Model of bridge players who make mistakes

Tomohito Otawa[1] *and Takao Uehara*[2]

Tokyo University of Technology, Japan

ABSTRACT

The contract bridge is an imperfect information game, but most computer bridges play cards assuming the best play by opponents. So, it is difficult for computer bridges to take advantage of mistakes made by less-than-perfect opponents. This paper proposes an opponent model with imperfect information and a heuristic algorithm to take advantage of mistakes by the opponent.

1. INTRODUCTION

The author proposed an agent model for a bidder of the contract bridge (Ando and Uehara, 2000). Since bridge auction is a task based on imperfect information, each agent has hypothetical reasoning ability and generates images of other players' hands by abduction from a bidding sequence (Uehara, 1995). The same architecture of the agent is useful as the opponent model in the card playing of bridge. The knowledge for guessing the other players' hands from the observed play is stored in the agent.

Application of the Monte Carlo sampling technique to the game of bridge was proposed by (Levy, 1989) and implemented by (Ginsberg, 1996). The card selection algorithm (Ginsberg, 2001) constructs a set of deals consistent with both the bidding and play thus far. The card to play is selected by evaluating the double dummy result for each deal. It is assumed that opponents have perfect information and do not make any mistake in each deal. It is well known that Ginsberg's approach has drawbacks (Frank and Basin, 1998). In order to remove the drawbacks, two heuristic algorithms, vector minimaxing and payoff reduction minimaxing, were proposed by (Frank, Basin, and Matsubara, 1998). They also use the assumption again as it represents the most conservative approach modeling the strongest possible opponents.

We propose to construct three sets of deals from each player's point of view which is consistent with the bidding and play thus far and also the card to play next. We think that an opponent makes mistakes because the player's hand constructed from the opponent's point of view are different from the real player's hand. Although algorithms to take advantage of opponent's mistakes may be implemented on any algorithms mentioned above, we implemented it as an extension of Ginsberg's algorithm. We are aware that Ginsberg's algorithm does not produce perfect or even good play in many situations but we believe it to be an adequate evaluation function for our opponent model.

2. MONTE CARLO CARD SELECTION ALGORITHM

The Monte Carlo card selection algorithm (Ginsberg, 2001) is introduced in this section. GIB (Ginsberg's computer bridge program) has a high performance (partition search) engine to analyze bridge's perfect-information variant (double dummy bridge), where all of the cards are visible and each side attempts to take as many tricks as possible. GIB uses Monte Carlo simulation for card-play (and bidding). The unseen cards are dealt at random, biasing the deal so that it is consistent with both bidding and with the cards played thus far. Then GIB analyzes the deal in double dummy and decides which of the possible plays is the strongest. It is expected that averaging over a large number of such Monte Carlo samples will allow us to deal with the imperfect nature of bridge information. Unfortunately this is not always true.

[Monte Carlo card selection algorithm] To select a move from a candidate set M of such moves:

[Step 1] Construct a set D of deals consistent with both the bidding and play of the deal thus far.

[Step 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 $s(m, d)$.

[1]email:otawa@ue.it.teu.ac.jp
[2]email:uehara@cc.teu.ac.jp

[Step 3] Return that m for which $\sum_d s(m, d)$ is maximal.

3. MODEL OF AN OPPONENT WHO MAKES NO MISTAKES

[Example 1]

```
          S 432
          H 32               Dealer: South
          D 32               Vul: NS
          C AJT987           Bidding: South  West   North  East
S 8765   +-------+ S QJT              2NT    Pass   3NT    Pass
H 654    |   N   | H KJT9             Pass   Pass
D 654    | W   E | D QJT9
C 543    |   S   | C K2      Play:    West   North  East   South
         +-------+                     H6     H2     HK     HA
          S AK9                        C3     C7     ?      CQ
          H AQ87
          D AK87
          C Q6
```

Figure 1: Deal, bidding and play history of example 1.

West leads the six of Hearts, and Declarer captures East's King with the Ace and runs the Queen of Clubs. If East is a human expert, he will play low with an air of innocence (Gardner and Mollo, 2001). If South tries the Club finesse again, he will be cut off from the Dummy. South cannot make his contract if Dummy yields him one trick only. If East is GIB, it plays the King of Clubs. Because GIB assumes that the opponent makes no mistakes, that is, East's hand is visible from other players as well as Dummy's hand. Figure 2 is a set of deals D consistent with both the bidding and play of the deal thus far. Number of deals used in GIB is about 50, but we use a small set of deals in order to simplify the explanation in this paper.

```
            S 432                                 S 432
  d1        H 32                      d2          H 32
            D 32                                  D 32
            C AJT987                              C AJT987
S 875     +-------+   S QJT          S 9765     +-------+   S QJT
H 654     |       |   H KJT9         H 654      |       |   H KJT9
D 8765    |       |   D QJT9         D 654      |       |   D QJT9
C 653     +-------+   C K2           C 543      +-------+   C K2
            S AK96                                S AK8
            H AQ87                                H AQ87
            D AK4                                 D AK87
            C Q4                                  C Q6
```

Figure 2: A set of deals D constructed from East's point of view.

Figure 3 is a part of the game-tree to evaluate the double dummy results of playing $C2$ and CK in the deal $d1$, where $C2$ is the Duce of Clubs, CK is the King of Clubs and CX is a small card of Clubs. If East plays low, South captures East's King of Clubs with the Ace and gets 5 tricks of Clubs. If East plays King of Clubs, South gets only 4 tricks of Clubs. The part of the game-tree for the deal $d2$ is almost the same. That is, $s(C2, d1) = -690$, $s(CK, d1) = -660$, $s(C2, d2) = -690$ and $s(CK, d2) = -660$. As a result GIB selects the King of Clubs, because $s(CK, d1) + s(CK, d2) > s(C2, d1) + s(C2, d2)$.

Figure 4 illustrates the opponent model of GIB using our agent. The East's agent constructs a deal according to the reasoning based on the bidding and play so far. The South's agent (the opponent) makes no mistake because it is assumed that all of the cards in the deal are visible.

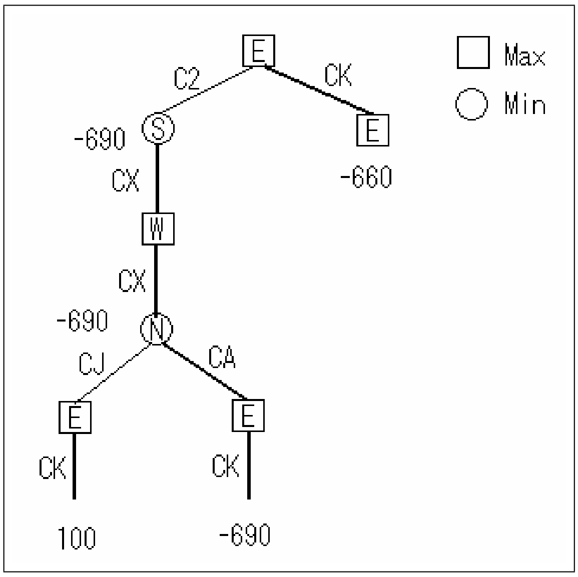

Figure 3: Game-tree for deal d1.

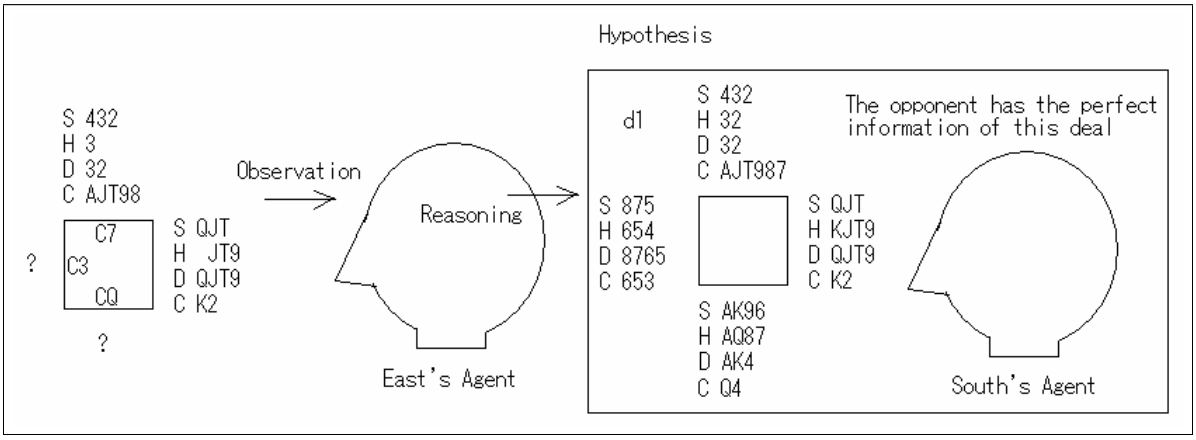

Figure 4: Model of the opponent (South) who makes no mistakes.

4. MODEL OF AN OPPONENT WHO MAKES MISTAKES

We propose an opponent model and a new card selection algorithm where the player can take an advantage of an opponent's mistake. Figure 5 illustrates our model. The East's agent constructs a deal but the other players cannot see his hand in our model.

The East's agent simulates two cases. In the case 1 East plays the Duce of Clubs and the South's agent constructs a sample of deals $d11@C2/S$ which is consistent with the biddings, plays so far (including the play of the Duce of Clubs) and the South's Hand in the deal $d1$ constructed by East's agent. In the case 2 East plays the King of Clubs and the South's agent constructs a sample of deals $d11@CK/S$ which is consistent with the biddings, plays so far (including the play of the King of Clubs) and the South's hand in the deal d1. West has King of Clubs in the deal $d11@C2/S$ constructed from the South's point of view after East plays the Duce of Clubs. If South considers such a case, he may try the Club finesse again mistakenly.

5. A NEW CARD SELECTION ALGORITHM

A card selection algorithm in the section 2 is extended by using the Monte Carlo samples constructed from other players' points of view. We call it ABC algorithm because it selects a move based on three players' (player A's, B's and C's) points of view.

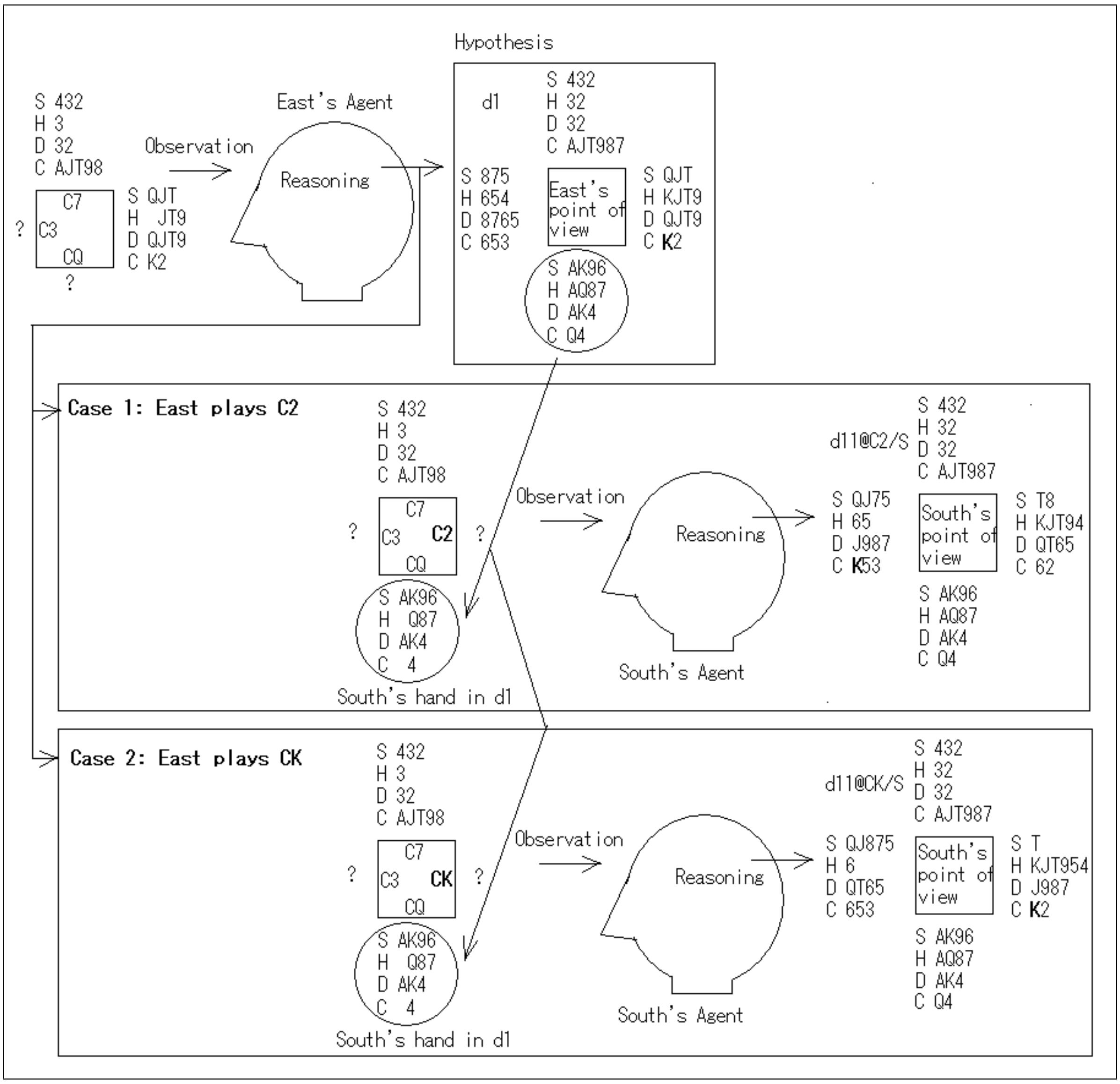

Figure 5: Model of the opponent (South) who makes mistakes.

[ABC algorithm] To select a move of the player A from a candidate set M considering other players' (player B's and C's) points of view:

[Step 1] Construct a set Da of deals consistent with both the bidding and play of the deal thus far from the player A's point of view.

[Step 2] For each move $m \in M$ and each deal $da \in Da$, evaluate the result of making the move m in the deal da according to the step 2-1 to 2-3.

[Step 2-1] For each deal $da \in Da$, construct two sets of deals $Db(m, da)$ and $Dc(m, da)$ consistent with both the bidding and play including m from the other players' (B's and C's except the Dummy's) points of view.

[Step 2-2] For each triple of deals $[da, db, dc]$ where $da \in Da$, $db \in Db(m, da)$ and $dc \in Dc(m, da)$, evaluate the result of making the move m where each player selects moves assuming the deal from his own point of view. Denote the score obtained by making these moves $s(m, [da, db, dc])$.

[Step 2-3] Denote the average score $\sum s(m, [da, db, dc])/n$ obtained by making this move $s(m, da)$, where n is the number of triples $[da, db, dc]$.

[Step 3] Return that m for which $\sum_d s(m, da)$ is maximal.

The details of Step 2-2 are as follows:

[Definitions]

$n0$ is the root node of the game-tree.
$player(n)$ is the player who selects a move at a node n of the game-tree.
The $player(n0)$ at the root has the name 'A'.
The other players excluding Dummy have names 'B' and 'C' individually.
$type(n)$ is the type of a node n and it is 'max' or 'min'.
We define the type(n0) is max.
If $player(n)$ is A or the partner of A, $type(n)$ is max.
If $player(n)$ is the opponent of the player A then $type(n)$ is min.
Values $a(n)$, $b(n)$, $c(n)$ at a node n are evaluated values by player A, B, and C respectively.
$suc(n)$ is a set of successors of a node n.
$payoff(n, d)$ is payoff at a leaf node n for a deal d. If it is not defined, $payoff(n, d) = *$.
We define max and min for $*$ as follows:
$max(*, x) = x$, $max(x, *) = x$, $min(*, x) = x$, $min(x, *) = x$.

[ABC game-tree search algorithm]To evaluate the score $s(m, [da, db, dc])$ in Step 2-2 of the ABC algorithm.

If node n is $suc(n0)$, corresponding to the move m at the root node,
return $a(n)$ as the $score(m, [da, db, dc])$.
If n is a leaf node, then
$a(n) = payoff(n, da)$,
$b(n) = payoff(n, db)$,
$c(n) = payoff(n, dc)$.
If $player(n)$ is 'A', then
$a(n) = max_{n' \in suc(n)} a(n')$,
$b(n) = max_{n' \in suc(n)} b(n')$,
$c(n) = max_{n' \in suc(n)} c(n')$.
If $player(n)$ is 'B', then $a(n)$ depends on the B's selection as follows:
$b(n) = type(n)_{n' \in suc(n)} b(n')$,
$a(n) = type(n)_{n' \in X} a(n')$
where X is the set of n' which gives the best value of $type(n)$ of $b(n')$,
$c(n) = type(n)_{n' \in suc(n)} c(n')$.
if $player(n)$ is 'C', then $a(n)$ depends on the C's selection as follows:
$c(n) = type(n)_{n' \in suc(n)} c(n')$,
$a(n) = type(n)_{n' \in X} a(n')$
where X is the set of n' which gives the best value of $type(n)$ of $c(n')$,
$b(n) = type(n)_{n' \in suc(n)} b(n')$.

6. EXAMPLES

[Example 1] ABC algorithm is applied to the example 1 in the section 3, where A, B and C are corresponding to East, South and West respectively.

Step 1: Construct a set Da of deals consistent with both the bidding and play of the deal thus far from East's point of view. We use a set D in figure 2 as Da.

Step 2: Evaluate each move in $M = \{C2, CK\}$ for each deal in $Da = \{d1, d2\}$. $d11@C2/S$ and $d11@C2/W$ in figure 6(a) are deals constructed from the points of B(South) and C(West) respectively assuming that East plays $C2$. $d11@CK/S$ and $d11@CK/W$ in figure 6(b) are deals constructed similarly assuming that the East plays CK.

The ABC game-tree search based on deals in figure 6 is illustrated in figure 7. At the node with the label N

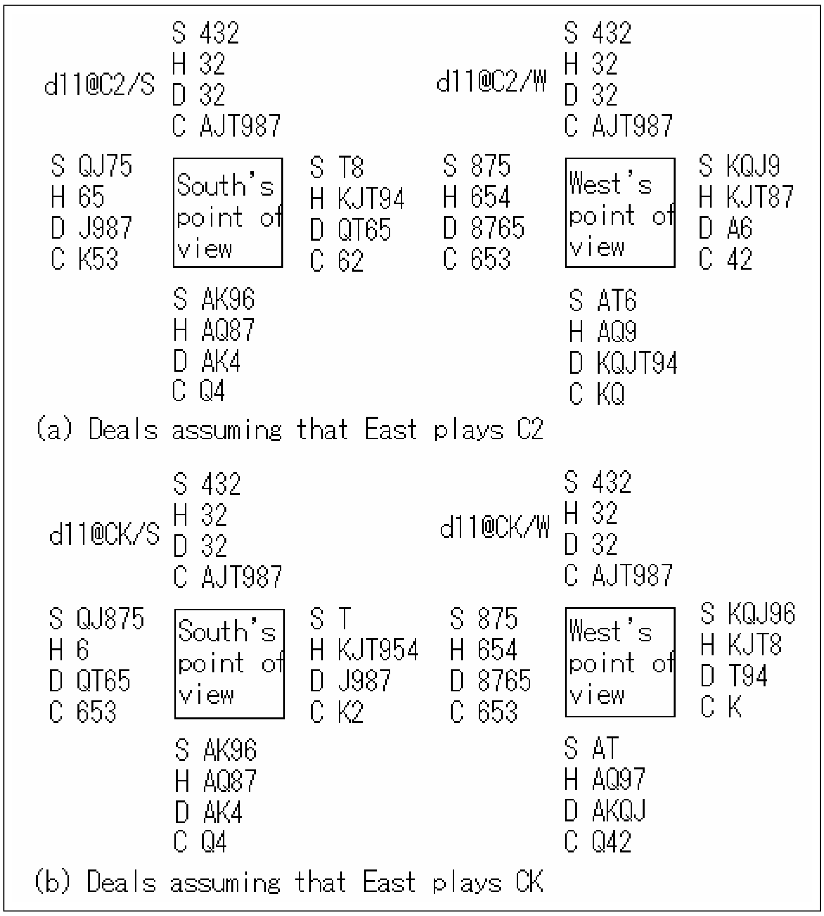

Figure 6: Deals constructed from the other players' point of view.

the declarer must select CJ or CA. Evaluations of CJ and CA depend on players' points of view. According to Declarer's point of view (the middle row of a triple) the evaluation values are -690 for CJ and 100 for CA. Declarer selects a card with minimum value, that is, CJ. It is good for East because the evaluation of CJ from East's point of view (the top row) is 100. As a result
$s(C2, d1, d11@C2/S, d11@C2/W) = 100$,
$s(CK, d1, d11@CK/S, d11@CK/W) = -660$.

Another sample of deals and the ABC game-tree search based on it is an follows.

$s(C2, d1, d12@C2/S, d12@C2/W) = -690$,
$s(CK, d1, d12@CK/S, d12@CK/W) = -660$.

If we finish Step2-2,

$$\begin{aligned}
s(C2, d1) &= s(C2, d1, d11@C2/S, d11@C2/W) + s(C2, d1, d12@C2/S, d12@C2/W)/2 \\
&= (100 - 690)/2 \\
&= -295 \\
s(CK, d1) &= s(CK, d1, d11@CK/S, d11@CK/W) + s(CK, d1, d12@CK/S, d12@CK/W)/2 \\
&= (-660 - 660)/2 \\
&= -660
\end{aligned}$$

Obtain $s(C2, d2)$ and $s(CK, d2)$ similarly.

In our experiments $s(C2, d2) = (-690 + 200)/2 = -245$, $s(CK, d2) = (-660 - 660)/2 = -660$.

Step 3: $s(C2, d1) + s(C2, d2) = -295 - 245 = -540$ $s(CK, d1) + s(CK, d2) = -660 - 660 = -1320$

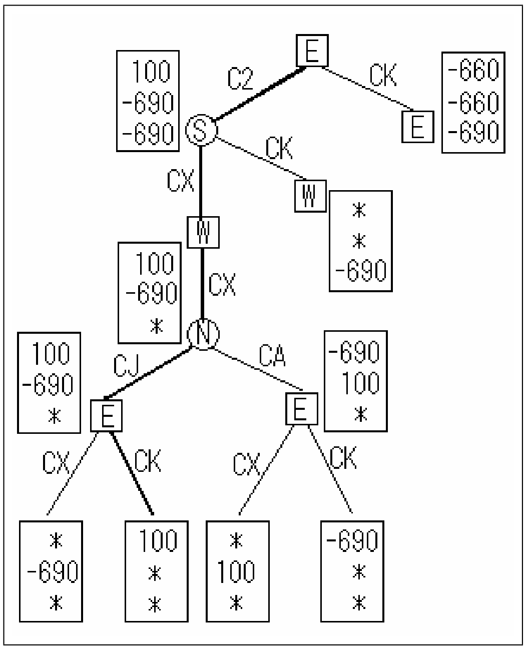

Figure 7: Game-tree search based on deals in figure 6.

So A(East) selects the maximum case, that is, $C2$ is selected.

If East has the nerve to play low, declarer is likely to finesse again and fails to make the contract. A bridge expert says (Kelsey, 1982), "Baring an honor card is much less dangerous than it looks, for it is hard for declarer to believe that you have done such a thing."

[Example 2]

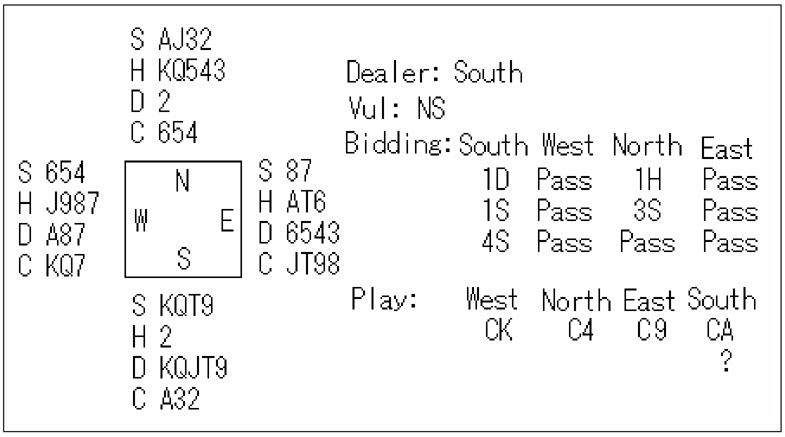

Figure 8: Deal, bidding and play history of Example 2.

West opens a Club against Four Spades. What card do you play after taking CA?

Experts (Gardner and Mollo, 2001) say, "Lead the highest of a sequence if you want opponents to take the trick, and the lowest if you hope that they will hold off." An agent in our computer bridge does not have such knowledge, but the agent uses the ABC algorithm to select the same card as an expert player of bridge.

A pair of deals constructed from South's point of view and West's point of view is illustrated in figure 9. Our new algorithm selects $D9$ expecting that West may not want to waste DA on such a small card.

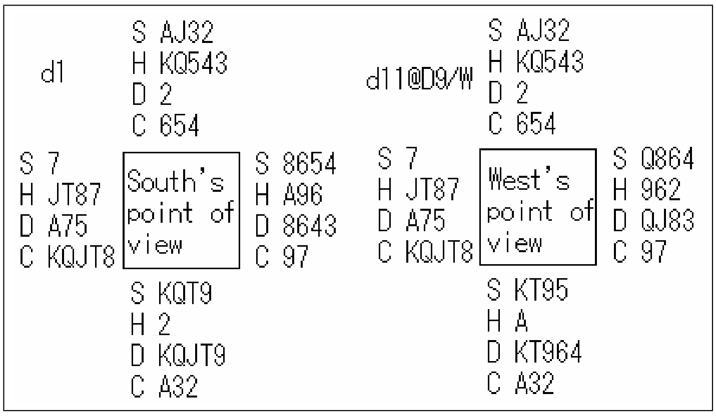

Figure 9: A pair of deals constructed from South's point of view and West's point of view.

7. CONCLUSIONS

We proposed an opponent model with imperfect information and a heuristic algorithm to take advantage of mistakes by the opponent. The model and the algorithm make deceptive plays possible. We made a prototype of computer bridge program and tested it for examples in the chapter 17 (Deceiving Defendeders) and the chapter 18 (Deceiving Declarer) of "Card Play Technique: The Art of Being Lucky" by V. Mollo and N. Gardener. None of the deceptions in the examples could not be played by existing computer bridges using the algorithm in the section 2, but most of the deceptive play problems in this book were solved by our prototype computer bridge. Sometimes our program estimated the risks of the deceptive play more than the authors of the book and did not select the recommended plays. Our prototype cannot play in real time. Now we are developing a new version of our computer bridge which is executed by 64 processors.

ACKNOWLEDGMENTS

The authors thank PARC of Imperial College for giving the license to use the constraint logic programming language ECL^{i}PS^{e}.

8. REFERENCES

Ando, T. and Uehara, T. (2000). *Reasoning by agents in computer bridge bidding.* Computer and Games 2000. Lecture Notes in Computer Science, Springer.

Frank, I. and Basin, D. (1998). *Search in games with incomplete information - A case study using bridge card play.* Artificial Intelligence. Vol.100, No.1-2.

Frank, I., Basin, D., and Matsubara, H. (1998). *Finding, Optimal Strategies for Imperfect Information Games.* AAAI.

Gardner, N. and Mollo, V. (2001). *Card play technique: the art of being lucky.* Batsford Bridge Books.

Ginsberg, M. L. (1996). *How computers willplay bridge.* The Bridge World.

Ginsberg, M. L. (2001). *GIB: Imperfect information in a computationally challenging game.* Journal of Artificial Intelligence Research.

Kelsey, H. (1982). *The tricky game: Deceptive plays to winning bridge.* Devyn Press Inc.

Levy, D. (1989). *The million pound bridge program.* Heuristic Programming in Artifical Intelligence —The First Computer Olympiad, Ellis Horwood.

Uehara, T. (1995). *Application of abduction to computer bridge.* Systems and Computers in Japan. Vol.26, No.9.