be destroyed by a Y type Blue platform;

\[ P^{BR}(X,Y) \] is the probability of the event that a Y type Blue platform will be destroyed by an X type Red unit;

\[ d_{i,a} \] is the distance between the \( i^{\text{th}} \) red unit and the \( n^{\text{th}} \) blue platform;

\[ s_n \] is the speed of the \( n^{\text{th}} \) blue platform;

\[ \delta_n = 1, \text{ if the } n^{\text{th}} \text{ blue platform has already been assigned a target. Otherwise } \delta_n = 0; \]

\[ S_n \] is the switching cost of the \( n^{\text{th}} \) blue platform.

Then, \( V_{p,\alpha}^R(P_{1,\text{1st}}^{\alpha}, P_{1,\text{1st}}^{\beta}) \) is the expected score the blue side will obtain if the \( i^{\text{th}} \) red target is assigned to the \( n^{\text{th}} \) blue platform. \( V_{p,\alpha}^B(P_{1,\text{1st}}^{\alpha}, P_{1,\text{1st}}^{\beta}) \) is the expected score the blue side will lose if the \( i^{\text{th}} \) red target is assigned to the \( n^{\text{th}} \) blue platform. \( w_i d_{i,a} \) is the weighted time cost. Finally, with a consideration of the target-switching cost,

\[
\sum_{i=1}^n w_i \left[ p_i^{1\text{st}} \left( V_{p,\alpha}^R(P_{i,\text{1st}}^{\alpha}, P_{i,\text{1st}}^{\beta}) - V_{p,\alpha}^B(P_{i,\text{1st}}^{\alpha}, P_{i,\text{1st}}^{\beta}) - w_{i,a} \frac{d_{i,a}}{s_n} \right) \right] - w_r^d \delta_n S_n \]

is the total “virtual” score the blue side will gain if the \( n^{\text{th}} \) blue platform is used. Here we use the word “virtual” due to the fact that only one target can be assigned to the \( n^{\text{th}} \) blue platform. Given that the same scores are gained by two blue units, the less the blue platform value, the better. That is why we put \( \frac{1}{V_{p,\alpha}^n} \) in front of the final score in the performance function.

Our target assignment approach has two steps. The first one is to give a priority sequence of the \( N^{\text{th}} \) Blue platforms by the value of the performance function from largest to smallest. This step de-centralizes the whole processing. The second one is to let the Blue platform “greedily” choose their target in the order of the rank created in the first step. In this step, the current \( k^{\text{th}} \) blue unit will choose the red target which has the biggest value of the following utility function

\[
Y_k(i) = \frac{1}{V_{p,\alpha}^k} \left\{ w_i \left[ p_i^{1\text{st}} \left( V_{p,\alpha}^R(P_{i,\text{1st}}^{\alpha}, P_{i,\text{1st}}^{\beta}) - V_{p,\alpha}^B(P_{i,\text{1st}}^{\alpha}, P_{i,\text{1st}}^{\beta}) - w_{i,k} \frac{d_{i,k}}{s_k} \right) \right] - w_r^d \delta_k S_k - w_r^d \delta_k^* \right\} 
\]

where the new variable \( w_r^d \) is a relative big number as a weight.

\[ \delta_k^* = 0 \text{ if the } i^{\text{th}} \text{ red unit has less than } m(=2) \text{ blue preys; otherwise, } \delta_k^* = 1. \]
Here, 
\[
\begin{align*}
   w_k^{(i)} & = \left[ P_{ik}^{(i)} \left( V_{ik}^{(i)} P_{ik}^{(i)} (P_{ik}^{(i)} + P_{ik}^{(i)}) - V_{ik}^{(i)} P_{ik}^{(i)} (P_{ik}^{(i)} + P_{ik}^{(i)}) - W_{ik} \frac{d_{ik}}{s_i} \right) ight] - w_k^{(i)} \delta_i S_k^{(i)},
\end{align*}
\]

having the same meaning as in (5), is the total expected score the blue side will gain if the \( j^{th} \) red target is assigned to the \( n^{th} \) blue platform. To prevent the \( k^{th} \) blue unit from choosing a target which has already \( m(=2) \) blue preys, we subtract a relatively large weight \( w_k^{(i)} \) if \( \delta_i^{(i)} = 1 \).

In order to improve the attacking efficiency, we proposed a jamming-decoy-weapon approach.

For the blue weapon platform, we assume that Blue decoys (100) replace weapons on 1:1 basis with GPS Bombs or Submunitions or 1:2 basis with Anti-radiation or Seeker Missiles. We also assume that only the Sensor platform has a jamming ability.

Our jamming objective is to develop the Blue jammer deployment strategy such that Blue platforms can avoid being detected by Red radars. Because Red radars can network, the ideal situation is to jam all of the Red radars. But sometimes due to the limited Blue jamming power, it is impossible to fulfill the ideal situation. Thus we reset the jamming goal to be jamming as many Red radars as possible. This is an exhaustive search problem and the computation depends on the number of both Blue jammers and Red radars.

A near-optimal static jamming strategy with less computation is developed as follows: Suppose that the initial states of all jammers are closed. In order to protect Blue Sensor platforms from Red attack, a conservative strategy is used: only the jammers loaded on the Sensor Platform which cannot be reached by all of the Red weapons are regarded as available. The information whether a Blue platform has been painted or not is known, but it does not know which Red radars painted it.

Step 1 is to guess the aim-point of each Red radar site. It is assumed that each Red radar site can track only one target Blue platform at a time. It is possible that several radar sites can simultaneously track the same blue platform. For the purpose of the jamming which target blue platform this radar is currently tracking. Although we cannot know exactly which blue platform is tracked by which radar, the probability of detection that a given blue platform is tracked by a certain radar site can be estimated according to the Blue platform type, radar type and distance between the target blue platform and the radar. Suppose that there are \( N \) jammers and \( M \) radar sites. Denote \( P_{ij} \) as the probability that the \( i^{th} \) radar could successfully detect the \( j^{th} \) blue platform, \( i=1, 2, \ldots, M \) and \( j=1, 2, \ldots, N \). Determine each Red radar’s target with the maximal detection probability criterion, that is to regard the \( j^{th} \) blue platform as the \( i^{th} \) radar’s target if

\[
   P_{ij}^* \geq P_{ij}, \quad (1 \leq i \leq M \quad \text{and} \quad 1 \leq j \leq N). \]

Step 2 is to sort the Red radar sites. We assign every Red radar a weight which is a function of radar type, target Blue platform type, target Blue platform value and detection probability. This weight describes the radar’s importance. For example, for the same type radars, the radar with a large detection probability and a large target Blue platform value has higher weight. Sort the radars by their weights. The larger weight the radar has, the earlier it should be jammed.

Step 3, we assign jammers for each Red radar that can be jammed by the current available jammers. In order to jam as many radars as possible, the proposed principle is to jam a radar using the least jamming power possible.

Step 3.1 For a given radar (indexed as \( i \)), we first calculate the jamming signal \( J_{ij} \) of each available jammer (indexed as \( j \)) to this radar.

Step 3.2 If all available jammers are selected, but still cannot jam a single radar, i.e., \( \sum J_{ij} \leq S \) (where \( S \) is the required jamming power for Radar \( i \)), then this radar is selected out, and marked as “cannot be jammed”.

Step 3.3 Otherwise this radar can be jammed, continue to the following steps.

Step 3.3.1 Sort these available jammers in ascending order by its contribution (jamming signal denoted as \( J_{ij} \)), such that \( J_{i1} \leq J_{i2} \leq \ldots \leq J_{in} \), \( n \) is the number of available jammers at the present time.

Step 3.3.2 If the jammer with the largest jamming signal \( J_{in} \) can jam this radar \( (J_{in} > S) \), then we choose the \( j^{th} \) jammer such that the \((j-1)^{th}\) jammer cannot jam the radar but the \( j^{th} \) jammer can, \( J_{ij} \leq S \) \( J_{ij+1} > S \). Mark this jammer as used. Note the number of available jammers decreases by one when a jammer is marked as used.

Step 3.3.3 If the jammer with the largest jamming signal \( J_{in} \) cannot jam this radar, i.e. \( J_{in} \leq S \), then we choose this jammer as the first one to jam this radar, and let \( S = S - J_{in} \) and \( n = n - 1 \).

Step 3.3.4 Repeat step 3.3.2 to step 3.3.3 to select the other jammers for this radar until it is jammed.
A typical scenario is shown in Fig. 2. The jamming procedure is illustrated in Fig. 3.

For the blue weapon platforms, the salvo size and decoy strategy are important to destroy as many red targets as possible and at the same time to save themselves. Suppose there are up to $TB$ teams for Blue and only one for Red. For each of the two forces we define at time stage $k$ objective functions to be maximized by each force correspondingly.
\[ J^B(k) = \sum_{i=1}^{TB} \left( w^B_i \sum_{X_i \in \mathcal{EX}} \left( f^X_i \hat{p}_i^X(k) \right) \right) + IW \cdot g^B(\alpha) \sum_{Y_i \in \mathcal{LY}} \left( g^Y_i \hat{p}_i^Y(k) \cdot (1 - p_{ri} - p_{ry}) \right) \]
\[-g^B(\alpha) \sum_{Y_i \in \mathcal{LY}} \left( g^Y_i \hat{p}_i^Y(k) \cdot p_{ri} \right) - g^B(\alpha) \sum_{Y_i \in \mathcal{LY}} \left( g^Y_i \hat{p}_i^Y(k) \cdot p_{ry} \right) \]

\[ J^R(k) = \sum_{Y_i \in \mathcal{LY}} \left( g^Y_i \hat{p}_i^Y(k) \right) - f^R(\beta) \sum_{X_i \in \mathcal{EX}} \left( w^R_i \hat{p}_i^R(k) \right) \]

In the above expressions, \( w^B_i \) and \( w^R_i \) are weights assigned to Blue team \( ti \) by Blue and Red forces, respectively. \( IW \) is a weight signifying the importance of White assets weighted by Blue, which is between 0 and 1. In other words, if avoiding collateral damage is very important for Blue, \( IW \) would be assigned a large value. On the other hand, if Blue does not care about collateral damage, \( IW \) should have a small value. \( p_{ri} \) is the likelihood of unit \( i \) to be a correct Red units and \( p_{ry} \) stands for the likelihood of unit \( i \) to be incorrect Red units. \( f^X_i \) and \( g^Y_i \) are nonnegative coefficients that account for the distribution of weights to assign relative importance to the terms in the objective function. \( \alpha \) and \( \beta \) are two scalar parameters whose values are respectively decided by Blue force and Red force during the non-cooperative game. \( g^B(\alpha) \) and \( f^R(\beta) \) are functions that account for changing the salvo size of the Blue and Red, which prevents the Blue or Red units from applying the maximum salvo size with negative consequences of running out of weapons for future salvos. \( \hat{p}_i^X(k) \) and \( \hat{p}_i^Y(k) \) are normalized platforms for all \( k = 0,1,\ldots,K \) : 
\[ \hat{p}_i^X(k) = p_i^X(k) / p_i^X(0), \quad \hat{p}_i^Y(k) = p_i^Y(k) / p_i^Y(0). \]

It can be observed that the objective functions are linear combinations of normalized and weighted platforms. The objective of Blue is to maximize its own platforms while minimizing its rival’s and avoid collateral damage to White. A pair \((\alpha^*, \beta^*)\) constitutes a Nash equilibrium solution to the above bi-matrix game \((J^B, J^R)\) if it satisfies

\[ J^B(\alpha^*, \beta^*) \geq J^B(\alpha, \beta) \]
\[ J^R(\alpha^*, \beta^*) \geq J^R(\alpha, \beta) \]

The mission planning level will allow the human operator to examine “what if” scenarios (new target priorities, different rules of engagement, different cultural and social idiosyncrasies, effects-based operator and collateral damage effects) by choosing different objective functions, or different weights and coefficient in objective functions in equations (7) and (8).

Although it seems obvious that the modeling of the opponent activities as controlled by an intelligent player would lead to better tactics decision, there are some mitigating factors. An obvious difficulty with the application of advanced techniques is the possible computational burden. One approach to the reduction of this burden is the use of hierarchical decompositions of the problem. Another difficulty with the application of advanced techniques is the numerical determination of the solution of closed-loop Nash equilibrium, which is sometimes impossible to obtain unique Nash solution. We have investigated co-evolutionary models of computation i.e., those where more than one evolution process takes place. In real life, this happens where there are different populations (e.g., parasites or predators) that interact with each other. In such systems the value function for one population may depend on the state of the evolution processes in the other population(s). In the method, there are two populations of individuals investigated, where each population corresponds to a particular decision-maker (Red or Blue). Each individual in a population represents a strategy for the game. Specialized “genetic” operators alter the genetic composition of children during reproduction and the evaluation functions in each population depend on the current state of such a co-evolution model.

### III. Simulation

To illustrate how our algorithm works, we developed a software package, which can connect to the MICA (Mixed Initiative Control of Automa-Teams) OEP (Open Experimental Platform) [2].

A typical scenario shown in Fig. 4 with several experiments is simulated on our software to evaluate the performance of our proposed Mission Planning algorithm for Multiple Aerial Platforms.
In the scenario, Molian rebels (Red Force) supported by terrorist organizations and a neighboring country (NNC) are overrunning the country of Molia. Molia has asked the US for support. Friendly Molian forces face numerically superior Molian Red Force plus volunteers from a neighboring country. Molian Rebels have integrated their air defense into the neighboring country’s EW Radars, SAMs and C2 structure. These EW Radars, SAMs and C2 structures are deemed acceptable targets but collateral damage must be avoided. Red Forces have overrun portions of the country. Blue ground forces consists of the Molian Army plus a limited number of US ground forces. The combine Blue Force is relying on UAV air power in support of these limited ground forces to contain the situation until additional US ground forces arrive.

For this scenario, after running our upper level knowledge/experience based static non-cooperative and non-zero Nash game, we obtain the following blue force allocation chart (Fig. 5). (Note that since the system is stochastic, all the figures below are based on the averages of 30 simulations) In this stage, four missions are assigned: one Close Air Support mission is assigned to team 1 (4 COMB UAVs, 1 Small Sensor UAV, 1 Small weapon UAV) to protect Blue Base; one Close Air Support mission is allocated to team 2 (2 COMB UAVs, 1 Small Sensor UAV, 1 Small weapon UAV) to guard Blue Concentration Area #1; one Close Air Support mission is sent to team 3 (2 COMB UAVs, 1 Small Sensor UAV, 1 Small weapon UAV) to guard Blue Concentration Area #2; and one SEAD mission is assigned to team 4 (5 small sensor UAVs, 5 small weapon UAVs, 6 large sensor UAVs, 6 large weapon UAVs) to attack Red Area #2. No blue force is sent to Red Area #1, #3 and NNC.

The result of the stage 1 is shown in Fig. 6. All the red forces (2 long SAM sites, 7 medium SAM sites, 1 Red Ground Troop composed of 4 Tanks, 2 EW radars, 2 Tracking radars and 2 search radar) in Red Area # 2 are
terminated with the cost of losing three Large Weapon UAVs and 2 Large Sensor UAVs partially damaged in Team 4. We also compare the result with one of other options, such as no jamming strategy, no decoy approach, and only weapon method. The result is shown in Fig. 7. (Only the damage information of team 4 is compared.)

![Fig. 6 The result of Stage 1: (a) global view; (b) details of Red Area #2](image)

![Fig. 7 The damage comparison of various options in the Blue force during Stage 1](image)

From the damage comparison results, we can see our proposed jamming-decoy-weapon approach is better than other methods. It also shows that decoy strategy is more efficient in saving the Blue forces than the Jamming strategy. We also notice that all blue forces are totally destroyed by the red force in the weapon only approach during the stage 2.

After Stage 1, we should call the upper level game again with the updated battle field information. In this step, the experience/knowledge functions will be adjusted too. The blue force allocation chart of stage 2 is shown in Fig. 8. The simulation result of stage 2 is illustrated in Fig. 9.
In Stage 2, there are two SEAD missions. Mission 1 is to attack the Red Area #1 and Mission 2 is to dominate the Red Area #3. During Mission 1, two small weapon UAVs are shot down by the Red force. In the SEAD Mission
2, one small sensor UAV and one large sensor UAV are totally destroyed by Red Integrated Air Defense System (IADS).

In Fig. 9, notice that all the blue force in team 2, which is assigned the Close Air Support to protect the Blue Area #1, are destroyed. There is one destroyed small weapon UAV in team 3, which has the task to protect Blue Concentration #2. The damage comparison results for various options during stage 2 are shown in Fig. 10. With the damage information we look back to the upper level game, it is not surprising that there are no Blue Forces assigned to guard Blue Area #1 in the Team Composition and Tasking output, which is shown in Fig. 11.

![Fig. 10 The damage comparison of various options in the Blue force during Stage 2](image)

![Fig. 11 Blue Force Allocation Chart of Stage 3](image)

The simulation result of stage 3 is shown in Fig. 12. All the Red forces in NNC are destroyed. It is surprisingly good. We think there are two reasons to interpret it. One is that we assigned two teams to attack NNC. The other is that most unites in NNC are red force support devices such as Radars and ESM sensors. We also found that there are some white objects that are damaged because some red units are very close to them as shown in Fig. 13.

![Fig. 12 Simulation Result of Stage 3](image)
Fig. 12 The result of Stage 3

Fig. 13 Dangerous red units very near white objects

Fig. 14 shows the damage comparison of various options in the Blue force during Stage 3. Clearly our proposed option achieved the minimal loss of Blue force. It should be noted that jamming strategy is more efficient in saving the Blue force than decoy strategy during this stage. This is different from the above two stages. We think that it is because most units in NNC are red force support devices such as Radars and ESM sensors.
After calling the upper level Team Composition and Tasking module again, we get the following Blue Force Allocation Chart of the final stage.

IV. Conclusion

In UAV cooperative control, mission planning is of great importance for the efficient destruction of the opposing force combat capabilities. In this paper, an effective TCT mechanism and an optimal TDT algorithm is developed under a hierarchical game theoretic framework. The simulations show that the game theoretic algorithm is capable of solving the coordinated mission planning problem.

References

1. José B. Cruz, Jr, Genshe Chen, Denis Garagic, Xiaohuan Tan, Dongxu Li, Dan Shen, Mo Wei, Xu Wang, “Team Dynamics and Tactics for Mission Planning”, Proceedings of the 42nd IEEE Conf. on Decision and Control, Hawaii, December 2003