BRIBIP: A BRIDGE BIDDING PROGRAM

Alan M. Stanier
University of Essex
Colchester, England.

Introduction

This paper describes a program, BRIBIP, capable *of* making decisions in an environment of imperfect information. It is frequently necessary that such decisions be made in real life - as, for example, the decisions made in the day-to-day running of a firm. As a paradigm of such situations, the game of Contract bridge (3) was chosen, for several reasons:

1) Despite being a game of imperfect information, it is well-defined: in particular, the scoring systems give a way of assessing the strength of any resulting program.

2) Although for any one player in the game, there is an environment of imperfect information, the four players between them have total knowledge of any situation. This leads naturally to a program based on explicit reasoning about why the other players react as they do, rather than a probablistic approach. This can be contrasted with games such as poker, where not all the cards are dealt, and thus a probablistic approach is necessary, even for humans. In fact, an earlier program (11), based on a probablistic approach, was written, and found to be unsatisfactory because the resulting data-structures were not subtle enough to reflect the logical processes needed to play a good game, and because the probablistic approach severely underestimated the need for a detailed plan. In view of these difficuties the current program, which can make explicit deductions in a Conniver-like fashion, has been built.

3) There is a great deal of literature on the theory of bridge, and hence a wealth of suggestions as to how the program should work. Many of these suggestions are not in an immediately useful form - for example "When it is hoped to add the possibility of a pseudo-squeeze to the other chances, one should first cash the winners in the suit where the length is visible in dummy" - but they can be modified to suit our needs. However, many are in an immediately useful format, and can be straightway represented in the program - for example "Open on 12 or more points and at least four of the suit", etc.

4) The author is interested in bridge to the extent of representing Essex University in club competitions.

The last two reasons are closely connected, and reflect a thesis which underlies this research: that the best way to produce an intelligent program is to code into it expert knowledge. This has been seen to be true of chess (2), and of checkers, where Samuel (9, 10) co-operated with expert players in order to determine the parameters in his evaluation functions.

The program is written in Popcorn (7), and occupies 26K of 36-bit core on the PDP-10, plus the POP2 interpreter and run-time system. The program bids to the standard of the average club player, and is capable of extracting from a bidding sequence the direct inferences of the hidden hands (but not, as yet, the indirect inferences based on the unmade bids), and is capable of maintaining a (usually) consistent data-base reflecting the hidden hands, which it can also modify in the light of seeing the first card played and the dummy hand. The program bids at approximately the speed of a human player. Several examples (in which the program bids the South hands):

```
                         NORTH

                     S:  A  J  8  7
                     H:  8  6  5
                     D:  K  J  8
                     C:  A  J  6

   WEST                                  EAST

S:  10 9  5  6                       S:  K  4  2
H:  A  K  Q                          H:  9  7  4  3
D:  4                                D:  10 7  5  3
C:  10 9  8  7  3                    C:  Q  5
                        SOUTH

                  S:  Q  3
                  H:  J  10 2
                  D:  A  Q  9  6  2
                  C:  K  4  2
```

BIDDING

```
                SOUTH  WEST  NORTH  EAST

                  -     -      -     NO
                 1 D    NO    2 S    NO
                 3 S    NO    4 C    NO
                 4 D    NO    4 S    NO
                 NO     NO
```

A good contract, which unfortunately goes two down after the failure of the trump finesse.

```
                    NORTH

                 S:  K  J
                 H:  A  K  Q
                 D:  8  6
                 C:  K  J  8  7  4  2
     WEST                            EAST

S:  A  8  7  4                S:  10  6  3
H:  8  5  3  2                H:  10  9  7  6
D:  Q  4                      D:  9  5  3
C:  6  5  3                   C:  A  10  9
                   SOUTH

                 S:  Q  9  5  2
                 H:  J  4
                 D:  A  K  J  10  7  2
                 C:  Q
```

BIDDING

SOUTH	WEST	NORTH	EAST
1 D	NO	2 C	NO
2 S	NO	2 NT	NO
3 NT	NO	NO	NO

A good contract, which is certain to be made, with at least one overtrick.

```
                  NORTH

              S:  A  J  9  4
              H:
              D:  A  K  Q  10  8  4
              C:  A  8  2
   WEST                                 EAST

S:  Q  10  8  7                 S:  3
H:  K  J  6  3                  H:  Q  8  7
D:  3                           D:  9  7  6  5
C:  Q  7  5  4                  C:  K  10  9  6  3
                SOUTH

            S:  K  6  5  2
            H:  A  10  9  5  4  2
            D:  J  2
            C:  J
```

BIDDING

SOUTH	WEST	NORTH	EAST
NO	NO	1 D	NO
1 H	NO	2 D	NO
2 H	NO	2 S	NO
4 S	NO	4 NT	NO
5 D	NO	5 NT	NO
6 D	NO	6 S	NO
NO	DBLE	7 D	DBLE
7 S	DBLE	NO	NO
NO			

The best possible contract. As the cards lie, 7 S can be made by taking the deep finesse in trumps: making 4 spade tricks, 6 diamonds, H A, C A and a club ruff.

General Approach

The program is based on two distinct but closely related ideas: firstly, close analysis of the bidding and the play to gain information previously unknown, and secondly, the utilisation of such information (and the programs knowledge of the game) in the creation of a plan which the program can follow.

In making a plan, the program will differ from most game-playing programs written to date (although Miller (8), Berliner and others have investigated a similar approach in the case of chess). The more usual tree-searching approach was rejected on several grounds:

1) I am interested in transferring to the program the knowledge gained by bridge-players in the past decades. The body of such knowledge is vast, and to ignore it seems extremely short-sighted. The author is looking for an artificial intelligence solution to the problem - and AI could well be defined as the implementation of knowledge/inference systems. It follows that instead of reducing this body of knowledge to a set of numbers, the author wishes explicitly to reason about it.

2) An evaluation function in a game of imperfect information would presumably return a probablistic result. This makes mini-maxing a complex task. *One* can conceive of single-valued evaluation functions, (such as the number of tricks won plus the expected number still to be won), but such functions *are* highly artificial, and a great deal of information is lost in the process.

3) Any tree-searching method that truncates the tree (as must happen in all but a few cases in the later stages of the game) can fall foul of what Berliner calls the 'Horizon Affect' - i.e. a false value is assigned to a given move; false because of an affect further down the tree, and thus invisible to the routine,

4) If one studies the concepts used by human bridge players, one finds such concepts as "bid to a slam", "draw trumps", "show first-rouna control", "finesse the queen of clubs", and so on. A successful bridge program will also need to have such concepts, and the author does not believe that such detailed planning is possible in an evaluation function. It is fairly clear, however, how mini-strategies of such a form can be generated *by* a reasoning program.

Information *Available*

Probably the most important activity performed by the program is the collection of information about the hidden hands. There are several opportunities for collecting such information:

Firstly, during the bidding stage, where the program attempts to gauge the strengths and weaknesses, perhaps even to the point of holding (or not holding) individual cards, of the hidden hands, from what bids they make. There art two aspects to this: the straightforward aspect of analysing the meaning of each bid, and the more subtle aspect of considering why a different bid was not made. For example, consider the partial sequence:

NORTH	EAST	SOUTH	
1 C	NO	1 S	...

What information is available from South's bid of 1 S? (here we assume that North/South use the bidding system known as Essex Standard ACOL). This can be simply interpreted as showing a high-card points count of at least 6, and at least 4 spades. However, a deeper analysis is possible, and yields more information. If we assume that South is making what he believes to be his best bid, then the bid of 1 S describes his hand more precisely than any other. from this, we may infer that he holds more spades than, say, diamonds, as with longer diamonds, or with an equal number, he would have bid 1 D (it being a cheaper bid). Similarly, he cannot hold more than three clubs, or he would have bid 2 C. More information is available from the facts that he did not bid 1 H, 1 NT, etc. etc. To facilitate the description of ones hand, there have evolved bidding conventions - pre-arranged bids with special meanings describing the hand. These conventions are gathered together in self-consistent systems, such as Goren (6), Acol (4), Roman (1) etc. While such systems provide useful guidelines for the bidding, they by no means make the process of bidding mechanical. Within any system there is a need for detailed logical thought in choosing the bid. Consequently, the planning section of the program is included as part of the bidding. Consider the following hand:

SPADES	A K Q J 9 7 4 2
HEARTS	A K Q
DIAMONDS	2
CLUBS	3.

Fairly obviously, with this hand, one can make b S with no help from partner, 6 S if partner holds one of the missing aces, and 7 NT should he hold both. The correct bid, therefore, would be an asking bid, enquiring of partner how many aces he held. However, the thought "I can make 5 S, plus a trick for every ace partner holds" implies the existence, and perhaps construction, of a rudimentary or partial plan for so doing.

The actual process of transferring information between the players is best considered in four sections, mirrored by four routines in the program. One routine is used once in every round of bidding to choose the programs own bids, another interprets the partners bidding, and the remaining two interpret the opponents bidding. At first sight, one might believe that the three interpreting routines are closely connected (and in the trivial sense that they all update the same data-base they are), but it is useful to separate them, as they have slightly differing effects concerned with inter-process communication.

Inter-process communication is modelled by messages which override the standard action of the receiving routine. for example, should the program make the 4 NT bid described above, it will send to the routine interpreting partners response a message saying "I have just made a Blackwood bid asking for aces. Interpret the bid you receive in that light".

All four routines have to access knowledge of the bidding system. There are several ways in which this could be done. In one possible method, the system is held as a data structure - most probably a tree - and then all the routines interpret it. This seems to be the most computationally feasible method, and has been used before (12), but is somewhat unsatisfactory in that the data structure will be only a vague approximation to the true tree - which would be of astronomic size. A preferred approach is to have the main parts of the system coded as mini-procedures (for example Conniver methods), and add these methods to, or delete them from, the data-base as they become relevant.

Hopefully, we should emerge from the bidding sequence with a reasonably accurate prediction about the hidden hands, and a tentative plan as to how the Contract could be made, or defeated, as the case may be. At this point, the playing stage is entered. After the first lead, the dummy hand is spread, and the dummy takes no further part in the play of this hand.

Assuming the program not to be dummy, it has two connected tasks to perform at this point. It should review the bidding, but this time knowing the dummy hand, and seeing whether any of its assumptions are either untenable, or nedd modifying - one can often pinpoint the location of several key cards in this way. Suppose the program holds the aces of hearts and clubs, and South has made a bid indicating the holding of one ace. Then, should dummy hold the ace of diamonds, we may place South with the ace of spades, and vice versa. This review will be made considerably more efficient if the program builds up, during the bidding stage, a list of cues for things for which to watch dummy. Concurrent with this improvement of the data-base, there should be an analogous improvement of the plan.

Program Details

After the comparative failure of BRIPP1 (the earlier, and unsuccessful, program based on probablistic techniques), it was decided to write BRIBIP with different design decisions as previously described.

The particular bidding systems implemented is a highly-modified version of ACOL used by some of the University of Essex bridge team. This is described in (5). This system was chosen as the author was familiar with it. Users of standard ACOL may find some bids unusual - this is due to eccentricities in the Bidding system, not to bugs in the program.

To demonstrate the operation in the program, we work through a deal, showing the various routines in operation. The program bids the South hand, and East is the dealer. The four hands (of which the program only knows its own, of course) are:

```
                    NORTH

                 S:  K  9  8  3
                 H:  K  J  10 8
                 D:  K  7  2
                 C:  A  4

     WEST                               EAST

S:  10 7  6  5                     S:
H:  4  2                           H:  Q  7  6  5
D:  Q  9  5  4  3                  D:  10 8  6
C:  J  7                           C:  Q  8  6  5  3
                   SOUTH

                 S:  A  Q  J  4  2
                 H:  A  9  3
                 D:  A  J
                 C:  K  10 2
```

Having received its hand - in this case from the teletype, but there is also the facility of computer-dealt hands - the program analyses its hand for various features.

The features routine returns seven features of the hand:

1) The highcard points count. This is used to determine whether or not the programs hand is strong enough to open the bidding, or to respond, or whether a special opening bid is permissable.

2) An estimate of the playing strength of each suit. The actual estimate returned is the length plus one-tenth the points count, with a weighting of 1.25 on majours. Thus Souths spades are rated at 1.25*(5 + 7/10) = 7.125.

3) A list of the suits in which the hand has first-round control, by which we mean an ace or void - the point being that if that suit is led, the hand can almost certainly win the trick.

4) The distribution points count.

5) Whether or not the hand is balanced. A balanced hand is a prerequisite for the program to consider a *natural* (as opposed to conventional) bid in NOTRUMPS.

b) 0 If the hand is not suitable for a weak two opening, or the possible suit if it is.

7) ¢ If the hand is not suitable for a preemptive bid, or the suit if it is. Thus, in this case features returns

```
19 (3.3 2.5 4.25 7.125) (4 3 2) 1 1 0 0,
```

and (as a side effect) adds to the data-base various facts about the hand - which cards are held, the length of each suit, the points in each suit, and the total points-count. It is true that the same facts are also held explicitly, but it is useful to have them duplicated in the data-base, as, in conniver-like languages such as POPCORN, Ifadded methods are used to interact with the facts in the data-base. For example, there is an Ifadded method invoked whenever any player shows first-round control. This method searches the data-base for the fact that any players (other than the one showing first-round control, but including the program itself) is known to hold the ace in the suit. If so, then we know the first-round control is based on a void.

Control then enters the bidding routine proper. It first initializes some variables and "pseudo-variables". These pseudo-variables appear to be standard variables, but behind the scenes they are functions that access and/or update the data-base. East opens 'NOBID'. BRIBIP is aware that 12 points or more is needed to open the bidding, and so concludes that East holds, at most, 11 points. This fact is added to the data-base. Since East, having nobid once, is likely to continue doing so (especially if West also notids). BRBIP henceforth expects East to nobid, and will only analyse other (in some sense 'surprizing') bids.

It is then the programs turn to bid. Having first checked that it does not hold a hand qualifying for a special opening bid, it considers the possibility of making a standard suit opening. The only requirement for this is at least 12 points, and this is eminently possible. Having satisfied itself that it has the strength to do so, it calculates which is its best suit (in this case, spades) and bids IS. It then interprets its own bid - at first sight a superfluous process, but in practise useful for the facts added to the data base. The interpretation adds the fact that South bid spades to the base, and sets the value *of* the pseudo-variable Weveopened. In fact, two databases are used - one to hold all facts BRIBIP knows, and another all facts BRIBIP has shown.

West nobids, and is marked as holding less than 12 points.

North responds 2S. This is analysed as a standard response, showing a points count of 6 or more. This is added to the data-base, along with the information that North bid spades. The addition of Norths points count triggers an Ifadded method that calculates safelevel, in this case 4, Safelevel is a pseudo-variable used to estimate whether there are good reasons to suppose a game or slam is on, and thus to decide whether there is any point in showing first-round controls. The addition of the fact that North bid spades triggers an Ifadded method that discovers that both sides of the partnership have bid in spades: thus spades is agreed as the trump suit.

East again nobids, as expected, (Both East and West will nobid throughout the auction, and will not be considered any more in this analysis.

It is now Souths turn to rebid. Safelevel indicates the possibility of game, and trumps have been decided, and therefore the program decides that it is time to show first-round controls. South holds first-round control in three suits (diamonds, hearts and spades), and so he makes the lowest first-round control bid open to him, i.e. 3 D.

North responds 4 C. This is interpreted as showing first-round control in clubs, and denying first-round control in hearts - had he first-round control in hearts, he would have made the cheaper bid of 3 H.

South now shows his first-round control of hearts, by bidding 4 H.

North, whose bidding has so far not shown the power of his hand, now decides that if South, by himself, believes that game is on, then a slam is a distinct possibility. Instead of signing off in spades, he therefore bids 4 NT. This is analysed as being a Blackwood bid, asking for aces. The interpreting routine passes this message to the bid-choosing routine.

When South again comes to bid, it sees the message, and thus invokes the Blackwood routine. Blackwood counts the aces, finds three, and so returns a bid of S C. (Note for Cognescenti: we use Roman Blackwood).

North is thus assured that the partnersnip holds all the aces, and so asks for kings by bidding 5 NT. This is interpreted by BRIBIP as being a Blackwood bid as before.

Therefore, South shows the holding of one king, by bidding b D.

North can then tell that the partnership holds all the aces, all the kings, at the very least 8 of the spades, and good opportunities for ruffing (for example, the clubs). He therefore bids 7 S, which is passed out, and can be made without difficulty.

Conclusions

The project was started in the hope of obtaining some general principles of decision-making in environments of imperfect information. To date, the only such principle found is the negative one; that probablistic methods are not in themselves sufficient. It was hoped to keep knowledge of bridge peripheral to the program, and use a general core, but it was found necessary to put this knowledge in the core itself. This leaves the author somewhat pessimistic about the possibility of a general program for decision-making.

References

1 **Belladonna & Avarelli**
Roman Club System of Distributional Bidding
Simon & Schuster, 1959.

2 **Berliner**
Chess as Problem Solving: The Development of a tactics analyzer.
Carnegie-Mellon University,1974.

3 **Cohen & Barrow**
The Bridge Players Encyclopedia.
Paul Hamlyn, 1967.

4 **Cohen & Reese**
Acol System of Contract Bridge.
Toiner & Steele, 1938.

5 **Everett, Mellor and Stanier,**
The Essex Tournament Bidding System.
Private Publication, 1973.

6 **Goren**
Point Count Bidding.
Simon & Schuster, 1951.

7 **Hardy**
The Popcorn Reference Manual.
CSM-1 Essex University, 1973.

8 **Miller**
Proposals for a Better Chess Program.
M.Sc. Thesis Essex University, 1973.

9 **Samuel**
Some Studies in Machine Learning Using the game of Checkers.
I.B.M. Journal of Research and Development, 3, 210-229 (1959).

10 **Samuel**
Some Studies in Machine Learning Using the Game of Checkers: 2 Recent Progress.
I.B.M. Journal of Research and Development, 11, 601-617 (1967)

11 **Stanier**
BRIPP - A Bridge-Playing Program
Essex University, 1974.

12 **Wasserman**
Realization of a Skillful Bridge Bidding Program.
FJCC, 70, 433-444 (1970)