Information-Theoretic Advisors in Invisible Chess.

A.E. Bud, D.W. Albrecht, A.E. Nicholson and **I. Zukerman**
{bud,dwa,annn,ingrid}@csse.monash.edu.au
School of Computer Science and Software Engineering, Monash University
Clayton, Victoria 3800, AUSTRALIA
phone: +61 3 9905-5225 fax: +61 3 9905-5146

Abstract

Making decisions under uncertainty remains a central problem in AI research. Unfortunately, most uncertain real-world problems are so complex that progress in them is extremely difficult. Games model some elements of the real world, and offer a more controlled environment for exploring methods for dealing with uncertainty. Chess and chess-like games have long been used as a strategically complex test-bed for general AI research, and we extend that tradition by introducing an imperfect information variant of chess with some useful properties such as the ability to scale the amount of uncertainty in the game. We discuss the complexity of this game which we call *invisible chess*, and present results outlining the basic game. We motivate and describe the implementation and application of two information-theoretic advisors, and describe our decision-theoretic approach to combining these information-theoretic advisors with a basic strategic advisor. Finally we discuss promising preliminary results that we have obtained with these advisors.

1 Introduction

Games and game theory model some properties of real-world situations and therefore allow analysis and empirical testing of decision-making strategies in these domains. In this capacity, games have long been used as a testbed for general AI research and ideas. In particular, chess has a long history of use because of its strategic complexity, and well-studied and understood properties.

Despite these advantages, chess has two significant drawbacks as a general AI testbed: the first is the success of computer chess players that use hard-coded, domain specific rules and strategies to play; and the second is the fact that standard chess is a perfect information domain. A number of researchers have tackled the first of these drawbacks. For example Berliner [2] investigated generalised strategies used in chess play as a model for problem solving, and Pell [16] introduced *metagamer* — a system for playing games arbitrarily generated from a set of games known as Symmetric Chess-Like games (SCL Games).

In this paper, we address the second drawback of chess as a general AI testbed — that of perfect information. We describe a missing (or imperfect) information variant of standard chess which we call invisible chess [4]. Invisible chess consists of a configurable number of invisible pieces, i.e., pieces that a player's opponent cannot see, and is thus a representative of the general class of strategically complex, imperfect information, two player, zero-sum games.

Many researchers have investigated games with missing information including poker ([7], [14], [13]), bridge ([10], [12], [21]), and multi-user domains [1]. With the exception of Albrecht *et al.* [1], who use a large uncontrollable domain, all of these domains are strategically simple given perfect information. Invisible chess retains the strategic complexity of standard chess with the addition of a controllable element of missing information. Invisible chess is related to kriegspiel ([15] and [5]), a chess variant in which all the opponent's pieces are invisible and a third party referee determines whether or not each move is valid.

The remainder of the paper is organised as follows. Section 2 discusses some related work. Sections 3 and 4 outline the game and our basic approach to invisible chess. Section 5 presents a brief analysis of play with various invisible pieces. Interestingly, we show that the relative values of minor invisible pieces differs from standard chess piece rankings. Section 6 discusses the relationship between uncertainty and the information-theoretic concept of entropy, and the application of entropy to our results. Section 6 also motivates the design of information-theoretic advisors, describes these advisors and presents preliminary results that demonstrate the efficacy of our approach. Finally, Section 7 contains conclusions and ideas for further work.

2 Related Work

Since the 1950s, when Shannon and Turing designed the first chess playing programs [19], computers have become better at playing certain games such as chess using large amounts of hand-coded domain specific information. In answer to the success of hard-coded algorithms, Pell [16] introduced a class of games known as Symmetric Chess Like (SCL) Games, and a system called Metagamer that plays games arbitrarily generated from this class using a set of advisors representing strategies in the class of games. Pell did not consider imperfect information games. Invisible chess is one game in the extension of Pell's class of SCL Games to the class of Invisible SCL Games, where one or more of a player's pieces is hidden from their opponent.

Koller *et al.* [13] investigated simple imperfect information games with a goal of solving them. However, their approach does not scale up to complex games such as invisible chess.

Smith *et al.* [21] wrote a bridge playing program using a modified form of game tree with enumerated strategies rather than actions, effecting a form of forward pruning of the game tree. Smith *et al.* stated that forward pruning works well for bridge, but not for chess. Ginsberg [11] introduced partition search to reduce the effective size of the game tree. He showed that this approach works well for bridge and other games with a high degree of symmetry. In bridge, the order in which the first n cards are played is irrelevant if the same cards remain in each player's hands. However, in chess, partly because pawns only move forward, and partly because the order of moves in invisible chess changes the flow of the game, the same positions do not tend to occur in a significant number of nodes in the game tree, and therefore partition search is not likely to be very effective on chess variants. More recently Ginsberg [12] included Monte Carlo methods to simulate many possible outcomes, choosing the action with the highest expected utility over the simulations. Ginsberg claims that trying to glean or hide information from an opponent is probably not useful for bridge. In contrast, we present results that show that both information hiding and gleaning can be useful in invisible chess.

Frank and Basin ([9] and [8]) have performed a detailed investigation of search in imperfect information games, concentrating almost exclusively on bridge. They focus on a number of search techniques including flattening the game tree which they have shown to be an NP-Complete problem, and Monte Carlo methods. Their investigation suggests that Monte Carlo methods are not appropriate for imperfect information games. This problem is exacerbated in invisible chess given the relative lack of symmetry and the size of the game tree.

Ciancarini *et al.* [5] considered king and pawn end games in the game of kriegspiel — an existing chess variant where neither player can see the opponent's pieces or moves. They used a game-theoretic approach involving substantive rationality and have shown promising results in this trivial version of kriegspiel. Thus far they have not applied their approach to a complete game of kriegspiel. Kriegspiel differs from invisible chess in a number of important areas. All the opponent's pieces are invisible to a player of kriegspiel. Thus there is no way to reduce the uncertainty in the domain without reducing the strategic complexity. By contrast, in invisible chess, the number and types of invisible pieces is configurable. Additionally, in kriegspiel, a player attempts moves until a legal move is performed; a third party arbitrator indicates whether or not each move is legal. In invisible chess, the player is given specific information when a move is impossible or illegal, and may miss a turn as a result of attempting an impossible move (see Section 3). These differences make kriegspiel a substantially more complex and less controllable domain than invisible chess. Specifically, exploring the relationship between uncertainty and strategic play would be more difficult in kriegspiel. Ultimately, the pathological case of invisible chess where all pieces are invisible approaches the complexity of kriegspiel. Thus invisible chess provides a convenient stepping stone to a much more difficult problem.[1]

[1]To our knowledge there exist no computer kriegspiel players that are able to play a full game of kriegspiel with which to compare our system.

3 Domain: Invisible Chess

Invisible chess is based on standard chess with the following difference. In invisible chess we differentiate between visible and invisible pieces and define them as follows: a *visible piece* is one that both players can see; an *invisible piece* is one that a player's opponent cannot see. Thus, every time a visible piece is moved, the board is updated as per standard chess. However, when a player moves an invisible piece, their opponent is informed which invisible piece has moved, but not where it has moved from or to. This information enables each player to maintain a probability distribution of their opponent's invisible pieces across the squares on the board. In this paper, we frame invisible chess pieces () to distinguish them from visible chess pieces ().

We define three terms for referring to moves in invisible chess: A **possible move** is any legal chess move given complete information about the board. That is, no invisible pieces are in the way of the move, and the piece can move to the desired position. An **impossible move** is an attempted move that violates the laws of chess because of an incorrect assumption as to the whereabouts of an invisible piece. For example, a player attempts to move a piece "through" an invisible piece, or attempts to use a pawn to capture an invisible piece that is not diagonally in front of it. See Figure 1. An **illegal move** is an impossible move that would not allow the game to continue. For example, a player is not allowed to move their king into check by an invisible piece. See Figure 2.

The rules of invisible chess are based on the rules of chess. The only modifications pertain directly to invisible pieces and their impact on the game. In general if a move is possible then it is accepted; if a move is impossible then it is rejected and the player's turn is forfeited; and if a move is illegal, some information regarding the reason the move is illegal is revealed to the player attempting the move, and the player must supply another move. Note that we do not allow invisible kings in this basic invisible chess, as a version of invisible chess with invisible kings would drastically modify the goal of the original game. Bud *et al.* [3] gives details of all scenarios where new rules come into effect.

3.1 Domain Complexity

The complexity of standard chess is well understood. Chess has an average branching factor of around 35. A typical game lasts approximately 50 moves per player so the entire game tree has approximately 35^{100} nodes ([19], p 123).

In the trivial case where the opponent has no invisible pieces, the game tree for invisible chess is exactly as large as that for standard chess. Once invisible pieces are introduced, each node must be expanded for each possible move for each possible combination of positions of the opponent's invisible pieces. In a game of invisible chess the player and opponent each have m invisible pieces. If each invisible piece IP_i has a positive probability of occupying n_i squares, then the branching factor is approximately the branching factor of chess multiplied by the combination of invisible squares that could be occupied, i.e., $Branching\ Factor = 35 \times n_1 \times n_2 \times ... \times n_m$

If each player has two invisible pieces, and each of those pieces has an average of four squares with a positive prob-

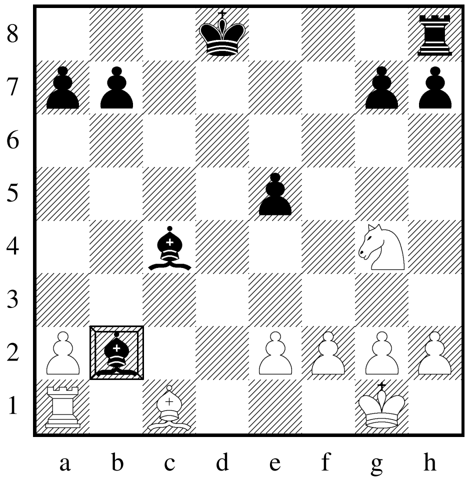

Figure 1: *An impossible move.* The black bishop on b2 is invisible. White attempts an impossible move by trying to move the bishop on c1 to a3.

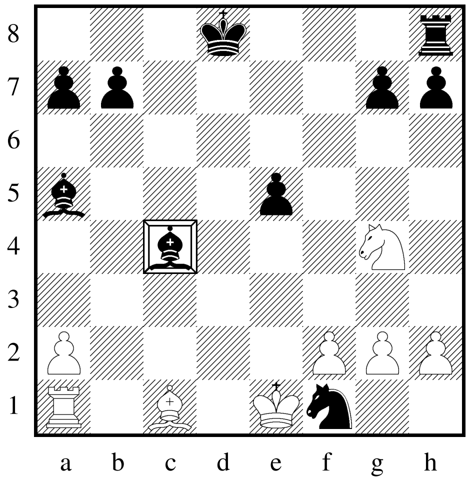

Figure 2: *An illegal move.* White is in check from the black bishop on a5, and attempts to capture the invisible black knight on f1 where it would be in check from the invisible black bishop on c4.

ability of occupation, then the average approximate branching factor of that game of invisible chess is $35 \times 16 = 560$. For three invisible pieces each, the branching factor is around $35 \times 64 = 2240$. Assuming that the invisible pieces are on the board and moving for approximately half the game, then the complete expanded game tree for three invisible pieces each is in the order of 2240^{50} (10^{150}) nodes.

To make the domain even more complex, chess has virtually no axes of symmetry that allow the size of the game tree to be reduced as in highly symmetrical games such as tic-tac-toe, and Smith and Nau [20] claim that forward pruning to reduce the branching factor of chess has been shown to be relatively ineffective.

In addition to dealing with the combinatorially expansive nature of the invisible chess game tree, a player must maintain beliefs about the positions of the opponent's invisible pieces. A player that uses any belief updating scheme that does not take into account every possible destination square when an invisible piece is moved, will lose important information about the flow of the game.

Assuming no strategic knowledge (i.e., each square that an invisible piece can move to is as likely to be visited as any other), and only one invisible piece, a player can easily maintain the precise distribution for that piece (see [3]). However, for multiple invisible pieces, the positions of invisible pieces are not conditionally independent with respect to each other. The probability of one invisible piece occupying any particular square affects the probability of another invisible piece occupying that square. Maintaining the probability distributions of multiple invisible pieces involves combinatorial calculations in the number of invisible pieces or the storage of combinatorially large amounts of data to maintain the complete joint distributions of all invisible pieces over the entire chess board.

4 Basic Design

To cope with the combinatorial explosion, and the strategic complexity of invisible chess, we employ a "divide and conquer" approach. We split the problem of choosing the next move into a number of simpler sub-problems, and then use utility theory [17] to recombine the calculations performed for these sub-problems into a move. We use information-theoretic ideas [18] to deal with the uncertainty in the domain, and standard chess reasoning to deal with the strategic elements.

This modular, hybrid approach is implemented with advisors or experts connected and controlled by a Game Controller and Distribution Maintenance Module (GCDMM), and a Maximiser. The GCDMM controls the game state, has knowledge of the positions of all invisible pieces and maintains distributions of invisible pieces on behalf of the two players. The GCDMM is responsible for deciding whether a move is legal, impossible or illegal and ensuring that the game progresses correctly. When it is a player's turn to move, the GCDMM requests the next move from the Maximiser for that player. The Maximiser, responsible for choosing the best move suggested by the available advisors, generates all possible boards and all possible moves, and requests utility values from each of the advisors for each move.

Each advisor evaluates the possible moves, across as many boards as possible in the available time, according to an internal evaluation function. The *strategic advisor*, a modified version of GNU Chess, returns all possible moves and their Expected Utilities (EU) from a strategic perspective.[2] The EU for each move or action (A) is calculated by multiplying its utility value by the probability of the game state (X) for which the utility value was calculated, and summing this across all possible game states (G) as follows.

$$EU(A) = \sum_{\forall X \in G} (Utility(A|X) \times Prob(X)) \tag{1}$$

The Maximiser multiplies the value returned from each of the

[2] Note that GNU Chess has no knowledge of invisible pieces. The modifications to GNU Chess are to perform fixed depth searching, and to return all moves and utilities rather than only the best move.

	1 B	1 N	1 R	1 Q	2 B	2 N	2 R	N.I.
1 B	X	61.0	65.0	35.0	17.2	41.5	43.6	89.8
1 N	39.0	X	53.4	39.2	13.2	37.4	27.6	89.0
1 R	35.0	46.6	X	38.2	25.4	34.4	27.4	93.0
1 Q	65.0	60.8	61.8	X	43.8	41.4	47.6	93.0
2 B	82.8	86.8	74.6	56.2	X	72.0	57.6	95.6
2 N	58.5	62.6	65.6	58.6	28.0	X	45.8	96.2
2 R	56.4	72.4	72.6	52.4	42.4	54.2	X	98.2
N.I.	10.2	11.0	7.0	7.0	4.4	3.8	1.8	X

Table 1: Win percentages for different combinations of invisible pieces.

advisors by the advisor weight and sums these values. The move with the highest overall utility value is passed to the GCDMM which implements that move and requests a move from the other player.

5 Invisible Chess with a Strategic Advisor

This section presents results where each player played invisible chess with a different configuration of invisible pieces, and used only a strategic advisor to decide the next move. Each result was obtained from a run of 500 games[3] with a particular configuration of invisible pieces. As white has a slight advantage in standard chess, to remove colour biases from the results, each set of 500 games was broken into two runs of 250 games each, the invisible piece configurations were swapped between white and black for each run, and the results were averaged between the two runs.

Table 1 shows results for games played with major (non-pawn) invisible pieces against each other and against no invisible pieces (N.I.). Reading across a row, it shows the percentage of games won by the combination of invisible pieces in the row heading against the invisible piece combination of a particular column. For example, one invisible bishop (1B) won 65% of the time against one invisible rook (1R), but only 43.6% of the time against two invisible rooks (2R).

These results show that the values of several invisible piece combinations differ from standard chess. For example, in standard chess, a rook is considered more valuable than a bishop. However, in invisible chess, one invisible bishop beat one invisible rook, and two invisible bishops beat every other combination of invisible pieces considered including invisible rooks. This apparent anomaly is due to the bishop's early involvement in the game. By causing uncertainty early in the game, a player with two invisible bishops has an early advantage against a player with two invisible rooks. Further analysis of these results is presented in [3].

6 Building Information Theoretic Advisors

A reasonable inference from the results above is that players with more information about the game tend to win more often; i.e., the closer a player's belief about a board position is to the true board position, the more likely the player is to play well strategically. In this section, we describe our use of information theory to quantify a player's uncertainty about the positions of an opponent's invisible pieces (Section 6.1), present two information-theoretic advisors (Section 6.2), and

[3]That ended in a win. Drawn games are discarded. They are largely the result of repeating moves continuously, and consist of less than 10% of games played.

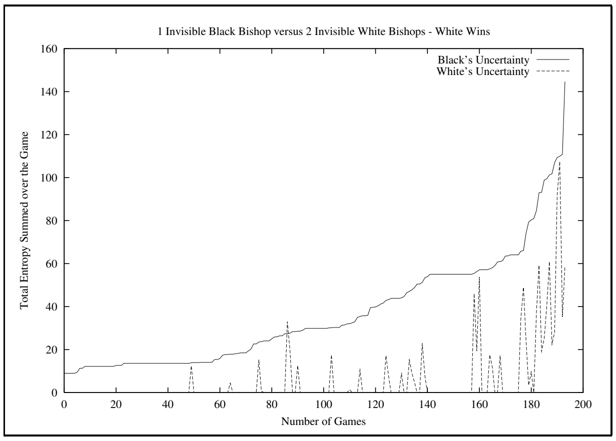

Figure 3: Uncertainty (as entropy) in games with one invisible black bishop against two invisible white bishops that were won by white.

discuss some preliminary results obtained with these advisors (Section 6.3).

6.1 Uncertainty and Entropy

Any probability distribution can be said to have an *entropy* or information measure associated with it. This entropy measure is bounded from below by zero, when there is only one possibility and therefore no uncertainty, and increases as the distribution spreads.

In invisible chess, given a probability distribution of the positions of the opponent's invisible pieces, it is possible to derive a probability distribution of all possible board positions. Thus we can calculate the entropy (H) of a set of board states together with their associated probabilities as follows:

$$H = -\sum_{\forall X \in G} Prob(X) \times \log_2(Prob(X)) \quad (2)$$

where X represents a single game state (one possible combination of positions of the opponent's invisible pieces), from the set of possible game states (G), and $Prob(X)$ represents the probability of that game state. As invisible pieces move, the number of squares they may occupy increases. This leads to an increase in the number of possible board states, and the entropy of the distribution of those board states, i.e., it increases the opponent's uncertainty.

Figure 3 shows the entropy in games that were won by white playing with two invisible bishops against black with one invisible bishop. The solid line shows blacks total entropy (the entropy of the distribution of the positions of white's invisible pieces) summed over the course of the game, sorted by entropy. The dashed line shows white's total entropy over each corresponding game. Of the 190 games won by white, only one game (number 86) shows greater entropy for black. This is not the case for games that were won by black. This example corroborates our intuition that more certain players are more likely to win, and underpins our information-theoretic advisors.

6.2 Information-Theoretic Advisors

Move Hide Advisor. Working on the premise that the more uncertain the opponent is, the worse they will play, the move hide advisor advises a player to perform moves that hide information from the opponent. That is, each move is scored

Weight	Hide		Seek	
	Q vs Q[4]	B vs B	Q vs Q[4]	B vs B
0.5	57.2	78.4	50.1	51.2
5.0	61.3	82.7	55.0	51.2
50.0	13.6	18.5	47.6	55.2

Table 2: The effect of the move hide and move seek advisors.

according to its expected effect on the entropy of the distribution of the positions of the player's invisible pieces from the opponent's perspective. This effect may be an increase, a decrease or no change in the entropy. Moves by invisible pieces that do not cause check result in an increase in entropy. Moves that cause check or moves by visible pieces may cause a decrease in entropy by revealing vacant squares.

To calculate this entropy, the move hide advisor needs to model the opponent's distribution update strategy. For a proposed move, the move hide advisor uses Equation 2 to calculate the entropy of the opponent model of the distribution of the player's invisible pieces prior to and following the move. The exponential of the perceived change in entropy is returned as the move hide utility. The exponential is taken to allow a comparison between the log based utilities returned by the move hide and move seek advisors, and the utility returned by the strategic advisor which scores moves on a linear scale.

Move Seek Advisor. The move seek advisor suggests that a player perform moves that are more likely to discover information about the positions of the opponent's invisible pieces. That is, each move is scored according to the expected decrease in the entropy of the opponent's invisible pieces following the move. This expected decrease in entropy must be greater than or equal to zero as no move can make a player less certain about the position of the opponent's invisible pieces.

A move that covers a large number of squares with a positive probability of occupation by the opponent's invisible pieces will yield a certain amount of information whether it is successful or not. Clearly this type of move will yield more information if it is successful, as the player now knows that all of those squares are vacant. On failure, a player can only conclude that at least one of those squares is occupied. On the other hand, a move that traverses only one square with a positive probability of occupation by the opponent's invisible pieces will yield more information if unsuccessful. That is, the player now knows that an invisible piece is on that square. The move seek advisor multiplies the projected decrease in entropy for each outcome by the probability of that outcome to get an expected utility. The exponential of this expected change in entropy is returned as the move seek utility.

6.3 Advisor Results

This section shows the individual effects of the move hide and move seek advisors with varying weights. Each result was obtained by playing 500 games separated into runs of 250 games as before, with one player using the strategic advisor only, against an opponent using one of the information-theoretic advisors and the strategic advisor.

Table 2 shows the results of adding the move hide or move seek advisor to an invisible queen each (Q vs Q) and an invisible bishop each (B vs B). The first column headed "Weight" shows the weight of the information-theoretic advisor relative to the strategic advisor. Thus, with a weight of 0.5, in games played with an invisible queen each, the player playing with the move hide advisor won 57.2% of the games.

Move Hide Results. The move hide advisor proved to be a significant advantage. In fact the win percentage with a weight of 5.0 for the invisible bishop games was 82.7% which is approaching the win percentage of an invisible bishop against no invisible pieces (89.8%).

This improvement is due to the increased number of invisible piece moves. With a small increase in the value of moves of invisible pieces, strategically reasonable invisible piece moves are greatly encouraged. This tends to maximise the opponent's uncertainty. However, as the weight of the move hide advisor increases past 5.0, the percentage of wins decreases. This behaviour is typical of all observed move hide runs, and results from the player performing highly valued move hide moves at the expense of strategy. Although the opponent may be slightly more uncertain when the move hide advisor is weighted highly, the player is making enough strategically poor moves to counter that advantage.

Move Seek Results. Table 2 (columns 5 & 6) shows the move seek advisor's effectiveness against an opponent's invisible pieces. With the appropriate weight, the move seek advisor does increase the player's win percentage by around 5%. However, for weights greater than or smaller than these values, the move seek advisor is either ineffective or detrimental.

Moves that traverse squares with a positive probability of occupation by the opponent's invisible pieces are valued by the move seek advisor. These moves may not correspond to good strategic moves. Thus, the higher the move seek advisor's weight, the fewer good strategic moves will be performed. The difference between the bishop-seeking and queen-seeking behaviour is related to the difficulty of finding the opponent's invisible piece. An invisible bishop can have a positive probability of occupying at most half the available squares, while an invisible queen can have a positive probability of occupying all available squares. Thus, the advice provided by the move seek advisor often aids in the early capture of the invisible bishop, thereby removing the uncertainty from the game. In contrast, following this advice when the invisible piece is an invisible queen leads to non-strategic moves. Thus the move seek advisor assists when the uncertainty in the game is low, but may be useless or detrimental when the uncertainty is high.

7 Conclusion and Future Work

The results presented in this paper are preliminary and further exploration of the domain is required. Specifically, more accurate prediction of the likely positioning of the opponent's invisible pieces is needed. This prediction could take the form of using strategic information about the likely destination of a moving invisible piece, or involve evaluating the complete search tree for more ply. Improving this distribu-

[4] The results in these column were generated with an old version of the program which did not calculate the value of impossible moves correctly. However, the partial results that have been generated for these columns with the new program followed the same trends as the results presented here.

tion would reduce its entropy and therefore improve strategic performance. A side effect of this type of distribution improvement is the possibility of incorporating bluffing into invisible chess, i.e., moving an invisible piece to an unlikely position to confuse an opponent.

Modelling the uncertainty regarding a player's pieces from the opponent's perspective for more ply would improve strategic performance. However, the problem of the combinatorial expansion in the search required as a result of this prediction needs to be resolved. Obtaining results in this domain is already a time consuming exercise and the addition of an extra layer of search would make rapid game play impractical. The most obvious way to manage this explosion is to find effective ways to prune the game tree. However, as mentioned earlier, this is not likely to be simple.

One significant difficulty of this type of approach is the problem of how to choose the relative advisor weights. It is likely that play may be improved by using statistical inference techniques such as those described in [6] to determine static weights for the advisors. However, our results show that the advisors have different levels of effectiveness in different situations. Thus if a player can infer what type of board position the current situation is, they can apply an appropriate weight vector to their advisor array. The type of current board situation may be based on the amount of entropy and the pieces that are causing it, the game move number, and other positional information. We intend to use statistical inference techniques to explore this possibility.

In summary, we have presented and discussed a complex, but controlled domain for exploring automated reasoning in an uncertain environment with a high degree of strategic complexity. We have motivated and introduced the use of information-theoretic advisors in the strategically complex imperfect information domain of invisible chess. We have shown that our distributed-advisor approach using a combination of information-theoretic and strategic aspects of the domain lead to performance advantages compared to using strategic expertise alone. Given the simplicity and generality of this approach, our results point towards the potential applicability to a range of strategically complex imperfect information domains.

Although the basic idea of information-theoretic advisors is intuitive, their application to this domain is not necessarily straightforward. There are complex multi-level interactions between the strategic advisor and the information-theoretic advisors. The balancing act required to maximise a player's performance almost certainly involves dynamically modifying the weights associated with the various advisors depending on both the invisible piece configuration and the game position. However, the preliminary results presented in this paper indicate that this is a domain worth exploring further.

Acknowledgements

The authors would like to thank Chris Wallace for his insight, advice and patience.

References

[1] D W Albrecht, I Zukerman, A E Nicholson, and A Bud. Towards a bayesian model for keyhole plan recognition in large domains. In *User Modelling. Proc. of the 6th Int. Conf. UM97*, number 383 in CISM Courses and Lectures, pages 365–376. Springer Wien, New York NY USA, 1997.

[2] Hans J. Berliner. *Chess as Problem Solving*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1974.

[3] A. Bud, I. Zukerman, D. Albrecht, and A. Nicholson. Invisible chess: Rules, complexity and implementation. Technical Report forthcoming, School of Computer Science and Software Engineering, Monash University, 2000.

[4] A.E. Bud, A.E. Nicholson, I. Zukerman, and D.W. Albrecht. A hybrid architecture for strategically complex imperfect information games. In *Proc. of the 3rd Int. Conf. on Knowledge-Based Intelligent Information Engineering Systems.*, pages 42–45. IEEE Press, 1999.

[5] P. Ciancarini, F. DallaLibera, and F. Maran. Decision Making under Uncertainty: A Rational Approach to Kriegspiel. In J. van den Herik and J. Uiterwijk, editors, *Advances in Computer Chess 8*, pages 277–298. Univ. of Rulimburg, 1997.

[6] G Farr and D Powell. Unsupervised learning in metagame. In N Foo, editor, *Proc. of the 12th Australian Joint Conference on AI (AI'99)*, number 1747 in Lecture Notes in AI, subseries of Lecture Notes in Computer Science, pages 24–35. Springer-Verlag, December 1999.

[7] Nicholas V. Findler. Studies in machine cognition using the game of poker. *Communications of the ACM*, 20(4):230–245, April 1977.

[8] I. Frank and D. Basin. A theoretical and empirical investigation of search in imperfect information games. In *Theoretical Computer Science — To appear.*, 2000.

[9] Ian Frank and David Basin. Search in games with incomplete information: A case study using bridge card play. *Artificial Intelligence*, 100:87–123, April 1998.

[10] Björn Gambäck, Manny Rayner, and Barney Pell. An architecture for a sophisticated mechanical Bridge player. In David N. Levy and Don F. Beal, editors, *Heuristic Programming in Artificial Intelligence — The Second Computer Olympiad 2*, pages 87–107, Chichester, England, 1991. Ellis Horwood.

[11] Matt Ginsberg. Partition search. In *Proc. of the 13th National Conf. on Artificial Intelligence*, pages 228–233. AAAI Press / MIT Press, 1996.

[12] Matt Ginsberg. GIB:steps toward an expert-level bridge-playing program. In *Proc. of the 16th Int. Joint Conf. on AI*, pages 584–589. Morgan Kaufmann, 1999.

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

[14] K B Korb, A E Nicholson, and N Jitnah. Bayesian poker. In K B Laskey and H Prade, editors, *Proc. of the 15th Conf. on Uncertainty in AI*, pages 343–350, Stockholm, Sweden, August 1999. Morgan Kaufmann.

[15] D Li. *Kriegspiel: Chess Under Uncertainty*. Premier Publishing, 1994.

[16] Barney Pell. *Strategy Generation and Evaluation for Meta-Game Playing*. PhD thesis, University of Cambridge, 1993.

[17] Howard Raiffa. *Decision analysis: introductory lectures on choices under uncertainty*. Addison-Wesley, 1968.

[18] Alfréd Rényi. *A diary on information theory*. John Wiley, 1984.

[19] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[20] S. J. J. Smith and D. S. Nau. Strategic planning for imperfect-information games. In *Games: Planning and Learning, Papers from the 1993 Fall Symposium*, pages 84–91. AAAI Press, 1993.

[21] S. J. J. Smith, D. S. Nau, and T. A. Throop. Total-order multi-agent task-network planning for contract bridge. In *Proc. of the 13th National Conf. on AI*, pages 108–113. AAAI Press / MIT Press, 1996.