Overconfidence or Paranoia? Search in Imperfect-Information Games

Austin Parker, Dana Nau, VS Subrahmanian
Department of Computer Science, University of Maryland
College Park, MD 20742, USA
{austinjp,nau,vs}@cs.umd.edu

Abstract

We derive a recursive formula for expected utility values in imperfect- information game trees, and an imperfect-information game tree search algorithm based on it. The formula and algorithm are general enough to incorporate a wide variety of opponent models. We analyze two opponent models. The "paranoid" model is an information-set analog of the minimax rule used in perfect-information games. The "overconfident" model assumes the opponent moves randomly.

Our experimental tests in the game of kriegspiel chess (an imperfect-information variant of chess) produced surprising results: (1) against each other, and against one of the kriegspiel algorithms presented at IJCAI-05, the overconfident model usually outperformed the paranoid model; (2) the performance of both models depended greatly on how well the model corresponded to the opponent's behavior. These results suggest that the usual assumption of perfect-information game tree search—that the opponent will choose the best possible move—isn't as useful in imperfect-information games.

Introduction

Only the paranoid survive. –Andrew Grove, Intel CEO.

Play with supreme confidence, or else you'll lose. –Joe Paterno, college football coach.

Imperfect-information games are characterized by one's lack of information about the current state of the world. To play such games, one must make assumptions about the world. When it is not possible to make "optimal" assumptions, one has a choice: to err on the side of paranoia or to err on the side of overconfidence.

In this paper we develop a game-theoretically sound algorithm for playing partial-information games. We call the algorithm *information-set search.* It is, of course, intractable for any game of interest as the decision problem in an imperfect-information game is complete in double exponential time (Reif 1984). Thus, we also present two sets of simplifying assumptions which bring our algorithm down to the realm of the tractable. One set of assumptions is overconfident: it arbitrarily assumes the opponent to make random moves. The other set of assumptions is paranoid, assuming an opponent who will always make the best possible move.

We implemented our algorithms and played them against each other in the game of kriegspiel, a imperfect-information version of chess that has recently garnered some research interest (Parker, Nau, & Subrahmanian 2005; Russell & Wolfe 2005; Ciancarini, DallaLibera, & Maran 1997; Sakuta & Iida 2000). We also played them against HS, the best of the kriegspiel algorithms in (Parker, Nau, & Subrahmanian 2005). We observed the following behavior:

- In general, information-set search outperformed HS. This suggests that information-set search may be a better way to search imperfect-information game trees than HS's approach (i.e., pretending that the imperfect-information game is a set of perfect-information games).
- On one hand, the performance of our two opponent models depended on how well they corresponded to the opponent's behavior. But on the other hand, the overconfident model usually outperformed the paranoid model. This suggests that the usual assumption that the opponent will choose the best possible move isn't as useful in imperfect-information games as in perfect-information games.

Notation

Let G be a finite two player turn-taking zero-sum imperfect-information game, whose starting state is s_0, and let the players be a_1 and a_2. The following definitions are based on (Osborne & Rubinstein 1994).

At each state s, let $a(s)$ be the player to move at s, $M(s)$ be the set of available moves, and $m(s)$ be the state produced by making move m in state s.[1] $a(s) = \emptyset$ if the game is over in s. A *history* is a sequence of moves $h = \langle m_1, m_2, \ldots, m_j \rangle$. We let H be the set of all possible histories for G.

For each player a_i, let $\mathcal{I}_i$ be a partitioning of H such that for every $I_i \in \mathcal{I}_i$ and every pair of histories $h, h' \in I_i$, if s and s' are the states produced by h and h' then either $a(s) = a(s') = a_i$ and $M(s) = M(s')$ or $a(s) = a(s') \neq a_i$. In other words, all histories in I_i in which it is a_i's turn lead to states that have the same set of available moves. Figure 1

[1]To model nondeterministic initial states and/or nondeterministic outcomes of moves, we can introduce an additional player a_0 who makes a nondeterministic move at the start of the game and/or after each of the other players' moves. To avoid affecting the other players' payoffs, a_0's payoff in terminal states is always 0.

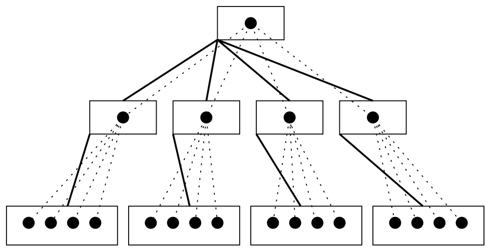

Figure 1: An information-set tree. Each dot represents a history, and each dotted line represents a move. Each box represents an information set, and the solid lines connect each information set with its children.

illustrates the correspondence between information sets and histories. I_i is called an *information set* because it is an abstract model of the information available to a player as a result of some observation.

In a perfect-information game, a player a_i's *strategy* is conventionally defined to be a function $\phi_i(m|s)$ that returns the probability p that a_i will make move m in state s.[2] If a_i uses a deterministic strategy then p will always be 0 or 1; if a_i uses a mixed strategy then p will be in the interval $[0, 1]$. For imperfect-information games, where a_i will not always know the exact state he/she is in, an information set I is used instead of the state s; hence $\phi_i(m|I)$ is the probability that a_i will make move m in the information set I. We say $M(I)$ to refer to the moves available in information set I.

If $h = \langle m_1, m_2, \ldots, m_n \rangle$ is a history, then its probability $P(h)$ can be calculated from the players' strategies. For example, in a two-player game, if a_1 has the first move and the players move in strict alternation, then

$$P(h) = \phi_1(m_1|h_0)\phi_2(m_2|h_1)\ldots\phi_i(m_j|h_{j-1}),$$

where $h_0 = \langle\rangle$, $h_1 = \langle m_1 \rangle$, and so forth. Furthermore, the probability of a history given an information set I is

$$P(h|I) = P(h)/\textstyle\sum_{h' \in I} P(h').$$

Determining Expected Utility

We now consider a two-player game from the point of view of one of the players. Without loss of generality, we will call this player a_1 and the other player a_2. All information sets referred to will be from $\mathcal{I}_1$. Suppose a_1 knows that a_2's strategy is ϕ_2, and wants to make a move that has the highest expected utility. We now discuss what this is and how to calculate it. There are three cases:

Case 1: every history $h \in I$ is terminal, i.e., h's last state s is an endpoint for the game. Then h's *utility value* $U(h)$ is the payoff a_1 gets in s. Hence, the *expected utility* $EU(I)$ is the expected value of the utilities of the histories in I:

$$EU(I) = \textstyle\sum_{h \in I} P(h|I)U(h)$$

Case 2: every history $h \in I$ ends at a state where it is a_2's move. Let $C(I)$ be the set of all *children* of I, i.e., all information sets that could result from a_2's move. In particular, $I' \in C(I)$ if there is an history $h = \langle m_1, \ldots, m_j \rangle \in I$ and a move m s.t. I' contains the history $h' = h \bullet m = \langle m_1, \ldots, m_j, m \rangle$, where $\bullet$ denotes concatenation. For every $I' \in C(I)$, the probability of I' occurring given h is:

$$P(I'|h) = \sum\{\phi_2(m|h) \mid h \bullet m \in I'\}.$$

It follows that

$$EU(I) = \sum_{h \in I} P(h|I) \sum_{I' \in C(I)} P(I'|h)EU(I').$$

In words, this is the sum, for all histories h in I, of the probability that h is the current history times the expected utility if h is the current history.

Case 3: Every history $h \in I$ ends at a state where it is a_1's move. For every child I' and every move m that is possible in I, the probability that m will lead to I' is

$$P(I'|I, m) = \textstyle\sum_{h \in I, h \bullet m \in I'} P(h|I).$$

Thus, the expected utility of I is

$$EU(I) = \max_{m \in M(I)} \left(\textstyle\sum_{I' \in C(I)} P(I'|I, m)EU(I') \right).$$

From the definition of an information set, if one of the histories in I leads to a state where it is a_1's (or a_2's) move, then the same is true for every history in I. Thus there are no states where both a_1 and a_2 can move simultaneously and the above three cases exhaust all of the possibilities.

Therefore, if I is any information set for a_1, then

$$EU(I) = \begin{cases} \displaystyle\sum_{h \in I} P(h|I)U(h), & \text{if } I \text{ is terminal,} \\ \displaystyle\sum_{h \in I} P(h|I) \sum_{I' \in C(I)} P(I'|h)EU(I'), & \\ & \text{if it's } a_2\text{'s move,} \\ \displaystyle\max_m \left(\sum_{I' \in C(I)} P(I'|I, m)EU(I') \right), & \\ & \text{if it's } a_1\text{'s move.} \end{cases} \tag{1}$$

Limited-Depth Recursion

For a finite game, the recursive computation in Eq. 1 will eventually terminate. But for most nontrivial games, the computation will be intractable. For example, in a 50-move game with two possible child information sets for every non-terminal information set, the computation will take 2^{50} recursive calls.

In perfect-information games, it is common to approximate the utility value of a state by searching to some limited depth d, applying a static evaluation function to the nodes at that depth, and pretending that the evaluation function values are the utility values of those nodes. We can approximate Equation 1 in a similar manner:

[2]Strictly speaking, ϕ_i is a partial function, because $\phi_i(m|s)$ is undefined when it isn't a_i's turn in state s.

$$EU_d(I) = \begin{cases} \mathcal{E}(I), & \text{if } d = 0, \\ \sum_{h \in I} P(h|I)U(h), & \text{if } I \text{ is terminal}, \\ \sum_{h \in I} P(h|I) \sum_{I' \in C(I)} P(I'|h) EU_{d-1}(I'), & \\ & \text{if it's } a_2\text{'s move}, \\ \max_m \left(\sum_{I' \in C(I)} P(I'|I,m) EU_{d-1}(I') \right), & \\ & \text{if it's } a_1\text{'s move}. \end{cases} \quad (2)$$

where $\mathcal{E}(I)$ is a heuristic function that returns an approximation of $EU(I)$.

Overconfidence and Paranoia

Equations 1 and 2 assume that a_1 knows a_2's strategy ϕ_2, an assumption that is quite unrealistic in practice. A more realistic assumption is that a_1 has a *model* of the opponent that provides an approximation of ϕ_2. For example, in perfect-information games, the well-known minimax formula corresponds to an opponent model in which the opponent always chooses the move whose utility value is lowest. We now consider two opponent models for imperfect-information games.

The first model assumes that a_2 will choose moves uniformly at random from the moves available. This corresponds to using Equations 1 and 2 with $\phi_2(m|I) = 1/|M(I)|$ for every $m \in M(I)$. We call this model the *overconfident* model, because an intelligent opponent is unlikely to play so badly.

The second model requires some modifications to Equations 1 and 2. It assumes that a_2 will always make a move which causes the next information set to be the information set $I' \in C(I)$ of minimum expected utility. This model, which we call *paranoid*, creates this minimax-like formula:

$$\text{Paranoid Utility of } I = PU_d(I) = \begin{cases} \mathcal{E}(I), & \text{if } d = 0, \\ \min_{h \in I} (U(h)), & \text{if } I \text{ is terminal}, \\ \min_{I' \in C(I)} PU_{d-1}(I'), & \text{if it's } a_2\text{'s move}, \\ \max_m \left(\min_{I' \in C(I)} PU_{d-1}(I') \right), & \text{if it's } a_1\text{'s move}. \end{cases} \quad (3)$$

Discussion

The question arises as to which is better: overconfident or paranoid play. The success of minimax in perfect information games might suggest that paranoid play would be the likely winner. Further analysis, however, muddies the issue.

The first important observation to be made is that in imperfect information games, we no longer have complete knowledge of where we are in the game tree. The information set tells us the many places where we *may* be in the game tree. Since each history has a unique information set per player, for every information set, we *must* make the same move for every history in the information set.

Paranoid play assumes that a_2 will make and have always made the worst move possible for a_1, *given a_1's information set*. This means that for any given information set, the paranoid player will find the history in the information set which is least advantageous to itself and make moves as though that were the game's actual history *even when the game's actual history is any other member of the information set*. There is a certain intuitively appealing protectionism occurring here: an opponent that happens to have made the perfect moves cannot trap the paranoid player. However, it really is not clear exactly how well a paranoid player will do in an imperfect information game, for the following reasons:

- There is no reason to necessarily believe that the opponent has made those "perfect" moves. The opponent has different information sets than the paranoid player, which may not give the opponent enough information to make the perfect moves paranoid play expects.
- The paranoid player is losing a lot of potentially winable games against a non-perfect player. The information set could contain thousands of histories in which a particular move is a win; if that move is a loss on just one history, and there is another move which admits no losses, then the first move will not be chosen.
- It is not clear that the opponent the paranoid player assumes is even possible. The opponent must behave the same for every board in the opponent's information set. However, the paranoid player assumes the opponent guesses the best move for the actual current history even if the best move is different for several of the histories in the opponent's information set, and even if this move isn't even possible given the true history of the game.

We should also note the relationship paranoid play has to the "imperfection" of the information in the game. A game with large amounts of information and small information sets should see better play from a paranoid player than a game with the opposite properties. The reason for this is that as we get more information about the actual game state, the more we can be confident that the move the paranoid player designates as "worst" for itself is the worst move the opponent could make in the actual game. The extreme of this is a perfect information game, where paranoid play has proven quite effective instantiated as minimax search. But without some experimentation, it is not clear to what extent smaller amounts of information degrade paranoid play.

Overconfident play, on the other hand, assumes that a_2 will, with equal probability, make all available moves regardless of what the available information tells a_2 about each move's expected utility. The effect this has on game play depends on the extent to which a_2's moves diverge from random play. Unfortunately for overconfidence, many interesting imperfect information games implicitly encourage non-random play. In these games the overconfident player is not adequately considering the risks of its moves. The overconfident player will many times not protect itself from a potential loss under the theory that the opponent is unlikely to make that particular move. These sorts of mistakes are the results of the overconfident assumptions, giving reasonable imperfect information game play an advantage against overconfidence.

But, we need also consider the amount of information in the imperfect information game. In particular, consider a situation where a_1, playing overconfidently, assumes the opponent is equally likely to make each of the ten moves available in a_1's current information set. Suppose that each move is clearly the best move for a reasonable opponent in exactly one tenth of the available histories. If the histories are equally likely, then, despite the fact that the opponent is playing a deterministic strategy, random play is a good opponent model given the information set. This sort of situation, where the model of random play is reasonable despite it being not at all related to the opponent's actual mixed strategy, is more likely to occur in games where there is less information. The larger the information set, the more likely it is that every move has enough histories to make that move as likely to occur as any other. Games which allow their players little information are therefore possibly giving a slight advantage to overconfidence.

Two things come from the above discussion: paranoid play should do better in games with "large" amounts of information, and overconfident play might do better in games with "small" amounts of information. But will overconfident play do better than paranoid play? Suppose we choose a game with small amounts of information and play a paranoid player against an overconfident player: what should the outcome be? Overconfident play has the advantage of probably not diverging as drastically from the theoretically correct expected utility of a move, while paranoid play has the advantage of actually detecting and avoiding bad situations – situations to which the overconfident player will not give adequate weight.

Overall, it is not at all clear from our analysis how well a paranoid player and an overconfident player will do relative to each other in a real imperfect-information game. Instead, experimentation is needed.

Kriegspiel: Our Testbed

For our experimental tests, we use the game of kriegspiel, an imperfect-information version of chess in which the players are unable to see where their opponent's pieces are. We choose kriegspiel because (i) it is clearly a game where each player has only a small amount of information about the current state, and (ii) due to its relationship to chess, it is complicated enough strategically to allow for all sorts of subtle and interesting play.

The specific ruleset we use for kriegspiel comes from (Russell & Wolfe 2005; Parker, Nau, & Subrahmanian 2005). Game play works as follows: a player makes a move, and is then told by a referee what observations the move caused. Observations represent events in the game: for instance, a piece capture elicits a "take" observation. Then the player waits for the opponent to move, after which the player is told by the referee the observations the opponent's move caused. Observations should, for the purposes of this paper, be thought of as defining the set of possible information sets. Any and all valid histories which have the same observations at each move and all the same moves for one of the players are in the same information set.

Illegal moves are specially handled in kriegspiel. For example, if I have a pawn at `e4`, I may try to make the diagonal pawn move `e4-d5`. This move is legal only if it results in a capture. If no capture occurs, the move is illegal and has no effect on the actual board. The referee will tell me that the move is illegal, which will allow me to update my information set by removing all histories in which `e4-d5` would not have given an "illegal" observation. I will not lose my turn for attempting this move: instead, I may try another move. Consequently, I may sometimes want to try moves that are likely to be illegal, in order to gather information that will help me to make other, better moves.

Implementation

Statistical Sampling

Information sets are quite large in kriegspiel, sometimes exceeding 10^{13} (Parker, Nau, & Subrahmanian 2005). We therefore must do something to have even a hope of making information-set search practical. To reduce the game tree's complexity, we approximate it by considering only a randomly chosen subset of the histories available in each information set. For example, suppose it is our turn, and we have an information set of size 10^5. We cannot hope to examine each history in the information set in 30 seconds. We can, however, choose some of the histories at random and do the expected utility calculations on that set of histories. To implement this, we use a technique similar to incrementalization from (Russell & Wolfe 2005). This allows us to sample one history at a time from the information set, folding its expected values into the tree. In this way our algorithm is able to sample as many histories from the information set as it can in the available time. Notice that there is a correlation between the search depth and the sample size: increasing the search depth increases the time it takes to examine one history exponentially, causing the number of histories that an algorithm can sample to be inversely proportional to an exponential of the search depth.

Statistical sampling in the kriegspiel implementation is our algorithms' only source of nondeterminism.

Heuristic Evaluation Functions

We developed a simple evaluation function called `KS` for the leaf nodes of an information-set tree. `KS` was designed for conservative kriegspiel play (our limited experience leads us to believe that conservative play is generally better in kriegspiel). The base board evaluation is the board's material value, with kriegspiel piece values (bishops, for instance, are worth more than knights in kriegspiel). It values protected pieces, pawn advances, and threatened opponent pieces and it punishes mildly for checking one's opponent without causing checkmate (the best way to lose a piece in kriegspiel is to check the opponent with it). We do not claim this to be an ideal evaluation function, and consider the development of a really good evaluation function to be outside

Time / Move		30 sec.			90 sec.		
Player	Depth	w	ℓ	EU	w	ℓ	EU
Overconfident Play	1	48	1	47	47	1	46
	2	50	1	49	55	2	53
	3	32	5	27	43	6	37
Paranoid Play	1	5	2	3	4	4	0
	2	29	3	26	31	1	30
	3	22	4	18	27	4	23

Figure 2: Percentage of wins (w), losses (ℓ), and expected utility (EU) for overconfident and paranoid play vs random play. The 95% confidence interval is at most ± 4.

the scope of this work.

Experiments

We implemented our kriegspiel playing algorithms in C++. In our experiments we tested the overconfident player and the paranoid player versus each other, and also versus two other players: (1) a player called *rand* that chooses its move from the available moves uniformly at random, and (2) an implementation of the HS algorithm described in (Parker, Nau, & Subrahmanian 2005).

We tested the play of both the paranoid and overconfident players at search depths of one, two, and three (the deepest our implementation could search in a reasonable amount of time and memory). We played games out to 160 ply. If the game reached 160 plys or if 50 plys occurred without a irreversible move (a pawn advance or a take), then we terminated the game and considered it a draw. We stopped at 160 ply because in trial runs, most games which ended in a checkmate ended before ply 160.

In order to see how the algorithms' performance would change if they could take larger samples of the histories in their information sets, we ran two sets of experiments: in our "short run" experiments we gave each algorithm 30 seconds per move, and in our "long run" experiments, we gave each algorithm 90 seconds per move. In either run an algorithm was given an extra two seconds after making an illegal move.

We ran these experiments on a cluster of machines, each capable of at least 500,000 KFlops according to Linpack benchmarks. The runs consisted of 500 games, giving each player 250 tries as both white and black. We report percentage wins, percentage losses and the algorithm's expected utility. We calculate the expected utility as wins minus losses divided by games played, multiplied by 100%.

As a benchmark, when $rand$ plays itself, white and black each win about 6% of games played.

Results

The experimental results showed overconfident play doing better than paranoid play in most cases.

In Figure 2, depth-one search against $rand$ does not do better when it is given more time per move. We attribute this to the fact that the statistical significance of a sample increases inversely proportionally to the square root of the sample's size. The 30 second depth-one search already has a statistically significant sample with 1683 boards (figure 4).

Time / Move		30 sec.			90 sec.		
Player	Depth	w	ℓ	EU	w	ℓ	EU
Over-confident Play	1	19	5	14	26	2	24
	2	16	7	8	18	5	12
	3	10	18	-8	13	14	-1
Paranoid Play	1	2	7	-5	1	6	-4
	2	5	5	0	7	4	3
	3	4	14	-10	5	13	-9

Figure 3: Percentage of wins (w), losses (ℓ), and expected utility (EU) for overconfident and paranoid play versus HS. The 95% confidence interval is at most ± 4.

Depth	1	2	3
30 sec. per move	1683	321	67
90 sec. per move	3145	662	190

Figure 4: Average sample size per move by depth for overconfident play versus random play. Values for like-depth paranoid play are similar.

		Overconfident Play					
%		w	ℓ	w	ℓ	w	ℓ
vs	Depth	1		2		3	
Paranoid Play	1	8	1	15	2	12	2
	2	22	4	19	5	16	17
	3	24	4	31	4	16	14

Figure 5: Win / loss percentages for paranoid play vs overconfident play. Moves were made in 30 seconds. All values are reported from overconfident play's perspective. The 95% confidence interval is at most ± 4.

For overconfident searches to depths two and three, however, quality of play against $rand$ increased as time per move increased (see Figure 2). We think this to be a result of the sample size's increasing and causing the results to more likely be statistically significant.

It may then seem counterintuitive that paranoid play's overall performance does not increase either against $rand$ nor against HS (Figures 2 and 3) as the time per move and number of samples increases. But paranoid play is not averaging the values of the samples it takes, and instead is trying to find the worst one. Paranoid play is able to find a sufficiently terrible history in the information set in 30 seconds, and does not need the extra time to find a worse one.

Figure 3 shows depth one overconfident play to do the best against HS. We attribute this to the fact that overconfident play's opponent model does a poor job of modeling HS's play. The deeper the search depth, the more the overconfident backup rule is applied, and the worse the results.

The fact that Figure 3 does not show paranoid play doing well against HS is evidence that in kriegspiel it is not always a good idea to assume the worst. Figure 5, which shows the results of head-to-head play between the overconfident and paranoid players, corroborates this. Of course, this data could mean that neither HS nor overconfident play are very good kriegspiel players and are unable to make the exceptionally good moves paranoid play expects – but that would mean that paranoid play is losing to *bad* kriegspiel play.

The fact that Figure 5 shows depth two overconfident search doing best against paranoid play is possible evidence of one of two things. Either depth three search does not sample enough boards to play effectively or the overconfident opponent model is not reasonable at guessing the moves make by paranoid play.

Related Work

There has been relatively limited work on the game of kriegspiel, with most of it targeting the endgame. Of that work, the most directly related is (Russell & Wolfe 2005), where a special case of information-set search, based on the concept of an AND-OR search is developed to handle the kriegspiel endgame positions with small information sets. The only work we know of that handles unrestricted kriegspiel game play through the entire game is (Parker, Nau, & Subrahmanian 2005). That paper develops a way of directly harnessing the large body of work done on standard chess to create a kriegspiel playing program. Other work on kriegspiel includes an algorithm from (Ciancarini, DalLaLibera, & Maran 1997) for solving endgames using metapositions as a state representation technique. Also, (Sakuta & Iida 2000) have implemented an endgame search strategy for an imperfect-information game similar to kriegspiel.

There is a larger body of work on imperfect-information card games such as bridge and poker. For example, (Billings *et al.* 2003) uses linear programming and abstraction techniques to create a successful poker-playing program. A technique from (Davidson 2002) called miximax uses similar probabilistically weighted averaging in the game of poker. (Frank & Basin 1998) is an analysis of bridge play using assumptions similar to our paranoid model. It shows specific cases in bridge and abstract imperfect-information games where such assumptions go wrong.

Conclusion

We have derived a general recursive formula for game-tree search in imperfect-information games, based on information sets. We have described two different backup rules for use in the formula: an "overconfident" one that backs up the values as if the opponent were making moves at random, and a "paranoid" opponent model that is an information-set analog of the well-known minimax backup rule.

The results of our experimental tests in the game of kriegspiel suggest the following:

- Even without spending much time optimizing the code for the overconfident model and the paranoid model, they both usually outperformed HS. Thus, information-set search might be better for imperfect-information games than HS's approach of pretending the imperfect-information game is a set of perfect-information games.
- In most cases, the overconfident model usually outperformed the paranoid model. This suggests that the usual assumption of perfect- information game tree search—that the opponent will choose the best possible move—isn't as useful in imperfect-information games as in perfect-information games.
- A player's performance depends on how well his/her opponent model models the opponent's behavior. Against the random player, the overconfident model did much better than the paranoid model. Against HS, the overconfident model did not do a lot better than the paranoid model, and the overconfident model's performance degraded as the search depth increased.
- In general, increasing the search depth did not improve an algorithm's performance nearly as much as it does in perfect-information games. In fact, in several cases it made the performance *worse* rather than better. There are two reasons for this. First, if a backup rule doesn't correspond to the opponent's likely behavior, then applying it more times should not necessarily help. Second, increasing the search depth without increasing the available time for the search forces the algorithm to use a smaller statistical sample of boards. If the statistical sample is too small, the search will not produce accurate results.

Acknowledgments. This work was supported by ARO grant DAAD190310202, ARL grants DAAD190320026 and DAAL0197K0135, ARL CTAs on Telecommunications and Advanced Decision Architectures, NSF grants IIS0329851, 0205489, IIS0412812, UC Berkeley subcontract SA451832441 of DARPA's REAL program, and ISLE subcontract 0508268818 of DARPA's Transfer Learning program. The opinions expressed in this paper are authors' and not necessarily opinions of the funders.

References

Billings, D.; Burch, N.; Davidson, A.; Holte, R.; Schaeffer, J.; Schauenberg, T.; and Szafron, D. 2003. Approximating game-theoretic optimal strategies for full-scale poker. In *IJCAI*, 661–668.

Ciancarini, P.; DallaLibera, F.; and Maran, F. 1997. Decision Making under Uncertainty: A Rational Approach to Kriegspiel. In van den Herik, J., and Uiterwijk, J., eds., *Advances in Computer Chess 8*, 277–298.

Davidson, A. 2002. Opponent modeling in poker: Learning and acting in a hostile environment. Master's thesis, University of Alberta.

Frank, I., and Basin, D. A. 1998. Search in games with incomplete information: A case study using bridge card play. *Artificial Intelligence* 100(1-2):87–123.

Osborne, M. J., and Rubinstein, A. 1994. *A Course In Game Theory*. The MIT Press.

Parker, A.; Nau, D.; and Subrahmanian, V. 2005. Game-tree search with combinatorially large belief states. In *IJCAI*, 254–259.

Reif, J. 1984. The complexity of two-player games of incomplete information. *Journal of Computer and Systems Sciences* 29:274–301.

Russell, S., and Wolfe, J. 2005. Efficient belief-state and-or search, with application to kriegspiel. In *IJCAI*, 278–285.

Sakuta, M., and Iida, H. 2000. Solving kriegspiel-like problems: Exploiting a transposition table. *ICCA Journal* 23(4):218–229.