Bayesian Games for Threat Prediction and Situation Analysis

Joel Brynielsson and Stefan Arnborg
Department of Numerical Analysis and Computer Science
Royal Institute of Technology
SE-100 44 Stockholm
Sweden
{joel, stefan}@nada.kth.se

Abstract – *Higher levels of the JDL model call for prediction of future development and awareness of the development of a situation. Since the situations handled by Command and Control systems develop by actions performed by opposing agents, pure probabilistic or evidential techniques are not quite sufficient tools for prediction. Game theoretic tools can give an improved appreciation of the real uncertainty in this prediction task, and also be a tool in the planning process. We review recent developments in game theory and apply them in a decision support tool for Command and Control situation awareness enhancements.*

Keywords: Bayesian Network, Influence Diagram, Game Theory, Bayesian Game, Command and Control, Situation Awareness

1 Introduction

The military domain is one of the purest possible game arenas, and history is full of examples of how mistakes in handling uncertainty about the opponent has had large consequences. For an entertaining selection, see, e.g., [1]. Commanders on each side have resources at their disposal, and want to use them to achieve their, mostly opposing, goals. In the NCW era, they are aided by vast amounts of information about the opponent from sensors and historic data bases, and the status of their own resources from their own IT infrastructure. In recently proposed infostructures for Command and Control (C2)[2], decision support tools play a prominent role. These tools seldom include game-theoretic tools. Gaming is a prominent feature of military training, however, and the regulated decision processes often assign the roles of red and blue players to staff officers in manual planning activities[3]. Gaming is thus a conceptual part of the planning process in many organizations. It must be emphasized, however, that there are significant differences between practice and theory in application of such regulations. It has been shown in studies that the Swedish defense organization practices a more naturalistic decision-making process than the recommended one[4]. A pure naturalistic planning process relies more on unobservable mental capabilities of decision-makers than rational analyses of alternative moves and their utilities[5]. The most common way to deal with uncertainty is however to make an assumption – and to forget that it was made.

These observations have been the starting point for introducing a less complex planning model – PUT (Planning Under Time-pressure). PUT is based on analyses of a few opponent alternatives and incremental improvement of ones own plans. It thus has potential for use of gaming tools, provided they are realized in a way that supports subjective improvement of decision situations and decision quality.

Information fusion aims at providing situation awareness at different levels for a commander. The well accepted JDL model[6, 7] proposes four fusion levels where the third level consists of higher level prediction of possible future problems and possibilities. We believe that the problem of predicting the future in a C2 context come in two variations that differ in complexity and dependencies. First, *influence diagrams* is an appropriate modeling technique when it comes to modeling everything that is *not* dependent on our own actions, meaning for example doctrine, terrain, etc. Efforts in this direction have been proposed, see for example [8] that discusses doctrine modeling using dynamic Bayesian networks[9]. Second, the problem of predicting decisions is also a game-theoretic problem. We outline a schematic model using influence diagrams to obtain parameters for a description of the situation in the form of a Bayesian game. This game is then solved with a tree look-ahead method. The result from the game is a description of equilibrium strategies for participants that can be incorporated in the influence diagram to form a Bayesian network (BN) description of the situation and its development.

We review some military applications of gaming and simulation in section 2 and describe our use of influence diagrams in section 3. Section 4 introduces the proposed game component and section 5 gives a short background on game theory. Section 6 contains an outline for the game component representation. Section 7 discusses solutions and addresses the problem of obtaining these solutions in a computationally feasible manner. In section 8 we illustrate the use of Bayesian game-theoretic reasoning for operations planning by transforming a decision situation into a Bayesian game that we solve. Section 9 discusses related work and section 10 is devoted to conclusions and discussion regarding future research.

2 Gaming in Command and Control

Among tools proposed to support the gaming perspective can be mentioned microworlds[10, 11], which are computer tools where several operators train together, and computer-intensive sensitivity analyses of simple models[12]. There are also large numbers of full and small scale simulation systems used to asses effectiveness of new types of equipment and ways to use them, with [13] being an interesting prototype system being built especially for the information fusion context. These microworld and simulation systems are used for off-line analyses to define recommended strategies in conceivably relevant situations. Recently, it has become possible to build Bayesian networks to identify the opponent's course of action (COA) from information fusion data using the plan recognition paradigm, which was extended from a single agent context to that of a composite opponent consisting of a hierarchy of partly autonomous units[8]. The conditions for this recognition to work is that the goals and rules of engagement of the opponent are known, and that he has a limited set of COAs to choose from given by the doctrines and rules he adheres to. The opponents COA can then be deduced reasonably reliably from fused sensor information, such as movements of the participating vehicles. The gaming component has thus been compiled out of the plan recognition problem. When the goals and resources are not known, these can be modeled as stochastic variables in a BN. However, this is not a strictly correct approach, since the opponents choice of COA should depend, in an intertwined gaming sense, on what he thinks about our resources, rules of engagement and goals. The situation is thus a classical Bayesian game, and should be resolved using game algorithms.

3 Influence Diagrams for the Representation of Situation Awareness

We address the problem of providing decision support for higher level C2. The areas of information fusion and decision support are closely connected. Solution concepts for future C2 decision-making should therefore be considered to be a joint effort with solutions emerging from both of these areas.

One goal of artificial intelligence (AI)[14] has been to create expert systems, i.e., systems that can, provided the appropriate domain knowledge, match the performance of human experts. Such systems do not yet exist, other than in highly specific domains, but AI research has meant that researchers from widely differing fields have come together in order to solve questions regarding knowledge representation, decision-making, autonomous planning, etc. These results provide a good ground for the construction of C2 decision support systems. During the last decade, the intelligent agent perspective has lead to a view of AI as a system of agents embedded in real environments with continuous sensory inputs. We believe that this is a viable way to reason about C2 decision-making and we adopt the agent perspective throughout this paper.

Agents make decisions based on modeling principles for uncertainty and usefulness in order to achieve the best expected outcome. The assumption that an agent always tries to do its best, is captured in the concept of *rationality*. The combination of probability theory, utility theory and rationality constitutes the basis for *decision theory*.

The basic elements that we use for reasoning about uncertainty are random variables. General joint distributions of more than a handful such variables are impossible to handle efficiently, and the way to model distributions as Bayesian networks has become a key tool in many modeling tasks.

A BN offers an alternative representation of a probability distribution with a directed acyclic graph where nodes correspond to the random variables and edges correspond to the causal relationships between the variables. Calculating the probability of a certain assignment in the full joint probability distribution using a BN means calculating products of probabilities of single variables and conditional probabilities of variables conditioned only on their parents in the graph. The BN representation is often exponentionally smaller[14] than the full joint probability distribution table and many inference systems use BNs to represent probabilistic information. Another advantage with the BN representation is that it facilitates the definition of relevant distributions from causal links that are intuitively understandable and, in the case of a dynamic BN, develop with time. Successful(?) uses of these networks include the implementation of the "intelligent paper clip" we struggle with in our everyday use of the Office Assistant in Microsoft Office[15].

An influence diagram is a natural extension to a BN incorporating decision and utility nodes in addition to chance nodes, and represent decision problems for a single agent. Decision nodes represent points where the decision-maker has to choose a particular action. Utility nodes represent terminal nodes where the usefulness for the decision-maker is calculated. These diagrams can be evaluated bottom up by dynamic programming to obtain a sequence of maximum utility decisions.

When designing decision-theoretic systems to be used for C2 decision-making, complex situations arise where one wants to represent knowledge, causality, and uncertainty at the same time as one wants to reason about the situation simulating different COAs in order to see the expected usefulness of the proposed move. We believe the influence diagram is the right choice for both representation and evaluation and propose a simplified schematic generic diagram in Fig. 1 for the C2 process. C is a discrete random variable representing the consequence of the decisions $D_1, \ldots, D_n$. D_1 represents our own decision and $D_2, \ldots, D_n$ represent the decisions of the other agents. G_1 is a discrete random variable that represents our own goals. U_1 is the utility that we gain after performing decision D_1 depending on the consequence C and our own goals G_1. G_i and U_i are defined similarly for the other agents where $2 \leq i \leq n$.

The diagram in Fig. 1 is a simplified representation, to be connected to models – encoded as BNs – of terrain, doctrine, etc., that can be implemented as sub-diagrams with causal relationships between different nodes of models. What is important is that the depicted influence diagram, or an extension of this diagram, is indeed the model that should be used.

A problem with the diagram in Fig. 1 is that it does not capture "gaming situations" where one wants to reason about opposing agents that act according to beliefs about ones' own actions. This is not possible to model in an in-

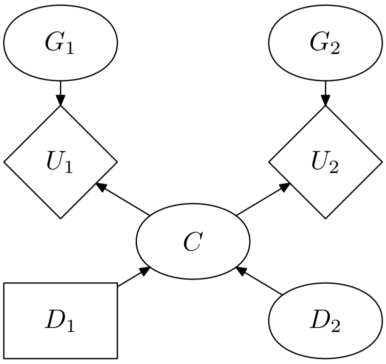

Fig. 1: The C2 process modeled as an influence diagram.

fluence diagram or BN without additional machinery.

Unfortunately, in higher level C2 we can be certain that large efforts are directed towards predicting the beliefs, desires, and intentions of the adversary – and there will not be a common agreed upon model of the situation and its utilities. This type of uncertainty can be modeled only by representations of Bayesian games or at least imperfect information games.

4 The Game Component

The decision situation that arises in decision node D_1 in the influence diagram depicted in Fig. 1 is characterized by its dependency on other actors' beliefs, desires and intentions. Standard AI tools for solving decision-making problems in complex situations, such as dynamic decision networks and influence diagrams, are not applicable for these kinds of situations, as the decisions are intimately related to the beliefs, desires and intentions of the other agents. 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.

In higher level C2, i.e., threat prediction in a data fusion context, the need of a game component becomes obvious[16]. Circular relationships are not allowed in influence diagrams (or other traditional agent modeling techniques) and therefore we cannot make the agents' decisions dependent on each other in the diagram in Fig. 1. On lower level C2 this need is not as obvious, because agents choices are to a large extent driven by standard operating procedures obtained by training and developed using off-line game analyses. On this level, like in helicopter dogfights, successful developments of strategies have been obtained with look-ahead in extensive form, i.e., perfect information game trees with zero-sum payoffs as reported in [17].

Decision-making in environments where multiple agents make decisions based on what they think the other agents might do is a difficult problem. Game theory provides a mathematical framework for determining rational behavior for agents when they interact in multi agent environments. The use of game theory for agent design has so far been limited due to lack of understanding and lack of standard implementation methods. We believe, however, that these barriers will be overcome as more research is focused on the use of game theory for agent design. The widely used AI book by Russell and Norvig[14] added a section on game theory just recently which indicates that the ideas are new and still need to be investigated more thoroughly.

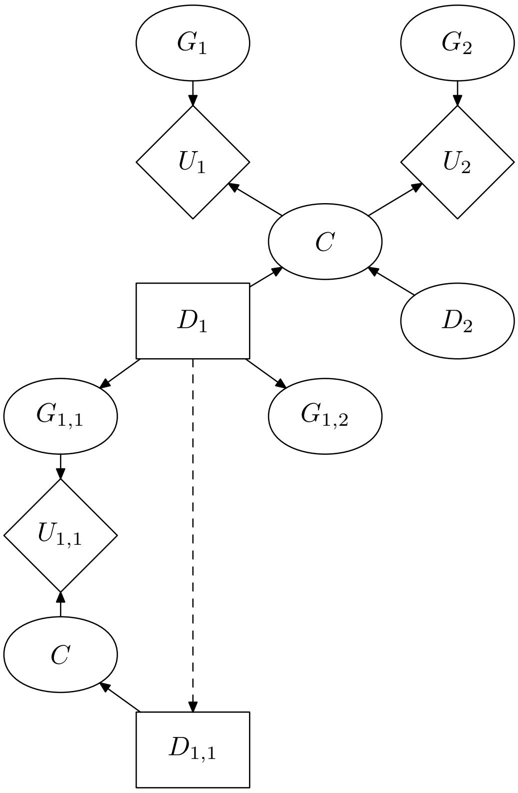

Fig. 2: Extended influence diagram for the C2 process. Gaming occurs at decision nodes in a top-down manner.

One of the earlier mentioned barriers that do exist when using traditional game theory for agent design is that it assumes that a player will definitely play a (Nash) equilibrium strategy. In some applications, such as the management of (own) mobile sensors[18] or the construction of algorithms for efficient network capacity sharing[19], where the game is a designed mechanism, this is for sure true. However, these situations must be considered being a small subset when looking at all the uncertain situations that occur in our everyday life where uncertainty regarding both the other actors and the world as a whole must be accounted for. In this work we aim at solving this problem using the Bayesian game technique, which is described below.

Other problems with game theory for agent design are the lack of methods for combining game theory with traditional agent control strategies[14] and the lack of standard computational techniques for game-theoretic reasoning[20].

In this work we propose the use of a Bayesian game for modeling higher-level agent interaction in an attempt to obtain better situation awareness in a C2 system. As situation awareness is obtained using data fusion techniques we believe that the game component is an integral part in the data fusion and provides information that is needed in the part of the data fusion that is to be found at level three according to the JDL model[6, 7]. A Bayesian game is a game with incomplete information, that is, at the starting point of the game the players may have private information about the game that the others do not know of. Also, each player expresses its prior belief about the other players as a probability distribution over what private information the other players might possess.

5 Structure of Games and their Representations

Recent developments in game theory and AI have made applications with significant game components feasible. Most of the work does however not address Bayesian games. Many description methods have been developed with algorithmic techniques being able to solve quite large games if they are of the right type. The *extensive form* of a game is a tree structure, where a non-terminal node can describe a chance move by nature (random draw) or a move possible for one of the participants, and a leaf node represents the end of the game and its payoff after evolving through the path to it. The immediate descendants of a non-leaf represent the alternative outcomes of a chance move (in which case the node is associated with a probability distribution) or the set of actions available for the player in turn at this point. This is adequate for leisure games like chess, a *perfect information game*, but the chess game tree does not fit into any computer. A deterministic game with full information (like generalized chess or checkers) can be solved if its game tree can be traversed, by bottom-up dynamic programming.

In games with *imperfect information*, the exact position in the game tree may not be known to players. This is the case in leisure games of cards, where the hand of a player is only available to her. The determination of optimal strategies must use a game tree where the decision is the same for a whole *information set*, a set of nodes for a player where the information available to her is the same. As an example, at the first bid of a game of contract bridge, each of the possible distributions of the cards not seen by the player is in the same information set. The bottom-up evaluation does not work, because at the lower levels of the game tree the players have information on the hidden information that was communicated by their opponents' choices of moves (like the initial round of bidding in bridge). This situation is solved by putting the game on *strategic form*, which means that each combination of moves for all of a players information-equivalent nodes in the tree, and all chance moves, are listed with their payoffs. Solutions can be found with numerical methods, linear programming for zero-sum games and the linear complementarity problem (LCP) for general games[21]. For the former, a unique mixed (randomized) strategy for each player is a non-controversial definition of the games solution. For the latter, the Nash equilibrium is the accepted solution concept[22]. A Nash equilibrium does always (under general assumptions) exist but is less non-controversial since sometimes several equilibria exist, and there are alternative proposals how to find one that is in a tangible way more relevant than the others. The payoff matrix is typically impossibly large, and games of this type, like standard variants of poker and bridge, have no known optimal solution. In the above games, all players know the exact structure and payoff system of the game. This is adequate for many purposes, but not for our application.

The concept of a Bayesian game is fairly complex. Many different views, and even a few misperceptions, abound in the literature. A Bayesian game, Γ^{b}, is defined by

$$\Gamma^{\mathrm{b}} = (N, (C_i)_{i\in N}, (T_i)_{i\in N}, (p_i)_{i\in N}, (u_i)_{i\in N}) \qquad (1)$$

where N is a set of players, C_i is the set of possible actions for player i, T_i is the set of player i:s possible types, p_i is a probability distribution representing what player i believes about the other players' types, and u_i is a utility function mapping each possible combination of actions and types into the payoff for player i.

This is a flat representation given originally in [23]. It seems as if it only states first-order beliefs of players about each other, but this is not a fair perspective. We want to consider all types of higher-order knowledge, such as what player 1 believes that player 2 believes that player 1 … believes. This type of information can indeed be modeled in a standard Bayesian game, under quite general conditions, as shown in a strictly mathematical and non-algorithmic argument in [24]. On the other hand, the amount of information required to perform such modeling can be infinite and thus not extractable from experts and decision-makers. Bayesian games can have infinite type sets even in simple cases like natural analysis of bargaining situations. We will restrict our attention to games with finite type sets and players, since otherwise general solution algorithms do not exist (games with infinite type sets must be analyzed manually to bring a finite solution algorithm). An important class of Bayesian games is games with consistent beliefs. In this case the players (conditional on his type) beliefs about other players types are all derivable from a global distribution over all players types by conditioning, i.e., $p_i(t_{-i}|t_i) = p(t_{-i}|t_i)$. Hence, this class is a subclass of imperfect information games.

6 Agent Interaction as a Bayesian Game

In this section we define the proposed game component using notation from [25]. The objective is to specify a game that in our view is suitable for the C2 domain. Our building blocks therefore rest on the following important criteria: 1) that the agents' decisions are based on their belief regarding the other agents' prior information, 2) that the game uses existing techniques for efficient computation, and 3) that the game is intended for prediction of the future. We achieve the first criterion by the use of a Bayesian game, the second criterion by using ideas and results introduced in a series of recently published articles regarding computational representation and solution of game-theoretic problems[26, 27], and the third criterion by using a game representation that gives us a property that resembles that of a look-ahead game tree.

The game that we propose is depicted in Fig. 3. It is constructed from the root downwards and solved from the leaves upwards. Instead of choosing the maximum or minimum utility in each node, as in minimax algorithms in for example chess programs, the algorithm solves a game in mixed strategies at each level, assuming that the child nodes have been solved and have defined probability distributions that are propagated upwards.

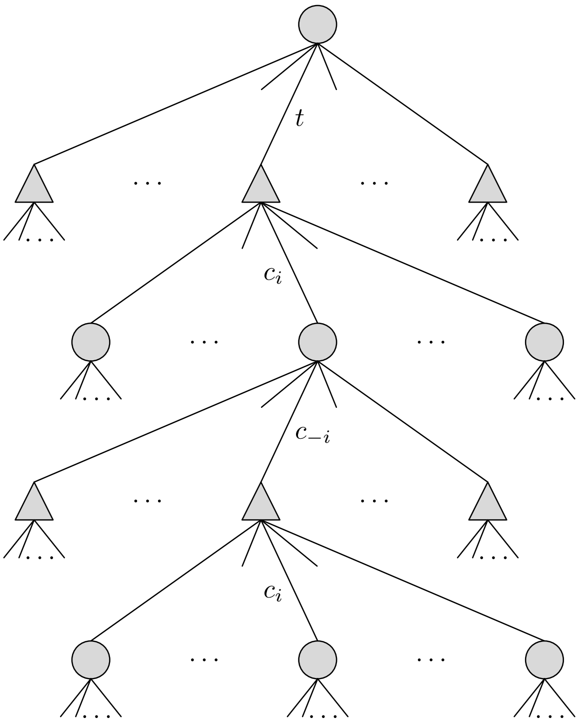

Fig. 3: The proposed game on extensive form.

6.1 Bayesian Game

The root node is a *historical chance node* that is used to implement the Bayesian property of the game. A historical chance node differs from an ordinary chance node in that the outcome of this node has already occurred and is partially known to the players when the game model is formulated and analyzed. For each set of possible types, the edges from the root node in our game correspond to the model that is used if the players were of this type. We say that a player i believes that the other players type profile is $t_{-i} \in T_{-i}$ with subjective probability $p_i(t_{-i}|t_i)$ given that player i is of type $t_i \in T_i$. Here the subscript $-i$ is a standard notation for a sequence of all players except player i, for example, t_{-i} is a list of types for all the other players.

For each type profile $t \in T$, an influence diagram, as in Fig. 2, describes the decision situation using random state variables. The different models differ in properties that cannot be seen in Fig. 2, consisting of other random variables describing for example terrain, doctrine, and belief regarding all kinds of properties that do not rely on other participating agents' decisions. In the context of our Bayesian C2 game the historical chance node is thus a lottery over the possible models that are represented as influence diagrams.

The Bayesian property of the game might seem trivial at first glance, but the historical chance node at the root of the tree poses a serious concern to us. To establish Nash equilibria for the game the normal representation in strategic form is needed, but the algorithm for the creation of this relies on the players being able to decide their strategies before the game begins, which is not true in a Bayesian game that is represented with a historical chance node. The solution, due to Harsanyi[23], is to reduce the game to *Bayesian form* and compute its *Bayesian equilibria*. This can in principle be accomplished by solving an LCP to obtain a mixed strategy for each type of each player. Although in game-theoretic studies, Bayesian games are often defined with infinite type and action spaces, we classify actions discretely after doctrines the players are trained to follow, and if the intuitive type of a player is a continuous variable we discretize it.

6.2 Extensive Imperfect Game with Chance Nodes

At level two, for each node represented by the distinct type profile $t_{-i} \in T_{-i}$, the node is the start of the model that the type profile $t_{-i} \in T_{-i}$ gives rise to. To represent this model we use a game on extensive form; that is, a game with players N, actions $(C_i)_{i \in N}$, and utility functions $(u_i)_{i \in N}$. The triangular nodes represent our own decisions and the round nodes are chance nodes where we are uncertain of the actions of the other players.

The (still Bayesian) game relates to the influence diagram in Fig. 1 in that N represent the n agents that are about to make decisions $D_1, \ldots, D_n$, C_i represent the actions available for agent i in decision node D_i, and u_i is the utility that is obtained in the diamond shaped utility node after the outcome in the random variable C in Fig. 1. The result of the game component will be the conditional probability table for the random variable C.

6.3 Tree Look-Ahead

The depth of the game tree corresponds to inference of agents' actions that are dependent of each other, i.e., a series of what-if questions such as "what is the usefulness if player $i \in N$ performs action $c_i \in C_i$ and players $N{-}i$ perform actions $c_{-i} \in C_{-i}$ which makes player $i \in N$ respond with action $c_i \in C_i$", etc.

Look-ahead algorithms are typically modeled using a discount factor $\gamma \in [0, 1]$ that reduces the utility with γ^d where d is the tree depth. For problems in which the discount factor is not too close to 1 a shallow search is often good enough to give near-optimal decisions[14].

Look-ahead game trees have been used successfully for reasoning in, possibly uncertain, games with perfect information where optimal solutions are obtained with the minimax algorithm. Examples of such games are chess, go, backgammon, and monopoly. In the context of C2 we deal with imperfect information which forces us to solve a more complex game, more similar to poker, since we cannot be sure of exactly where we are in the game tree. Although ordinary minimax algorithms cannot be used in our context it is still likely that the ideas from ordinary game play algorithms, such as the famous alpha-beta pruning, can be re-used effectively. This is interesting as these ideas rest on almost a century of research and experience[14, 20].

7 Equilibria and Complexity

While modeling and representing a C2 situation is interesting in its own right, a primary concern is the use and interpretation of the model. In game theory the concept of Nash equilibria define game solutions in the form of strategy profiles in which no agent has an incentive to deviate from the specified strategy. Without doubt, defining equilibria is the foremost goal in game theory. Fortunately, this means that we can lean on well-established results in our effort to find equilibria for the C2 situation.

For a Bayesian game, Harsanyi[23] defined the *Bayesian equilibrium* to be any set of mixed strategies for each type of each player, such that each type of each player would be maximizing his own expected utility given that he knows his own type but does not know the other players' types. Existence of a Bayesian equilibrium solution in mixed strategies follows from the famous existence theorem for extensive games, which is due to Nash[22].

In a series of articles[26, 27] published during the last decade the *sequential form* as a replacement for the strategic form has provided a representation suitable for efficient computation of equilibria in an extensive imperfect game with chance nodes. The idea is to keep the game in extensive form and look at the leaves in order to compute equilibria in a bottom-up fashion. As the creation of the matrix for the strategic form requires exponential space the computational complexity is reduced from a **PSPACE**-complete one into a problem that is linear in the height of the tree. This method is presently applicable to Bayesian games with consistent beliefs. These results give hope for solutions that solve finite and small Bayesian games in reasonable time.

8 A Small Example

At a certain point in battle, a blue (male) unit controls an asset (equipment or territory). When a red (female) unit appears on the scene the blue unit knows immediately whether its own forces are inferior or superior. The red unit on the other hand, does not know anything regarding the capabilities of the blue unit. The blue unit has the choice to engage in battle or to remain passive. If he remains passive the red unit will use her sensors to detect whether he is superior or not and if he is inferior she will force him to give up the asset. On the other hand, if the blue unit choose to engage the red unit she will be faced with an opportunity to flee or to engage. If the blue unit is superior and the red unit chooses to engage him, he will both defeat the red unit and keep control of the asset. If the blue unit is inferior and the red unit chooses to engage him he will both loose the battle and the asset. If the red unit flees the blue unit will keep control of the asset whether he is superior or not.

We use the card game of Myerson[25] to illustrate the depicted situation. As indicated in the situation description, we follow the convention that odd-numbered players are male and even-numbered players are female. This is common practice in game theory. At the beginning of the game both players put a dollar (the asset) in the pot. Player 1 (the blue force) looks at a card from a shuffled deck which may be red (he is superior) or black (he is inferior). Player 2 (the red force), on the other hand, does not know the color of the card but maintains a belief of this in the form of a probability distribution in her influence diagram, i.e., a belief of the possibility of player 1 being superior or inferior. Player 1 moves first and has the opportunity to fold (F) or to raise (R) with another dollar, i.e., remain passive or engage in battle. If he raises, player 2 has the opportunity to pass (P) or to meet (M) with another dollar in the pot, i.e., flee or engage in battle.

We let $\alpha \in (0, 1)$ denote player 2's belief of player 1 being superior. The situation can then be modeled with a Bayesian game Γ^{b}, as defined in Eq. (1), with $N = \{1, 2\}$, $C_1 = \{F, R\}$, $C_2 = \{M, P\}$, $T_1 = \{1.\mathrm{a}, 1.\mathrm{b}\}$, $T_2 = \{2\}$, $p_1(2|1.\mathrm{a}) = p_1(2|1.\mathrm{b}) = 1$, $p_2(1.\mathrm{a}|2) = \alpha$, $p_2(1.\mathrm{b}|2) = 1 - \alpha$ and $(u_1(c_1, c_2, t_1), u_2(c_1, c_2, t_1))$ as in Table 1.

$t_1 = 1.\mathrm{a}$	M	P
R	$2, -2$	$1, -1$
F	$1, -1$	$1, -1$

Superior

$t_1 = 1.\mathrm{b}$	M	P
R	$-2, 2$	$1, -1$
F	$-1, 1$	$-1, 1$

Inferior

Table 1: Payoff matrices for the Myerson card game.

Solving the game using the technique described by Harsanyi[23] involves introducing a historical chance node, a "move of nature", that determines player 1's type, hence transforming player 2's incomplete information regarding player 1 into imperfect information. The Bayesian equilibrium of the game is then precisely the Nash equilibrium of this imperfect information game. The Harsanyi transformation of Γ^{b} is depicted in Fig. 4 on extensive form.

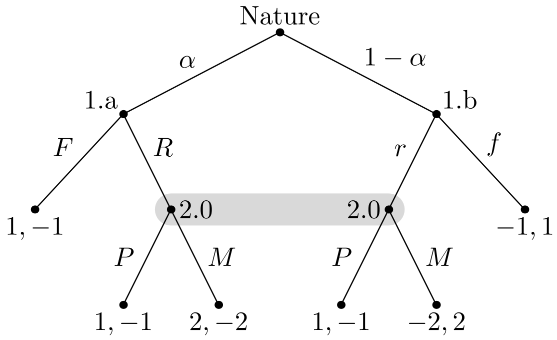

Fig. 4: The Harsanyi transformation of the game in Table 1.

Note that there are two decision nodes denoted "2.0" that belong to the same information set representing the uncertainty of player 2 regarding player 1's type. Also, note that the move labels on the branch following the "1.a" node do not match the move labels on the branches following the "1.b" node, representing that player 1 is able to distinguish between these two nodes. The normal way of solving such a game is to look at the strategic representation, as seen in Table 2.

	M	P
Rr	$4\alpha-2, 2-4\alpha$	$1, -1$
Rf	$3\alpha-1, 1-3\alpha$	$2\alpha-1, 1-2\alpha$
Fr	$3\alpha-2, 2-3\alpha$	$1, -1$
Ff	$2\alpha-1, 1-2\alpha$	$2\alpha-1, 1-2\alpha$

Table 2: The strategic form of the game in Fig. 4.

In order to solve the game, first note that Fr is dominated by Rr and that Ff is dominated by Rf regardless of the value of α, i.e., player 1 will always raise if in a superior position. Second, if $3/4 \leq \alpha < 1$ we have that P dominates M so that player 2 will always choose to pass, which, in turn, implies that player 1 will always choose to raise. Hence, $([Rr], [P])$ is the one and only equilibrium strategy profile for $3/4 \leq \alpha < 1$. For $0 < \alpha < 3/4$ there are no equilibria in pure strategies (just check all four remaining possibilities) and we have to look for equilibria in mixed strategies. Let $q[Rr] + (1-q)[Rf]$ and $s[M]+(1-s)[P]$ denote the equilibrium strategies for players 1 and 2 respectively, where q denotes the probability that player 1 raises with a loosing card and s the probability that player 2 meets if player 1 raises. A requirement for an equilibrium for player 1 is that his expected payoff is the same for both Rr and Rf, i.e., $s(4\alpha-2) + (1-s)1 = s(3\alpha-1) + (1-s)(2\alpha-1) \Rightarrow s = 2/3$. Similarly, to make player 2 willing to randomize between M and P, M and P must give her the same expected utility against $q[Rr] + (1-q)[Rf]$ so that $q(4\alpha-2) + (1-q)(3\alpha-1) = q1 + (1-q)(2\alpha-1) \Rightarrow q = -\alpha/(3(\alpha-1))$.

We can now use the equilibrium strategy of the imperfect information game in order to derive the Bayesian equilibrium of the game Γ^{b}. A Bayesian equilibrium specifies a randomized strategy profile containing one strategy $\sigma_i(\cdot|t_i)$ for all combinations of players and types. Hence, the unique Bayesian equilibrium of the game Γ^{b} is $\sigma_1(\cdot|1.\mathrm{a}) = [R]$, $\sigma_1(\cdot|1.\mathrm{b}) = q[R] + (1-q)[F]$, $\sigma_2(\cdot|2) = 2/3[M] + 1/3[P]$ for $0 < \alpha < 3/4$ and $\sigma_1(\cdot|1.\mathrm{a}) = [R]$, $\sigma_1(\cdot|1.\mathrm{b}) = [R]$, $\sigma_2(\cdot|2) = [P]$ for $3/4 \leq \alpha < 1$.

Although this simple game presents a solution that is not entirely trivial, it is simpler than our full family of games in that it is zero-sum with only two players (and thus has a unique Nash equilibrium that is computationally easy to find), and in that both sides know the value of α (making it equivalent to an imperfect information game).

9 Related Work

Development of game tools is an active area in AI. In the Gala system of [20], tools exist for defining games with imperfect information. A tractable way to handle games with recursive interaction in a strategic form was developed in Gmytrasiewicz[28], where the potentially infinite recursion of beliefs about opponents is represented approximately as a finite depth discrete utility/probability matrix tree defining the players beliefs about each other. The solution emerging from this modeling is not a Bayesian game equilibrium, however.

There is a significant body of work on multi-agent interactions in the intelligent agents literature. A survey of methodological and philosophical problems appears in [28]. The principle of bounded rationality can be taken as an excuse to use simpler solution concepts than Bayesian game solutions. In our case, there is no reason to assume that the opponent is not rational – there would be few excuses if he turned out to be so. This does not mean that it is not necessary to take advantage of opponents mistakes when they occur. Plans must foresee this and have opportunities of opponent mistakes as a part; but these options should not be executed until the evidence of the mistake is sufficient. The recursive modeling of multi-agent interaction of [28] (mostly developed for cooperative rather than competitive interaction) is thus not appropriate in our application. The proposal in [17] is to use game theory with zero-sum game-tree look-ahead for C2 applications. Although this approach was successful for analysis of lower level game situations, we have argued above that it is not enough in a complete higher level C2 tool.

In [29] the MAID representation is defined, which partitions the decision and utility variables by agent, so that utilities and decisions of many agents can be described. However, in this type of model it must be assumed that the structure of the game and its utilities is understood and accepted by all participants. An extension of the MAID framework is the NID – Network of Influence Diagrams. In the version described in [30], several MAIDs – or other game representations – can be connected in a directed acyclic graph, where outgoing arcs are labelled with a probability distribution. This gives a possibility to define situations were agents do not all use the same model, but there is no way to describe in an acyclic graph a situation where there is mutual uncertainty and inconsistent beliefs about the game structure and the opponents goals. A development of the NID to handle general Bayesian games is however announced in [30].

10 Conclusions

Game-theoretic tools have a potential for situation prediction that takes real uncertainties in enemy plans and deception possibilities into consideration. Developments in game algorithms and descriptions of situation development with dynamic BNs, and decision situations with influence diagrams, make realistic modelling from sensor information possible, and the resulting games solvable. For realizing this on full scale problems, further development of efficient game solving methods to Bayesian games will be required. For usability, it is important to design game tools to match actual planning methods used. Several problems remain to analyze. The possibility of using zero-sum payoffs would lead to a much simpler game solution, at the price of less accurate modeling. Can it be used in some cases? If not, are there application specific reasons to use one of the several proposed criteria for selecting a particular equilibrium? We presently assume that robustness (insensitivity to small changes in parameters) is an important property of a relevant equilibrium.

Acknowledgments

The influence diagram in section 3 originates from discussions within the Decision Support Group[1] and contain the joint efforts of Ronnie Johansson, Robert Suzić, Klas Wallenius, and the authors. We acknowledge the insights into the problem area made possible by Berndt Brehmer, who invited us to the research seminars at the Swedish Defence College. The discussions in these seminars have helped us understand the usability aspects of our problem.

References

[1] Erik Durschmied. *The Hinge Factor: How Chance and Stupidity Have Changed History*. Coronet Books, London, 1999.

[2] Thomas P. Coakley. *Command and Control for War and Peace*. National Defense University Press, Washington, District of Columbia, 1991.

[3] US Army. *Intelligence Preparation of the Battlefield*. Field Manual FM 34–130, Headquarters, Department of the Army, Washington, District of Columbia, July 1994.

[4] Peter Thunholm. *Military Decision Making and Planning: Towards a New Prescriptive Model*. PhD thesis, Stockholm University, Sweden, June 2003.

[5] Gary A. Klein. Recognition-primed decisions. In William B. Rouse, editor, *Advances in Man-Machine Systems Research*, pages 47–92. JAI Press, Greenwich, 1989.

[6] Alan N. Steinberg and Christopher L. Bowman. Revisions to the JDL Data Fusion Model. In David L. Hall and James Llinas, editors, *Handbook of Data Fusion*, chapter 2. CRC Press, 2001.

[7] Franklin E. White. Managing data fusion systems in joint and coalition warfare. In Mark Bedworth and Jane O'Brien, editors, *Proc. of the Int. Conf. on Data Fusion (EuroFusion98)*, pages 49–52, October 1998.

[8] Robert Suzić. Representation and recognition of uncertain enemy policies using statistical models. In *Proc. of the NATO RTO Symposium on Military Data and Information Fusion*, Prague, Czech Republic, October 2003.

[9] Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[10] Henrik Friman and Berndt Brehmer. Using microworlds to study intuitive battle dynamics: A concept for the future. In *Command and Control Research and Technology Symposium*, 1999.

[11] Joel Brynielsson and Klas Wallenius. Game environment for command and control operations (GECCO). In *Proc. of the First Int. Workshop on Cognitive Research With Microworlds*, pages 85–95, November 2001.

[12] Gary E. Horne. Maneuver Warfare Distillations: Essence not Verisimilitude. In P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, editors, *Proc. of the 1999 Winter Simulation Conf.*, 1999.

[13] Simon Ahlberg, Pontus Hörling, Karsten Jöred, Christian Mårtenson, Göran Neider, Johan Schubert, Hedvig Sidenbladh, Pontus Svenson, Per Svensson, Katarina Undén, and Johan Walter. The IFD03 information fusion demonstrator. In *Proc. of the Seventh Int. Conf. on Information Fusion (FUSION 2004)*, Stockholm, Sweden, 2004.

[14] Stuart J. Russell and Peter Norvig. *Artifical Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, second edition, 2003.

[15] Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In Gregory F. Cooper and Serafín Moral, editors, *Proc. of the Fourteenth Conf. on Uncertainty in Artificial Intelligence (UAI-98)*, pages 256–265, Madison, Wisconsin, July 1998.

[16] Klas Wallenius. On support for situation awareness in command and control. In *Proc. of the Seventh Int. Conf. on Information Fusion (FUSION 2004)*, Stockholm, Sweden, 2004.

[17] Amnon Katz and B. Butler. "Game commander" – applying an architecture of game theory and tree lookahead to the command and control process. In *Proc. of the Fifth Annual Conf. on AI, Simulation, and Planning in High Autonomy Systems (AIS94)*, Gainesville, Florida, 1994.

[18] Ronnie Johansson. *Information Acquisition in Data Fusion Systems*. Licentiate thesis, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, November 2003.

[19] Rina Azoulay-Schwartz and Sarit Kraus. Negotiation on data allocation in multi-agent environments. *Autonomous Agents and Multi-Agent Systems*, 5(2):123–172, June 2002.

[20] Daphne Koller and Avi Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1):167–215, July 1997.

[21] Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. *The Linear Complementarity Problem*. Academic Press, 1992.

[22] John F. Nash. Non-cooperative games. *Annals of Mathematics*, 2(54):286–295, 1951.

[23] John C. Harsanyi. Games with incomplete information played by "Bayesian" players. *Management Science*, 14(3,5,7):159–182, 320–334, 486–502, 1967–1968.

[24] Jean-Francois Mertens and Shmuel Zamir. Formulation of Bayesian analysis for games with incomplete information. *International Journal of Game Theory*, 14(1):1–29, 1985.

[25] Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.

[26] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast algorithms for finding randomized strategies in game trees. In *Proc. of the 26th Annual ACM Symposium on the Theory of Computing*, pages 750–759, Montréal, Québec, Canada, May 1994.

[27] Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, June 1996.

[28] Piotr J. Gmytrasiewicz and Edmund H. Durfee. Rational coordination in multi-agent environments. *Autonomous Agents and Multi-Agent Systems*, 3(4):319–350, December 2000.

[29] Daphne Koller and Brian Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45(1):181–221, October 2003.

[30] Ya'akov Gal and Avi Pfeffer. A language for modeling agents' decision making processes in games. In *Proc. of the Second Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'03)*, pages 265–272, Melbourne, Australia, July 2003.

[1] `http://www.nada.kth.se/theory/dsg/`