Challenges in Applying Game Theory to the Domain of Information Warfare[†]

Samuel N. Hamilton, Wendy L. Miller, and Allen Ott
ORINCON Information Assurance
9363 Towne Centre Drive, San Diego, CA 92121-3017

O. Sami Saydjari
SRI Computer Science Laboratory
3601 43rd Street South, Wisconsin Rapids, WI 54494

There are significant advantages to utilizing game theory in the domain of information warfare. The algorithms sculpted to create programs capable of beating some of the best humans in the world in games such as chess and backgammon provide the ability to analyze millions of possibilities, model opponent characteristics, and self-generate what-if scenarios. Unfortunately, the difference between the domain of information warfare and traditional game domains make application of some of these techniques challenging. In this paper, we identify the major challenges presented by the domain of information warfare, and how these challenges interact with traditional game theoretic techniques.

1 Introduction

The success of game theory in creating world-class competitors in domains such as chess, checkers, backgammon, and othello is well known. The strongest effect this has had in the gaming community is not the entry of these programs in competition, but in helping human players with preparation and analysis. By considering huge numbers of possibilities, computers have been shown capable of finding exceptions to general rules and exploiting them mercilessly. In some cases, whole new theories on how to play are developed because of this [1]. We believe game theory will not ultimately replace the human analyst in the domain of information warfare, but can supply him with a powerful tool for suggesting approaches and techniques that might not occur to him, and provide the ability for better and more detailed analysis. Reference [2] provides a nice example of the role we envision game theoretic techniques taking in information warfare.

There is a significant difference, however, between the domain of information warfare and that of traditional games such as chess. In this paper we outline the domain characteristics of information warfare, and define the challenges involved in a straightforward application of game theoretic techniques to this domain.

2 Challenges in information warfare

While the literature is bursting with well-refined techniques for playing games like chess, checkers, and othello, there are numerous differences between the domain of information warfare and these games. In this section, we define the most significant issues, and what effect these issues have on implementing game theoretic techniques in this domain. We assume familiarity with current game theoretic techniques. For those without a background in game theory we suggest referring to [2], which provides a summary of the relevant literature.

Throughout our enumeration and explanation of the fundamental challenges, we adopt the position that a tree search must use a max-max′ based propagation technique [2], not a mini-max based propagation technique. The assumption in mini-max is that that the opponent evaluation function ε_o is the opposite of your own evaluation function ε_s. Thus, $\varepsilon_o = - \varepsilon_{s.}$ In the domain of command and control planning, there has been some limited success based on the assumption that $\varepsilon_o = - \varepsilon_s$ [3]. We believe, however, that in the domain of information warfare this assumption is not justified. There are a variety of reasons for $\varepsilon_o \neq - \varepsilon_s$. First, your opponent will likely have different goals and priorities than you. Second, they may not have full information about your network configuration. This makes it more difficult for them to judge how close they are to reaching their goals, which is the purpose of ε_o.

Within this context, the fundamental issues are:

[†] This material is based on work sponsored by DARPA and supported by the Air Force Research Laboratory under Contract F30602-00-C-0165.

1. There are limited examples to draw from.
2. Players can make multiple, simultaneous moves.
3. Opponents are under no time control constraints.
4. Opponents may have different end goals from us.
5. The set of known legal moves may change during the game.
6. Opponent resources and end goals may change during the game.
7. Timing for move and state updates is not well defined.

2.1 Limited examples

Unlike traditional games, there are not large databases of games between established masters to draw from. This limits the ability to provide static automated evaluation function tuning, which many top programs rely on [4,5,6]. The alternatives are to have domain experts handcraft evaluation functions, or to devise methodologies for tuning the evaluation function during play. In general, handcrafted results have proven inferior to automated learning. The most dramatic example of this comes from backgammon, where programs played well below master strength using handcrafted evaluations until a program utilizing neural network based learning was introduced [5]. In information warfare, where new techniques for attack are frequently introduced, online learning using a handcrafted evaluation function as a starting point appears to be a superior approach. There is an inherent danger, however, in relying on data from an opponent to train. The opponent may be able to construct data that will train your program in bad habits, which the opponent can later exploit. Programs susceptible to this are referred to as *leadable*. One approach to combating leadability is to learn only from mistakes, not from successes. This approach limits the opponent's ability to use rewards for bad moves to train your system.

2.2 Multiple simultaneous moves

In most games, players alternate moves. In information warfare, this is not necessarily true. An opponent can easily launch multiple, simultaneous attacks. While this has been looked at in Markov model approaches using linear programming [7], the solution is not scalable. There are two obvious approaches to modeling simultaneous moves. One is to enumerate all possible move combinations, which can be explosive. The other is to create a null move in the search, which would in effect allow players to move multiple times in a row by skipping the opponent's move.

While null-move searches are already in use in many chess programs, there is a significant difference between those null-move searches and ones in this domain. In chess, the null move is used to check what would happen if a player did nothing, on the assumption that whatever move is chosen, it should be better than doing nothing. Thus, it is used to improve pruning. In our domain, the null move is forced upon us by the opponent. This conflicts with a standard max-max′ search, which assumes a player can always choose which move to make, including the null move.

A straightforward patch would be to create a new type of move, the opponent null move, and to adjust the max-max′ search so that instead of returning Max(children) it would return Max′ (opponent null,Min′ (children)). Thus, the opponent picks whichever is best for him, making a simultaneous move, or letting you do what he thinks you would do. While this would work, some care would have to be taken to make sure that multiple simultaneous moves were searched at an adequate depth. If not, the introduction of opponent null moves could lead us to searching multiple simultaneous moves less deeply than other moves, despite the fact that they are potentially more dangerous.

2.3 Opponent under no time constraints

Most mini-max and max-max′ searches iteratively search deeper and deeper depths until a time limit is reached. Due to the explosive nature of the search, each new iteration takes much longer than the previous iteration. The practical effect of this is that the longer the time between moves, the less effective these search techniques are. For example, while strong chess programs can give even top grand masters a tough game, your average postal chess master still dominates computer opponents given a day or two to think about each move, particularly if the postal master gets to use his own computer as an exploratory tool.

This characteristic implies that more powerful pruning methods are called for in this domain, to combat the explosive nature of the search. This is particularly challenging, since standard pruning techniques for max-max′ searches are weaker than in alpha-beta based searches [8,9]. Best-first approaches are also not feasible, since the unbounded time between opponent moves amplifies the already troublesome memory requirements for best-first based searches.

The lack of time constraints also affects techniques used in opponent modeling. One of the factors in the opponent modeling literature is how deeply the opponent is calculating. The methods used to infer this rely on an opponent spending approximately the same amount of time on each move, since given more time opponents can calculate deeper. Since in information warfare opponent time between moves may be an hour, or may be a month, current techniques for modeling opponent depth no longer apply.

2.4 Opponent may have different end goals from us

The max-max′ searches in the literature all run under the assumption that at the end of the game one person loses, and the other wins [5,6]. Thus, while the opponent evaluation function may differ from ours, both are trying to represent the same thing: how likely are we to win, and our opponent to lose. In information warfare, this symmetry is broken. Our opponent may have an end goal compatible with ours. Or they may be trying for something which we don't like, but does not compromise our mission critical objectives.

Having differing goals violates an assumption inherent in current pruning techniques, which rely on our evaluation and their evaluation varying within a fixed amount. It also decreases the likelihood that the set of heuristics included in the opponent evaluation function are a subset of our heuristics. One fix would be to create a domain database that listed goals and heuristics correlated with obtaining those goals. Opponent modeling could then draw from the heuristics included in that database, instead of from one's own heuristics. In this way not only could the opponent evaluation functions be modeled, their likely goals could be identified.

2.5 Set of known legal moves may change

In most games, the rules are known by all participants a priori. In information warfare, anything goes. New techniques for compromising cyber assets are discovered on a regular basis, and any system that is unable to cope with that is at a severe disadvantage.

While this situation varies significantly from games such as chess, it may fit in with the concept of *quiescence*, which has been explored. Quiescence captures the idea that in some positions the evaluation function is much more accurate than in others. For example, in chess, a position is not quiescent if there are a lot of possible captures and checks, since in the next few moves the evaluation will change drastically thereby calling into question the accuracy of the current evaluation.

By extending this idea to the domain of information warfare, we may be able to capture the idea that while unknown moves may occur at any time, in some positions they are more likely than others. For example, if an intruder is believed to have physical access to a machine within a network, the chances of being able to do a previously unknown move using that machine are much greater.

While quiescence does capture the concept of unknown legal moves at the theoretical level, not all techniques for addressing quiescence are applicable. In chess, the most common method is to conduct searches at extra depth when checks and captures are possible, so that positions can be judged in more quiescent positions. This solution is not applicable here, of course, since an unknown move cannot be searched.

2.6 Opponent end goals may change

Different opponents come and go, which in some cases can be considered equivalent to an opponent changing goals. In fact, a single opponent may decide to change objectives. By changing goals during the game, the opponent often adopts a radically new evaluation function. Current techniques for learning opponent evaluation functions do not take this into account [8,9]. Thus, new techniques must be devised to deal with these instances.

2.7 Time for moves and state updates are not well defined

In most games once a move occurs, it occurs instantaneously. In information warfare this is not true. In fact, it can take a variable amount of time to carry out a move. Not only that, but the move may not actually have the intended effect on the position. In chess, this would be like picking up your Bishop and accidentally moving it to the wrong square. Of course, no serious chess program considers this possibility. In the domain of information warfare, however, there is a big difference between an intended course of action and a completed course of action.

One approach to dealing with time is to give each move a fixed amount of time to complete, and time stamp positions. Then, the move generation algorithm takes care of determining which moves are possible given the position, which now includes moves that are pending completion. While this works, it does not take into account that moves actually may take a variable amount of time, and may not be carried out as expected. Addressing these issues would likely require a new search type which, instead of a node taking the Max(children), would take a probability weighting of the children. Multiple new moves of the same type could then be created, each with a different timing characteristic. The challenge with this approach is that it would require very strong pruning capabilities, since searching a range of outcomes for each move could explode the move space.

3 Summary

While the idea of applying game theory techniques to the domain of information warfare is extremely promising, it is by no means trivial. Information warfare contains numerous characteristics that conflict with the straightforward implementation of standard search techniques. In this paper, we have elucidated the challenges, and in some cases outlined potential approaches to addressing these challenges. It is our belief that while there are numerous hurdles to overcome, game theoretic techniques can be modified to revolutionize the domain of information warfare.

References

[1] G. Tesauro. Temporal Difference Learning and TD-Gammon. *Communications of the ACM*, 38(3):58-68, 1995.

[2] S. N. Hamilton, W. L. Miller, A. Ott, and O. S. Saydjari, The Role of Game Theory in Information Warfare, *The Information Survivability Workshop*, 2001, submitted.

[3] A. Katz and B. Butler, “Game Commander”-Applying an Architecture of Game Theory and Tree Lookahead to the Command and Control Process, Proceedings of the Fifth Annual Conference on AI, Simulation, and Planning (AIS94), Florida, 1994.

[4] A. L. Samuel. Some studies in machine Learning using the Game of Checkers. *IBM Journal of Research and Development*, 3(3):211-229, 1959.

[5] G. Tesauro. TD-Gammon, a Self-Teaching Backgammon Program, reaches master-level play. *Neural Computation*, 6(2):215-219, 1994.

[6] F. Hsu, S. Anantharaman, M. S. Campbell, and A. Nowatzyk. Deep Thought. In T.A. Marsland and J. Schaeffer, editors, *Computer, Chess, and Cognition*, p. 55-78. Springer Verlag, 1990.

[7] M. L. Littman, Markov Games as a Framework for Multi-agent Reinforcement Learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, p. 157-163, 1994.

[8] D. Carmel and S. Markovitch, Learning and using Opponent Models in Adversary Search, Technical Report CIS9606, 1996.

[9] H. H. L. M. Donkers e.t. al, Implementing β-pruning Opponent-Model Search, Technical Reports in Computer Science., CS 00-05. IKAT, Universiteit Maastricht, Maastricht, The Netherlands, May 2000.