GIB: Steps Toward an Expert-Level Bridge-Playing Program

Matthew L. Ginsberg
CIRL
1269 University of Oregon
Eugene, OR 97403-1269
ginsberg@cirl.uoregon.edu

Abstract

This paper describes GIB, the first bridge-playing program to approach the level of a human expert. (GIB finished twelfth in a hand-picked field of thirty-four experts at an invitational event at the 1998 World Bridge Championships.) We give a basic overview of the algorithms used, describe their strengths and weaknesses, and present the results of experiments comparing GIB to both human opponents and other programs.

1 Introduction

Of all the classic games of mental skill, only card games and Go have yet to see the appearance of serious computer challengers. In Go, this appears to be because the game is fundamentally one of pattern recognition as opposed to search; the brute-force techniques that have been so successful in the development of chess-playing programs have failed almost utterly to deal with Go's huge branching factor. Indeed, the arguably strongest Go program in the world (Handtalk) was beaten by 1-dan Janice Kim (winner of the 1984 Fuji Women's Championship) in the 1997 Hall of Champions after Kim had given the program a monumental 25 stone handicap.

Card games appear to be different. Perhaps because they are games of imperfect information, or perhaps for other reasons, existing poker and bridge programs are extremely weak. World poker champion Howard Lederer (Texas Hold'em, 1996) has said that he would expect to beat any existing poker program after five minutes[1] play.*[1] Perennial world bridge champion Bob Hamman, six-time winner of the Bermuda Bowl, once summarized all of the commercial bridge programs by saying that, "They would have to improve to be hopeless."+

In poker, there is reason for optimism: the GALA system [Koller and Pfeffer, 1995], if applicable, promises to produce a computer player of unprecedented strength by reducing the poker "problem" to a large linear optimization problem which is then solved to generate a strategy that is nearly optimal in a game theoretic sense. Schaeffer, author of the world champion checkers program CHINOOK [Schaeffer, 1997], is also reporting significant success in this domain [Billings *et* al., 1998].

The situation in bridge has been bleaker. In addition, because the American Contract Bridge League (ACBL) does not rank the bulk of its players in meaningful ways, it is difficult to compare the strengths of competing programs or players.

In general, performance at bridge is measured by playing the same deal twice or more, with the cards held by one pair of players being given to another pair during the replay and the results then being compared.[2] A "team" in a bridge match thus typically consists of two pairs, with one pair playing the North/South (N/S) cards at one table and the other pair playing the E/W cards at the other table. The results obtained by the two pairs are added; if the sum is positive, the team wins this particular deal and if negative, they lose it.

In general, the numeric sum of the results obtained by the two pairs is converted to International Match Points, or IMPS. The purpose of the conversion is to diminish the impact of single deals on the total, lest an abnormal result on one particular deal have an unduly large impact on the result of an entire match.

Jeff Goldsmith+ reports that the standard deviation on a single deal in bridge is about 5.5 IMPS, so that if two roughly equal pairs were to play the deal, it would not be surprising if one team beat the other by about this amount. It also appears that the difference between an average club player and a world class expert is about 2 IMPS (per deal played). The strongest bridge playing programs thus far appear to be slightly weaker than average club players.

Progress in computer bridge has been slow. A recent incorporation of planning techniques into Bridge Baron,

1Many of the citations here are the results of personal communications. Such communications are indicated simply by the presence of a + in the accompanying text.

[2]Space restrictions prevent my describing the rules of bridge. Descriptions can be found in other AI papers dealing with bridge, and there are many excellent texts available [Sheinwold, 1996]. Articles on chess-playing programs never describe the rules; hopefully bridge will be treated similarly as it becomes a more regular topic for AI research.

for example, appears to have led to a performance increment of approximately 1/3 IMP per deal [Smith *et* al., 1996]. This modest improvement still leaves Bridge Baron far shy of expert-level (or even good amateur-level) performance.

Existing programs have attempted to duplicate human bridge-playing methodology in that their goal has been to recognize the class into which any particular deal falls: finesse, end play, squeeze, etc. Smith et.al.'s work uses planning to extend this approach, but the plans continue to be constructed from human bridge techniques. In retrospect, perhaps we should have expected this approach to have limited success; certainly chess-playing programs that have attempted to mimic human methodology, such as PARADISE [Wilkins, 1980], have fared poorly.

GIB works differently. Instead of modeling its play on techniques used by humans, GIB uses brute-force search to analyze the situation in which it finds itself. Monte Carlo techniques are then used to suggest plays by combining the results of analyzing instances of bridge's perfect-information variant. This approach appears to have been first suggested by Levy [Levy, 1989].

Card play is only half of bridge; there is bidding as well. It is possible to use search-based techniques here also, although there is no escaping the fact that a large database of bids and their meanings is needed by the program. (Bidding is, after all, a communicative process; the meanings of the bids need to be agreed upon.) GIB'S success here has been more modest; the overall approach is promising but is, for technical reasons that we will describe, unusually vulnerable to gaps or other inaccuracies in the bidding database itself.

GIB currently seems to be about halfway between Bridge Baron and world class, beating Bridge Baron by something over 2 IMPS per deal played and losing to strong human players by approximately half that. Unlike previous programs, however, it it still improving rapidly; there are many straightforward additions that are likely to enhance its performance substantially.

The outline of this paper is as follows: We begin in the next section by describing a Monte Carlo approach to card play, outlining its strengths and weaknesses, and providing details on its performance. Section 3 describes the use of a similar approach to bidding, explaining why it is so vulnerable to database errors and describing several possible ways around this vulnerability. We end with a summary of the GIB project, including details on its overall performance and suggestions for future work.

2 Card play

In order to understand the card play phase of a bridge deal, consider first bridge's perfect information variant, the game where all of the players are playing "double dummy" in that they can see which cards the other players hold. In this case, the game tree is a fairly straightforward minimax tree, although there are some minimizing nodes with minimizing children, since the player playing last to one trick may well play first to the next. The raw branching factor of the tree appears to be about four; alpha-beta pruning and the introduction of a transposition table bring it down to about 1.7. Augmenting the move ordering heuristic to exploit narrowness[3] reduces the branching factor further to approximately 1.3, corresponding to a search space of some 10^6 nodes per deal. The introduction of partition search [Ginsberg, 1996] and the killer heuristic reduce the space further to some 18,000 nodes per deal.

One way in which we might now proceed in a realistic situation would be to deal the unseen cards at random, biasing the deal so that it was consistent both with the bidding and with the cards played thus far. We could then analyze the resulting deal double dummy and decide which of our possible plays was the strongest. Averaging over a large number of such Monte Carlo samples is one possible way of dealing with the imperfect nature of bridge information.

Algorithm 1 (Monte Carlo card selection) *To select a move from a candidate set M of such moves:*

1. *Construct a set D of deals consistent with both the bidding and play of the deal thus far.*
2. *For each move $m \in M$ and each deal $d \in D$, evaluate the double dummy result of making the move m in the deal d. Denote the score obtained by making this move s(m,d).*
3. *Return that m for which $\sum_d s(m,d)$ is maximal*

The Monte Carlo approach has drawbacks that have been pointed out by a variety of authors, including Roller+ and others [Frank and Basin, 1998]. Most obvious among these is that the approach never suggests making an "information gathering play." After all, the perfect-information variant on which the decision is based invariably assumes that the information will be available by the time the next decision must be made! Instead, the tendency is for the approach to simply defer important decisions; in many situations this may lead to information gathering inadvertently, but the amount of information acquired will generally be far less than other approaches might provide. In spite of this, GIB'S card play is at the level of a human expert.

Performance was measured initially using *Bridge Master* (BM), a commercial program developed by Gitelman. BM contains 180 deals at 5 levels of difficulty. Each of the 36 deals on each level is a problem in declarer play. If you misplay the hand, BM moves the defenders' cards around if necessary to ensure your defeat.

BM was used for the test instead of randomly dealt deals because the signal to noise ratio is far higher; good plays are generally rewarded and bad ones punished. Every deal also contains a lesson of some kind; there are

[3] The narrowness heuristic suggests placing early in the move ordering those moves to which the opponents have few legal responses, thereby keeping the size of the game tree small. This heuristic is apparently well known in the chess community but is poorly cited in the academic literature. A recent paper [Plaat *et* al., 1996] suggests that the idea is rooted in that of conspiracy search [McAllester, 1988].

no completely uninteresting deals where the line of play is irrelevant or obvious. There are drawbacks to testing GIB'S performance on nonrandomly dealt deals, of course, since the BM deals may in some way not be representative of the problems a bridge player would actually encounter at the table.

The test was run under Microsoft Windows on a 200 MHz Pentium Pro. As a benchmark, Bridge Baron (BB) version 6 was also tested on the same deals using the same hardware.[4] BB was given 10 seconds to select each play, and GIB was given 90 seconds to play the entire deal with a Monte Carlo sample size of 50.[5] New deals were generated each time a play decision needed to be made.

These numbers approximately equalized the computational resources used by the two programs; BB could in theory take 260 seconds per deal (ten seconds on each of 26 plays), but in practice took substantially less. GIB was given the auctions as well; there was no facility for doing this in BB. This information was critical on a small number of deals. (The *auction* is the sequence of bids made by the players.)

Here is how the two systems performed:

Level	BB	GIB
1	16	31
2	8	23
3	2	12
4	1	21
5	4	13
Total	33	100
	18.3%	55.6%

Each entry is the number of deals that were played successfully by the program in question.

GIB'S mistakes are illuminating. While some of them are of the sort that have already been mentioned (failing to gather information), most are quite different.

GIB is very good (nearly optimal, in fact) at identifying specific possibilities that will allow a contract to be made or defeated, since such possibilities are overlooked only if they don't appear in the Monte Carlo sample being used. What it is weak at is *combining* such possibilities. As an example, suppose that you are playing a hand and you can take one of four possible lines. Each of the first two banks on a specific (but different) distribution of the opposing cards. The third line simply defers the guess by doing something random, and the fourth line is a clever one that succeeds in either of the first two cases, independent of which actually transpires.

GIB chooses randomly between the third and fourth possibilities in this situation, assuming that if it can defer the guess, it will make it correctly in the future! (And on a double dummy basis, it would.) This pattern accounts for virtually all of GIB'S mistakes; as BM's deals get more difficult, they more often involve combining a variety of possibly winning options and that is why GIB's performance falls off at levels 2 and 3.

At still higher levels, however, BM typically involves the successful development of complex end positions, and GIB'S performance rebounds. This appeared to happen to BB as well, although to a much lesser extent. It was gratifying to see GIB discover for itself the complex end positions around which the BM deals are designed, and more gratifying still to witness GIB'S recent discovery of a maneuver that had hitherto not been identified in the bridge literature.

Experiments such as this one are tedious, because there is no text interface to a commercial program such as Bridge Master or Bridge Baron. As a result, information regarding the sensitivity of GIB's performance to various parameters tends to be only anecdotal.

GIB solves an additional 16 problems (bringing its total to 64.4%) given additional resources in the form of extra time (up to 100 seconds per play, although that time was very rarely taken), a larger Monte Carlo sample (100 deals instead of 50) arid hand-generated explanations of the opponents' bids and opening leads. Each of the three factors appeared to contribute equally to the improved performance.

Other authors are reporting comparable levels of performance. Forrester, working with a different but similar benchmark [Blackwood, 1979], reports[6] that GIB solves 68% of the problems given 20 seconds/play, and 74% of them given 30 seconds/play. Deals where GIB has outplayed human experts are the topic of a series of articles in the Dutch bridge magazine *IMP* [Eskes, 1997, and sequels].[7] Based on these results, GIB was invited to participate in an invitational event at the 1998 world bridge championships in France; the event involved deals similar to Bridge Master's but substantially more difficult. GIB joined a field of 34 of the best card players in the world, each player facing twelve such problems over the course of two days. GIB was leading at the halfway mark, but played poorly on the second day (perhaps the pressure was too much for it), and finished twelfth.

The human participants were given 90 minutes to play each deal, although they were penalized slightly for playing slowly. GIB played each deal in about ten minutes, using a Monte Carlo sample size of 500. Michael Rosenberg, the eventual winner of the contest and the pre-tournament favorite, in fact made one more mistake than did Bramley, the second place finisher. Rosenberg played just quickly enough that the time penalties gave him the victory. The scoring method thus favors GIB slightly.

There are two important technical remarks that must be made about the Monte Carlo algorithm before proceeding. First, note that we were cavalier in simply saying, "Construct a set *D* of deals consistent with both the

[4]The current version is Bridge Baron 9 and could be expected to perform guardedly better in a test such as this. Bridge Baron 6 does not include the Smith enhancements.

[5]GIB's Monte Carlo sample size is fixed at 50 in most cases, which provides a good compromise between speed of play and accuracy of result.

[6]Posting to rec.games.bridge on 14 July 1997.

[7]http://www.imp-bridge.nl

bidding and play of the deal thus far."

To construct deals consistent with the bidding, we first simplify the auction as observed, building constraints describing each of the hands around the table. We then deal hands consistent with the constraints using a deal generator that deals unbiased hands given restrictions on the number of cards held by each player in each suit. This set of deals is then tested to remove elements that do not satisfy the remaining constraints, and each of the remaining deals is passed to the bidding module to identify those for which the observed bids would have been made by the players in question. This process typically takes one or two seconds to generate the full set of deals needed by the algorithm.

To conform to the card play thus far, it is impractical to test each hypothetical decision against the cardplay module itself. Instead, GIB uses its existing analyses to identify mistakes that the opponents might make. As an example, suppose GIB plays the ♠5. The analysis indicates that 80% of the time that the next player (say West) holds the ♠K, it is a mistake for West not to play it. If West in fact does not play the ♠K, Bayes' rule is used to adjust the probability that West holds the ♠K at all. The probabilities are then modified further to include information revealed by defensive signalling (if any), and the adjusted probabilities are finally used to bias the Monte Carlo sample, replacing the evaluation $\sum_d s(m,d)$ with $\sum_d w_d s(m,d)$ where w_d is the weight assigned to deal d. More heavily weighted deals thus have a larger impact on GIB's eventual decision.

The second technical point regarding the algorithm itself involves the fact that it needs to run quickly and that it may need to be terminated before the analysis is complete. For the former, there are a variety of greedy techniques that can be used to ensure that a move m is not considered if we can show $\sum_d s(d,m) \leq \sum_d s(d,m')$ for some m'. The algorithm also uses iterative broadening [Ginsberg and Harvey, 1992] to ensure that a low-width answer is available if a high-width search fails to terminate in time. Results from the low- and high-width searches are combined when time expires.

Also regarding speed, the algorithm requires that for each deal in the Monte Carlo sample and each possible move, we evaluate the resulting position exactly. Knowing simply that move mi is not as good as move m_2 for deal d is not enough; m1 may be better than m_2 elsewhere and we need to compare them quantitatively. This approach is aided substantially by the partition search idea, where entries in the transposition table correspond not to single positions and their evaluated values, but to *sets* of positions and values. In many cases, m1 and m_2 may fall into the same entry of the partition table long before they actually transpose into one another exactly.

3 Bidding

The purpose of bidding in bridge is twofold. The primary purpose is to share information about your cards with your partner so that you can cooperatively select an optimal final contract. A secondary purpose is to disrupt the opponents' attempt to do the same.

In order to achieve this purpose, a wide variety of bidding "languages" have been developed. In some, when you suggest clubs as trumps, it means you have a lot of them. In others, the suggestion is only temporary and the information conveyed is quite different. In all of these languages, *some* meaning is assigned to a wide variety of bids in particular situations; there are also default rules that assign meanings to bids that have no specifically assigned meanings. Any computer bridge player will need similar understandings.

Bidding is interesting because the meanings frequently overlap; there may be one or more bids that are suitable (or nearly so) on any particular set of cards. Existing computer programs have simply tried to find the bid that is the best match for the cards that the machines hold, but world champion Chip Martel reports* that human experts take a different approach.[8,9]

Although expert bidding is based on a database such as that used by existing programs, close decisions are made by simulating the results of each candidate action. This involves projecting how the bidding is likely to proceed and evaluating the play in one of a variety of possible final contracts. An expert gets his "judgment" from a Monte Carlo-like simulation of the results of possible bids, often referred to in the bridge-playing community as a *Borel* simulation. GIB takes a similar approach.

Algorithm 2 (Borel simulation) *To select a bid from a candidate set* B, *given a database Z that suggests bids in various situations:*

1. *Construct a set D of deals consistent with the bidding thus far.*
2. *For each bid b ∈ B and each deal d ∈ D, use the database Z to project how the auction will continue if the bid b is made. (If no bid is suggested by the database, the player in question is assumed to pass.) Compute the double dummy result of the eventual contract, denoting it s(b,d).*
3. *Return that b for which $\sum_d$ s(b,d) is maximal.*

As with the Monte Carlo approach to card play, this approach does not take into account, the fact that bridge is not played double dummy. Human experts often choose not to make bids that will convey too much information to the opponents in order to make the defenders' task as difficult as possible. This consideration is missing from the above algorithm.

Unfortunately, there are more serious problems also. Suppose that the database Z is somewhat conservative

[8]The 1994 Rosenblum Cup World Team Championship was won by a team that included Martel and Rosenberg.

[9]Frank suggests [Frank, 1997] that the existing machine approach is capable of reaching expert levels of performance. While this appears to have been true in the early 1980's [Lindelof, 1983], modern expert bidding practice has begun to highlight the disruptive aspect of bidding, and machine performance is no longer likely to be competitive.

in its actions. The projection in step 2 leads each player to assume his partner bids conservatively, and therefore to bid somewhat aggressively to compensate. The partnership as a whole ends up ot/ercompensating.

Worse still, suppose that there is an omission of some kind in Z; perhaps every time someone bids 7♢, the database suggests a foolish action. Since 7♢ is a rare bid, a bidding system that matches its bids directly to the database will encounter this problem infrequently.

GIB, however, will be much more aggressive, bidding 7♢ often on the grounds that doing so will cause the opponents to make a mistake. In practice, of course, the bug in the database is unlikely to be replicated in the opponents' minds, and GIB'S attempts to exploit the gap will be unrewarded or worse.

This is a serious problem, and appears to apply to any attempt to heuristically model an adversary's behavior: It is difficult to distinguish a good choice that is successful because the opponent has no winning options from a bad choice that *appears* successful because the heuristic fails to identify such options.

There are a variety of ways in which this problem might be addressed, none of them perfect. The most obvious is simply to use GIB'S aggressive tendencies to identify the bugs or gaps in the bidding database, and to fix them. Because the database is large (some 7400 rules),[10] this is a slow process.

Another approach is to try to identify the bugs in the database automatically, and to be wary in such situations. If the bidding simulation indicates that the opponents are about to achieve a result much worse than what they might achieve if they saw each other's cards, that is evidence that there may be a gap in the database. Unfortunately, it is also evidence that GIB is simply effectively disrupting its opponents' efforts to bid accurately.

Finally, restrictions could be placed on GIB that require it to make bids that are "close" to the bids suggested by the database, on the grounds that such bids are more likely to reflect improvements in judgment than to highlight gaps in the database.

All of these techniques are used, and all of them are useful. GIB'S bidding is substantially better than that of earlier programs, but not yet of expert caliber.

The bidding was tested as part of the 1998 Baron Barclay/OKBridge World Computer Bridge Championships. Each program bid deals that had previously been bid and played by experts; a result of 0 on any particular deal meant that the program bid to a contract as good as the average expert result. There were 20 deals in the contest; although card play was not an issue, the deals were selected to pose challenges in bidding and a standard deviation of 5.5 iMPs/deal is still a reasonable estimate. One standard deviation over the 20 deal set could thus be expected to be about 25 IMPS.

GIB'S final score in the bidding contest was +2 IMPS, as it narrowly edged out the expert field against which it was compared.[11] The next best program finished with a score of -35 IMPS, not dissimilar from the -37 IMPS that had been sufficient to win the bidding contest in 1997.

4 Overall remarks

4.1 GIB compared

GIB participated in the 1998 World Computer Bridge Championships, along with six other computer programs, including Bridge Baron. The event consisted of a complete round robin, with each program playing each other and the results being converted to "victory points." After the round robin, the four leading programs advanced to a knockout phase, which was designed to favor slightly the program that won the round robin.

GIB won every match it played in the round robin, accumulating 95 out of a possible 120 victory points. In the knockout phase, it beat Bridge Baron by 84 IMPS over 48 deals (a 2.2 standard deviation event had the programs been evenly matched) and then beat Q-Plus Bridge in the finals by 63 IMPS over 64 deals (a 1.4 standard deviation event). GIB also played a 14 deal demonstration match against human world champions Zia Mahmood and Michael Rosenberg[12], losing by a total of 6.4 IMPS (a 0.3 standard deviation event). GIB also plays on OKBridge, an internet bridge club with some 15,000 members.[13] After playing thousands of deals against human opponents of various levels, it is losing at the rate of 0.2 IMPs/deal.

4.2 Other games

This has been very much a paper about bridge; I have left essentially untouched the question of to what extent the basic Monte Carlo technique could be applied to other games of imperfect information. Although I can make educated guesses in this area, the experimental work on which this paper is based deals with bridge exclusively.

The primary drawback of the Monte Carlo approach appears to be that it does not encourage information gathering actions, instead tending to defer decisions on the grounds that perfect information will be available later. This leads to small but noticeable errors in GIB's cardplay. Hearts appears to be similar to bridge in this area, and I would expect it to be possible to translate GIB'S success from one game to the other.

The Monte Carlo approach is known to be successful in both backgammon and Scrabble, where the strongest machine players simulate possible dice rolls or tile draws

GIB uses the database that is distributed with Meadowlark Bridge.

[11] This is in spite of the earlier remark that GIB's bidding is not of expert caliber. GIB was lucky in the bidding contest in that all of the problems involved situations that it understood. When faced with a situation that it does not understand, GIB's bidding deteriorates drastically.

[12] Mahmood and Rosenberg have won, among other titles, the 1995 Cap Volmac World Top Invitational Tournament. As remarked earlier, Rosenberg would also go on after the GIB match to win the Par Competition in which GIB finished 12th.

* http://www.okbridge.com

several moves ahead in order to select a move. These games clearly meet the criteria of the previous paragraph, since it is impossible to gather information in advance about the stochastic processes underlying the game.

For other games, however, the problems may be more severe. Poker, for example, depends heavily on the ability to make information gathering maneuvers. How effective Monte Carlo techniques are in cases such as this remains to be seen.

4.3 Future work

GIB has matured to the point that new ideas can be tested by having it play itself overnight over 100 deals. The chess community has already observed that it is easy to use this approach to overfit, so GIB'S self-testing is used only to evaluate coarse features of the approach such as the question of whether a Monte Carlo simulation be used during the bidding at all.[14]

There are a variety of straightforward extensions to GIB that should also improve its performance substantially. Principal among these is the further development of GIB'S (i.e., Meadowlark's) bidding database, and the inclusion of a facility that allows GIB to think on its opponents' time. None of these modifications requires substantial technical innovation; it's simply a matter of doing it. Martel has predicted that GIB will achieve expert levels of performance around 2000, and be stronger than any human player within two or three years after that. The prospects for doing this seem fairly bright.

Acknowledgement

The GIB work has been supported by Just Write, Inc; during its development, I have received invaluable help from members of both the bridge and computer science communities. I am especially indebted to Chip Martel, Rod Ludwig, Alan Jaffray, Hans Kuijf and Fred Gitelman, but also to Bob Hamman and Eric Rod well, to David Etherington, Bart Massey and the other members of GIRL, to Jonathan Schaeffer and Rich Korf, and to Jeff Goldsmith, Thomas Andrews and many other members of the rec.games.bridge community.

The work has also been supported by DARPA and AFRL under agreements F30602-97-1-0294 and F30602-98-2-0181. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, AFRL, or the U.S. Government.

References

[Billings *et al*, 1998] Darse Billings, Dennis Papp, Jonathan Schaeffer, and Duane Szafron. Opponent modeling in poker. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence,* pages 493499, 1998.

[Blackwood, 1979] Easley Blackwood. *Play of the Hand with Blackwood.* Bobbs-Merrill, 1979.

[Eskes, 1997] Onno Eskes. GIB: Sensational breakthrough in bridge software. *IMP,* 8(2), 1997.

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

[Frank, 1997] Ian Frank. Bridge. Technical report, ETL, 1997. http://www.etl.go.jp/etl/suiron/ ianf/Publications/bridge.

[Ginsberg and Harvey, 1992] Matthew L. Ginsberg and William D. Harvey. Iterative broadening. *Artificial Intelligence,* 55:367-383, 1992.

[Ginsberg, 1996] Matthew L. Ginsberg. Partition search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence,* 1996.

[Roller and Pfeffer, 1995] Daphne Roller and Avi Pfeffer. Generating and solving imperfect information games. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence,* pages 1185-1192, 1995.

[Levy, 1989] David N.L. Levy. The million pound bridge program. In D.N.L. Levy and D.F. Beal, editors, *Heuristic Programming in Artificial Intelligence,* Asilomar, CA, 1989. Ellis Horwood.

[Lindelof, 1983] Torbjorn Lindelof. *COBRA: The Computer-Designed Bidding System.* Gollancz, London, 1983.

[McAllester, 1988] David A. McAllester. Conspiracy numbers for min-max searching. *Artificial Intelligence,* 35:287-310, 1988.

[Plaat *et al,* 1996] Aske Plaat, Jonathan Schaeffer, Wim Pijls, and Arie de Bruin. Exploiting graph properties of game trees. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence,* pages 234-239, 1996.

[Schaeffer, 1997] Jonathan Schaeffer. *One Jump Ahead: Challenging Human Supremacy in Checkers.* Springer-Verlag, New York, 1997.

[Sheinwold, 1996] Alfred Sheinwold. *Five Weeks to Winning Bridge.* Pocket Books, 1996.

[Smith *et al,* 1996] Stephen J.J. Smith, Dana S. Nau, and Tom Throop. Total-order multi-agent task-network planning for contract bridge. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence,* Stanford, California, 1996.

[Wiikins, 1980] David E. Wilkins. Using patterns and plans in chess. *Artificial Intelligence,* 14:165-203, 1980.

[14] The simulation does appear to be useful; GIB bidding with it beats GIB without it by 1 iMP/deal.