Refinements of the Command and Control Game Component

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

***Abstract*— Prediction of future courses of events is a necessary ingredient in tomorrow's Command and Control centers. This is being identified in higher levels of, e.g., the JDL model where awareness of the development of a situation is crucial for providing a complete and comprehensible situation picture. To cope with gaming situations, i.e., situations where commanders make decisions based on other commanders' reasoning about one's own reasoning, traditional AI methods for inference need to be extended with algorithms stemming from game theory.**

In this article we formalize the ideas of an information fusion "game component". Also, we review current state of the art when it comes to computational game theory and discuss the time constraints from an information fusion perspective along with a discussion regarding the solution/equilibrium selection problem. Furthermore, results from computer simulations and analysis of computational bottlenecks are presented.

I. Introduction

The goal of any Command and Control (C2) system is to maintain and use vast amounts of information, with varying relevance and accuracy, in a proper and timely manner to support planning and decision-making. Information fusion spans the full information path from sensor reports to a commander grasping the situation. However, research has until now been focused on lower levels, focusing primarily on fusing sensor data into tracks or vehicles. Some efforts have also been put on the subsequent tasks of aggregation into larger formations in order to create a refined situation picture that can be presented to the commanders. However, merely presenting a comprehensible description of the situation does not give a complete understanding of the development of a situation. Hence, the last step in the information fusion process is the task of predicting other actors' plans and to suggest future courses of actions (COAs).

We believe that the challenges and difficulties when it comes to prediction are fundamentally different depending on available time, resources and level of abstraction. In a short-term tactical decision task the commanders are probably best off with a comprehensible prediction of forthcoming troop movements. Lately, a few interesting research papers focusing on this objective have started to appear in the information fusion community[1, 5, 11]. However, decision tasks depending on speculations about opponents' long term planning require reasoning and modeling regarding opponent intentions. This work focuses on this latter form of prediction, which ought to be an integral part of higher level C2 decision support, with few results seen within information fusion research although proposed in for example [15, 24].

The main contribution of our work is that of the application and combination of techniques for the information fusion context. General game-theoretic methods for prediction are, in most cases, intractable for the generic case. The best known method, the Lemke-Howson algorithm, yields exponential worst case running time and does not find all solutions. Continuation methods, following bifurcations of perturbed problems, are delicate at best[3]. By building a game component especially suited for prediction in information fusion, we can eliminate certain problems that give rise to computationally infeasible problems. Moreover, by representing data using extended influence diagrams used in the AI community we maintain a reasonably good understanding of the decision problem at hand.

In section II we give some background on building blocks from the AI community that we build upon and in section III we outline the game component that we are targeting. Section IV gives a description of common computational methods for Nash equilibria computation and discusses computational complexity. Experiments and results from computer simulations are found in section V. Finally, section VI concludes and discusses further research issues.

II. Lower versus Higher Level Prediction

One goal of artificial intelligence (AI)[20] has been to create expert systems, i.e., systems that can match the performance of human experts provided the appropriate domain knowledge. 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 to solve problems 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 article.

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 (BNs) 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 or statistical 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[20] 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.

An influence diagram is a natural extension to a BN incorporating decision and utility nodes in addition to chance nodes, and represents decision problems for a single agent[10]. 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 proposed moves. 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.

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. While these sub diagrams are interesting in their own right, they are not the topic for this article. Hence, we have chosen to think of them as existing models that influence the decisions we are modeling.

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 one's own actions. This is not possible to model in an influence diagram or BN without additional machinery. At this point it should also be noted that the diagram in Fig. 1 should not be considered to be very useful in its own right. Rather, it is a statement of the problem we are trying to solve. Among other things, the diagram is not *regular* which is a requirement for algorithms that evaluate influence diagrams[21]. Regularity assumes a total ordering of all of the decisions, a reasonable condition when there is a single decision maker. In this work we use the influence diagram to develop a generalized technique that solves problems for multiple decision makers.

In higher level C2 we can be certain that great 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.

III. The Game Component

In [4] a "game component" suitable for prediction in higher level information fusion was introduced. The basic element of the game component is an influence diagram, as seen in Fig. 1, which gives the commander the opportunity to model the decision situation *given* that he knows what type of opponent he is facing, i.e., he makes an assumption regarding the opponent's beliefs, desires and intentions. Since the commander knows the properties of the opponent he is facing he will be able to estimate the payoffs in the influence diagram using domain knowledge.

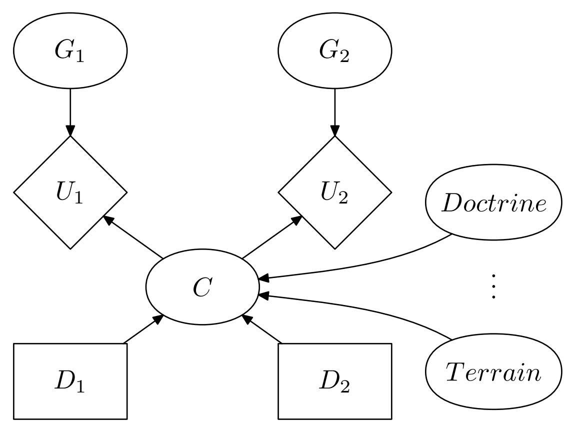

Fig. 1. The C2 process modeled in an influence diagram. Terrain data bases and doctrine are examples of sub diagrams that characterize a certain type profile.

In our application, which is reasoning in a C2 center, we believe there are a few, say less than ten, different possible interesting opponent models one ought to consider with a final "no information model"[8] representing irrational behavior and lack of information.

The architecture consists of a probability distribution over the possible models, depicted in Fig. 2, forming a game with incomplete information. Hence, the decision maker is also supposed to estimate the prior probability regarding which model is accurate. The result from this is a one stage Bayesian game that describes the whole decision problem and solutions are obtained in the form of mixed strategy Nash equilibria. An example of such a game can be seen in Fig. 3, which is an example of the game we have experimented with, where each of the two players reason about two possible opponents and have the choice of performing four different actions/plans, respectively.

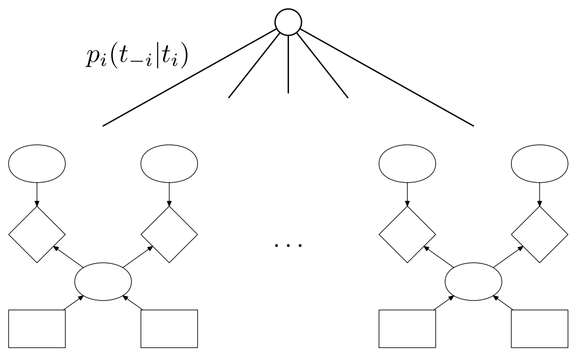

Fig. 2. Architecture overview. Models are represented by influence diagrams that yield payoff values for a Bayesian game.

IV. Computational Aspects

The representation and formulation of the C2 game mechanism is without doubt both interesting and important in its own right. However, the calculation of an optimal solution, i.e., the task of proposing a COA to the commander in question, is of course the next task to undertake. Relative to the study of game mechanisms and their properties, which has been pursued by economists for decades, the study of computational methods for calculation of Nash equilibria is still a fairly young research area with innovative results seen during the last decade.

In game theory the concept of Nash equilibria defines game solutions in the form of strategy profiles in which no agent has an incentive to deviate from the specified strategy. The appropriate algorithm to use depends on several factors, e.g., the structure of the game, the number of players in the game and the number of equilibria one wants to find. The game we are looking at gives rise to a bimatrix game, i.e., a two-person, non-zero-sum game with a finite number of pure strategies.

The problem of finding Nash equilibria in bimatrix games can be formulated as a linear complementarity problem (LCP)[6] with the best known solution method being the Lemke-Howson algorithm[14]. The Lemke-Howson algorithm starts with a dummy equilibrium point and finds subsequent equilibria by following paths of "almost" equilibria relaxing one constraint at a time. One can find the set of all accessible equilibria by tracing out the entire network of such "almost" equilibria. It should, however, be kept in mind that the set of accessible equilibria does not necessarily include all Nash equilibria.

Finding *all* equilibria of a bimatrix game will remain a hard problem[7]. Several methods exist with the algorithm proposed by Mangasarian[17] being the most widely adopted method. This method enumerates all of the extreme points of the components of the set of Nash equilibria, using the algorithm proposed by Balinski[2], and therefore finds all equilibria.

For a game on extensive form, the traditional solution method has been to transform the game into strategic (matrix) form and apply standard solution methods as described above. However, the creation of the matrix for the strategic form may cause a combinatorial explosion due to the fact that each value in the matrix representation of a strategic form game represents the payoff for a *complete* strategy. Hence, even though a game tree typically contains widely different decision alternatives in different sub trees the decisions in the other sub trees still need to be considered and therefore the strategic form matrix dimension grows for each node that is traversed. As an alternative to the transformation into strategic form the sequence method can be used. This method re-formulates the game by replacing the game's (pure) strategies with new strategies being represented by sequences of moves, i.e., paths from the root node down to the leaves. As the creation of the matrix for the sequence form relies on payoffs that are already in the tree the problem complexity is reduced from a $\mathbf{PSPACE}$-complete problem into a problem that is linear in the size of the tree. However, it should be kept in mind that general game trees often share decision alternatives and, hence, do not exhibit a full scale combinatorial explosion. The resulting reduced game matrix can then be solved using Lemke's algorithm as described in [12, 13, 22].

Solution methods for general-sum game-theoretic problems are, at least in theory, intractable for the generic case. The Lemke-Howson algorithm exhibits exponential worst case running time for some, even zero-sum, games although this does not seem to be the typical case[23]. Interior point methods are not known for linear complementarity problems arising from games. Methods amounting to examining all equilibria, such as finding an equilibrium with maximum payoff, have unfortunately been proven $\mathbf{NP}$-hard[7], so for these kinds of problems no efficient algorithm is likely to exist.

V. Computer Experiments

To get an understanding of the games and their properties that the C2 game component gives rise to we have performed computer experiments with algorithms found in Gambit[18]. Gambit is an equilibrium computation package incorporating both GUI and program libraries. Although version 1.0 has still not been finished, the Gambit project probably contains the most well-developed code available today for equilibria computation. Development started already in the mid 1980s with a total rewrite of the code into C++ in 1994.

The games considered are two-person one-stage Bayesian games[9] with two possible types for each player and four possible actions or plans that need to be considered. Payoffs consist of uniformly chosen integer values ranging from -10 to 10. The prior probabilities $p_i(t_{-i}|t_i)$ are chosen uniformly in the interval ranging from $1/10$ to $9/10$ with rational precision and only using even tenths. 100 such games were created randomly. An example extensive form game is shown in Fig. 3. The corresponding normal form game will be a

16×16-matrix since there exist 4×4 pure strategies for each player.

Sampling probabilities from a uniform distribution instead of some other distribution indicate our belief that these probabilities are in reality estimated by a commander and should reflect a commander's subjective belief regarding what kind of opponent he is actually facing. If, for example, the commander is fairly certain regarding the situation he might give the model reflecting his belief the probability $9/10$ while he still wants to model an unlikely, but possibly dangerous, opponent with probability $1/10$. If the commander on the other hand is totally uncertain regarding the situation he might assume that two models are equally likely, i.e., he chooses $[1/2, 1/2]$ for the probabilities. The objective of this study has primarily been to study the amount and properties of the games' resulting equilibria in order to investigate if a tool based on game theory can actually be used to give advice in a C2 situation. Although not part of the main goal for this study, we have also compared the 100 games versus each other with respect to running time in order to get an understanding of how computation time varies depending on game type and number of equilibria.

We believe that the described games are good examples of games that might need to be considered in the C2 game component as described briefly in section III and more thoroughly in [4]. We have two primary reasons for making this assumption. First, the games are Bayesian, i.e., they give the commanders the possibility to reason about several opponent models according to their prior belief. Second, the plans and opponent models that need to be considered in higher level C2 are likely to be conceptual and, hence, fairly limited. It is important to notice that it is sufficient to vary a game's payoff function and, in the Bayesian game case, its type probabilities to create games that exhibit fundamentally different characteristics. Therefore we feel confident that the 100 games included in this study represent the characteristics of the games needed to be considered in a C2 game component.

The methods described in section IV are found in Gambit as `EnumMixedSolve` and `LcpSolve`. `EnumMixedSolve` implements the algorithm for finding all Nash equilibria described in [17] while `LcpSolve` formulates the problem of finding equilibria as a linear complementarity problem and solves the game using this approach. `LcpSolve` can also be used directly on an extensive form game where it implements the sequence method and solves the game using the approach described in [12, 13, 22]. The 100 random games were solved in four different ways:

- `EnumMixedSolve` was used to get hold of all equilibria for comparison reasons,
- `LcpSolve` on the transformed normal form game,
- `LcpSolve` directly on the extensive form game using the sequence method,
- `LcpSolve` using rational precision due to problems with numerical stability.

The number of (actual) equilibria are shown in Table I along with a comparison with the two versions of `LcpSolve`. The column "`EnumMixed`" represents the actual amount

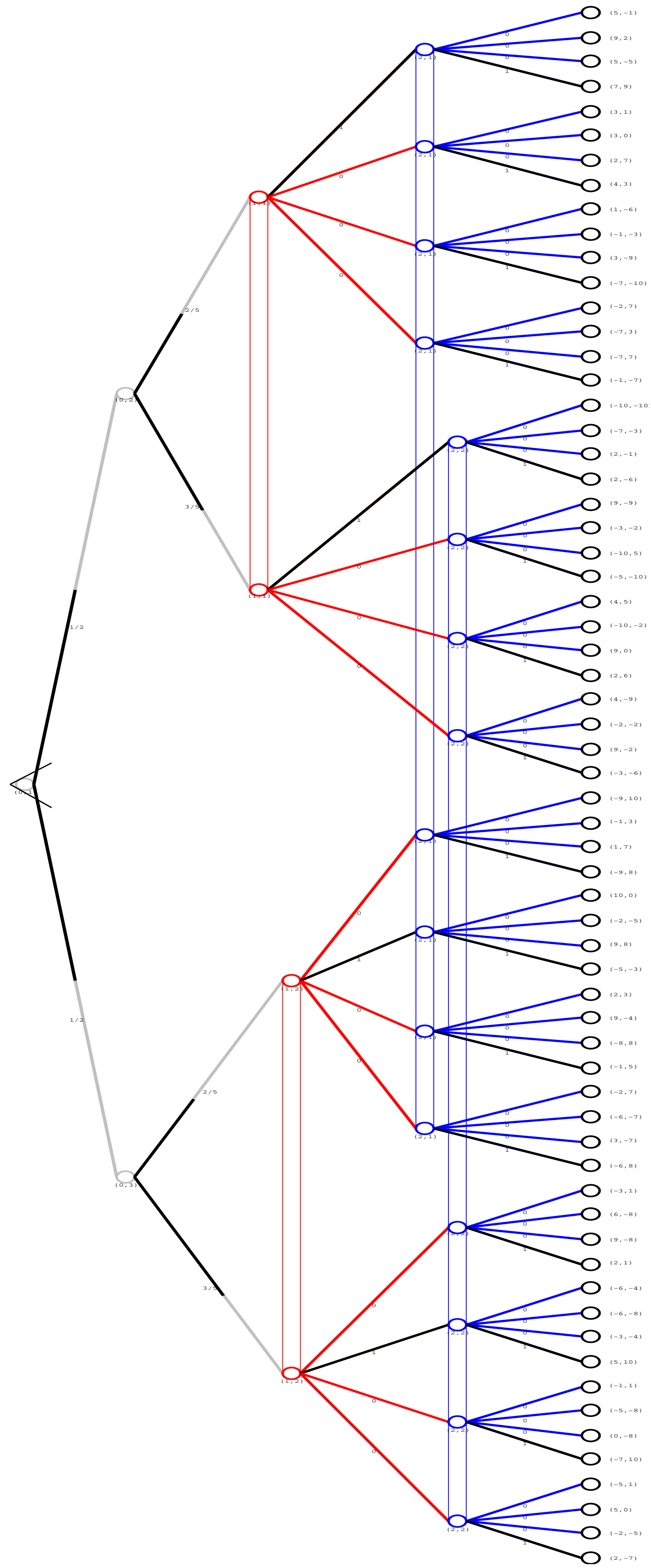

Fig. 3. One of the 100 games that were used in the simulations. The probabilities shown next to the arcs show the only pure strategy equilibrium out of nine Nash equilibria.

of equilibria found using the `EnumMixedSolve` algorithm while columns denoted "`Lcp`" indicates the percentage of the actual solutions that are found correctly using the two different versions of `LcpSolve`. "NFG" means the normal form game transformation and "EFG" means `LcpSolve` implements the sequence method on the extensive form game.

# of eq.	`EnumMixed`	`Lcp` [NFG]	`Lcp` [EFG]
1	37	100%	100%
2	2	0%	0%
3	25	92%	96%
4	4	0%	0%
5	15	93%	73%
6	3	0%	0%
7	6	100%	83%
8	0	–	–
9	6	50%	67%
10	0	–	–
11	1	100%	100%
12	1	0%	0%

TABLE I

THE AMOUNT OF EQUILIBRIA THAT DIFFERENT EQUILIBRIA COMPUTATION ALGORITHMS FINDS.

As indicated in Table I the Lemke-Howson algorithm always yields an odd number of equilibria. This is part of the algorithm.

VI. CONCLUSIONS

Game-theoretic tools have a potential for situation prediction that takes real uncertainties in enemy plans and deception possibilities into consideration. Development of game algorithms using influence diagram situation descriptions, make realistic modeling and reasoning from sensor information possible.

This work has focused on the computationally hard task of calculating optimal solutions in a game similar to one that must be solved in a C2 decision situation. The concept of a Bayesian game makes it possible for a commander to incorporate any prior beliefs regarding his opponents and, hence, seems to be a good choice for representing reality. Computer simulations show that computation of optimal solutions seems to be tractable in reasonably sized C2 decision problems. Moreover, despite the intractability of finding all optimal solutions there exist fast algorithms that often finds all, or nearly all, solutions.

However, several problems remain to analyze. We have only started looking at what can be done to choose among multiple equilibria. This is a well-studied subject with several proposed criteria for selecting a particular equilibrium, for example using principles of maximum payoff or Pareto efficiency. However, a generic method has not been found and each proposed criteria seems to have a counterexample that results in unexpected behavior. As an example, consider the game-theoretic classic known as the Prisoners' Dilemma (see, for example, [16, 19] for the background story). In this game the notion of Pareto efficiency, where an outcome of the game is said to be *Pareto efficient* if and only if there is no other outcome that would make all players better off, applies to all outcomes of the game except for the game's unique Nash equilibrium. Our hope is that it will become possible to establish guidelines regarding the small subset of games that appear in a C2 setting.

Moreover, we have not primarily been interested in how large games one can actually solve. We believe the size of the game reflects *one possible outcome* of the influence diagram utilities described in section III. The games we have studied are only large enough to require one or two minutes to solve. In a subsequent study, it would be interesting to look at how large games can actually be solved using hardware and time limits that are likely to be found in a state-of-the-art C2 center.

ACKNOWLEDGMENTS

We would like to thank Per Svensson, Swedish Defence Research Agency, for commenting on this work. Furthermore, we wish to acknowledge the many fruitful discussions with other researchers at the FUSION 2004 conference which led to an increasing understanding and awareness of the nature of higher level information fusion problems. In particular, we wish to mention Subrata K. Das, Charles River Analytics Inc., who provided beneficial thoughts on the historically prevailing philosophy of influence diagrams being a tool solely intended for the single agent decision problem.

REFERENCES

[1] 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 *Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004)*, Per Svensson and Johan Schubert, Eds., vol. 2, Stockholm, Sweden, June 28–July 1, 2004, pp. 936–943.

[2] Michel L. Balinski, "An algorithm for finding all vertices of convex polyhedral sets," *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, no. 1, pp. 72–88, Mar. 1961.

[3] Ben Blum, Christian R. Shelton, and Daphne Koller, "A continuation method for Nash equilibria in structured games," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003)*, Georg Gottlob and Toby Walsh, Eds., Acapulco, Mexico, Aug. 2003, pp. 757–764.

[4] Joel Brynielsson and Stefan Arnborg, "Bayesian games for threat prediction and situation analysis," in *Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004)*, Per Svensson and Johan Schubert, Eds., vol. 2, Stockholm, Sweden, June 28–July 1, 2004, pp. 1125–1132.

[5] Joel Brynielsson, Mattias Engblom, Robert Franzén, Jonas Nordh, and Lennart Voigt, "Enhanced situation awareness using random particles," in *Proceedings of the Tenth International Command and Control Research and Technology Symposium (ICCRTS)*, McLean, Virginia, June 2005.

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

[7] Itzhak Gilboa and Eitan Zemel, "Nash and correlated equilibria: Some complexity considerations," *Games and Economic Behavior*, vol. 1, no. 1, pp. 80–93, Mar. 1989.

[8] Piotr J. Gmytrasiewicz and Edmund H. Durfee, "Rational coordination in multi-agent environments," *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 4, pp. 319–350, Dec. 2000.

[9] John C. Harsanyi, "Games with incomplete information played by 'Bayesian' players," *Management Science*, vol. 14, no. 3,5,7, pp. 159–182, 320–334, 486–502, 1967–1968.

[10] Ronald A. Howard and James E. Matheson, "Influence diagrams," in *Readings on the Principles and Applications of Decision Analysis*, Ronald A. Howard and James E. Matheson, Eds. Menlo Park, California: Strategic Decisions Group, 1984, pp. 721–762.

[11] Ronnie Johansson and Robert Suzić, "Particle filter-based information acquisition for robust plan recognition," in *Proceedings of the Eighth International Conference on Information Fusion (FUSION 2005)*, Philadelphia, Pennsylvania, July 2005.

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

[13] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel, "Efficient computation of equilibria for extensive two-person games," *Games and Economic Behavior*, vol. 14, no. 2, pp. 247–259, June 1996.

[14] Carlton E. Lemke and Joseph T. Howson, Jr., "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 2, pp. 413–423, June 1964.

[15] James Llinas, Christopher Bowman, Galina Rogova, Alan Steinberg, Ed Waltz, and Frank White, "Revisiting the JDL data fusion model II," in *Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004)*, Per Svensson and Johan Schubert, Eds., vol. 2, Stockholm, Sweden, June 28–July 1, 2004, pp. 1218–1230.

[16] R. Duncan Luce and Howard Raiffa, *Games and Decisions*. New York: John Wiley and Sons, 1957.

[17] Olvi L. Mangasarian, "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 4, pp. 778–780, Dec. 1964.

[18] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. (2004) Gambit: Software tools for game theory, version 0.97.0.7. [Online]. Available: `http://econweb.tamu.edu/gambit/`

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

[20] Stuart J. Russell and Peter Norvig, *Artifical Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 2003.

[21] Ross D. Shachter, "Evaluating influence diagrams," *Operations Research*, vol. 34, no. 6, pp. 871–882, Nov.–Dec. 1986.

[22] Bernhard von Stengel, "Efficient computation of behavior strategies," *Games and Economic Behavior*, vol. 14, no. 2, pp. 220–246, June 1996.

[23] Bernhard von Stengel, "Computing equilibria for two-person games," in *Handbook of Game Theory with Economic Applications*, ser. Handbooks in Economics, Robert J. Aumann and Sergiu Hart, Eds. Elsevier Science, 2002, vol. 3, ch. 45, pp. 1723–1759.

[24] Klas Wallenius, "Support for situation awareness in command and control," in *Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004)*, Per Svensson and Johan Schubert, Eds., vol. 2, Stockholm, Sweden, June 28–July 1, 2004, pp. 1117–1124.