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 Goren In a Box (GIB), the first bridge-playing program to approach the level of a human expert. 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 earlier programs.

Introduction

Of all the classic games of 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 was beaten by Janice Kim in the AAAI-97 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 has said that he would expect to beat any existing poker program after five minutes' play.†[1] Perennial world bridge champion Bob Hamman has summarized all of the commercial 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.†

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.[2]

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.[3] 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 hand 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

[1] Many 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] The ACBL measures performance by allowing players to accumulate "master points". Since master points never expire, they tend to measure longevity as opposed to skill. While this is undoubtedly sound marketing for an organization whose average member is now of age 66, it provides little useful information on playing strength.

[3] 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.

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 application of planning techniques into Bridge Baron, 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 performance, but was sufficient for it to win the 1997 World Computer Bridge Championships in Albuquerque.

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, avoidance, 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 attempting to use techniques similar to those 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.

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 1 IMP per deal played and losing to strong human players by a similar amount. 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. The "bidding" section 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, a time line of GIB's development, and suggestions for future work.

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 and the introduction of a transposition table bring it down to about 1.7. Augmenting the move ordering heuristic to exploit narrowness[4] reduces the branching factor further to approximately 1.3, corresponding to a search space of some 10^6 nodes per deal. Partition search (Ginsberg 1996) reduces the space further to some 50,000 nodes per deal; Kuijf[†] reports similar results using a careful implementation of the killer heuristic.

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 hand 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

[4]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).

Koller† and others (Frank and Basin 1997). 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! In spite of this, GIB's performance as a card player is at the level of a human expert.

Performance was measured using *Bridge Master* (BM), a commercial program developed by Gitelman. BM contains 180 hands 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 hands because the signal to noise ratio is far higher; good plays are generally rewarded and bad ones punished. Every hand also contains a lesson of some kind; there are no completely uninteresting hands where the line of play is irrelevant or obvious. There are drawbacks to testing GIB's performance on nonradomly dealt hands, 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 hands using the same hardware.[5] 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. 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.

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.

[5]The current version is Bridge Baron 8 and, as discussed in the introduction, could be expected to perform guardedly better in a test such as this. Bridge Baron 6 does not include the Smith enhancements.

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. 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 independent of the opponents' holdings.

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 (Ginsberg 1997). GIB's performance in these situations is clearly at the level of a human expert.

Experiments such as this one are extremely tedious to perform, 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 60) and hand-generated explanations of the opponents' bids and opening leads. Each of the three factors appeared to contribute equally to the improved performance. Galligan† also reports that GIB's card play is "much worse" if the sample size is reduced from 100 deals to 50.

Other authors are reporting comparable levels of performance. Forrester, working with a different but similar benchmark (Blackwood 1979), reports[6] that

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

GIB solves 68% of the problems given 20 seconds/play, and 74% of them given 30 seconds/play. Hands where GIB has outplayed human experts are the topic of an ongoing series of articles in the Dutch bridge magazine *IMP* (Eskes 1997, and sequels).[7] GIB is regularly used to compare different lines of play in the bridge newsgroup rec.games.bridge and assisted with the commentary at the 1997 world championships in Tunisia. No other program is even considered for this purpose.

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 bidding and play of the hand 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 adjusted probabilities are then used to bias the Monte Carlo sample used by the algorithm.

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.[8]

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 m_1 is not as good as move m_2 for deal d is not enough; m_1 may be better than m_2 elsewhere and we need to compare them quantitatively. This approach is therefore 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, m_1 and m_2 may fall into the same entry of the partition table long before they actually transpose into one another exactly.

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

[8] In practice, results from the low- and high-width searches are combined when time expires.

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.[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 or, in some cases, evaluating how the play is likely to go 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 sug-*

[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 (Lindelöf 1983), modern expert bidding practice has begun to highlight the disruptive aspect of bidding, and machine performance is no longer likely to be competitive.

gests bids in various situations:

1. *Construct a set D of deals consistent with the bidding thus far.*
2. *For each bid $b \in B$ and each deal $d \in D$, 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 as well. Suppose, for example, that the database Z is somewhat conservative in its actions. Each player will conclude that it pays to be somewhat aggressive because partner is assumed (step 2) to bid conservatively. Both players end up overcompensating.

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 move (i.e., bid) that is successful because the opponent has no winning options from a bad move 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 indentify the bugs or gaps in the bidding database, and to fix them. Because the database is large (some 6500 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 being effective in 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 appears to be slightly better than Bridge Baron's, but clearly not yet of expert caliber.

Overall remarks

GIB compared

As mentioned earlier, direct comparisons between GIB and commercial programs are difficult because the interface must be handled manually. In the most recent test, a 16-deal match compared the most recent version of Bridge Baron with the current version of GIB. GIB won by 2.2 IMPs per deal (1.6 standard deviations over the 16 deal match). GIB was given an average of ninety seconds per hand (two minutes is the tournament norm), and Bridge Baron was given a comparable 10 seconds per play (it crashed at any higher setting).

GIB also plays on OKBridge, an internet bridge club with some 10,000 members.[11] After playing some 800 hands against human opponents, it is losing at the rate of 0.66 IMPs/deal.

One final note: GIB's dependence on double-dummy analysis means that it lacks the ability to operate deceptively in either bidding or card play. How much of a difficulty this is in practice remains to be seen.

Timeline and future work

A short chronology of the GIB project's main milestones appears in the table on the next page.[12]

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.[13]

[10] GIB uses the database that is distributed with Meadowlark Bridge, a commercial product.

[11] `http://www.okbridge.com`

[12] During GIB's development, there was a short period during which I did not understand GIB's propensity to propel itself into gaps in the bidding database. The program exhibiting this behavior is referred to as "experimental GIB," or EGIB, in the table.

[13] The simulation does appear to be useful; GIB bidding with it beats GIB without it by 1 IMP per deal.

Date	Event
6/1/94	work begins on GIB
11/7/95	double dummy engine solves complete deals in 200,000 nodes on average
1/8/96	GIB makes its first non-double dummy play
7/4/96	GIB plays (human bidding, machine card play) in a club game, finishes third
11/8/96	GIB solves 64% of the Bridge Master deals
12/4/96	faced with GIB, world champion Zia Mahmood retracts a celebrated Levy-like $1 million bet
5/8/97	GIB plays (with bidding) in a club game, finishes about average
5/17/97	world champion Chip Martel predicts GIB will outperform existing programs in 1998, be an expert around 2000
6/29/97	GIB finishes 7th in an ACBL tournament
7/31/97	EGIB plays world champions Meckstroth and Rodwell, loses by 33 IMPs in 10 deals
7/31/97	EGIB finishes last in World Computer Bridge Championships
9/19/97	GIB plays world champion Bobby Goldman, loses by 6 IMPs in 8 deals
10/16/97	GIB discovers a new card play ending

There are a variety of straightforward extensions to GIB that should also improve its performance substantially. Principal among these is the possibility of signalling on defense; as described earlier, GIB tries to figure out what its partner (or any other player) has simply by constructing hands for which the observed plays make sense. Defensive signalling is a more effective way of achieving this goal. GIB also needs to be extended to deal with forms of bridge scoring other than IMPs, and to think on its opponents' time.

None of these modifications requires substantial technical innovation; it's simply a matter of doing it. The prospects for an expert level computer bridge player in the time frame Martel has suggested (see the table) 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 Rodwell, to David Etherington, Bart Massey and the other members of CIRL, to Jonathan Schaeffer and Rich Korf, and to Jeff Goldsmith, Thomas Andrews and many other members of the rec.games.bridge community.

References

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

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

Ian Frank and David Basin. Search in games with incomplete information: A case stude using bridge card play. *Artificial Intelligence*, 1997. To appear.

Ian Frank. Bridge. Technical report, ETL, 1997. `http://www.etl.go.jp/etl/suiron/%7Eianf/Publications/bridge`.

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

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

Matthew L. Ginsberg. A computer discovers something new. In preparation, 1997.

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

Torbjörn Lindelöf. *COBRA: The Computer-Designed Bidding System.* Gollancz, London, 1983.

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

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.

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

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

David E. Wilkins. Using patterns and plans in chess. *Artificial Intelligence*, 14:165–203, 1980.