Cases and Clusters in Reuse Policies for Decision-Making in Card Games

Gustavo B. Paulus
Graduate Program in Computer Science
Federal University of Santa Maria
Santa Maria – RS, Brazil
gustavobpaulus@gmail.com

Joaquim V. Carvalho Assuncao
Department of Applied Computing
Federal University of Santa Maria
Santa Maria – RS, Brazil
joaquim@inf.ufsm.br

Luis A. L. Silva
Graduate Program in Computer Science
Federal University of Santa Maria
Santa Maria – RS, Brazil
luisalvaro@inf.ufsm.br

***Abstract*— This work investigates the combination of cases and clusters in the reuse of game actions (e.g., cards played, bets made) recorded in the cases retrieved for a given query in Case-based Reasoning (CBR) card-playing agents. With the support of the K-MEANS clustering algorithm, clustering results detailing problem states/situations and game outcomes relationships recorded in cases from the case base guide the execution of augmented reuse policies. These policies consider the game actions recorded in the retrieved cases in the selection of the clusters to be used. Then, the cases that belong to the selected clusters are used in the determination of which game action is reused as a solution to the current game problem situation. With this two-step reuse process, the proposed policies rely on the majority with clusters, the probability with clusters, the number of points won with clusters and the chance of victory with clusters. To evaluate these proposals, card-playing agents implemented with different reuse policies competed against each other in duplicated game matches where all of them played using the same set of cards.**

Index Terms – Reuse policies. Case-based reasoning. Clustering. Card games. Truco game.

I. Introduction[1]

Card games have stochastic characteristics that pose a challenge for Artificial Intelligence (AI). As stated in [1], the random distribution of playing cards and the players' partial vision of the cards of the opponents prevent one to formulate data-driven models of decision-making, making it difficult for the AI to make the best decisions in a given game scenario. In this context, Case-based Reasoning (CBR) [2] relies on a lazy learning approach to support decision-making permitting to better clarify how past experiences of human game playing can be utilized in the resolution of new game problem situations.

In CBR, the techniques used to choose a solution to a current game problem are often captured as part of *reuse policies*. In [3], for instance, the majority rule, the probabilistic choice, and the best outcome reuse policies are evaluated in the playing of Texas Hold'em Poker. Despite the very competitive card-playing agents presented there, there still room for further research since these CBR policies tend to rely on local decisions in the game, without taking into consideration a broader view of the overall game problem situations along with their executed *game actions* (e.g., the cards played, the bets made, etc). As a contribution to the improvement of CBR-based card-playing agents, a better examination of the humans' playing patterns used in different game contexts as recorded in cases from game playing case bases can be investigated through the exploration of clustering algorithms [4, 5].

This work shows that card-playing problem states and their outcomes can support the proposition of augmented reuse policies, which are here described as a) the majority with clusters (MJC): the choice of clusters and cases is supported by a voting technique that is guided by the different game actions recorded in the retrieved cases, b) the probability with clusters (PC): the choice of clusters and cases rely on probability estimates computed from the game actions recorded in the retrieved cases, c) the number of points won with clusters (NPC): the computed number of points won for each one of the game actions recorded in the retrieved cases supports the choice of clusters and cases and d) the chance of victory with clusters (CVC): the computed chances of victory for each one of the different game actions recorded in the retrieved cases guide the choice of clusters and cases. This paper detail how such clusters and cases can be integrated in order to support the construction of improved reuse techniques for CBR, which are implemented by card-playing agents to support the making of bets and the playing of cards in the game of Truco [6].

II. Background to this Work

The usual cycle of CBR [2] consists of retrieval, reuse, revise, and retain steps. Once a list containing the most similar cases is retrieved from the case base, the reuse step provides a solution to the current problem situation, since solutions recorded in the past cases can be reused according to different reuse policies. As investigated in this work, different methods of reusing the retrieved case solutions are explored in the selection of the solution to be applied to the new problem.

In computer game applications, clustering algorithms are most relevant in the identification of playing patterns over logs of game competitions [4]. By enabling pattern recognition, clustering reduces the complexity of the data and provides

[1] This work was partly supported by the Federal Institute of Education, Science and Technology-IFRS and CAPES/Brazil.

conclusions about the types of behaviors present in games (e.g. [7]). As cited in [4], an important contribution provided by these clustering algorithms is the increase in the possibilities of collecting contextual information about the behavior of the players. In [8], for instance, clustering algorithms are used to aid in the process of identifying sequences of plays that provide better results for given game situations. Once such knowledge about game playing patterns is made explicit, what emerges from such kinds of clustering investigations is that these playing patterns should not be underused (or even lost) in the development of better game-playing agents.

There is a myriad of works that attempted to create CBR-based agents capable of making good decisions in adversarial games. In particular to this work, [3] compares the performances of three different reuse policies applied to Poker: a) reuse of the Poker playing action used in the most similar retrieved case, b) reuse of the Poker playing action used in the majority of the retrieved cases and c) probabilistic reuse of Poker playing actions used in the retrieved cases, where the probability of each existing action in the retrieved cases is computed and used in the probabilistic choice of the game action to be reused in the solution of the current game problem. Among the forms of reuse presented in [3], the majority-based reuse policy provided the best performance. In [9], it is used a decision model that explores the information of the number of wins, profits, and losses that each decision category in the game provided; however, it does not contain a definition of how to use the consequences from different decisions made in the game. In effect, we investigate this approach in this paper.

III. The Truco Card Game

The cases and cluster techniques proposed here were implemented in a CBR agent that can play the game of Truco [6] - a very popular card-playing game in the southern regions of South America. Similar to Poker, Truco involves cognitive challenges due to uncertainty. Unlike Poker, however, Truco has several decision-making steps due to its multiple game interactions per hand. Truco players have a partial view of the states of the game, where two or more, adversarial players interact during the rounds or *hands of the game*. To approach the basics of a two-player game, we briefly describe the Truco game. For a more in-depth introduction to Truco, see [6].

Truco is divided into multiple hands, where the game ends when one player reaches 24 points. In each Truco hand, players receive three cards in which they dispute a certain number of points. This dispute can have at most three *turns of card-playing*. It also means that players take turns when playing a hand, where the winner of a turn starts playing the next turn. To win the hand, the player has to win at least two of the three turns of the game hand. The main actions in this game can be divided into "playing cards" and "making bets". Given a hand, players aim to drop a card in the table which is higher than the card played by the opponent. Bets are made to increase the number of points which are disputed in the game hand. Although bets are usually advanced when the players have a strong set of cards, they are also advanced to force the opponent to fold in the case of a bluff. There are two *betting stages* in a Truco hand: the ENVIDO and the TRUCO (which gives the name of the game). While the TRUCO stage of betting is based on the relative strength of the cards in the TRUCO game, the ENVIDO is based on numerical scores calculated from equal card suits in the player hand. For each of these bets, players can call, fold and raise the proposed bets. In the TRUCO betting mode, bets must follow a flow of interaction between the opponents since it is not possible to start with the maximum bet. In the event of a bet being denied, in any of the betting stages, the winner receives the points from the last bet called, or one point, in the case of the first bet is denied. In the TRUCO betting mode, when a player wins, the competition in the current hand ends. The ENVIDO betting mode can be made before the TRUCO betting mode and it does not end the hand competition. So, it is possible to define a set of common terms to be used in Truco bets: Bet: an initial bet, Call: accept the last bet, Raise: increase the bet and Fold: deny the bet and give the opponent the points involved in the last bet. Similar to Poker, the two distinct betting stages of the Truco game end up revealing different information about the strength of cards that the players might have. After each card is played, the players collect more information from that hand. The first player to play is called "hand player", he/she sees only his/her own cards. The second player is called "foot player", this player can see the first card of the opponent unless the opponent makes a bet before play any card. In summary, Truco cases recorded in case bases reflect the actual behavior of human players, with their strengths and weaknesses.

IV. CBR and Clustering Approach to the Truco Game

A web-based system to play Truco was used in the collection of game hands played by different human players. In the end, a case base containing 2044 game hands was constructed. The cases in this case base were represented as pairs of attributes and values as described in Table I. With this case base, the recorded cases were used as input on the execution of clustering analysis tasks [5]. In general, the overall goal was to find clusters that could reflect the different playing contexts and their outcomes which are the result of the two-player dispute scenarios recorded in the Truco game. As presented in other works [4, 7, 8], that clustering enables the identification of patterns/tactics of game-playing to be used in the improvement of our CBR implementations.

As described in [4], the K-MEANS algorithm is usually used in the analysis of game logs. This algorithm requires the number of groups as a parameter. When it is not known how many groups exist in a given dataset, the "Elbow" technique (e.g. [10]) allows a few divisions to be found according to the natural characteristics of the data. In this work, this algorithm was explored in the clustering of the Truco playing cases recorded in the case base, where the number of clusters was based on the results of the Elbow technique (considering the within-cluster sum of errors). In the end, the information available for each one of different game scenarios was used as input in the various executions of this clustering algorithm. For instance, the game actions executed while playing the turns of a Truco hand were used as input (e.g., cards played in each turn of a Truco hand) in the clustering of cases. As a result, the formed clusters revealed alternative patterns of game playing which were represented accordingly. Table II illustrates the kind of clusters obtained for the first card-playing action in the Truco game. With such clusters available, the group that each case belonged to, for each of the different game actions, was added as a case attribute in the structure of the cases of the case base.

TABLE I. CASE REPRESENTATION FOR THE TRUCO GAME

Case attributes	Explanation
Received cards	The players' received cards are represented in the case structure as numerical attributes. Cards are encoded in a non-linear numerical scale varying from 1 to 52. The values in this scale capture the relative importance of the cards used in the Truco game [6].
Player that started playing in each turn of a Truco hand	Either Player 1 or Player 2 is the "hand player"
Cards played in each turn of a Truco hand	The three received cards in a Truco hand are encoded as high, medium and low cards. Using this scale, this attribute captures which card was played in each turn of a Truco hand.
Player that won each Truco hand	Either Player 1 or Player 2
Bets made by the players	It represents which player made each one of the bets made in a Truco hand
Number of Truco points won / lost as a result of each action made in a Truco hand	Number of points resulting from each game action played in a Truco hand
Cluster labels according to each game action executed	Clusters that the case is part of according to game actions recorded in the case structure

TABLE II. CLUSTERING RESULTS FOR PLAYING THE FIRST IN THE TRUCO GAME IN THE FOOT GAME SCENARIO

Cluster	Cases	Cluster characteristics
Game tactics used when all three received cards have a low value in the game	2.55%	The card with the highest value is played by the Player 1. Then, Player 2 plays the lowest card
Game tactics used when the received cards have a balanced value in the game	38.00%	The card with the lowest value is played by Player 1. Player 2 plays the lowest card that beats the card played by the opponent
Game tactics relating to bluff	10.38%	When all three received cards have a low value, the hand competition is ended even before the first card is played in most of the situations. When Player 1 plays the first card without making any bet, Player 2 makes a bet. Player 1 doesn't call and the hand is ended. In some situations, Player 1 raises the bet which forces Player 2 to not call
Game tactics used when only one of the three received cards has a high value in the game	49.07%	Player 1 plays either a medium or low value card. Player 2 plays the card that has the lowest value

In general, each one of the three different turns of a game hand has distinct pieces of information that are visible for Truco players. In this case, only such visible game information – which captures the state of the game for a given agent - is used in the formation of the various CBR queries executed by our card-playing agents during the dispute of a Truco game hand. Similarly, these different game contexts were also explored in the formation of clusters in this work.

V. THE AUGMENTED REUSE POLICIES FOR DECISION-MAKING IN CARD GAMES

To support the implementation of CBR-based card-playing agents, alternative reuse policies for decision-making in card games were approached. In a two-step approach for the decision-making, the implemented policies first analyze the clusters that the retrieved cases for a given query are inserted into (see Listing 1, the *chooseCluster()* function). There is a cluster attribute represented in the case structure which indicates what cluster (i.e., the cluster label) the retrieved cases belong to. Once the clusters present in the list of retrieved cases are identified, one of them is selected according to certain criteria, which are (as tested in this work) the majority voting, the selection based on probability estimates and the new criteria based on the analysis of the game outcomes; e.g., to obtain the highest number of points in the game. Then, a similar selection is applied again over the game actions recorded in the retrieved cases that belong to the selected cluster (see Listing 1, the *chooseGameAction()* function). Retrieved cases belonging to other clusters that are not selected in the first step are no longer considered in the reuse of solutions applied to the resolution of the current game problem.

Algorithm 1 : The proposed two-step reuse policy

data: Given: *CB*, case base; *C*: case clusters; query, query considering the current game problem situation; *st*, similarity threshold; aReusePolicy, reuse policy ∈{MJC, PC, NPC, CVC}
result: *aGameAction*, game action selected.
  begin
    $R = \{case_1, case_2, \ldots, case_N \in CB \mid (sim(query, case_n) > st)\}$
    selectedCluster = chooseCluster(*R*, *C*, aReusePolicy)
    $R' = \{case_n \in R \mid case_n.clusterLabel == \text{selectedCluster}\}$
    aGameAction = chooseGameAction(*R'*, aReusePolicy)
    return aGameAction
  end

Listing 1: Pseudocode used in the computation of the two-step approach where cases and clusters are considered in the computation of augmented reuse policies for decision-making in card games.

In the proposed two-step reuse approach, the use of the majority voting technique enables the computation of the majority with clusters (MJC) reuse policy. In the first step, this policy identifies which kind of game action was the most frequently used in the retrieved cases. With it, a cluster from the set of clusters capturing that particular state of the game (e.g., clusters for disputes happening in the first turn of the game hand, or for the second turn, and so on) is selected. Then, a filter is applied to the retrieved cases, where only the cases that belong to the most voted cluster are maintained in the retrieval results. Finally, the same voting technique applied to the selection of a cluster is computed again over the game actions used in the past as recorded in the retrieved cases belonging to the selected cluster. In the end, the result of this second step of voting indicates which game action (e.g., which card should be played, how to make bets or provide answers for bets) is reused.

In [3, 11], a probabilistic reuse technique is presented, named as probabilistic solution (PS). This technique computes the probabilities for each one of the existing game actions recorded in the retrieved cases. In practice, probabilities regarding each game action in each turn of a game hand can be computed. In doing so, probability estimates are derived from the number of times that each game action appeared in such retrieved cases. With probability estimates available, a lottery function is evaluated, resulting in the choice of a particular game action to be reused. In this context, the probability with clusters (PC) reuse policy involves the application of the probabilistic reuse policy in two-steps, both to the choice of which cluster to select and to the identification of which game action to reuse. In effect, the number of retrieved cases belonging to each cluster is used in the computation of probability estimates, which are later used in the lottery computation leading to the selection of a cluster to be used. Similar to other augmented reuse policies detailed here, only the retrieved cases from the selected cluster are used in the second step of probabilistic evaluation of which the game action is reused.

To allow the CBR-based card-playing agent to choose a game action to be used, the consequences in the game resulting from the use of each game action recorded in the case structure can be explored [9]. To compute the choice of game actions with the highest chance of victory (i.e., victory as an outcome in the game hand), for instance, it is explored a case attribute that represents the outcome of the played game actions, whether this outcome is either a victory or a defeat in that hand. Then, the number of victories and defeats for each different game action recorded in the retrieved cases is computed, and divided (i.e., normalized) by the number of retrieved cases in which each of those game actions was played. In the end, the game action with the highest chance of victory as recorded in the retrieved cases is returned (reused) as a solution to the current problem. From this "chance of victory" (CV) reuse policy, we propose the "chance of victory with cluster (CVC) reuse policy. When the CVC policy is computed according to the two-step reuse approach, the first step is directed to the choice of which cluster contains the retrieved cases with the highest chance of victory. With the selection of the cluster and the retrieved cases belonging to it, the CV reuse policy is applied in the choice of which game action to reuse.

To allow the agent to choose a game action solution for game situations involving bets, the cases record in their structure the number of points that were either won or lost as a result of players' actions of betting. To compute the choice of game actions with the highest number of points won, we considered the different kinds of betting actions identified in the retrieved cases. For each betting action, the sums of the points won and lost are recorded. Then, these resulting sums are divided by the number of retrieved cases in which each of such betting actions was played. In the end, the betting action with the highest rate of points won is returned (reused) as a solution to the current problem. Alternatively, the betting action with the lowest rate of points lost is returned (reused) in case it is not possible to win points in that hand as indicated by the retrieved cases. In this work, this policy is named "the number of points won for each game solution" (NPS). When this NPS reuse policy is applied in two steps, the same selection criteria is applied twice, one for the selection of the cluster containing the retrieved cases with the highest number of points won, and another for the selection of which game action to reuse as far as the filtered set of retrieved cases is concerned. This amounts to explore the augmented AS reused policy named "the number of points won for each game action with clusters" (NPC).

VI. Experiments and Results

The experimental approach used in this work is based on what the ACPC (Annual Computer Poker Competition) [1] proposes: the analysis of the results from the dispute of duplicated game matches, where all the players compete in these matches using the same set of cards. In the tests, 50 duplicated Truco matches were played by agents to assess the effectiveness of the proposed reuse policies for decision-making. There, 6 different reuse policies were used by CBR-based Truco player agents when playing one against each other. In total, 500 matches were played by each agent in this simulated competition, where the reuse policies tested were: (i) the majority with clusters (MJC) – it is based on the voting of the majority in which clustering results are considered in the two steps of decision-making; (ii) the probability with clusters (PC) – it is based on the probability of the game actions in which clustering results are considered in the two steps of decision-making; and (iii) the chance of victory and the number of points won with clusters (CNPC) – considering clustering results in the two steps of decision-making, the reuse policy is based on the number of points won to find solutions for game actions that involve bets (NPC). In it, this reuse policy is also based on the chance of victory to find solutions for game actions that involve playing cards (CVC).

To contrast the techniques proposed in this work with others [3, 11], three other techniques for reusing solutions and supporting the decisions of player agents were implemented and adjusted to the Truco game. In them, clustering results were not explored in the decision-making of what game actions to reuse. These standard reuse policies are: (a) the majority of solutions (MJS) – it is based on the voting of the majority; (b) the probability of solutions (PS) – it is based on probability of the different game actions and (c) the chance of victory and the number of points won (CNPS) - to game actions that involve bets, the reuse policy is based on the number of points won. Alternatively, to game actions that involve playing cards, the reuse policy is based on the chance of victory.

In the proposed tests, the final competition results of the 1500 Truco matches are summarized in Fig. 1, showing the number of matches that each Truco agent won in the competition. There, the chance of victory and the number of points won reuse policy with clusters (CNPC) had the highest number of victories. In fact, this policy permitted to select the clusters of game behaviors that resulted in the highest benefits in the game, enabling to focus on the reuse of game actions primarily recorded in these groups. Conversely, the chance of victory and the number of points won policy without the use of clusters was not so promising in the competition. Without using the clustering results, the list of retrieved cases used in the reuse of game actions contained a quite varied list of game situations (retrieved cases capturing different game contexts). As far as the chance of victory and the number of points won were concerned, reusing solutions from this larger set of distinct kinds of retrieved cases was not so effective in comparison to the reuse of solutions from a smaller set of retrieved cases representing more homogeneous game situations.

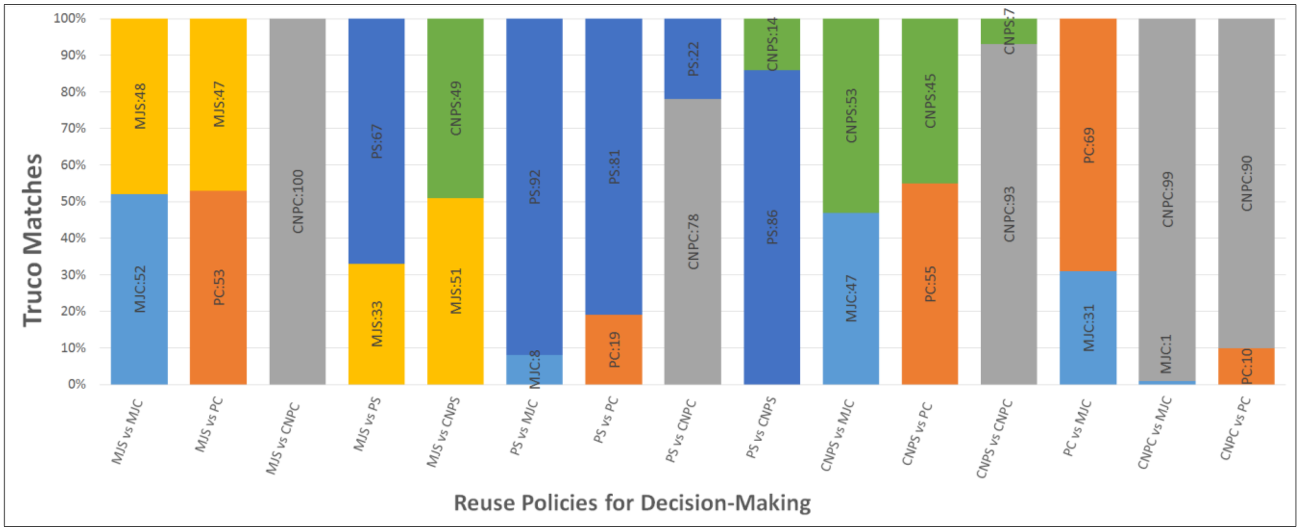

Fig. 1 . The competition results of the duplicated matches played between the different Truco player agents implemented.

The Truco player agents relying on the probability-based reuse policy without clusters (PS) showed an unpredictable behavior in practice, and it resulted in a better competition score than the similar policy with the use of clusters. Among other reasons, there were too many problem situations in which the probabilistic-based lottery developed on the choice of clusters could not select the cluster that was associated with the highest probability according to the retrieved cases. In the proposed two-step approach, this second step of lottery sorting introduced an additional degree of randomness in the reuse of game actions, resulting in the reuse of actions that could not be so adequate to the solution of the current game problems. The majority with clusters (MJC) reuse policy had a small lead on the competition score when it was compared with the same technique without the use of clusters (MJS). However, the Truco player agents based on the MJC reuse policy were not so aggressive. We also noticed that certain opponent agents of the MJS-based agents explored such lack-of-aggressiveness. That is because the case base contains cases from “conservative” human players mainly. This is a disadvantage to a cluster-based technique since it tends to reduce the heterogeneity of playing patterns used as input for the computation of the solution reuse. Without using clusters, there was a higher number of game actions considered in the voting of solutions to be reused (i.e. the clusters contained a more homogeneous set of game scenarios and game outcomes), and these different options of playing resulted in more unpredictable player behaviors.

VII. Concluding Remarks

The stochastic and partial view nature of card games, in this work illustrated and experimented in the Truco card game, presents a challenging scenario for the exploration of cases and clusters in the proposition of augmented reuse policies to be used by CBR-based card-playing agents. This paper describes how to explore clustering algorithms in the organization of the different states of a game problem and the game outcomes emerging from the cards played and bets proposed in these contexts. The improved reuse policies proposed, based on the game patterns and captured with the exploration of clustering algorithms, allowed the CBR card-playing agents to make better decisions as shown in the developed experiments. As observed, these policies are most useful to card-playing games when the success from game actions can be measured according to the number of points won and the number of victories.

References

[1] J. Rubin and I. Watson, "Case-based strategies in computer poker," *AI Communications,* vol. 25, pp. 19-48, 2012.

[2] R. Lopez De Mantaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. T. Cox, and K. Forbus, "Retrieval, reuse, revision and retention in case-based reasoning," *The Knowledge Engineering Review,* vol. 20, pp. 215-240, 2005.

[3] J. Rubin and I. Watson, "Similarity-Based Retrieval and Solution Re-use Policies in the Game of Texas Hold’em," presented at the Int. Conf. on Case-Based Reasoning (ICCBR 2010), Alessandria, Italy, 2010.

[4] C. Bauckhage, A. Drachen, and R. Sifa, "Clustering Game Behavior Data," *IEEE Trans. on Comp. Intelligence and AI in Games,* vol. 7, pp. 266-278, 2015.

[5] A. K. Jain, M. M. N., and F. P. J., "Data Clustering: A Review," *ACM Computing Surveys,* vol. 31, pp. 264-323, 1999.

[6] L. L. Winne, *Truco*. Ciudad Autónoma de Buenos Aires Ediciones Godot, 2017.

[7] R. Sifa, A. Drachen, C. Bauckhage, C. Thurau, and A. Canossa, "Behavior evolution in Tomb Raider Underworld," presented at the Conf. on Computational Inteligence in Games (CIG 2013), Niagara Falls, ON, Canada 2013.

[8] D. Wehr and J. Denzinger, "Mining game logs to create a playbook for unit AIs," presented at the Conf. on Computational Intelligence and Games (CIG 2015), Tainan, Taiwan 2015.

[9] A. Sandven and B. Tessem, "A case-based learner for poker," presented at the The Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006), Helsinki, Finland, 2006.

[10] P. Bholowalia and A. Kumar, "EBK-Means : A Clustering Technique based on Elbow Method and K-Means in WSN," *Int. Journal of Computer Applications,* vol. 105, pp. 17-24, 2014.

[11] J. Rubin and I. Watson, "Investigating the Effectiveness of Applying Case-Based Reasoning to the Game of Texas Hold'em," presented at the Florida Artificial Intelligence Research Society Conference (FLAIRS 2007), Key West, Florida, USA, 2007.