ISSN:2249-5789
M Dharmalingam et al, International Journal of Computer Science & Communication Networks,Vol 2(1), 140-146

Techniques and Programs Used in Bidding and Playing phases of Imperfect Information Game – an Overview

M DHARMALINGAM
Research Scholar, Department of Computer Science
D J Academy for Managerial Excellence
Bharathiar University, Coimbatore
Tamil Nadu – 641032, India
emdharma@gmail.com

Dr E CHANDRA
Director, Department of Computer Science and Applications
Dr SNS Rajalakshmi Arts and Science College
Bharathiar University, Coimbatore
Tamil Nadu – 641049, India

Abstract—**Bridge is an international imperfect information game played with similar rules all over the world and it is played by millions of players. It is an intelligent game; it increases creativity and knowledge of human mind. Many of the researchers analyses the Bridge bidding and playing phases, and they developed many programs for getting better results. The programs were performing well and it is also matter of time before the computer beats most human bridge players. As such, the researchers mainly focused on the techniques and computer programs which were used in bridge bidding and playing phases**.

Keywords **— Artificial Neural Network, Game of Bridge, DDBP, Bridge Bidding, Playing Bridge, Monte – Carlo Approach.**

I. INTRODUCTION

The game of bridge is a card game for four players, who are split into two pairs. Members of each pair sit opposite to one another, so that North-South forms one pair and East-West in the other. The bridge game is played using a standard deck of 52 cards divided into four suits, each containing 13 cards from A down to 2 (Ace, King, Queen, Jack, 10….2). Bridge is an imperfect information game, which requires multiple skills and knowledge [1]. Because no player knows exactly what moves other players are capable of making. Bridge being an imperfect information game, it is well defined in particular; the scoring systems give a way of assessing the strength of any resulting program. Artificial neural networks trained only on sample deals without presentation of any human knowledge or even rules of the game and used to the estimate the number of tricks to be taken by one pair of bridge players in the so-called Double Dummy Bridge Problem [2]. This paper briefly reviewed about the computer programs and languages which were used to play the bridge game.

II. ARTIFICIAL NEURAL NETWORKS IN BRIDGE GAME

Artificial Neural Network for making an opening bid in the game of contract bridge. The neural network captures the implicit mapping in bidding a bridge hand adopting a standard convention which acts as a guide or weak constraint on the mapping function. Artificial Neural networks acquire knowledge by a process of learning from example hands, it is possible to use hierarchy of networks in which each network recognizes only a limited number of bids. This network has bid correctly for about 90-92% of the test hands, which were not part of the training set. On the training hands the performance of the networks varies from 95% to 100% [3].Writing any computer program designed to generate or check proofs in any complicated mathematical system [4].

III. PHASES IN BRIDGE GAME

Bridge is also a multiplayer game with four players instead of two, although the players are paired into two teams. Bridge is a game of imperfect information; one pair of player's cards is hidden from the other pair of players. The game of bridge consists of two phases viz., bidding and playing. Bidding is the system to convey the maximum amount of information for cases which are more frequent [5]. Each player can bid in its turn and propose a trump suit and the level of the contract. The bids are called conventions and used for interaction between the partners. After bidding phase is complete, the side that suggested the final contract is the declaring side. Of the two members of declaring side, the one who first suggested the final trump suit is the declarer. Play begins with the player to the declarer's left leading to the first trick. The scoring depends on the number of tricks taken by the declarer and the final contract [6].

A. *Double Dummy Bridge Problem*

To estimate the number of tricks to be taken one pair of bridge players is called Double Dummy Bridge Problem (DDBP). A bridge problem presented for entertainment, in which the solver is presented with all four hands and is asked to determine the course of play that will achieve or defeat a particular contract. The partners of the declarer, whose cards are placed face up on the table and played by declarer. Dummy has few rights and may not participate in choices concerning the play of the hand [7].

Double-dummy Bridge is intermediate in complexity and the bridge tree is too large for an exhaustive search. The new programming technique Proof Checking Program was introduced, which has met with considerable success in solving no-trump double-dummy bridge problems and checking solutions which have been generated by humans. The proof- checking program is able to generalize the conditions under which the heuristic strategy works and construct rigorous proofs by non-exhaustive methods. When coupled with a heuristic program which makes intelligent guesses as to the location of the functions maximum, the combined prover- heuristic program can be used to locate the maximum of the function. This approach often results in a rigorous determination of the maximum of the function in far less time than would be required by an exhaustive search [4].

IV. BRIDGE BIDDING

The purpose of bidding in bridge is twofold. The primary purpose is to share information about cards with the partner and to 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 has been developed. Existing computer programs have simply matched possible bids against large data base giving their meanings, searching for that bid best matches the cards that the machines hold. Playing stage proved to be relatively easy for computers, and many programs have shown good playing performance [8].

A. *Redouble in Bridge Bidding*

The model based on natural linguistics and natural language processing technique was developed and tested on a case of bridge card play bidding. The computational linguistics language activity is described as a special process of communication between intelligent active agents. The active agent must know classical game theory and decision theory as well as reasoning of various kinds on various types of information sets. The machine bidding program was compared with human expert players and machine bidding program used were REDOUBLE. Deep analysis of human player performance and the game itself leads to quite good results in games with cooperation and incomplete information, REDOUBLE bids at a beginner level [1].

B. *Partial Information Decision Algorithm in Bridge Bidding*

Bridge bidding is one of the most difficult problems of game playing program, it involves four agents including cooperative (co-agents) and competitive agents (opp-agents). Partial Information Decision Making (PIDM) algorithm is most suitable for bridge bidding. The algorithm allows models to be used for opp-agents and co-agents and utilizing, a novel model-based Monte Carlo Sampling method to overcome the problem of hidden information. PIDM solves the problem implicitly through the simulation process. The simulation process covers the opposite case, where the player asks the partner for information. An "information – request" bid will be recognized down the look ahead tree by the simulated partner, who will return the required information through its own bid. This will reduce the space of possible partner hands and will yield more accurate Monte Carlo Simulation that will result in better bid [8].

C. Bridge Bidding Program (BRIBIP)

BRIBIP (Bridge Bidding Program) was developed by using POPCORN language. The Program is based on two distinct but closely related ideas. Primarily, close analysis of the bidding and the play to gain information previously unknown, secondarily, the utilization of such information in the creation of plan. The program bids approximately at the speed of a human player [9].

D. A-EBL Theory in Opening Bid

An important problem which frequently surfaces when applying explanation-based learning (EBL) to imperfect theories is the multiple inconsistent explanation problems. A solution to this problem would be an important step toward integration of explanation-based and similarity-based approaches to learning. The multiple inconsistent explanation problems occur when a domain theory produces multiple explanations for a training instance, only some of which are correct. Extension of explanation-based learning called *Abductive Explanation-Based Learning* (A-EBL) which solves the multiple inconsistent explanation problems. A-EBL makes use of additional positive examples and negative examples to choose among the multiple explanations. The behavior of A-EBL has been evaluated in two ways namely, formal evaluation and empirical evaluation. The empirical evaluation is in the domain of bridge opening bids. A-EBL inferring the exact conditions under which heuristic bidding rules should be applied from knowledge of those rules and examples. A-EBL is successful in improving the accuracy of the original domain theories. The experimental results were reported on an application of A-EBL to learning correct rules for opening bids in the game of contract bridge [10].

V. BRIDGE PLAYING

Once bidding has finished, the declarer needs to make the required number of tricks to achieve the contract or the defender need to stop the declarer. The partner of the opening bidder knows more about the combined strength of the two hands after hearing the opening bids and looking his own hand [7].

A. SPAN: System Program used in Bridge Playing

SPAN is a system for integrating a large variety of problem solving tactics into a coherent package. At the top level of the SPAN planning system there is a scheduler with an agenda of tasks. These tasks are domain independent or more accurately, their domain of expertise is planning. SPAN is the system proposed is the case with which it can handle more complex problems in the game of bridge [11].

B. Proof Planning technique in Bridge Playing

The system FINESSE was introduced, which forms plans for declarer play in the game of bridge. FINESSE is a modular body prolog code consisting of a *Pre planner*, *a Planner* and *Interpreter*. FINESSE generalizes the proof-planning technique, which produced a plan that consists of tree and tactics. Proof- planning ideas have been successfully applied to the problem of finding proofs for mathematical theorems, computer system configuration and to program verification. FINESSE proved to be a very competent system consistently identifying the highest probability of making the maximum number of tricks. FINESSE also produces correct analyses in situations where declarer holds so many cards in a suit [12].

C. Monte – Carlo Sampling: Technique in Bridge Playing

Monte - Carlo sampling is the technique used in imperfect information game. Monte -Carlo sampling consists of guessing the possible world and then finding a solution to the game tree for this complete information sub-problem. The result showed that, for very simple game trees the chance of finding the optimal strategy with Monte-Carlo sampling rapidly approaches zero as the number of moves in the game increases. The authors explained this sub-optimality by identifying the different kinds of errors that can arise and by analyzing their interplay and explained the sources of these errors in terms of strategy fusion and non-locality. A new algorithm, payoff- reduction minimaxing (*prm*) was formulated, which does not suffer from strategy fusion at MAX nodes and also significantly reduces the incidence of non-locality [13].

D. Complexity and Heuristics: Best Defence Model in bridge playing

Bridge problems have been formalized in a best defence model, which presents the strongest possible assumptions about the opponent. This is used by human players because modelling the strongest possible opponents provides a lower bound on the pay off that can be expected when the opponents are less informed. The new heuristics of *beta-reduction* and *iterative biasing* were introduced and represents the first general tree search algorithm capable of consistently performing at and above expert level in actual card play. The effectiveness of these heuristics, particularly when combined with payoff-reduction minimaxing to produced the *iprm-beta* algorithm. The problems from the game of bridge, *iprm-beta* actually makes less errors than the human experts that produced the model solutions. It thus represents the first general search algorithm capable of consistently performing at and above expert level on a significant aspect of bridge card play [14].

Payoff reduction minimaxing and new heuristics of *Beta Reduction* and *Interative Biasing* are modelled after perfect – information search techniques, to be applied to an imperfect information tree. The combination of *prm* and beta proved to solve defensive problems at a much higher rate, but the cost of which was a ten – fold performance reduction, making the use of these techniques unfeasible [14].

E. Forward pruning techniques in bridge playing

Forward pruning techniques may produce reasonably accurate result in bridge game. Two different kinds of game trees viz., N-Game trees and N-Game like trees were used to investigate, how forward pruning affects the probability of choosing the correct move. The results revealed that, minimaxing with forward pruning did better than ordinary minimaxing in cases where there was a high correlation among the minimax values of sibling nodes in a game tree. The result suggested that forward pruning may possibly be a viable decision-making technique in bridge games [15].

F. Bridge Baron Program in Bridge play

The Bridge Baron is generally acknowledged to be the best available commercial program for the game of Contract Bridge. The Bridge Baron program was developed by using Domain Dependent Pattern-matching Techniques which has some limitations. Hence there was a need to develop more sophisticated AI techniques to improve the performance of the Bridge Baron which was supplemented by its previously existing routines for declarer play with routine based on Hierarchical Task-Network (HTN) planning techniques. The HTN planning techniques used to develop game trees in which the number of branches at each node corresponds to the different strategies that a player might pursue rather than the different cards the player might be able to play [16].

G. SIBL Domain Theory

Game playing may be viewed as a planning problem, with an initial state, one or more goal states and set of operators that transforms one state into another. A plan consists of an ordered set of operators that transforms the initial state into a goal state. Some planning approaches, such as non-linear and hierarchical planners, rely on complete domain knowledge to derive plans [17]. Sequential Instant-Based Learning (SIBL) theory learns to select actions based upon sequence of consecutive states. The algorithms rely primarily on sequential observations rather than a complete domain theory. SIBL learns and makes decisions based on sequential dependency. Each approach incrementally constructs a Database (DB) from a set of problem solving experience, and uses it to make decisions. Once DB has been constructed, SIBL algorithm makes decisions based upon it. The authors tested each SIBL decision maker on the 936 withheld instances, noting how often it selected the correct card. If the program selected a card in a different suit, the result was scored as 0 (error). In case the program selected the identical card, the result was scored as 1 (correct). If the program selected a card in the same suit as the correct card and consecutively adjacent to the target feature, the result was also considered correct as scored as 1 [18].

VI. GIB (Ginsberg's Intelligent Bridge Player) in card play

GIB is a production program, expected to play bridge at human speeds. GIB used Monte Carlo methods exclusively to select an action based on the Double Dummy analysis. All other competitive bridge-playing programs have switched their card play to similar methods, although GIB's double dummy analysis is substantially faster than most of the other programs and its play are correspondingly stronger. If the bidding simulation indicates that the opponents are about to achieve a result much inferior 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 troublesome its opponents efforts to bid accurately. GIB's bidding is substantially better than that of earlier programs but not yet of expert caliber [19].

VII. PARTITION SEARCH IN DDBP

Double Dummy Bridge Problems are solved using a technique known as Partition Search. Bidding and playing rests on an ability to analyze bridge's perfect-information variant, where all of the cards are visible and each side attempts to take as many tricks as possible. Partition search was tested by comparing the number of nodes expanded using partition search and conventional methods. The result showed that, in both partition and conventional cases, a binary zero-window search was used to determine the exact value to be assigned to the hand, which the rules of bridge constrain to range from 0 to the number of tricks left. The hands generated using a full deck of 52 cards were not considered because the conventional method was in general incapable of solving them [20].

VIII. PYTHON PROGRAM IN DDBP

PYTHON, a rule based program for recognizing squeezes, an advanced strategy in the game of bridge. PYTHON is an application of logic programming which demonstrates the power of formulating a solution to a simple problem as a logic program. A powerful expert-system was created by translating textbook knowledge into a logic program. The development was facilitated by the knowledge engineer having considerable knowledge in the domain. Using PROLOG and an essentially top-down program development methodology led to readable code, which was easy to design and change, and is consistent with the way humans approach problem solving. The result reported that the example code is glosses over representational details, namely the exact form of the underlying data structures. The problem could be described without worrying about the implementation if low-level structures. The more advanced squeezes were implemented almost immediately. The cleanness of this approach was noticeable in the insignificant amount of code that was discarded during the development of the program. To observed the improvement in bridge play attributable to building PYTHON [21].

IX. CONCLUSION

Computer Bridge players have surpassed humans in the time they take to solve the double dummy problems and the best programs are attaining expert status as card players. Artificial neural networks turned out to be very effective in estimating the number of tricks to be taken by one pair of players in the Double Dummy Bridge Problem. Bridge being an imperfect information game, it is well defined in particular; the scoring systems give a way of assessing the strength of any resulting program. Various techniques were discussed with reference to bridge bidding and playing which were give better results. In future new techniques and algorithms will be developed for getting better results in bridge bidding and playing.

REFERENCES

[1] W. Jamroga, "Modeling Artificial Intelligence on a case of bridge card play bidding"In Proce. 8th International Workshop on Intelligent Information System, Poland, pp 276-277, 1999.

[2] Krzysztof Mossakowski and Jacek Mandziuk, "Weight Patterns in the network trained to solve Double Dummy Bridge Problem" in issues in Intelligent systems. pp 155-165. 2005

[3] B. Yegnanarayana, D. Khemani and M. Sarkar, "Hierarchical neural networks - An application in contract bridge bidding", In Proc. ICAUTO, Indore, 1995.

[4] E.R. Berlekamp, "Program for Double – Dummy Bridge Problems - A New Strategy for Mechanical Game Playing, Journal of the ACM, Vol.10, No.3, pp 357-364, 1963.

[5] B.Yegnanarayana,D.Khemani,and M.Sarkar, "Neural networks for contract bridge bidding," Sadhana, Vol.21, No.3, pp.395-413, 1996.

[6] Jacek Mandziuk, Text Book on "Knowledge Free and Learning Based Methods in Intelligent Game Playing", 2010

[7] Jacek Mandziuk and Krzysztof Mossakowski, "Neural networks compete with expert human players in solving the double dummy bridge problem" Proc. of 5th Int. Conf. on Computational Intelligence and games pp 117-124. 2009

[8] A.Amit and S.Markovitch, "Learning to bid in bridge" Machine Learning, Vol.63, no.3, pp 287-327, 2006.

[9] A.M.Stanier, "BRIBIP: A bridge bidding program" in Proce. 4th Int. Joint. AI, Georgia, pp 374-378, 1975.

[10] W.W. Cohen, "Abductive Explanation-Based Learning: A solution to the multiple inconsistent explanation problem" Machine Learning, Vol. 8 pp 167-219, 1992.

[11] D.L.S. Berlin, "Span: Integrating problem solving tactics" in Proce. International Joint Conference AI, Los Angeles, pp 1047-1051, 1985.

[12] I. Frank, D.A.Basin and H.Matsubara, " An adaption of proof-planning to declarer play in bridge" in Proce of the 10th European Conference in AI, New York, pp 72-76, 1992.

[13] I. Frank, D.A.Basin and H.Matsubara, "Monte-Carlo Sampling In Games With Imperfect Information: Empirical Investigation And Analysis" 1997.

[14] I. Frank and D.A. Basin, "Optimal play against best Defence: Complexity and Heuristics" in Lecture Notes in Computer Science Germany: Springer - Verlag, Vol.1558, pp 50-73, 1999.

[15] S.J.J. Smith and D.S. Nau, "An Analysis of Forward Pruning" in Proce of the National Conference on AI, pp 1386-1391, 1994.

[16] S.J.J. Smith, D.S. Nau and T.A. Throop, "Success in Spades: Using AI Planning Techniques to Win the World Championship of Computer Bridge" in Proce of the National Conference on AI, pp 1079-1086, 1998.

[17] K.J.Hendler, A.Tate, and M.Drummond, AI planning: Systems and Techniques. AI Magazine, 11(2): pp 61-77, 1990.

[18] S.L. Epstein and J.Shih, "Sequential Instance - Based Learning" in Lecture Notes in Computer Science, Germany: Springer-Verlag, Vol.1418, pp 442-454, 1998.

[19] M.L. Ginsberg, "GIB: Imperfect Information in a Computationally Challenging Game" Journal of Artificial Intelligence Research, Vol. 14, pp 303-358, 2001.

[20] M.L.Ginsberg, "Partition Search," in Proc.13th National. Conf. Artif. Intell Vol.2, pp 228-233, 1996.

[21] L. Sterling and Y. Nygate, "PYTHON: An Expert Squeezer" J. Logic Program, Vol.8, No1-2, pp 21-39, 1990.