

2006 CCRTS

The State of the Art and the State of the Practice

A Game Theoretic Approach to Threat Intent Inference

Assigned Track: C2 Modeling and Simulation
Assigned Paper Number: C-076

Dan Shen*
The Ohio State University
205 Dresses Laboratory, 2015 Neil Ave
Columbus, OH 43202
Ph: (614)292-2572
Email: shen.100@osu.edu

Genshe Chen (Principal point of contact)
Intelligent Automation, Inc.
15400 Calhoun Drive, Suite 400
Rockville, MD 20874
Tel: 301 294 5218 (direct)
Fax: 301 294 5201
Email: gchen@i-a-i.com

Jose B. Cruz, Jr.,
The Ohio State University
205 Dresses Laboratory, 2015 Neil Ave
Columbus, OH 43202
Ph: (614)292-1588
Email: cruz.22@osu.edu

Chiman Kwan
Intelligent Automation, Inc.
15400 Calhoun Drive, Suite 400
Rockville, MD 20874
Tel: 301 294 5238 (direct)
Fax: 301 294 5201
Email: ckwan@i-a-i.com

ABSTRACT

A game theoretic approach to threat intent inference

In the adversarial military environment, it is important to efficiently and promptly predict the enemy's tactical intent from the lower level spatial and temporal information. In this paper, we propose a decentralized Markov game (MG) theoretic approach to estimate the belief of each possible enemy COA (ECO), which is utilized to model the adversary intents. It has the following advantages: 1) Decentralized. Each cluster or team makes decisions mostly based on the local information. We put more autonomies in each group allowing for more flexibilities; 2) Markov Decision Process (MDP) can effectively model the uncertainties in the noisy military environment; 3) Game model with three players: red force (enemies), blue force (friendly forces), and white force (neutral objects); 4) Deception. With the consideration that asymmetric enemy may manipulate the information available to friendly force; we integrate the deception concept in our game approach to model the action of purposely rendering partial information to increase the payoffs of the enemy. 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.

1. Introduction

Game theory provides a framework for modeling and analyzing various interactions between intelligent and rational decision makers, or players in conflict situations, in which every individual decision maker is not in complete control of other decision units entering into the environment. The idea of the game can be tracked back to the Babylonian Talmud which is the compilation of ancient law and tradition set down during the first five centuries A.D.. However, it was not until 1944 that the mathematical theory of games was invented by John von Neumann and Oskar Morgenstern [1].

Mathematically, Game theory is used to study strategic situations where players choose different actions in an attempt to maximize their returns, which depend upon the choices of other individuals. To make optimal movement in multi-agent systems, strategies of other agents should be taken into account and therefore it is essential to be able to model the behavior of the opponents.

In the adversarial military environment, it is important to efficiently and promptly predict the enemy's or adversary tactical intent from the lower level spatial (terrain) and temporal information. Standard AI tools for solving decision-making problems in complex situations, such as dynamic decision networks and influence diagrams, are not applicable to these kinds of situations. Game theory, on the other hand, provides a mathematical framework designed for the analysis of agent interaction under the assumption of rationality where one tries to identify the game equilibria as opposed to traditional utility maximization principles. A game component in multi-agent decision-making thus uses rationality as a tool to predict the behavior of the other agents [11-15].

In this paper, the focus is on the application of Markov Game [2], the multi-agent extensions of Markov Decision Processes (MDPs), to the estimation of enemy course of actions (COAs) [3], which approximately model the intent of targets. By successfully assessing possible future threats from the adversaries, the decision makers can make more effective targeting decisions, plan friendly COAs, mitigate the impact of unexpected adversary actions, and direct sensing systems to better observe adversary behaviors. We have achieved the following important results. First, we developed a highly innovative framework for threat intent prediction in an urban warfare setting based on Markov (Stochastic) game theory. It consists of three closely coupled activities: 1) the processing and integration of information from disparate sources to produce an integrated object state; 2) the reasoning and grouping the cooperative objects which perform common tasks; 3) predicting the intentions and CoAs of asymmetric threats. Second, we have implemented an adversary Markov game model with three players: red force (enemies), blue force (friendly forces), and white force (neutral objects). This is a significantly extension of existing game theoretic tools for modeling and control of military air operations, which does not explicitly consider the neutral force (or civilian) as an intelligent player [13]. Inherent information imperfection is considered and implemented in two methods: 1) decentralized decision making scheme; and 2) deception with bounded rationality. Third, a software prototype has been developed with connectivity to the MICA (Mixed Initiative Control of Automa-Teams) Open

Experimental Platform [4] (ontology-based virtual battlespace) to demonstrate the performance of our proposed approach. It has verified that our proposed algorithms are scalable, stable, and perform satisfactorily according to the situation awareness performance metric.

The paper is organized as follows. In section 2, we will summarize the technical approach, which includes basic ideas of Markov game, threat intent inference framework, and moving horizon control approach for game solution. Section 3 describes the experimental results. Section 4 concludes the paper.

2. Markov Game Framework

We propose a Markov game (MG) theoretic approach to estimate the belief of each possible enemy COA (ECO) because 1) Decentralized. Each cluster or team makes decisions mostly based on the local information. We put more autonomies in each group allowing for more flexibilities; 2) Markov Decision Process (MDP) can effectively model the uncertainties in the noisy military environment; 3) Game model with three players: red force (enemies), blue force (friendly forces), and white force (neutral objects). Game framework is an effective and ideal model to capture the nature of military conflicts: the determination of one side's strategies is tightly coupled to that of the other side's strategies and vice versa. With the consideration that an asymmetric threat (terrorist) may act like a neutral or white object (civilian), we also model the actions of white units in our game framework; 4) Deception [10]. With the consideration that asymmetric enemy may manipulate the information available to friendly force; we integrate the deception concept in our game approach to model the action of purposely rendering partial information to increase the payoffs of the enemy.

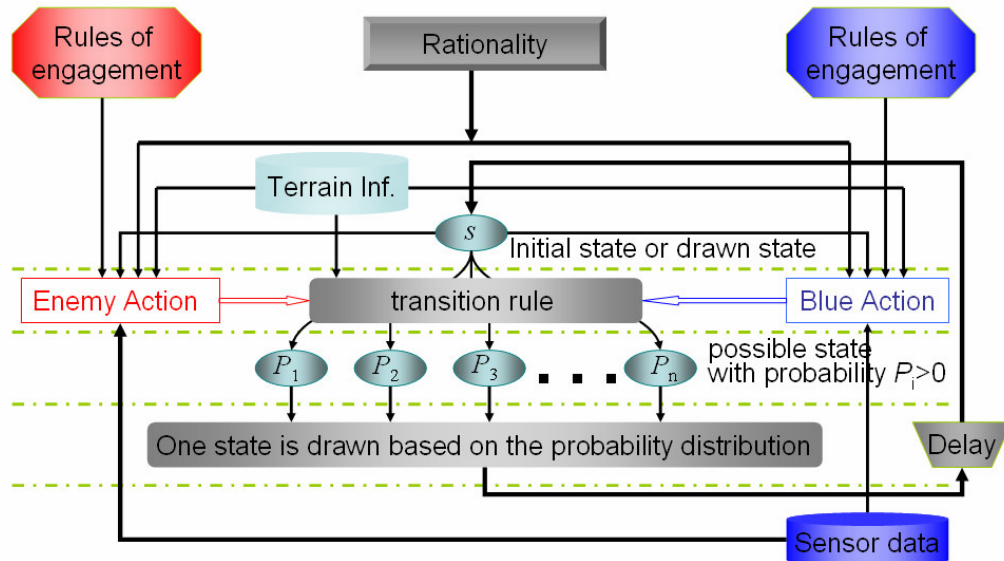


Fig. 1 Stochastic game model for intent inference

The game intent inference framework is shown in Fig. 1. It is constructed from the initial state and evolved according to the transition rule. At each time k , blue and red actions are

decided according to the various sensor data, rules of engagement, rationality, terrain information and current state. These actions decide the updated probability distribution over the state space according to the transition rules, which also takes terrain information as one of the inputs. Only one state is selected as the next state. In the real battle, the state is chosen by the nature while in our model it is drawn based on the probabilities of all possible states. The bigger the probability is, the easier it will be drawn.

By definition, a Markov (stochastic) game is given by (i) a finite set of players N ; (ii) a finite set of states S ; (iii) for every player $i \in N$, a finite set of available actions A^i (we denote the overall action space $A = \times_{i \in N} A^i$); (iv) a transition rule $q: S \times A \rightarrow \Delta(S)$, (where $\Delta(S)$ is the space of all probability distributions over S); and (v) a payoff function $r: S \times A \rightarrow R^N$. For the intent inference problem, condition (i) is obviously satisfied. Conditions (ii) and (iii) hold because we assume that rules of engagement and terrain information are known and each player has a limited set of COAs given by the doctrines, and terrain information, etc. For example, a river will limit the actions of ground forces. (iv) and (v) are designed according to the specific situations including terrain information. For our threat prediction problem, we obtain the following discrete time decentralized Markov game:

Players (Decision Makers) --- Although, in our distributed (decentralized) Markov game model, each group (cluster, team) makes decisions, there are three main players: enemy, friendly force, and white objects. All clusters of enemy (friendly force, or white objects) can be considered as a single player since they have a common objective.

State Space --- All the possible COAs for enemy and friendly force consist of the state space. An element $s \in S$ is thus a sample of enemy and friendly force COAs composed of a set of triplets (Resource, Action Verb, and Objective). As an example, an enemy COA might be: the red team 1 (Resource) attacks (Action Verb) the blue team 2 (Objective). In the context of this report, it is assumed that, for the enemy COAs, the Resource is always an adversary entity while the Objective is a friendly asset. Similarly, for the friendly force COAs, Resource is a friendly asset and Objective is an adversary entity. $s = (s^B, s^R, s^W)$ and $S = S^B \times S^R \times S^W$, where $s^B \in S^B$ is the COAs of Blue (friendly) force and $s^B = \{(r_i^B, a_i^B, o_i^B) \mid r_i^B \in R^B, a_i^B \in A^B, o_i^B \in O^B\}$, R^B, A^B, O^B are the set of the resource, action, and objective of blue force, respectively. On the other hand, $s^R \in S^R$ is the COAs of Red (enemy) force and $s^R = \{(r_i^R, a_i^R, o_i^R) \mid r_i^R \in R^R, a_i^R \in A^R, o_i^R \in O^R\}$. $s^W \in S^W$ is the COAs of White force (neutral objects).

Action Space --- At every time step, each blue group choose a list of targets with associated actions and confidences (probability distribution over the list of targets, i.e., the sum of the confidences should be equal to 1) based on its local battle field information, such as the unit type and positions of possible targets, from level 2 data fusion. Let D_i^B denote the action space of the i^{th} blue team. Each element d_i^B of D_i^B is defined as

$$d_i^B = \{(a_i^B, t_i^B, p_i^B) \mid a_i^B \in A^B, t_i^B \in O^B, 0 < p_i^B \leq 1, \sum p_i^B = 1\} \quad (1)$$

where p_i^B is the probability of the action-target couple (a_i^B, t_i^B) , which defined as the action a_i^B to target t_i^B . Therefore, the action space of blue side $A^1 = \times_{i \in R^B} D_i^B$. As an example, for the blue small weapon UAV 1 in blue team 1, its action might be $d_1^B = \{(\text{attack}, \text{red fighter 1}, 0.3), (\text{fly to}, \text{red fighter 2}, 0.5), (\text{avoid}, \text{red fighter 3}, 0.2)\}$.

Similarly, each red cluster (obtained from the level 2 data fusion) determines a probability distribution over all possible action-target combinations. Let D_i^R denote the action space of the i^{th} red cluster. Each element d_i^R of D_i^R is defined as

$$d_i^R = \{(a_i^R, t_i^R, p_i^R) \mid a_i^R \in A^R, t_i^R \in O^R, 0 < p_i^R \leq 1, \sum p_i^R = 1\} \quad (2)$$

where p_i^R is the probability of action a_i^R to target t_i^R . Therefore, the action space of red force $A^2 = \times_{i \in R^R} D_i^R$. A possible action for red platform 1 (red fighter 1) is $d_1^R = \{(\text{attack}, \text{small weapon UAV 1}, 0.6), (\text{move to}, \text{blue solider 2}, 0.2), (\text{avoid}, \text{blue solider 1}, 0.2)\}$. Remark: Action and Action Verb are different concepts. Action is a set of triplets with associated probabilities while Action Verb is just a component of triplet composed of Resource, Action Verb and Objective. All Actions are included in A^1 for player 1 (Blue force) and A^2 for player 2 (Red force). All Action Verbs are enumerated in A^B for player 1 (Blue force) and A^R for player 2 (Red force).

The actions of white objects are relatively simple. The main action type is the movement. Let D_i^W denote the action space of the i^{th} white unit. Each element d_i^W of D_i^W is defined as

$$d_i^W = \{(a_i^W, t_i^W, p_i^W) \mid a_i^W \in A^W, t_i^W \in O^W, 0 < p_i^W \leq 1, \sum p_i^W = 1\} \quad (3)$$

where p_i^W is the probability of action a_i^W to target t_i^W .

Transition rule --- Due to the uncertainty properties of military environments, we assume that the states of the Markov game have inertia so that the decision makers have more chance in pursuit of the objective of the previous actions. We define an inertia factor vector $\eta^i = (\eta_1^i, \eta_2^i, \dots, \eta_{m_i}^i)^T$ for player i , where m_i is the number of the teams or clusters of player i , and $0 \leq \eta_j^i \leq 1, 1 \leq j \leq m_i$. So, for the j^{th} team of the i^{th} player, there is a probability of η_j^i to keep the current action-target couple and a probability of $(1-\eta_j^i)$ to use the new action composed of action-target couples.

There are two steps to calculate the probability distribution over the state space S , where s_k, s_{k+1} are states of time step k and $k+1$ respectively, a_k^1, a_k^2, a_k^3 , are the decisions of player 1 (blue force or friendly force), player 2 (red force or enemy), and play 3 (white force) respectively, at time step k .

- Step 1: with the consideration of inertia factor vector η^i , we combine the current state with decisions of both players to obtain fused probability distributions over

all possible action-target couples for red and blue forces. To do this, we first decompose the current state into the action-target couples for each team of each player (red force or blue force). Let $\Psi_j^i(s_k)$ denote the resulting action-target couple related to the j^{th} team of the i^{th} player. For example, if there is one triplet of (blue team 1, attack, red fighter 2) in the current state s_k , then the action-target couple for blue team 1 (the first team of blue force) is $\Psi_1^1(s_k) = (\text{attack}, \text{red fighter 2})$. Secondly, for each specified team, say the j^{th} cluster of player 2 (enemy force), we fuse its action-target couples via modifying the probability of each possible action-target couple based on the following formula

$$\bar{p}((a_j^R, t_j^R) | s_k) = \begin{cases} p_j^R(1-\eta_j^2) & , (a_j^R, t_j^R, p_j^R) \in d_j^R \text{ and } (a_j^R, t_j^R) \notin \Psi_j^2(s_k) \\ p_j^R(1-\eta_j^2) + \eta_j^2 & , (a_j^R, t_j^R, p_j^R) \in d_j^R \text{ and } (a_j^R, t_j^R) \in \Psi_j^2(s_k) \\ \eta_j^2 & , (a_j^R, t_j^R, p_j^R) \notin d_j^R \text{ and } (a_j^R, t_j^R) \in \Psi_j^2(s_k) \\ 0 & , (a_j^R, t_j^R, p_j^R) \notin d_j^R \text{ and } (a_j^R, t_j^R) \notin \Psi_j^2(s_k) \end{cases} \quad (4)$$

There are four cases in Eq (6): 1) The action-target couple (a_j^R, t_j^R) only occurs in the current action of j^{th} cluster of player 2 and is not in the current state s_k , which can be mathematically represented by $(a_j^R, t_j^R, p_j^R) \in d_j^R \text{ and } (a_j^R, t_j^R) \notin \{\Psi_j^2(s_k)\}$. Then we know the probability of (a_j^R, t_j^R) in current state s_k is 0 and probability of (a_j^R, t_j^R) in current action is p_j^R . So, according to the definition of inertia, the fused probability of the action-target couple (a_j^R, t_j^R) is $p_j^R(1-\eta_j^2) + 0(\eta_j^2) = p_j^R(1-\eta_j^2)$. 2) The action-target couple (a_j^R, t_j^R) happens both in the current action of j^{th} cluster of player 2 and in the current state s_k . Then we know the probability of (a_j^R, t_j^R) in current state s_k is 1 and probability of (a_j^R, t_j^R) in current action is p_j^R . So, according to the definition of inertia, the fused probability of the action-target couple (a_j^R, t_j^R) is $p_j^R(1-\eta_j^2) + 1(\eta_j^2) = p_j^R(1-\eta_j^2) + \eta_j^2$. 3) The action-target couple (a_j^R, t_j^R) only occurs in the current state s_k , and then we know the probability of (a_j^R, t_j^R) in current state s_k is 1 and probability of (a_j^R, t_j^R) in current action is 0. So, according to the definition of inertia, the fused probability of the action-target couple (a_j^R, t_j^R) is $0(1-\eta_j^2) + 1(\eta_j^2) = \eta_j^2$. 4) The action-target couple (a_j^R, t_j^R) occurs neither in the current state s_k nor in the current action of j^{th} cluster of player 2, and then we know the probability of (a_j^R, t_j^R) in current state s_k is 0 and probability of (a_j^R, t_j^R) in current action is 0. So, according to the definition of inertia, the fused probability of the action-target couple (a_j^R, t_j^R) is $0(1-\eta_j^2) + 0(\eta_j^2) = 0$.

Similarly, the new probability distribution for the j^{th} team of player 1 (blue force) is

$$\bar{p}((a_j^B, t_j^B) | s_k) = \begin{cases} p_j^B (1 - \eta_j^1) & , (a_j^B, t_j^B, p_j^B) \in d_j^B \text{ and } (a_j^B, t_j^B) \notin \{\Psi_j^1(s_k)\} \\ p_j^B (1 - \eta_j^1) + \eta_j^1 & , (a_j^B, t_j^B, p_j^B) \in d_j^B \text{ and } (a_j^B, t_j^B) \in \{\Psi_j^1(s_k)\} \\ \eta_j^1 & , (a_j^B, t_j^B, p_j^B) \notin d_j^B \text{ and } (a_j^B, t_j^B) \in \{\Psi_j^1(s_k)\} \\ 0 & , (a_j^B, t_j^B, p_j^B) \notin d_j^B \text{ and } (a_j^B, t_j^B) \notin \{\Psi_j^1(s_k)\} \end{cases} \quad (5)$$

The new probability distribution for j^{th} team of player 3 (white force) is

$$\bar{p}((a_j^W, t_j^W) | s_k) = \begin{cases} p_j^W (1 - \eta_j^3) & , (a_j^W, t_j^W, p_j^W) \in d_j^W \text{ and } (a_j^W, t_j^W) \notin \{\Psi_j^3(s_k)\} \\ p_j^W (1 - \eta_j^3) + \eta_j^3 & , (a_j^W, t_j^W, p_j^W) \in d_j^W \text{ and } (a_j^W, t_j^W) \in \{\Psi_j^3(s_k)\} \\ \eta_j^3 & , (a_j^W, t_j^W, p_j^W) \notin d_j^W \text{ and } (a_j^W, t_j^W) \in \{\Psi_j^3(s_k)\} \\ 0 & , (a_j^W, t_j^W, p_j^W) \notin d_j^W \text{ and } (a_j^W, t_j^W) \notin \{\Psi_j^3(s_k)\} \end{cases} \quad (6)$$

- Step 2: we determine the probability distribution over the all possible outcomes of state s_{k+1} ,

$$q(s_{k+1} | s_k, a_k^1, a_k^2, a_k^3) = \prod_{i=1}^{m_1} \bar{p}((a_i^B, t_i^B) | s_k) \prod_{j=1}^{m_2} \bar{p}((a_j^R, t_j^R) | s_k) \prod_{j=1}^{m_3} \bar{p}((a_j^W, t_j^W) | s_k) \quad (7)$$

when $s_{k+1} = \bigcup_{i=1}^{m_1} \{(r_i^B, a_i^B, t_i^B)\} \bigcup_{j=1}^{m_2} \{(r_j^R, a_j^R, t_j^R)\} \bigcup_{j=1}^{m_3} \{(r_j^W, a_j^W, t_j^W)\}$. Otherwise,

$$q(s_{k+1} | s_k, a_k^1, a_k^2, a_k^3) = 0.$$

where m_1 is the number of the teams or clusters of player 1 (blue force), m_2 is the number of the teams or groups of player 2 (red force) and m_3 is the number of the units of player 3 (white force). $\{(r_i^B, a_i^B, t_i^B)\}$ is the set of the all possible (with positive probability) triplets for the i^{th} team of player 1 (blue). Therefore $\bigcup_{i=1}^{m_1} \{(r_i^B, a_i^B, t_i^B)\}$ contains all the possible (with positive probability) triplets for the blue force. From the step 1, we know that the fused probability of each specified (a_j^B, t_j^B) is $\bar{p}((a_j^B, t_j^B) | s_k)$ defined in equation (5). With the assumption that all teams of blue force are independent, we obtain the overall probability of blue force, $\prod_{i=1}^{m_1} \bar{p}((a_i^B, t_i^B) | s_k)$. Similarly, $\prod_{j=1}^{m_2} \bar{p}((a_j^R, t_j^R) | s_k)$ and $\prod_{j=1}^{m_3} \bar{p}((a_j^W, t_j^W) | s_k)$ are the overall probabilities of the enemy and white force, respectively. So the probability distribution over the all possible outcomes of state s_{k+1} (composed of all possible sub-states of blue force and red force) can be calculated via equation (7).

Payoff Functions --- In our proposed decentralized Markov game model, there are two levels of payoff function for each player (enemy, friendly force, or white force).

- The lower level payoff functions are used by each team, cluster or unit to determine the team actions based on the local information. For the j^{th} team of blue force, the payoff function is defined as $f_j^B(\tilde{s}_j^B, d_j^B, W_k^B)$, where $\tilde{s}_j^B \subseteq s$ is the local information obtained by the team, and W_k^B , the weights for all possible action-target couples of blue force, is announced to all blue teams and determined according the top level payoff functions by supervisor of Blue force.

$$f_j^B(\tilde{s}_j^B, d_j^B, W_k^B) = \sum_{(a_i^B, t_i^B, p_i^B) \in d_j^B} w^B(j, a_i^B, t_i^B, W_k^B) p_i^B g^B(j, a_i^B, t_i^B, \tilde{s}_j^B) \quad (8)$$

where, $w^B(j, a_i^B, t_i^B, W_k^B)$ will calculate the weigh for any specified action-target couple for j^{th} team of blue force from the W_k^B , p_i^B is the probability of the action-target couple (a_i^B, t_i^B) , and $g^B(j, a_i^B, t_i^B, \tilde{s}_j^B)$ will determine the gain from the action-target couple (a_i^B, t_i^B) for j^{th} team of blue force according to the positions and features, such as platform values and defense/offense capability, of blue and aimed platforms. Similarly, we obtain the lower level payoff functions for the j^{th} team of red and white force,

$$f_j^R(\tilde{s}_j^R, d_j^R, W_k^R) = \sum_{(a_i^R, t_i^R, p_i^R) \in d_j^R} w^R(j, a_i^R, t_i^R, W_k^R) p_i^R g^R(j, a_i^R, t_i^R, \tilde{s}_j^R) \quad (9)$$

$$f_j^W(\tilde{s}_j^W, d_j^W, W_k^W) = \sum_{(a_i^W, t_i^W, p_i^W) \in d_j^W} w^W(j, a_i^W, t_i^W, W_k^W) p_i^W g^W(j, a_i^W, t_i^W, \tilde{s}_j^W) \quad (10)$$

Remark 1: For some asymmetric threats, such as suicide bombers, the payoff functions may only consider the loss of the blue side. For some camouflage, and concealment entities, their objectives are to hide themselves and move close to the blue units. Other deception units will do some irrational movements to hide their true goals with the cost the time.

Remark 2: The white units only care about their possible losses. For an example, when a dangerous spot is detected, normal white units will find a COA to keep themselves as far as possible from the spot.

- The top level payoff functions are used to evaluate the overall performance of each player.

$$J^B = \sum_k \left[\sum_{j=1}^{m_1} f_j^B(\tilde{s}_j^B, d_j^B, W_k^B) \right] \quad (11)$$

$$J^R = \sum_k \left[\sum_{j=1}^{m_2} f_j^R(\tilde{s}_j^R, d_j^R, W_k^R) \right] \quad (12)$$

$$J^W = \sum_k \left[\sum_{j=1}^{m_3} f_j^W(\tilde{s}_j^W, d_j^W, W_k^W) \right] \quad (13)$$

where k is the time index. In our approach, the lower lever payoffs are calculated distributedly and sent back to commander/supervisor via communication networks.

Remark 3: Since the gain functions $g^B(j, a_i^B, t_i^B, \tilde{s}_j^B)$ for blue force, $g^R(j, a_i^R, t_i^R, \tilde{s}_j^R)$ for red force and $g^W(j, a_i^W, t_i^W, \tilde{s}_j^W)$ for white force are different functions, asymmetric force and cost utilities can be straightforwardly represented in our model. In addition, after an irregular adversary is detected, a different type of gain function will be assigned dynamically.

The strategies --- In this paper, we try several well known types of strategies.

- Min-max strategies [5]. This kind of strategies will give a conservative solution to minimize the possible maximum “loss”. Actually, in our problem, it is a max-min solution in the sense that each player maximizes the possible minimum his payoffs. So, this kind of strategies is also called safest solutions.
- Pure Nash Strategies with finite horizon. In game theory, the Nash equilibrium (named after John Nash [6] who proposed it) is a kind of optimal collective strategy in a game involving two or more players, where no player has anything to gain by changing only his or her own strategy. If each player has chosen a strategy and no player can benefit by changing his or her strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash equilibrium. In our approach, we use a game search tree (shown in Fig. 2) to find the solution. In our proposed approach, the solution to the Markov game tree is obtained via a K time-step look-ahead approach, in which we only optimize the solution in the K time-step horizon. The suboptimal technique is used successfully for reasoning in games such as chess, backgammon and monopoly.

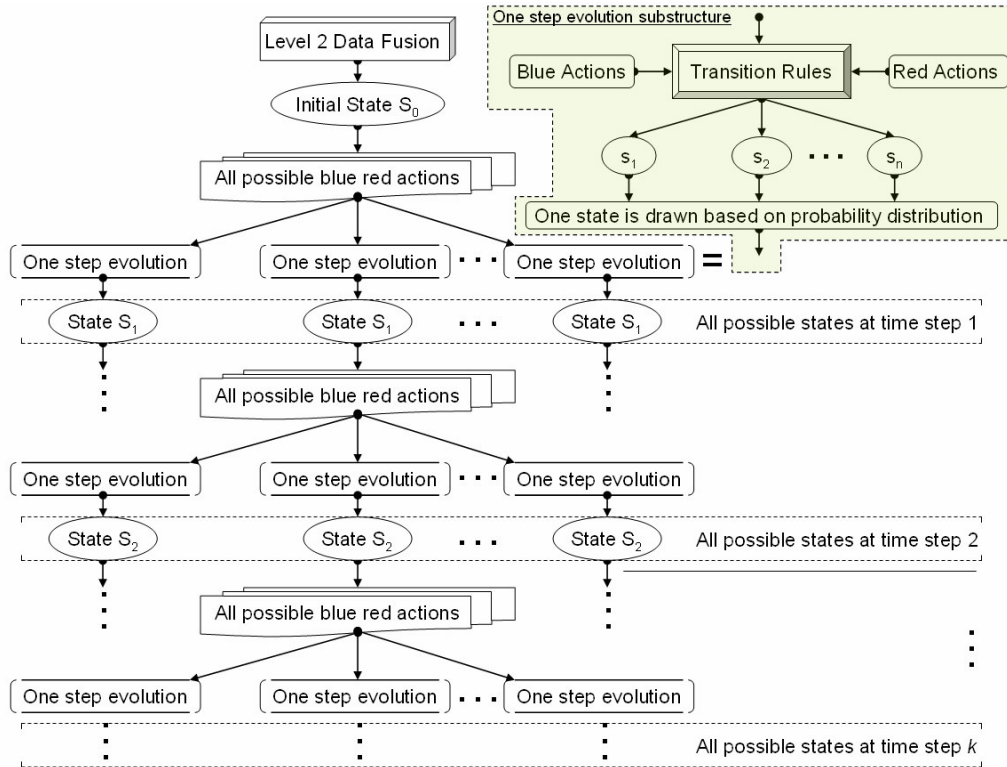


Fig. 2 A game tree search approach to find pure Nash strategies with finite planning horizon (moving planning window)

- Mixed Nash Strategies. A mixed strategy is used in game theory to describe a strategy comprised of possible actions and an associated probability, which corresponds to how frequently the action is chosen. Mixed strategy Nash

equilibria are equilibria where at least one player is playing a mixed strategy. It was proved by Nash that that every finite game has Nash equilibria but not all has a pure strategy Nash equilibrium.

- Correlated Equilibria [7]. Unlike Nash equilibria, which are the concept of equilibria formulated in independent strategies, the correlated equilibria were developed from the correlated strategies in noncooperative games. The correlated equilibrium of a Markov game describes a solution for playing a dynamic game in which players are able to communicate but are self-interested. Based on the signals generated by the correlated devices and announce to the each decision maker, players choose their actions according to the received private signals. There are two types of correlation devices: autonomous and stationary devices. An autonomous correlation device is a pair $\mathcal{D} = (((M_n^i)_{i \in N}, d_n)_{n \in N})$, where (i) M_n^i is a finite set of signals for player i at time step n , and (ii) $d_n : M(n) \rightarrow \Delta(M_n)$, $M_n = \times_{i \in N} M_n^i$ and $M(n) = M_1 \times M_2 \times \dots \times M_{n-1}$. A stationary correlation device is a pair $\mathcal{D} = (((M^i)_{i \in N}, d))$, where $d \in \Delta(M)$ and $M = \times_{i \in N} M^i$. Actually, a stationary correlation device is a special case of an autonomous correlation device, where M_n^i is independent of n and d_n is a constant function that is independent of n .

Given a correlation device \mathcal{D} , we define an extended game $G(\mathcal{D})$. The game $G(\mathcal{D})$ is played exactly as the original game, but at the beginning of each stage n , a signal combination $m_n = (m_n^i)_{i \in N}$ is drawn according to the probability function $d_n(m_1, m_2, \dots, m_{n-1})$ and each player i is informed of m_n^i . Then each decision maker must base his choice of actions on the received signal. Any deviator will be punished via his min-max value. The punishment only occurs if a player disobeys the recommendation of the device. It was proved in [7] that every Markov game with an autonomous correlated device admits a correlated equilibrium.

- Sequential Nash Strategies [8]. A sequential game is one in which players choose their strategies following a certain predefined order, and in which at least some players can observe the moves of other players who make decisions preceded them. The sequential game is a natural framework to address some real problems, such as the Action-Reaction-Counteraction paradigm used in military intelligence and advertising campaigns strategies of several competing firms in economics. In our approach, we use a turn-by-turn scheme shown in Fig. 3. At each step k , the control strategy from only one player, say, player 1 (blue force), is applied and the corresponding outcome will be observed by player 2

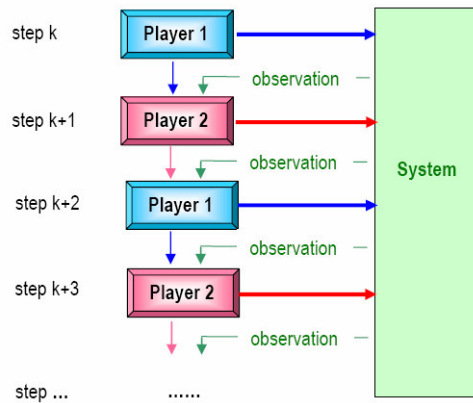


Fig. 3 Two-player Sequential Game

(enemy force) before it decides its next action. This is helpful in estimating the opponent's intent because each time only one action is applied and the state changes from the action is observed.

- Leader-Follower Strategies [9]. With the consideration of the limited and non-perfect communication, we use the Stackelberg conception to model the cooperation part between the commander and the local teams. The commander (called the leader) declares incentives to the local teams (called followers) in order to induce them to accept his desired system behavior as the common desired behavior. The Leader-Follower strategy is useful in our clearly defined hierarchical systems which have asymmetric information structures (in our case the leader, or commander know the cost functional of every decision maker, or local teams while the followers know only their own).

To efficiently reason the enemy's intent or COAs, we divide our approach into two phases: training phase and reasoning phase. In the first one, we measure or observe the enemy's actions and compare them with the actions generated by our model. The results are used to tune or adjust the transition rules. In the reasoning phase, we fix the transition rules and use the generated red actions as the intent of enemy.

Deceptions [10] are used to make the other player act in our own advantage by making it believe that the game is in a state other than the actual one. It is only possible in partial information games. We propose an equilibrium approach to deception where deception is defined to be the process by which actions are chosen to induce erroneous inferences so as to take advantage of them. This framework differs from the earlier literature on multi-stage games with incomplete information in that whether the player has a perfect understanding of the strategy employed by his opponent. We introduced two types of deceptions: cognitive type and strategy environment.

Cognitive types are defined as follows. Each player i forms an expectation about the behavior of the other player by pooling together several nodes in which the other player must move. Each such *pool* of nodes is referred to as a *class of analogy*. Players are also differentiated according to whether or not they distinguish between the behaviors of the various types of their opponent. Formally, a cognitive type of player i is characterized by (An_i, d_i) where An_i stands for player i 's analogy partition and d_i is a dummy variable that specifies whether or not type t_i distinguishes between the behaviors of the various types t_j of player j . We let $d_i = 1$ when type t_i distinguishes between types t_j 's behaviors and $d_i = 0$ otherwise.

A strategic environment is described by (Y, u_i, p) where p denotes the prior joint distribution on the type space $Q = Q_1 \times Q_2$. To simplify notation we will assume that the types of the two players are independently distributed from each other, and we will refer to $p_i = (pt_i)_{ii}$ as the prior probability of player is type where pt_i denotes the prior probability of type t_i .

3. Simulations and Experiments

In the Simulation part, we build a virtual battle-space and a typical urban scenario based on the Ontology concept, which is an explicit, formal, machine-readable semantic model that defines the classes (or concepts) and their possible inter-relations specific to some specified domain. To simulate our data fusion approach, we implemented and tested our battle-space, scenario and algorithms on the MICA Open Environment Platform (OEP) based on the Boeing C4Isim simulation, which models the collection, processing, and dissemination of battlefield information.

We used a scenario shown in Fig. 4 to demonstrate the performance of our proposed threat prediction and situation awareness algorithm. In the shown urban environment, the blue force's missions are to capture and secure two bridges which are guarded by the red force. The two bridge locations are well connected via wide roads highlighted by dashed lines. The red force includes armed vehicles, fighters and asymmetric forces hiding in and acting like the white objects (the civilians and vehicles). The blue force consists of a few fighters with close air support provided by several unmanned aerial vehicles (UAVs) such as small sensor UAVs and small weapon UAVs, which will, if needed, do some searching and fighting tasks too. We assume the total offense force and total defense force are almost at the same level. There is no dominant one. There are several choices for the red force to guard these objectives efficiently. They can deploy all red units to protect one location. However, the blue force can capture other places first. The blue force faces the same dilemma. So the main challenge for both sides is to understand the situation from the fused sensor data and predict the intent of the opponent under the "believed" war situation.

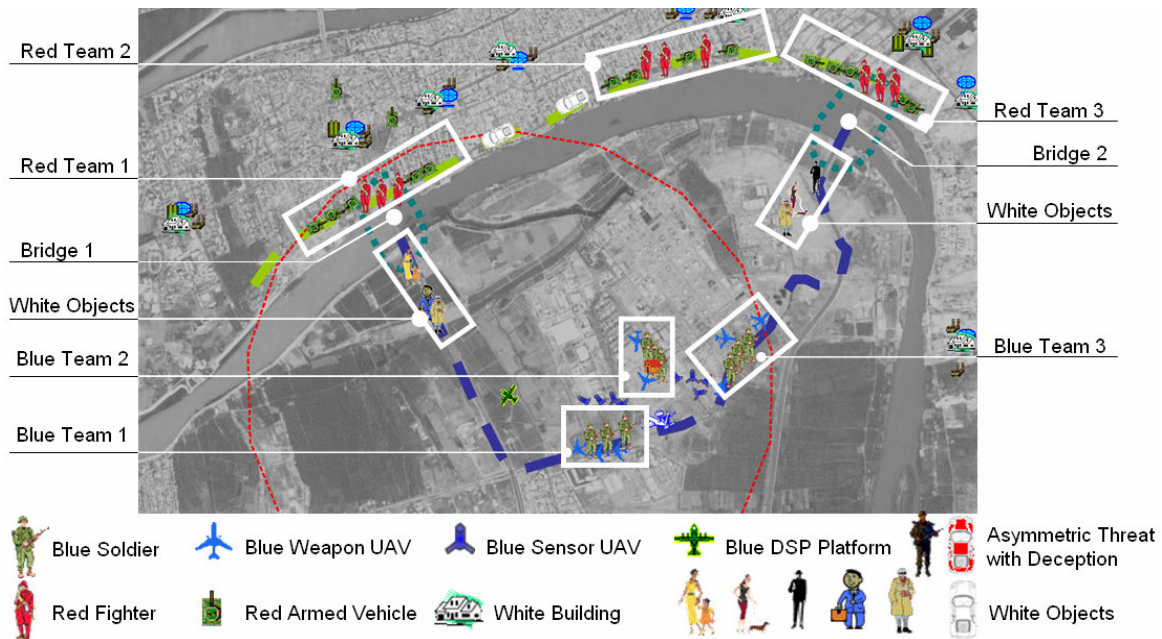


Fig. 4: A Simulated Scenario

For this scenario, in a specific simulation run (correlated equilibrium method) as shown in Fig. 5, blue team 1 and blue team 3 were assigned to capture Bridge 1 and Bridge 2, respectively, almost in the whole simulation period of 30 minutes. On the other hand, Red team 1 and Red Team 3 were guarding Bridge 1 and Bridge 2 almost all the time. Blue Team 2 was strategically moving based on the threat prediction from the Markov game. At the same time, the Red Force was dynamically deploying Red Team 2 and trying to keep a balance between the Blue force and the Red force at each bridge. As shown in the movie of the demo, we can see that after about 15 minutes of movement, at Bridge 1, the Blue force achieves domination with 2 Blue teams (Blue Team 1 and Blue Team 3) to attack 1 red team (Red Team 1). The Blue force captures and secures Bridge 1 before Red Team 3 reaches Bridge 1 to help Red Team 1. During this Phase 1 battle, our algorithm detected two asymmetric adversaries with deception (a person and a vehicle) which were hidden in a vast of background harmless civilian activity. After the capture of Bridge 1, the Blue side deployed a part of the remaining force of Team 1 and Team 3 to secure the bridge and others to help Team 2 to capture Bridge 2. Finally, the Blue side won the urban battle at a cost of 5 Blue soldiers and 6 small weapon UAVs. Another hidden asymmetric threat with deception (terrorist) was detected and killed in the Phase II battle of capturing Bridge 2.

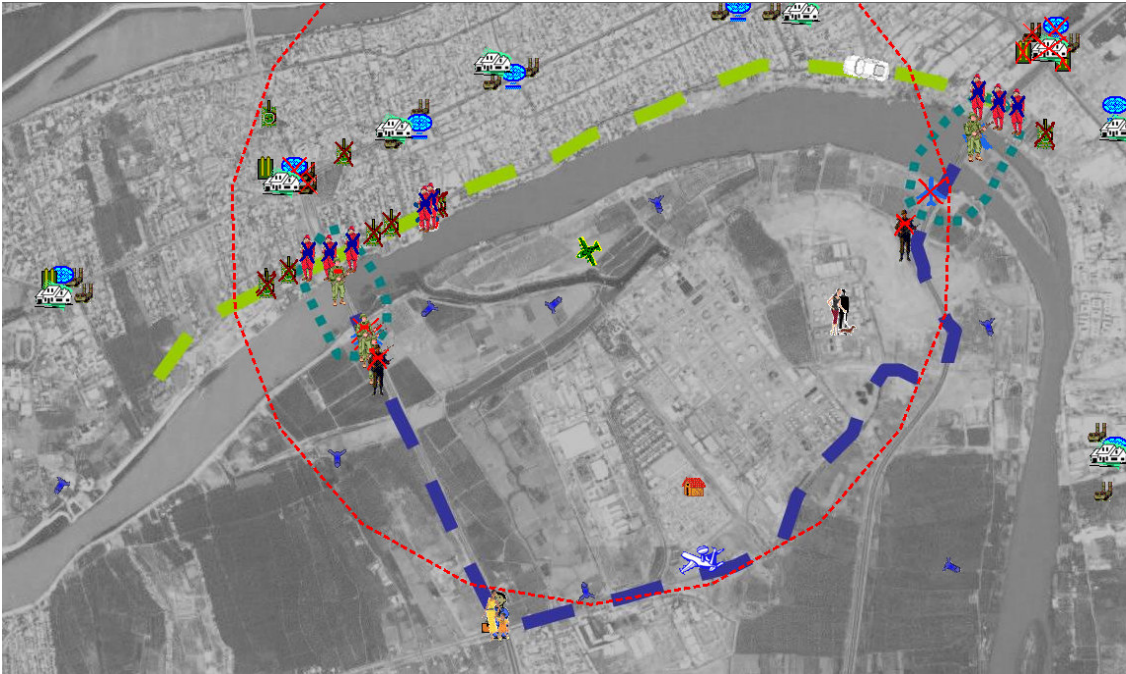


Fig. 5: Result of a simulation run

In addition to the explained run, we performed many experiments. We compared the results using the various options, such as without game theoretic fusion (without levels 2 and 3), without asymmetric Threat Prediction (with level 2 but the payoff function of game model at level 3 doesn't change dynamically), Game approach with mixed Nash Strategy, and game approach with Correlated Equilibria. The results (since the simulation is stochastic, results are the mean of 10 runs for each case) are shown in Fig. 6 (Only the damage information of the Blue side is shown). From the damage comparison results, we can see that our proposed Markov game framework with correlated equilibrium and deception consideration for threat detection and situation awareness is better than the other methods.

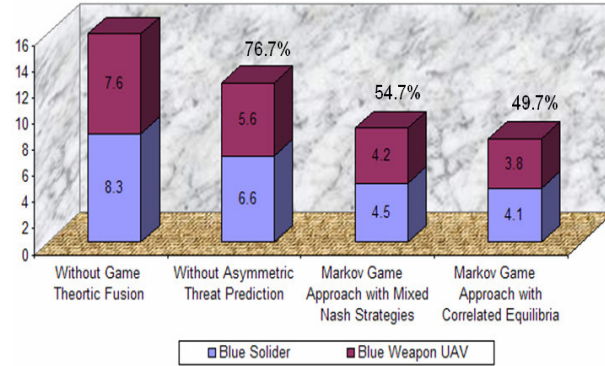


Fig. 6: Damage comparison of various options

4. Conclusions

Game theoretic tools have a potential for threat prediction that takes real uncertainties in Red plans and deception possibilities into consideration. In this paper, we have evaluated the feasibility of the advanced adversary intent inference algorithm and their effectiveness through extensive simulations. It has verified that our proposed algorithms are scalable, stable, and perform satisfactorily.

Acknowledgements

This research was supported by the US Navy under contract number N68335-06-C-0105. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Navy.

References

- [1] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944.
- [2] E. Solan and N. Vieille, "Correlated Equilibrium in Stochastic Games", *Games and Economic Behavior*, vol. 38, no. 2, pp. 362-399, Feb. 2002.
- [3] North Atlantic Treaty Organization, "The Land C2 Information Exchange Data Model", *ATCCIS Baseline 2.0*, March 2002
- [4] User Guide for the Open Experimental Platform (OEP), version 1.3, Boeing, Mar, 2003
- [5] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, SIAM Series in Classics in Applied Mathematics. Philadelphia, second edition, January, 1999.
- [6] John Nash, "Noncooperative games", *Annals of Mathematics*, vol. 54, pp. 286-295, 1951.
- [7] E. Solan and N. Vieille, "Correlated Equilibrium in Stochastic Games", *Games and Economic Behavior*, vol. 38, no. 2, pp. 362-399, Feb. 2002
- [8] D. M. Kreps and R. Wilson, "Sequential equilibria", *Econometrica*, vol. 50, no. 4, pp. 863-894, July 1982.
- [9] M. Simaan and J. B. Cruz, Jr., "On the Stackelberg Strategy in Nonzero-Sum Games," *Journal of Optimization Theory and Applications*, Vol. 11, No. 5, May 1973, pp. 533-555.
- [10] J. P. Hespanhaz, Y. S. Ateskan, and H. H. Kizilocak, "Deception in Non-Cooperative Games with Partial Information", In *Proc. of the 2nd DARPA-JFACC Symposium on Advances in Enterprise Control*, July 2000.
- [11] Cruz, J. B., Jr., Simaan, M. A., Gacic, A., Jiang, H., Letellier, B., Li, M., and Liu, Y., "Game-Theoretic Modeling and Control of a Military Air Operation", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 4, October, 2001.
- [12] Cruz, J. B. Jr., Simmaan M. A., Gacic, A., Liu, Y., "Moving horizon Nash strategies for a military air operation", *IEEE Transactions on Aerospace and Electronic Systems*, 38(3), 2002.
- [13] Cruz, J. B., Jr., Chen, G., Garagic D., Tan, X., Li, D., Shen, D., Wei, M., Wang, X., "Team dynamics and tactics for mission planning," *Proceedings, IEEE Conference on Decision and Control*, pp. 3579-3584, December 2003.
- [14] D. Shen, J. B. Cruz, G. Chen, and C. Kwan, S. P. Riddle, S. Cox, and C. Matthews, "A Game Theoretic Approach to Mission Planning For Multiple Aerial Platforms", *AIAA infotech@aerospace*, Sept. 26-29, 2005, Arlington, VA.
- [15] G. Chen, D. Shen, J. B. Cruz, C. Kwan, S. Cox, S. P. Riddle, and C. Matthews, A novel cooperative path planning for multiple aerospace vehicles, *Proceedings of AIAA infotech@aerospace*, Sept. 26-29, 2005, Arlington, VA.