Volume 4, No. 2, Jan-Feb 2013

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Supervised Learning in Imperfect Information Game

M Dharmalingam*
Research Scholar, Department of Computer Science
Sri Vasavi College, Erode
Bharathiar University, Coimbatore
Tamil Nadu – 638316, India
emdharma@gmail.com

R Amalraj
Associate Professor, Dept. of Computer Science
Sri Vasavi College, Erode
Bharathiar University, Coimbatore
Tamil Nadu – 638316, India
r_amalraj05@yahoo.co.in

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, which increases the creativity with multiple skills and knowledge of human mind. Because no player knows exactly what moves other players are capable of making. It is viewed as an imperfect information game. It is well defined in particular; the scoring system gives a way of assessing the strength of any resulting program. Artificial Neural Networks (ANN) is trained on sample deals without any idea of 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 (DDBP). Supervised learning was used in Back-propagation neural network to take best tricks in Double Dummy Bridge Problem. The target data was trained and tested using Log Sigmoidal transfer function and Hyperbolic Tangent Sigmoid functions, in the intermediate layer of the network. The results of both the functions seem to be more convincing regarding the results of the game. In summary, the study described in this paper provides a detailed comparison between Log Sigmoid transfer function and Hyperbolic Tangent Sigmoid functions which were used to train and test the data

Keywords: Double Dummy Bridge problem, BPN, Supervised learning, Log Sigmoid transfer function, Hyperbolic Tangent Sigmoid function.

I. INTRODUCTION

Card games are interesting for many reasons besides their connection with gambling. Among the huge variety of games, the researchers focus on those in which cards dealt randomly at the beginning of the game, with each player receiving a hand of cards that is not visible to the other players. Bridge, being a game of imperfect information, it is to be well defined with good decision making strategies [1]. The imperfect information games are in contrast with the perfect information game, in which the players are not having the complete knowledge of the game. Also one player does not know exactly what cards the opponent hold. The Card game is skillful and knowledgeable; it increases the creativity of the human mind. There are extremely powerful Artificial Neural Network (ANN) approaches in which playing agents are equipped with carefully designed evaluation functions.

Artificial Neural Networks (ANNs) are non – linear mapping structures based on the function of the human brain. Neural networks are type of artificial intelligence that attempts to imitate the way a human brain works rather than using a digital model [2]. The Feed-Forward Neural Networks (FFNN) are one of the most common types of neural network in use and these are often trained by the help of supervised learning supported by Back-propagation algorithm [3]. Many of the feed-forward neural networks were trained to solve the Double Dummy Bridge Problems (DDBP) in bridge game [4]. Among the various neural networks, in this paper we mainly focus Back-propagation Network (BPN) for training and testing the data. Neural networks were trained to solve the Double Dummy Bridge Problems in bridge game [3].

II. BRIDGE GAME

Nature of the card turned on to any individual or the fashion in which an individual selects or discards any particular card. Contract Bridge is card game which is simply known as bridge. As with virtually all card games, one of its primary features is that it is a game of imperfect information in that different players have different information about the actual state of the game. Bridge game is played by four players in two fixed pairs. Partners sit facing each other. Players are referred to according to their position at the table as North (N), East (E), South (S) and West (W), hence competing pairs are composed by (N, S) and (W, E) players. A standard 52 card pack is used. In each suit cards ranked from highest to lowest is the following Ace (A), King (K), Queen (Q), Jack (J), 10, 9, 8, 7, …3, 2. The whole deck is dealt out, so each player receives 13 cards. A bid specifies a number tricks and trump suit. The side which bids highest will try to win at least that number of tricks bid, with the specified suit as trumps. There are five possible trump suits: Spades (S), Hearts (H), diamonds (D), Clubs (C) and No-Trump which is the term for contracts played without a trump. After three consecutive passes, the last bid become the contract. The team who made the final bid will now try to make a contract. The first player of this team who mentioned the denomination of contract becomes the declarer. The declarer partner is known as Dummy [5]

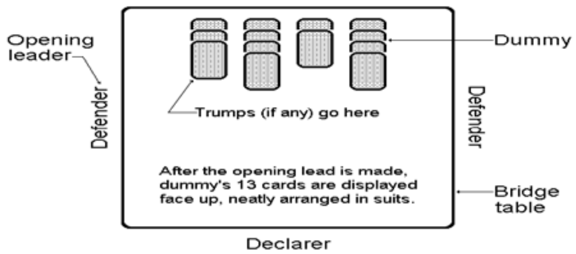


Figure 1. Bridge Table

A. Bridge Game Representation:

In bridge games, basic representation include value of each card (*Ace,…three,two*) and suit as well as the assignment of cards into particular hands and into public or hidden subsets, depending on the game rules. In the course learning, besides acquiring this basic information several other more sophisticated game features need to be developed by the learning system. Typical examples include the relevance of groups of cards of the same value of sequences of cards within one suit. A discussion on game representation is concluded by a case study related to the influence of deal representation on the effectiveness of NN approach to solving the DDBP [6]

B. Double Dummy Bridge Problem:

To estimate the number of tricks to be taken as 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].

Estimating hands strength is a decisive aspect of the bidding phase of the game of bridge, since the contract bridge is a game with incomplete information and during the bidding phase. This incompleteness of information force considering many variants of a deal cards distributions. The player should take into account all these variants and quickly estimate the expected number of tricks to be taken in each case [8]

C. The Bidding phase:

The bidding phase is a conversation between two cooperating team members against an opposing partnership. It aims to decide who will be the declarer. Each partnership uses an established bidding system to exchange information and interpret the partner's bidding sequence. Each player has knowledge of his own hand and any previous bids only.

A very interesting aspect of the bidding phase is cooperation of players in a North with South and West with East. In each, player is modeled as an independent, active agent that takes part in the communication process. The agent-based algorithm to use of achieve in appropriate learning, a bidding ability close to that of a human expert [9, 10, 11, 12, 13].

D. The Play Phase:

In the game, the play phase seems to be much less interesting than the bidding phase. Artificial Intelligence (AI) approaches tried to imitate human strategy of the play by using some “tactics”. The new system was able to find a strategy of play and additionally a “human” explanation of it [14, 15].

The player to the left of the declarer leads to the first trick and may play any card. Immediately after this opening lead, the dummy's cards are exposed. Play proceeds clockwise. Each of the other three players in turn must, if possible, play a card of the same suit that the leader played. A player with no card of the suit led may play any card. A trick consists of four cards, one from each player, and is won by the highest trump in it, or if no trumps were played by the highest card of the suit led. The winner of a trick leads to the next and may lead any card. Dummy takes no active part in the play of the hand and is not permitted to offer any advice or comment on the play. Whenever it is dummy's turn to play, the declarer must say which of dummy's cards is to be played, and dummy plays the card as instructed. Finally, the scoring depends on the number of tricks taken by the declarer team and the contract [16].

III. SOFT COMPUTING

Soft computing involves partnership of several fields, the most important being Artificial Neural Networks (ANN), Fuzzy Logic (FL), Genetic Algorithm (GA) and Evolutionary Computations (EC). Among the above fields, Artificial Neural Networks used to solve the Double Dummy Bridge Problem in Contract Bridge [11].

IV. ARTIFICIAL NEURAL NETWORKS

Neural networks are predictive models loosely based on the action of biological neurons. An artificial network is a highly simplified model of the structure in the biological neural network. Artificial Neural Network consists of several processing units which are interconnected according to some topology to accomplish a pattern classification task. An Artificial Neural Network is configured for a specific application, such as pattern recognition or data classification through learning process. ANNs are nonlinear information processing devices, which are built from interconnected elementary processing devices called neurons [17].

ANNs are based on Artificial Intelligence (AI) techniques that attempt to imitate the way a human brain works. A neural network is a massively parallel – distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. Either humans or other computer techniques can use neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, to extract patterns and detect trends that are too complex to be noticed. A trained neural network can be thought of as an expert in the category of information it has been given to analyses. [18]

A. Activation Functions:

The activation function is used to determine the output response of a neuron. The sum of the weighted input signal is applied with an activation to obtain the response. For neurons in same layer, same activation functions are used. There may be linear as well as Non - Linear activation functions. The Non – Linear activation functions are used in a Multilayer Neural Net

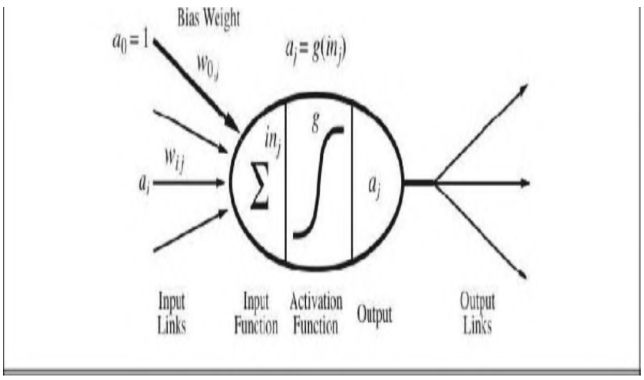


Figure 3. Simple mathematical model for activation function

Neural networks are composed of nodes or units connected by directed links. Fig 3. denotes, a link from unit *j* to unit *i* serves to propagate the activation a_i from *j* to *i*. Each link also has a numeric weight $W_{j,i}$ associated with it, which determine the strength and sign of the connection. Just as in linear regression models, each unit has a dummy input a_u=1 with an associated weight $W_{i,j}$. Each unit *i* first computes a weighted sum of its inputs [19]

$$ini = \sum_{j=0}^{n} Wj,i\ aj$$

Then it applies an activation function *g* to this sum to derive the output

$$ai = g(ini) = g \sum_{j=0}^{n} Wj,iaj \qquad (1)$$

There are several activation functions namely as follows Identity function, Binary step function, Bipolar step function, Sigmoidal functions and Ramp functions.

Among the above activation functions, the Sigmoidal functions are widely used in Back-propagation nets. Because of the relationship between the value of the functions at a point and the value of derivative at that point which reduce the computational burden during training. There are two types of sigmoidal functions viz., binary sigmoidal function and bipopular sigmoidal functions [20]. In this paper we have focused only Binary Sigmoidal function. If the network uses a binary it is better to convert it to bipolar form and use the bipolar sigmoid function activation function or hyperbolic tangent function shown in the Fig 4.

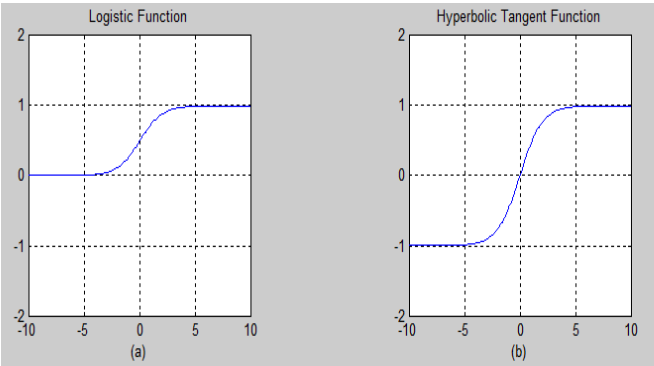


Figure 4. Logistic function and Hyperbolic Tangent function

B. Learning:

Learning or training is a process by means of which a neural network adopts itself to stimulus by making proper parameter adjustments, resulting in the production of desired response. The Learning in an Artificial Neural Networks can be generally classified into three categories viz., Supervised Learning, Unsupervised Learning and Reinforcement Learning. Among these three categories we mainly focused on Supervised Learning in this paper.

In Artificial Neural Networks following the supervised learning, each input vector requires a corresponding target vector, which represents the desired output. The input vector along with the target vector is called *training pair*. In supervised learning, a supervisor is required for error minimization. Hence the network trained by this method is said to be using supervised learning methodology. In supervised learning, it is assumed that the correct target output values are known for each input pattern [11].

C. Architecture of Back-propagation Network (BPN):

A Back-propagation network is a multilayer, Feed-forward Neural Network (FFNN) with an input layer, a hidden layer and an output layer which is shown in the Fig 5. The neuron in the hidden and the output layers has biases which are connections from units whose output is always is 0 to 1.

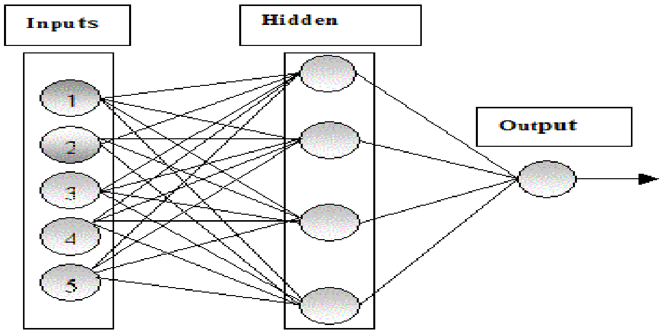


Figure 5. Architecture of BPN

BPN is a multi – layer forward network using extended gradient-descent based delta-learning rule, commonly known as Back-propagation rule. Back-propagation provides a computationally efficient method for changing the weights in a feed-forward network, with differentiable activation function units to learn a training set of input output patterns. The Back-propagation network implements the generalized delta rule. A gradient – descent method, it minimize the total squared error of the output computed by the network. The network is trained by supervised learning method [21].

D. BPN training algorithm:

The training algorithm of Back-propagation invokes four stages viz., Initialization of weights, Feed-forward, Back-propagation of errors and Updating of the weights and bias.

The detailed algorithm is shown below in Fig 6.

function BACK-PROP-LEARNING(*example*, *network*) **returns** a neural network
inputs: *example*, a set of examples, each with input vector **x** and output vector **y**
network, a multilayer network with L layers, weights $W_{j,i,}$ activation function *g*
repeat
for each *e* **in** *examples* **do**
for each node *j* in the input layer **do** $aj \leftarrow xj\ [e]$
for ℓ = 2 **to** *M* **do**
$\text{ini} \leftarrow \sum j\ Wj,i\ \ aj$
$ai \leftarrow g\ (ini)$
for each node *i* in the output layer **do**
$\Delta i \leftarrow g'\ (inj)\ \times\ (yi\ [e] - ai)$
for ℓ = *M* – 1 *to* 1 *do*
for each node *j* in layer ℓ **do**
$\Delta j\ \leftarrow g'\ (inj)\ \sum i\ Wj,i\ \ \Delta i$
for each node *i* in layer ℓ + 1 **do**
$Wij \leftarrow Wj,i\ \ + \alpha\ \times\ aj\ \times\ \Delta i$
until some stopping criterion is satisfied

return NEURAL-NET-HYPOTHESIS(*network*)

Figure 6. The back-propagation algorithm for learning in BPN networks

For the mathematically inclined, the back-propagation equations are derived from first principals. The squared error on a single example is defined as

$$E = \frac{1}{2}\sum i\ (yi - \alpha i)\,2 \qquad (2)$$

Where the sum is over the nodes in the output layer. To obtain the gradient with respect to a specific weight $W_{j,i}$ in the output layer, one should need only expand out the activation $\ _i$ as all other terms in the summation are unaffected by $W_{j,i}$:

$$\frac{\partial E}{\partial Wj,i} = -(yi - \alpha i)\frac{\partial ai}{\partial Wj,i} = -(yi - \alpha i)\frac{\partial g(ini)}{\partial Wj,i}$$

$$= -(yi - \alpha i)\,g'(ini)\frac{\partial ini}{\partial wj,i} =$$

$$-(yi - \alpha i)\,g'(ini)\frac{\partial}{\partial wj,i}\left(\textstyle\sum_j Wj,i\ \alpha j\right)$$

$$= -(yi - \alpha i)\,g'(ini)\,a_j \qquad (3)$$

With $\ _i$ defined as before. To obtain the gradient with respect to the $W_{k,j}$ weights connecting the input layer to the hidden layer, to keep the entire summation over i because each output value a_i may be affected by changes in $W_{k,j}$. Activations of a_j also be expanded and the derivative operator propagates back through the network

$$\frac{\partial E}{\partial Wk,j} = -\sum_i (yi - ai)\frac{\partial ai}{\partial Wk,j} = -\sum_i (yi - ai)\frac{\partial g(ini)}{\partial Wk,j}$$

$$= -\sum_i (yi - ai)\,g'(ini)\frac{\partial ini}{\partial Wk,j} = -\sum_i \Delta i\frac{\partial}{\partial Wk,j}\left(\sum_j Wj,i\ ai\right)$$

$$= -\sum_i \Delta i\,Wj,i\frac{\partial ai}{\partial Wk,j} = -\sum_i \Delta i\,Wj,i\frac{\partial g(ini)}{\partial Wk,j}$$

$$= -\sum_i \Delta i\,Wj,i\ g'(ini)\frac{\partial ini}{\partial Wk,j}$$

$$= -\sum_i \Delta iWj,i\ g'(ini)\frac{\partial}{\partial Wk,j}\left(\sum_k Wk,j\ ak\right)$$

$$= -\sum_i \Delta iWj,i\ g'(ini)ak = -ak\ \Delta j \qquad (4)$$

Where $\ _j$ is defined as before. The update rules were obtained earlier from intuitive considerations. It is also clear that the process can be continued for networks with more than one hidden layer, which justifies the above general algorithm [19]

V. NEURAL NETWORKS AND BRIDGE GAME

The game domains NNs are a popular way to represent an evaluation function. A neural architecture used in this task is typically a feed-forward network implemented as an MLP (Multilayer Perceptron) or its modification. The topology of the network is usually tuned to the specific game. The MLP is usually trained with a variant of back-propagation method. Typically, the learning process is based on successive presentation of the training vectors according to some fixed, randomly chosen order, in an iterative multiple-epoch manner [22].

VI. THE DATA REPRESENTATION OF GIB

The data used in this game of DDBP was taken from the Ginsberg's Intelligent Bridge (GIB) Library. The data created by Ginsberg's Intelligent Bridge player, the winner of computer bridge champion in 1998 and 1999 [23]. In our research for implementing GIB library data we used MATLAB 2008a.

The GIB library includes 7,17,102 deals and for each of them provides the number of tricks to be taken by N S pair for each combination of the trump suit and the hand which makes the opening lead. Together there are 20 numbers of each deal i.e. 5 trump suits by 4 sides. Here 5 trump suits are No-trumps, spades, Hearts, Diamonds and Clubs, No-trump which is the term for contracts played without trump. Four sides are West, North, East and South. So North and South are partners playing against East and West [24].

A. *Implementation:*

In this paper 20 sample data were used for training in MATLAB R2008a. Only one output neuron was used and in order to get the result, decision boundaries were defined the range of 0.1 to 0.9denoting particular number of tricks. The rank of the card was transformed using a Uniform Linear Transformation to the range from 0.1 to 0.9 with biggest values to lowest values. The decision boundaries were defined a prior and target ranges for all possible number of tricks from 0 to 13 were pair wise equal length.

Gradient descent training function was used to train the data and gradient descent weight/bias learning function was used for learning the data. For training and learning the data, two transfer functions viz., Log Sigmoid transfer function and Hyperbolic Tangent Sigmoid functions were used.

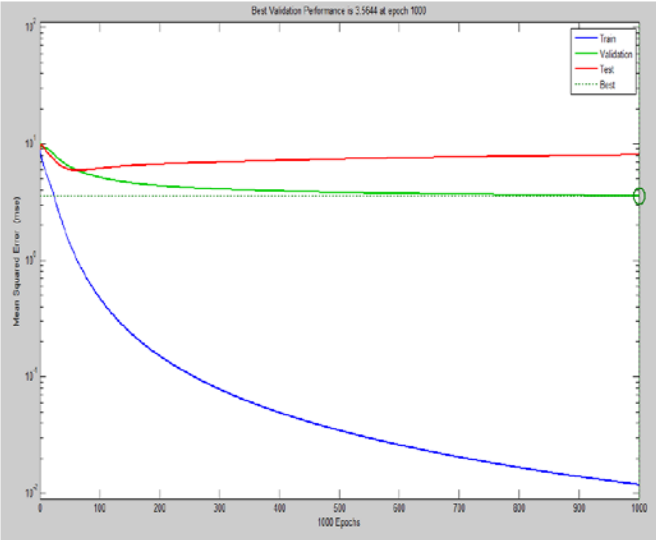
In this paper we compared the output data results received from these two transfer functions along with GIB library target data and got the following results.

Table 1. Comparison of target tricks with Log Sigmoid transfer function and Hyperbolic Tangent Sigmoid function

S. No	GIB library target data	Log Sigmoid transfer function	Hyperbolic Tangent Sigmoid function
1	09	10.115	09.955
2	10	10.629	07.144
3	12	10.744	06.758
4	10	06.167	10.026
5	09	08.692	10.970
6	06	07.050	09.890
7	09	10.767	07.434
8	06	07.132	08.521
9	10	10.375	09.187
10	07	06.214	07.045
11	12	06.843	08.290
12	07	09.437	06.443
13	06	09.265	07.386
14	11	06.152	07.979
15	06	07.984	07.015
16	06	06.647	07.740
17	10	10.381	06.247
18	08	11.340	08.187
19	07	09.993	08.123
20	10	11.041	10.450

B. Representation of Target Tricks:

The results presented in the table shown that the comparison of target tricks along with Log Sigmoid transfer function and Hyperbolic Tangent Sigmoid function. While comparing the trained data along with target data, the result indicated that, trained data shown significantly better results in both transfer functions, which minimized the total mean square error shown in Fig 7.



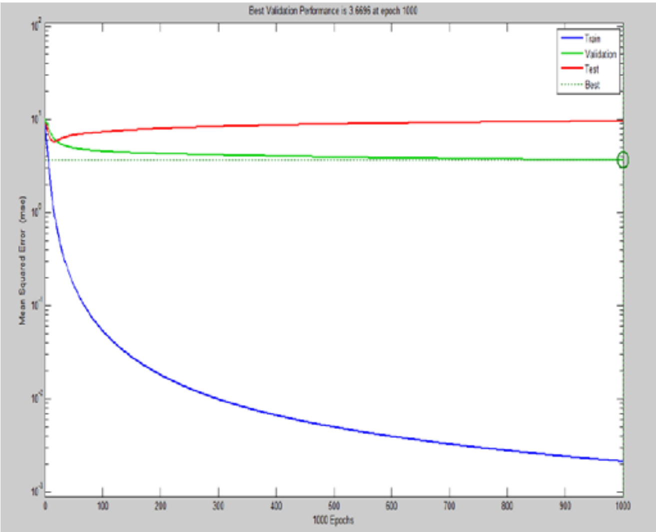


Figure 7. Performance comparison of Mean square representation of both Log Sigmoid Transfer function and Hyperbolic Tangent Sigmoid functions

The result also revealed that, Log Sigmoid transfer function and Hyperbolic Tangent Sigmoid functions were compared with each other and Log Sigmoid transfer function was given significantly superior results than Hyperbolic Tangent Sigmoid function.

VII. CONCLUSION

Back-propagation algorithm was used which minimizes the error in output while implementing the data in MATLAB R2008a. Gradient descent method is used to minimize the total mean square error of the output computed by the neural network.

These extracted numerical features may be compared with human hand scoring systems and potentially lead to development of some new ideas in human bridge playing and be helpful for semiprofessional players for improving their bridge skills.

VIII. REFERENCES

[1]. I. Frank, D. A. Basin, and H. Matsubara, “Finding Optimal Strategies for Imperfect Information Games,” in Proceeding of AAAI, pp 500-507, 1998.

[2]. Kishan Mehrotra, Chilukuri, Mohan K and Sanjay Ranka “Elements of artificial Neural Networks”, pp 65-96, 1996.

[3]. S.N Sivanandan and S.N Deepa, “Principles of Soft Computing”, Wiley India Ltd, First Edition, 2007.

[4]. Jacek Mandziuk and Krzysztof Mossakowski, “Looking Inside Neural Networks Trained to Solve Double-Dummy Bridge Problems,” Int. Proceeding 5th Game – On Computer Games: Artif. Intell., U.K. pp: 182-186, 2004.

[5]. W H Root, The ABCs of Bridge. Three Rivers Press, 1998.

[6]. Jacek Mandziuk and Krzysztof Mossakowski, “Example – based estimation of hands strength in the game of bridge with or without using explicit human knowledge”, In Proc. IEEE Symp. Comput. Intell. Data mining, pp.413-420, 2007.

[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]. Krzysztof Mossakowski and Jacek Mandziuk, “Learning without human expertise: A case study of Double Dummy Bridge Problem”, IEEE Transactions on Neural Networks, Vol.20, No.2, pp 278-299, 2009.

[9]. 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.

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

[11]. T. Ando, Y. Sekiya and T. Uehara, “Partnership bidding for computer bridge,” Systems *and Computers in Japan*, vol. 31, No. 2, pp. 72–82, 2000.

[12]. T. Ando and T. Uehara, “Reasoning by agents in computer bridge bidding,” in *Computers and Games*, Vol. 2063, pp. 346–364, 2001.

[13]. T. Ando, N. Kobayashi and T. Uehara, “Cooperation and competition of agents in the auction of computer bridge,” *Electronics and Communications in Japan, Part 3*, vol. 86, no. 12, pp. 76–86, 2003.

[14]. D. Khemani, “Planning with thematic actions.” in *AIPS*, pp. 287–292, 1994.

[15]. I. Frank, and D. A. Basin, “Strategies explained” in Proceeding 5th Game programming Workshop in Japan. pp 1-8, 1999.

[16]. Ali Awada, May Dehayni and Antoun Yaacoub, “An ATMS-Based Tool for Locating Honor Cards in Rubber Bridge” Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No.5, 2011.

[17]. S.N Sivanandan and M Paulraj, "Introduction Artificial Neural Networks", Vikas Publishing House Private Limited, 2011.

[18]. B. Yegnanarayana, "Artificial Neural Networks" Printice Hall. 2010.

[19]. J.Stuart , Russell and Peter Norvig " Artificial Intelligence A- Modern Approach" Chapter 18, pp 727-736, Third Edition., USA Printice Hall 2009.

[20]. S.N Sivanandam ,S Sumathi, S.N Deepa, "Introduction to Neural Network Using MATLAB 6.0", Tata McGraw-Hill Publishing Company Limited, New Delhi 2008.

[21]. Satish Kumar "Neural Networks A Class Room Approach", Tata McGraw-Hill Education Private Limited, 2011.

[22]. Jacek Mandziuk, "Knowledge-free and Learning – Based methods in Intelligent Game Playing" Book Chapter 5, pp 53 – 58, 2010.

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

[24]. 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.

