IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol-6, Special Issue-4, May 2018 E-ISSN: 2347-2693

Supervised Learning Architecture for Solving Double Dummy Bridge Problem

Dharmalingam Muthusamy

Dept.of Computer Science, Bharathiar University Arts and Science College, Modakkurichi, Erode, Tamil Nadu, India.

*Corresponding Author: emdharma@gmail.com, Tel.: +91 9942155553

Available online at: www.ijcseonline.org

Abstract— The bridge game is one of the most generally known card games comprising many mesmerizing aspects, such as bidding, playing and winning the trick including estimation of human hand strength. The harmonizing input data based on the human knowledge of the game to improvement the quality of tricks. The bridge game classification under a game of imperfect information is to be equally well-defined. The decision made on any stage of the game is simply based on the assessment that was made on the immediate preceding stage. The intelligent game of bridge incompleteness of information, the real spirit of the card game in proceeding further deals of the game are taking into many forms especially during the distribution of cards for the next deal. The cascade correlation neural network architecture with supervised learning implemented in resilient back - propagation algorithm to train data and therefore to test data it is together along with the bamberger point count method and work point count methods.

Keywords— Cascade-correlation neural network, Resilient back-propagation algorithm, Bridge game, Double dummy bridge problem, Bamberger point count method, Work point count method.

I. Introduction

The contract bridge game is a trick-taking card game, where - on each of a number of deals, the different sides first compete in a bidding auction for the right to approach the contract for that deal, with the side winning the auction being known as the declaring side. The contract is a swap of the right to create which suit is a trump for a responsibility to win at least the number of tricks individual by the highest bid [1, 2].

The neural networks are based on non-linear activation function approximations which make them appropriate for most of the applications specifically games. The number of tricks to be taken by a one pair of bridge players is called double dummy bridge problem (DDBP). In this paper is generally on confirmation of neural network abilities to be trained the evaluation function to solve the Double Dummy Bridge Problems (DDBP) fairly than decision the solution to the problem [3]. There are a group of feed forward neural networks available which are trained in the bridge game [4,5,6] and have been dignified in the best justification model, with the strongest feasible assumptions about the opponent [7,8]. This is used by human players for the reason that modeling the strongest potential opponents provides an inferior bound on the pay off that can be predictable when the opponents are less knowledgeable.

The two methods known as the point count method [9,10] and the distributional point method [11] to evaluate the hand strength during the game, is an special and a fashionable method used to bid a final contract in a bridge game [12,13]. The bidding and playing are the two phases of contract bridge and both should be played optimally well to gain the best possible result. It can be matched with that of the training and testing phases of artificial neural network. The best possible training of the neurons can alone reflect the level of accuracy of inference in the testing phase of the network [14,15].

The overview of this paper is as follows. Section II provides a brief description of the game of bridge followed by a definition of the double dummy bridge problem. The Section III details about the cascade-correlation neural network architecture in a diagrammatical representation with resilient back-propagation algorithm. The process flow of the Bamberger point count method and Work point count method in bridge game, 52-32-1 through the cascade-correlation neural network model with implementation in Section IV. The experimental results and discussion with the defined architecture is discussed with a sample of data in Section V. The Section VI, conclusions of this work.

II. GAME OF BRRIDGE

The bridge game is one of the well-known card games played in the worldwide. The bridge game with randomly dealt cards, which are makes it is also a game of chance. In which, it is more accurately, a tactical game with inherent uncertainty, imperfect information and classified communication. The impulsiveness and imperfect information built-in the game is representation intelligence of number of the researchers and Computational Intelligence (CI) methods are applied to focus generally on the feature of learning in the game playing methods [16,17,18]. The bridge is a partnership game requiring four players, each player sits opposite to his partner and it is conventional to refer to the players according to their position at the table as North, East, South and West, so North and South are partners playing against East and West. It is played with a standard deck of 52 playing cards, where one of the players deals all of the cards, 13 to each player, in clockwise turning round, starting with the player to the left of the dealer. In bridge games, basic representation includes value of each card as (Ace (A), King (K), Queen (Q), Jack (J), 10, 9, 8, 7, 6, 5, 4, 3, 2) and suit as (♠ (Spades), ♥ (Hearts), ♦ (Diamonds), ♣(Clubs), depending on the game rules[19,20,21,22,23,24].

A.The bidding and playing phases

The game then proceeds through bidding and playing phases and the principle of the biding phase is the classification of trumps and declarer of the contract [25]. The playing phase consists of 13 tricks with each player causative one card to each trick in a clockwise fashion with another level bid to decide who will be the declarer. The side which bids the highest will try to win at least that number of tricks bid, with the specified suit as trumps. There are 5 possible trump suits: spades (♠), hearts (♥), diamonds (♦), clubs (♣) and 'no-trump' which is the term for contracts played without a trump. In continue three successive passes the last bid becomes the contract. The bidding phase is an exchange between two cooperating team members in opponent to an opposite partnership which aims to choose who will be the declarer [26]. Each partnership uses an recognized bidding method to exchange information, understand the partner's bidding progression as each player has comprehension of his own hand and an remarkable aspect of the bidding phase is the cooperation of players in North with South and West with East [27,28].

The play phase seems to be much less inspiring than the bidding phase. The play takings clockwise and each of the other three players in turn must, play a card of the same suit that the human being in-charge played [29]. A player with no card of the suit may play any card of his selection. The winner of a trick leads consequently with any card as the dummy takes no dynamic role in the play and not allowed to offer any suggestion. Finally, the scoring depends on the number of tricks taken by the declarer team and the contract [30,31,32,33,34,35,36,37].

B. Point Count Methods

The proposed methods are Bamberger Point Count Method (BPCM) and Work Point Count Method (WPCM) are surprising, most important and fashionable method which are used to bid a final contract in bridge game. The bamberger is a point count method that requires 52 points to produce a probable slam on power alone. The bamberger point count method which scores 7 point for Ace, 5 point for King, 3 point for Queen and 1point for a Jack is followed in which no points are counted for 10 and below. The Work Point Count Method (WPCM) which scores 4 points for Ace, 3 points for King, 2 points for Queen and 1point for a Jack is followed in which no points are counted for 10 and below. During the bidding phase of contract bridge, when a team reaches the combined score of 26 points, they should use WPCM for getting final contract and out of thirteen tricks in contract bridge, there is a possibility to make use of eight tricks by using WPCM.

III. CCNN ARCHITECURE

The game of bridge has not attracted much attention of the researchers of soft computing; there are many interesting nuances of the game of bridge from the point of view of artificial neural networks. The cascade-correlation architecture was introduced by [38] defined with number of input neurons, output neurons represented in the input layer and output layer correspondingly and hidden neurons are added to the network depends on the inevitability of the precision of the results. The cascade-correlation begins with a minimal network, then perfunctorily trains and adds new hidden units one by one, creating a multi-layer pattern as in Figure 1. The previously a new hidden unit has been added to the network, its input-side weights are frozen. The new hidden neuron is added in each training set and weights are adjusted to minimize the magnitude of the correlation between the new hidden neuron output and the residual error signal on the network output that has to be eliminated. The cascade-correlation neural network architecture has many rewards over its complement, as it learns at a faster rate; the network determines its own element and topology.

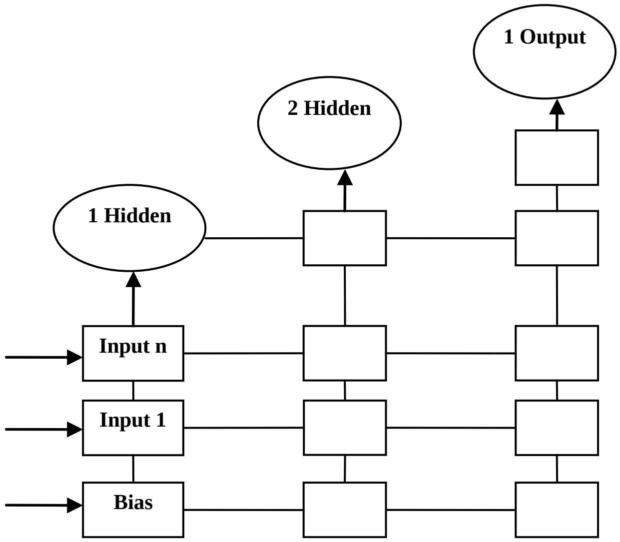

Figure 1 The architecture of cascade-correlation neural network.

The neuron to be added to the presented network can be made in the following two steps: (i) The candidate neuron is connected to all the input and hidden neurons by trainable input associations, but its output is not associated to the network. Then the weights of the candidate neuron can be trained while all the other weights in the network are frozen. (ii) The candidate is associated to the output neurons and then all the output associations are trained. The whole progression is repeated until the desired network accurateness is obtained. In Equation (1) the correlation parameter 'S' defined as below is to be maximized.

$$S = \sum_{0=1}^{O} \left| \sum_{p=1}^{P} (V_p - \bar{V})(E_{po} - \overline{E_o}) \right| \quad \textbf{(1)}$$

where 'O' is the number of network outputs, 'P' is the number of training patterns, 'V_p' is output on the new hidden neuron and 'E_{po}' is the error on the network output. In the Equation (2) the weight change for the new neuron can be found by gradient descent rule as

$$\Delta_{w_i} = \sum_{0=1}^{O} \sum_{p=1}^{P} \sigma_o \left(E_{po} - \overline{E_o}\right) f_{p}' \, x_{ip} \quad \textbf{(2)}$$

The output neurons are trained using the generalized delta learning rule for faster convergence in resilient back-propagation algorithm. The each hidden neuron is trained just once and then its weights are frozen. The network learning procedure is fulfilled when satisfy results are obtained. The cascade-correlation architecture needs only a forward sweep to compute the network output and then this in sequence can be used to train the candidate neurons.

A. The Resilient back-propagation algorithm

The resilient back-propagation algorithm is a local adaptive learning scheme, performing supervised batch learning in cascade-correlation neural network. The basic principle of resilient back-propagation algorithm is to eliminate the harmful influence of the size of the partial derivative on the weight step. As significance, only the sign of the derivative is considered to indicate the direction of the weight update. The Equation (3) for each weight, if there was a sign change of the partial derivative of the total error function compared to the last iteration, the update value for that weight is multiplied by a factor η−, where 0 <η− < 1. If the last iteration produces the same sign, the update value is multiplied by a factor of η +, where η+ > 1.

The update values are calculated for each weight in the above manner, and finally each weight is changed by its own update value, in the opposite direction of that weight's partial derivative. This is to minimize the total error function. η+ is empirically set to 1.2 and η− to 0.5.

The above description mathematically we can start by introducing for each weight its individual update value (t), which exclusively determines the magnitude of the weight-update. This update value can be expressed mathematically according to the learning rule for each case based on the observed behavior of the partial derivative during two successive weight-steps by the following formula:

$$\Delta_{ij}(t) = \begin{cases} \eta^{+}.\Delta_{ij}(t-1), & if \ \dfrac{\partial E}{\partial w_{ij}}(t).\dfrac{\partial E}{\partial w_{ij}}(t-1) > 0 \\ \eta^{-}.\Delta_{ij}(t-1), & if \ \dfrac{\partial E}{\partial w_{ij}}(t).\dfrac{\partial E}{\partial w_{ij}}(t-1) < 0 \\ \Delta_{ij}(t-1), & else \end{cases} \quad \textbf{(3)}$$

Where 0< <1< .

The Equation (4) it is obvious that every time the partial derivative of the equivalent weight varies its sign, which indicates that the last update was large in magnitude and the algorithm has skipped over a local minima, the update-value (t) is decreased by the factor η−. If the derivative holds its sign, the update - value will to some extent increase in order to speed up the convergence in shallow areas. When the update-value for each weight is settled in, the weight-update itself tracks a very simple rule. The Equation (5) that is if the derivative is positive, the weight is decreased by its update value, if the derivative is negative, the update-value is added.

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}(t), & if \quad \frac{\partial E}{\partial w_{ij}}(t) > 0 \\ \Delta_{ij}(t), & if \quad \frac{\partial E}{\partial w_{ij}}(t) < 0 \\ 0, & else \end{cases} \quad \textbf{(4)}$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad \textbf{(5)}$$

However, there is one exception. The Equation (6) if the partial derivative changes sign that is the previous step was too large and the minimum was missed, the previous weight-update is reverted

$$if \frac{\partial E}{\partial w_{ij}}(t) . \frac{\partial E}{\partial w_{ij}}(t-1) < 0 \quad \textbf{(6)}$$

due to that backtracking weight-step, the derivative is assumed to change its sign once again in the following step. In order to avoid a double penalty of the update-value, there should be no adaptation of the update-value in the succeeding step. In practice this can be done by setting in the update-rule above.

The Equation (7) partial derivative of the total error is given by the following formula:

$$\frac{\partial E}{\partial w_{ij}}(t) = \frac{1}{2}\sum_{p=1}^{p} \frac{\partial E_p}{\partial w_{ij}}(t) \quad \textbf{(7)}$$

Hence, the partial derivatives of the errors must be accumulated for all training patterns. This indicates that the weights are updated only after the presentation of all of the training patterns [39].

B. Process flow of point count methods

In bridge games, though the basic demonstration includes value of each card as (Ace (A), King (K), Queen (Q), Jack (J), 10, 9, 8, 7, 6, 5, 4, 3, 2) for assignment of cards into selective hands and into public, a uniform linear transformation in the range 0.10 through 0.90 where 0.10 is assigned to the smallest card value 2 with an increment of 0.067 to the next card value i.e., 3 and so on till 0.90 for the highest card value A is assigned as represented in Table 1.

The suit cards such as (♠ (Spades), the highest, ♥ (Hearts), ♦ (Diamonds), ♣(Clubs), the lowest) are assigned a real number using the following mapping: Spades (0.3), Hearts (0.5), Diamonds (0.7) and Clubs (0.9). There are 52 input values and each value represents one card from the deck and the positions of cards in the input layer are fixed. Apart from the usual card values as input to the neurons in the input layer which are multiplied with individual weights of their associates to the hidden neurons and hence from hidden layer to the output layer, the human knowledge is represented by various numerical estimators of hand's strength used by experienced human bridge players in order to declare the best possible contract. The human estimators of hand strength can be divided into two categories such as point count methods and distributional point methods.

Table 1. The range of values assigned to each card of the deck

S. No.	Rank Card	Rank Card Value
1	2	0.10
2	3	0.17
3	4	0.23
4	5	0.30
5	6	0.37
6	7	0.43
7	8	0.50
8	9	0.57
9	10	0.63
10	J	0.70
11	Q	0.77
12	K	0.83
13	A	0.90

The human point count methods are based on calculating the strength of a hand as a sum of single cards strength and the value of each card depends only on card's rank. Though there are many human point count methods such as Work point count [40], Collet point count, Four aces points, Polish points etc., are available, Bamberger point count method and Work point count method are employed here refer Table 2. The other category of human hand's strength estimators contains distributional points, in which the patterns are scored based on its existence in a set of cards assigned to one hand. The most important patterns are suit lengths and existence of groups of honors in one suit.

Table 2. Comparisons of Card values in BPCM and WPCM

S.No	Rank card name	BPCM	WPCM
1	A (Ace)	7	4
2	K (King)	5	3
3	Q (Queen)	3	2
4	J (Jack)	1	1

IV. 52-32-1 OF CCNN ARCHITECTURE

Though several neural network architectures have been used to solve the double dummy bridge problem, in this paper, mainly the CCNN architecture with 52, (13x4) input neurons for solving the DDBP is attempted and the results are discussed. The 52 input card representation deals are implemented in the CCNN architecture as shown in Figure 2.

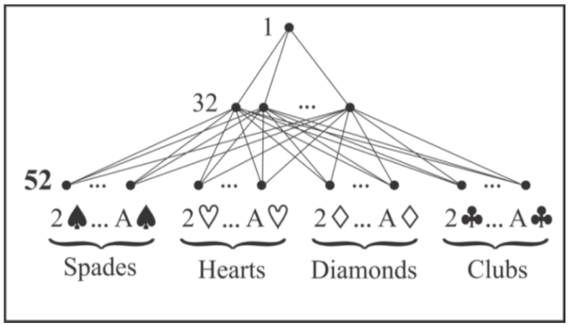

Figure 2 CCNN architecture with 52-32-1 input patterns.

Layers are fully associated, i.e., in the 52-32-1 network 52 input neurons are associated to all 32 hidden neurons and all hidden neurons are associated to a single output neuron. Though the number of hidden neurons to a scrupulous problem is still decided by a rule of thumb, when the number of neurons is minimum, the model may take too much of time to learn or may not be able to learn at all resulting in a underprivileged performance during the training session. On the other hand if the number of neurons in the hidden layer is equivalent to the input neurons, then the aim of the training phase itself may become obsolete and instead of learning during training session, the network might memorize the patterns, which will result, very badly in the testing phase of the network. Thus, it is decided to have half the size of the input neurons as a rule of thumb and in the implementation phase after a trial with 25 neurons, 26 neurons, 32 neurons, it is concluded to stick with 32 neurons since it is slightly more than half the size of the input neurons. For training and learning the data, two activation functions viz., log sigmoid transfer function and hyperbolic tangent sigmoid functions are used. The resilient back propagation algorithm is used for training and testing through MATLAB 2013a.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A total number of five thousand deals from the GIB library[41] for training and two thousand five hundred among the trained data were used for testing on CCNN with fifty two input neurons, thirty two hidden neurons and one output neuron (52-32-1). There are 20 numbers for each deal i.e. 5 trump suits confidential as no-trumps, spades, hearts, diamonds and clubs by 4 sides. The mean squared error during the training phase and testing phase using log sigmoid as the activation function is illustrated in Figure 3, while the mean squared error during the training and testing phases using hyperbolic sigmoid as the activation function is illustrated in Figure 4.

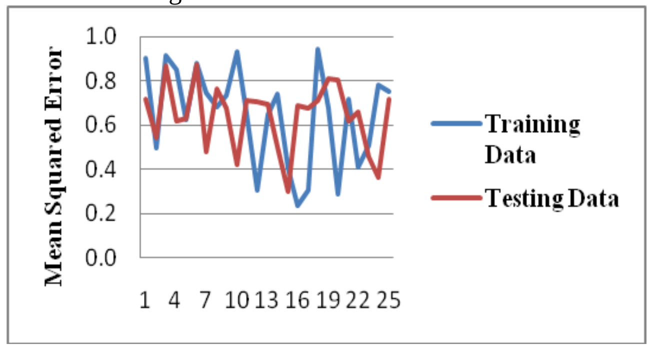

Figure 3 Mean Squared Error (MSE) during training and testing phase of log sigmoid function.

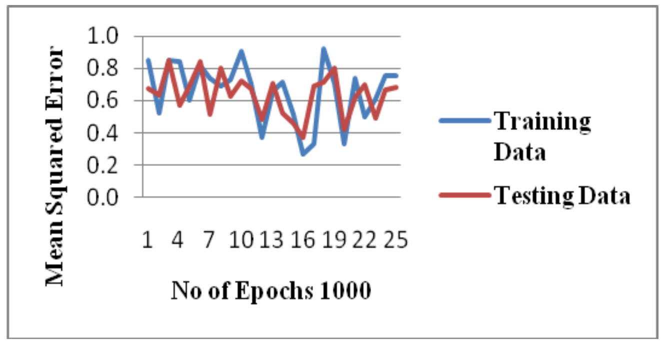

Figure 4 Mean Squared Error (MSE) during training and testing phase of hyperbolic sigmoid function.

In the CCNN model, BPCM and WPCM are used as contract bridge separately with resilient back- propagation algorithm, compared with each other and inferred that the BPCM produced better results when compared to WPCM. The results revealed that, the data tested through CCNN architecture show better performance and the time taken for training and testing are relatively minimum which is converging towards the possible minimum error during the iterations.

The first sample deal, presented in Figure 5, shows the benefit of the (52) CCNN architecture over the networks using the 52 inputs. In this deal the N S pair is able to take 10 tricks when playing spades contract. Though, 52 the normal networks estimated only 7 and 8 tricks, respectively. Both tested networks using the (52) coding were absolutely right, and the enlarging CCNN network size to 52-32-1, allowed to estimate the correct number of tricks. The result of analysis this deal shows that N S pair has together only 15 WPCM. There is a void in clubs on

South and two singletons in hearts and diamonds on North. These short suits extremely strengthen N S and enable them to hold 10 tricks.

A. 4♠ with 15 WPCM

♠ A K Q 9
♡ A 10 5
◇ 6 4 2
♣ A Q 5

West		East
♠ 8 6	N	♠ J 10 7 4
♡ 8 7 6 2	W E	♡ J 9 4
◇ A 10	S	◇ K Q 5 3
♣ K J 10 9 8		♣ 4 2

♠ 5 3 2
♡ K Q 3
◇ J 9 8 7
♣ 7 6 3

Figure 5 The first sample deal (NS pair in Spades contract with West opening lead)

B. Advantage of BPCM

In the second sample deal example, presented in Figure 6 it is quite easy to point out 8 tricks for N S pair (3 in ♠, 3 in ♥ and 2 in ♣), and '8 tricks' was the most frequent answer given by humans. The correct number of tricks is 9. Since the WPCM estimation also suggests 8 tricks (the N S pair plays on 25 points) it is quite interesting that the networks were able to 'find' this 'missing' trick (in ◆).

♠ K J 6 5 4
♡ 5
◇ 5
♣ K 10 9 8 4 2

West		East
♠ 10 9	N	♠ A 8
♡ A 6 4 2	W E	♡ K Q 10 8 3
◇ 10 2	S	◇ K Q 8 6
♣ Q 7 6 5 3		♣ A J

♠ Q 7 3 2
♡ J 9 7
◇ A J 9 7 4 3
♣

Figure 6 The second sample deal (NS pair in ♠ contract with South opening lead)

C.Comparison of BPCM and WPCM

The sample deal representation by adding human estimation didn't improve the best overall result accomplished by pure 52-32-1 in the case of BPCM and only slight improvement was notified in the case of WPCM contracts. This observation suggested that the relevance of additional information related to suit lengths and point distribution in particular hands has been autonomously discovered by the best 52-32-1 CCNN model during the training process. The human players are visibly better at solving the no-trump contracts than the suit ones and the opposite conclusion is also valid in the case of neural networks. The CCNN can be trained to capture the implicit reasoning used for bidding a hand in bridge.

Thus to validate the convergence of the algorithm in the CCNN architecture with 52 (13x4) input neurons for solving the DDBP, the problem is attempted with resilient back-propagation algorithm. The performance during training and testing phases of the sample deal of BPCM and WPCM of the same data from GIB library used in this architecture is illustrated for the purpose of comparison in Figure 7.

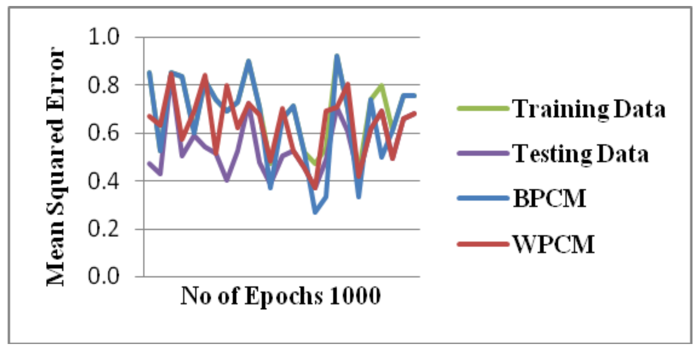

Figure 7 Mean Squared Error during training and testing phase of BPCM and WPCM.

VI. CONCLUSION

In CCNN, for the duration of training process new hidden nodes are added to the network one by one. When the node is being added to the network, the input weights of hidden nodes are frozen, only the output associations are trained frequently. The CCNN model with resilient back-propagation algorithm is compared with back-propagation algorithm. The CCNN model used in contract bridge, the BPCM and WPCM are compared with each other during training process. The numbers of hidden nodes used are 25, 26 and 32. The result reported that, both the methods BPCM and WPCM minimized the Mean Square Error (MSE), reduced the time taken for playing and increase the number of tricks taken in DDBP. The problem of representative a particular acquaintance gained through the learning process is extremely specialized and it is inferred that the estimated method bamberger point count method is a superior information method that provides some new ideas to the bridge players and encouraging for basic and semi professional players as well as humanizing their bridge skills. Furthermore the bamberger point count method can be comprehensive to be taking into deliberation different methods in contract bridge using different architectures and algorithms to solve DDBP more professionally and effectively.

REFERENCES

[1] B.Yegnanarayana, D.Khemani, M.Sarkar, "*Neural networks for contract bridge bidding*", Sadhana, Vol.21, Issue.3, pp.395-413, 1996.

[2] A.Amit, S.Markovitch, "*Learning to bid in bridge. Machine Learning*", Vol.63, Issue.3, pp.287-327, 2006.

[3] J.Mandziuk, K.Mossakowski, "*Looking Inside Neural Networks Trained to Solve Double-Dummy Bridge Problems*", In the 5th Game-on International Conference on Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004),U.K, 2004.

[4] J.Mandziuk, K.Mossakowski, "*Neural networks compete with expert human players in solving the Double Dummy Bridge Problem*", In the Proceedings of the 2009 IEEE symposium on computational intelligence and games (CIG 2009), Italy, pp. 117-124, 2009.

[5] K.Mossakowski, J.Mandziuk, "*Artificial neural networks for solving double dummy bridge problems*", Lecture Notes in Computer Science (LNAI), Springer, Heidelberg, Vol. 3070, pp. 915-921, 2004.

[6] M.Sarkar, B.Yegnanarayana, D.Khemani, "*Application of neural network in contract bridge bidding*", In the Proceedings of the 1995 national conference on neural networks and fuzzy methods, India, pp.144-151,1995.

[7] M.Dharmalingam, R.Amalraj, "*Neural Network Architectures for Solving the Double Dummy Bridge Problem in Contract Bridge*", In the Proceedings of the 2013 PSG-ACM national conference on intelligent computing, India, pp. 31-37, 2013.

[8] M.Dharmalingam, R.Amalraj, "*Artificial Neural Network Architecture for Solving the Double Dummy Bridge Problem in Contract Bridge*", Advanced Research in Computer and Communication Engineering, Vol.2, Issue.12, pp.4683-4691, 2013.

[9] H.Francis, A.Truscott, D.Francis, "*The Official Encyclopedia of Bridge*", 6th ed. Memphis, American Contract Bridge League, 2001.

[10] K.Mossakowski, J.Mandziuk, "*Learning without human expertise: A case study of Double Dummy Bridge Problem*", IEEE Transactions on Neural Networks. Vol.20, Issue.2, pp.278-299, 2009.

[11] B.Seifert, "*Encyclopedia of Bridge*", Polish Scientific Publishers PWN, Warsaw, 1996.

[12] S.Haykin, "*Neural Networks: A Comprehensive Foundation*", Prentice Hall, New York, 1998.

[13] K.Mossakowski, J.Mandziuk, "*Neural networks and the estimation of hand's strength in contract bridge*", Lecture Notes in Computer Science (LNAI), Springer, Heidelberg, Vol.4029, pp. 1189-1198, 2006.

[14] S.N.Sivanandam, S.N.Deepa, "*Principles of Soft Computing*", Wiley, New Delhi, 2007.

[15] B.Yegnanarayana, "*Artificial neural networks*", Prentice Hall, New Delhi, 2010.

[16] J.Mandziuk, "*Computational intelligence in mind games*", In the Studies in computational intelligence, Springer, Heidelberg, 2007.

[17] J.Mandziuk, "*Some thoughts on using computational intelligence methods in classical mind board games*", In the Proceedings of the 2008 International joint conference on neural networks (IJCNN 2008), China, pp. 4001-4007, 2008.

[18] J.Mandziuk, "*Knowledge-free and learning- based methods in intelligent game playing*", Springer, Berlin, Heidelberg, 2010.

[19] M.Dharmalingam, R.Amalraj, "*Back-Propagation Neural Network Architecture for Solving the Double Dummy Bridge Problem in Contract Bridge*", In the Proceedings of the 2014 IEEE international conference on intelligent computing applications (ICICA 2014), India, pp. 454-461,2014.

[20] M.Dharmalingam, R.Amalraj, "*A Solution to the Double Dummy Bridge Problem in Contract Bridge Influenced by Supervised learning module Adapted by Artificial Neural Network*", In the Special Issue on distributed intelligent method and applications, Vol.5, Issue.1, pp.836-843, 2014.

[21] M.Dharmalingam, R.Amalraj, "*Artificial Neural Network Architectures: Work Point Count System Coupled with Back-Propagation and Resilient Back - Propagation Algorithm for Solving Double Dummy Bridge Problem in Contract Bridge*", International Journal of Innovative Science, Engineering and Technology, Vol.2, Issue.8, pp.142-157, 2015.

[22] M.Dharmalingam, R.Amalraj, "*Artificial Neural Network Architectures for solving Double Dummy Bridge Problem in Imperfect Information Game*", International Journal of Computational Intelligence and Informatics, Vol.5, Issue.3, pp.188-197, 2015.

[23] M.Dharmalingam, "*Point Count Systems in Imperfect Information Game*", International Research Journal of Engineering and Technology, Vol.3, Issue.1, pp.362-370, 2016.

[24] M.Dharmalingam, "*Artificial Neural Network Architectures for Solving the Contract Bridge*", International Journal of Advanced Information and Communication Technology, Vol.4, Issue.3, pp.01-09, 2017.

[25] W.H. Root, "*The ABCs of Bridge*", Three Rivers Press, 1998.

[26] M.Dharmalingam, R.Amalraj, "*Supervised Learning in Imperfect Information Game*", Advanced Research in Computer Science, Vol.4, Issue.2, pp.195-200, 2013.

[27] T.Ando, N.Kobayashi, T.Uehara, "*Cooperation and competition of agents in the auction of Computer Bridge*", Electronics and Communications in Japan, Part 3, Vol.86, Issue.12, pp.76-86, 2003.

[28] T.Ando, T.Uehara, "*Reasoning by agents in computer bridge bidding*", In the Computers and Games, Lecture Notes in Computer Science, Springer, Berlin, Vol. 2063, pp. 346-364, 2001.

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

[30] T.Ando, Y. Sekiya, T.Uehara, "*Partnership bidding for Computer Bridge*", Systems and Computers, Japan, Vol.31, Issue.2, pp.72-82, 2000.

[31] I.Frank, D.A.Basin, "*Strategies explained*", In the Proceedings of the 5th Game programming workshop, Japan, pp1-8, 1999.

[32] I.Frank, D.A.Basin, "*A Theoretical and Empirical Investigation of Search in Imperfect Information Game*", Theory of Computer Science, Vol.252, Issue.1, pp.217-256, 2001.

[33] W.Jamroga, "*Modeling artificial intelligence on a case of bridge card play bidding*", In the Proceedings of the 1999 international workshop on intelligent information method, pp.276-277, 1999.

[34] D.N.Levy, "*The million pound bridge program*", In the Heuristic Programming in artificial intelligence - First Computer Olymiad, pp.95-103, 1989.

[35] J.Mandziuk, K.Mossakowski, "*Example-based estimation of hand's strength in the game of bridge with or without using explicit human knowledge*", In the Proceedings of the 2007 IEEE symposium on computational intelligence in data mining (CIDM 2007), Los Alamitos, pp. 413-420, 2007.

[36] S.J.J. Smith, D.S.Nau, T.A.Throop, "*Computer bridge - a big win for AI planning*", Artificial intelligence, Vol.19, Issue.2, pp.93-106, 1998.

[37] S.J.J. Smith, D.S.Nau, T.A.Throop, "*Success in Spades: Using AI Planning Techniques to Win the World Championship of Computer Bridge*", In the Proceedings of the 1998 national conference on artificial intelligence, pp.1079-1086, 1998.

[38] S.E.Fahlman, V.Lebiere, "*The cascade-correlation learning architecture*", In the Advances in neural information processing methods, pp.524-532, 1990.

[39] M.Riedmiller, H.Braun, "*A direct adaptive method for faster back propagation: the RPROP algorithm*", In the Proceedings of the 1993 IEEE international conference on neural networks (ICNN 1993), pp. 586- 591, 1993.

[40] R.Amalraj, M.Dharmalingam, "*A work point count method coupled*

[41] *with back-propagation for solving double dummy bridge problem*", Neurocomputing, Vol.168, pp.160-178, 2015.
[42] M.L.Ginsberg, "*GIB:Imperfect information in a computationally challenging game*", Artificial intelligence research, Vol.14, pp.303-358, 2001.

Author Profile

Dr. Dharmalingam Muthusamy is currently working as an Assistant Professor in the Department of Computer Science in Bharathiar University Arts and Science College at Modakkurichi in Erode, Tamil Nadu, India. He has published more than 20 research papers in reputed international and national journals. His main research work focuses on Neural Networks, Data Mining and Computational Intelligence based games. He has 15 years of teaching experience and 8 years of research experience.