Exploration and Analysis of Ayo Game Tournament using Case-based Probability Distance Clustering and Awale Shareware

[1]*Ibidapo O. Akinyemi, [2]Adesola O. Anidu, and [2]Adefemi T. Adeniran
[1]Department of Computer Science & Mathematics,
Mountain Top University, Ibafo, Ogun State
[2]Department of Mathematical Sciences
Augustine University, Ilara-Epe, Lagos State
Email: [1]ioakinyemi@mtu.edu.ng, [2]adesola.anidu@augustineuniversity.edu.ng,
[3]adefemi.adeniran@augustineuniversity.edu.ng

*Corresponding Author

ABSTRACT

Game playing has become one of the most interesting areas of Artificial Intelligence (AI) application in problem solving. It is an event that enables people to compete against each other, and more often than not brings about entertainment. Ayo is a combinatorial game that requires rigorous calculations and strategies, and has captured the attention of many AI researchers, mathematicians and computer scientists. A number of AI techniques have been used in time past to solve Ayo game unfortunately, the exact solution has not been found as result of the irregular pattern of game space as the play progresses. This work proposes a Case-Based Probability Distance Clustering technique to evolve an Ayo game player and the results obtained from the technique was tested (as a comparison) against Awale shareware Program for which appealing results were obtained. The Case-Based Probability Distance Clustering technique was implemented using C++ Builder 6.

Keywords: *Ayo game, Probability distance clustering, CDG, Case-based reasoning.*

African Journal of Computing & ICT Reference Format:

Ibidapo O. Akinyemi, Adesola O. Anidu and Adefemi T. Adeniran (2019), Exploration and Analysis of Ayo Game Tournament using Case-Based Probability Distance Clustering and Awale Shareware,
Afr. J. Comp. & ICT, Vol.12, No. 1, pp. 28 - 39.

I. INTRODUCTION

Ayo game is the most popular board game among the Yoruba people who occupy roughly the south-western states of Nigeria and parts of Republic of Benin [1, 2]. Ayo game is one of the oldest game of strategy known. It is a game that requires calculation and strategy, with the aim of capturing seeds while keeping to the rules, agreed by the players. Due to the number of strategies and amount of calculation involved, Ayo game has captured the attention of many Artificial Intelligence (AI) researchers and computer scientists [3, 4].

Ayo is a game played by two players, which is traditionally played on a plank of wood (board) with twelve pits that are arranged into two rows (north and south) of six pits each and the board is put in between the players with 48 seeds, as shown in Figure 1 [5]. The objective of the game is to capture as many seeds as possible. Each seed is worth one victory point, and when one of the players gets at least 25 seeds he wins. To start, four seeds are placed in each of the twelve pits on the board. The player on the north side must then choose one of the pits on his side of the board. The player scoops up all four seeds and sows them in a counter-clockwise fashion around the board. This is done by placing one seed in the pit to the right of the original pit, then one seed in the pit to the right of that one, and so on until there are no more seeds left in the player's hand. When a player gets to the last pit on his side, he simply places a seed in his opponent's pit and keeps going counter-clockwise. If a player has enough seeds to go all the way around the board (generally known as *kroo* move), he skips the pit he started with.

Seeds are captured when the last seed sown is placed in an opponent's pit with one or two seeds. In addition, if the previous pit has two or three seeds in it after sowing, those seeds are collected as well. And so on, up to five pits' worth, therefore it is possible to totally wipe out an opponent's side of seeds this way. During the game, players' alternate turn; a player never takes two turns in a row.

Previous research into Ayo game (a variant of Mancala) has looked at exploring good heuristics, good strategies and in some work, solving the game. However, it is still unknown as to how applicable certain heuristics are across related variants. Certain heuristics are not applicable due to rule changes; other heuristics may be weakened or strengthened across variants due to rule changes. Our aim in this paper is to adopt a Case-Based Probability Distance Clustering method to evolve Ayo game player thereby compare its performance with a well-known Computer Ayo game player - Awale shareware.

For the purpose of clarity and readability, the rest of this paper is succinctly organized thus. Section 2, discusses the related works on Ayo Game. In Section 3, the concept of Completely Determined Game (CDG) is discussed. We describe the proposed Case-Based Probability Distance clustering technique in section 4. In Section 5, we present the experimentation result and present a concluding remark in Section 6.

II. RELATED WORKS

The strongest Ayo program today is believed to be Awale, developed by Didier and Olivier Guillion of Myriad software [6], but the technique upon which this shareware was built has not been widely made public. Endgame databases [7] have been offered for evolving Ayo/Awari player. Lithindion [8] is an artificial Awari player that uses a combination of alpha-beta search algorithm and endgame database. Marvin [9] is an Awari player that uses a hybrid method (called drop-out expansion) of depth-first and breadth-first search. Softwari [10] is an Awari program that constructs a large endgame database. These methods have one commonality, they all focus on searching and database utilization and gave little attention to evaluation function.

Generally, endgame databases can only be computed when few pieces remain on board and they can require solution lengths that defy the capabilities of minimax based searches for optimal play.

Retrograde Analysis (RA) [6, 7] has been offered for finding optimal play for all possible board positions. Retrograde is applicable to search spaces, which can be completely enumerated within the memory of a computer. The technique was used in Awari [7] and recently used to solve Awari by searching the entire state-space on a parallel computer with 144 processors [8]. The exact solution of Awari has not been obtained, albeit by use of more intensive computational resources to be drawn, assuming both players play optimally. CPU time of 51 hours on a distributed supercomputer with 72 1GHz PIII nodes, each with 1 Gb RAM was used. The researchers reported that the state space contains 889,063,398,406 positions and was searched using RA [11, 12]. They admitted that no single computer has enough processing power and memory to search the entire state-space, but even on a modern, parallel computer the problem was extremely challenging [11]. Both endgame databases and retrograde approaches can be very expensive to implement and since Awari positions occur in several billions, such methods cannot be easily implemented on a small memory device like wireless handset.

In contrast, a GA-based technique was proposed in [13] to mine endgame databases for relevant features that are useful in the construction of a static evaluation function (simply called evaluator). A hybrid method of co-evolution and minimax was investigated for evolving Awari player [14]. They proposed an evaluator that was based on a linear combination of six Ayo features with associated weights representing the current game position. The weights were evolved using evolution strategy and the output of the evaluator was used in a minimax search. The method reported in [15] extended the six Ayo features presented in [14] to twelve for performance improvement and the associated weights were evolved using genetic algorithm. While a search tree of depth seven was used in [14], only depths three and five were considered in the Ayo program implementation discussed in [15]. Their result showed a considerable improvement over that presented in [14].

III. COMPLETELY DETERMINED PATTERN OF AYO GAME

CDG is a game configuration in which there exists a series of move and capture strategies for a player X on the action of player Y such that the move process continues for the player X until only one seed remains on the board and the seed belongs to player Y. As the concept of the CDG is defined, its usefulness for ending a game should be apparent, and can be configured (or arranged) as the number of seeds on the board reduces to twenty-one.

In *Ayo* game, a CDG play exists if the configuration of seeds on the board satisfies the two conditions given below;

i. all the holes on the side of player Y (the first player) are empty except the first hole that contains only one seed
ii. As player Y moves, player X takes a move that results in a capture of two seeds.

This strategy of move and capture continues until player X has captured all the seeds except one seed that is left for player Y to ensure that the golden rule is obeyed. Figure 1 below depicts the initial configuration of player X and Y for the CDG.

The CDG configuration can be reduced iteratively into another such that its continuity is not hampered in any way until the last possible CDG possible is obtained. This process is referred to as CDG-vector reducibility [16]. Maxima and minima CDG-vectors enable us to define a transformation function between vectors. The reducibility has been efficiently computed by implementing the maximal CDG-vectors as contained in the algorithm in Figure 2.1.

IV. THE CASE-BASED PROBABILITY DISTANCE CLUSTERING (CBPd-Clustering)

Case-based reasoning (CBR), according to Aamodt & Plaza [17], is an intelligent reasoning method that guides action based on past experience. It has widely been used to solve a number of problems that span different areas of human endeavours, such as product design [18], pattern classification [19, 20], regression prediction [21, 22], and intelligent control [23], with a remarkable application success. The goal of CBR application research is to improve the learning performance [24, 25] and the accuracy of the problem solving [26]. When used for solving game problems particularly in suggesting move strategies, there is need to adapt a classification/clustering of the strategies into good or bad, hence probability distance clustering is eminent. Probabilistic distance clustering (PD-

clustering) is an iterative, distribution free, probabilistic clustering method. PD-clustering assigns units to a cluster according to their probability of membership, under the constraint that the product of the probability and the distance of each point to any cluster center is a constant. PD-clustering is a flexible method that can be used with non-spherical clusters, outliers, or noisy data.

The fundamental principle underlying the proposed CBPd-Clustering is to avoid looking into a dataset of Ayo game strategies for suggesting a move. This can be accomplished by training a machine learning system to predict the outcome of such a move. In this way, Ayo game player is evolved that fits into the main memory of a computer. The proposed case-based PD-clustering technique implemented three principal components, build Game Tree, compute Game value and predict Strategy [27]. The build Game Tree is the component that gives the structure for which the game tree is built and computes Game value using Equation (1) to compute the game value. The predict Strategy is the component that uses a refinement procedure [28] to deal with equipotent strategies that can result from computing game value. A refinement is a mapping that accepts a set of strategies, which are to be evaluated and the mapping returns a strategy with the best advantage [28]. We considered Aggregate Mahalanobis distance function as the refinement procedure in this work on the account that the general problem of data clustering is to partition a dataset into *m-clusters* of similar data points. The majority of the existing clustering techniques can be classified into distance based clustering such as k-means [29] and probabilistic based clustering such as expectation maximization [30]. The third category of clustering technique is the pd-clustering, which relates probability and distance using a simple inverse principle.

The pd-clustering principle is illustrated as follows.

Let $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^N$ (1)

be a given vector of data points and suppose a dataset D consists of N data points $\{x_1, x_2, ..., x_N\}$, for each $x \in D$, cluster centroid C_k and constant K, the probability $p_i(x)$ that x belongs to D is given as:

$$\frac{P_k(x)d_k(x)}{q_k} = k \quad (2)$$

Equation (2) has been shown to be the solution of the extrema problem [31] stated in the Appendix as Equation (3), where d_1x and d_2x are distances of the data point x to the clusters of sizes q1 and q2 and p_1x and p_2x are the cluster probabilities. In order to solve Equation (3), the Lagrangian of the problem is defined as Equation (4) in the Appendix.

The distance metric d(x, y) measures the closeness of the vectors x and y and is usually given as Equation (5).

The proposed CBPd-Clustering was designed such that *Ayo* strategies were classified into two clusters ς_1 and ς_2 of good and bad strategies respectively. A good strategy will always lead a player towards winning whilst a bad strategy leads the player towards losing the game. In order to provide an understanding of good and bad *Ayo* strategies, consider the following CDG strategy {7, 5, 3, 1, 2, 2, 0, 1, 0, 0, 0, 0}, which is a fictitious *Ayo* strategy [34, 35]. The current player, Max will always make a move to capture (*n-1*) seeds such that the opponent, Min player has no choice than to keep making on available move. The parameter n is the total number of seeds on board. In order to prevent the Min player from ever capturing seeds as the game progresses, the Max player will try to make a good move each time. According to the CDG reducibility strategy for example, a good strategy is in {7, 5, 3, 1, 2, 0, 1, 0, 0, 0, 0, 0}, is such that best move is suggested to be that you play (or move) form house 6 and every alternative strategy is regarded as a bad strategy, thereby disallow the reducibility of CDG strategy.

Given an *Ayo* strategy *x*, the refinement procedure calculates Mahalanobis distances d(x, ς_1) and d(x, ς_2) between *x* and the clusters of good and bad *Ayo* strategies. The clusters are represented by their centroids c_1 and c_2 respectively. The aggregate Mahalanobis distance D^m of the strategy x to the clusters ς_1 and ς_2 is then calculated as Equation (8).

The value of D^m lies between 0 and 1, with value close to 0 suggesting that x is the worst strategy and value close to 1 suggesting that x is the best strategy respectively. The decision rule of the case-based PD-clustering system suggests a strategy with the highest aggregate Mahalanobis distance.

V. EXPERIMENTAL TEST AND RESULT

The CBPd-Clustering technique was implemented in C++ using C++ Builder 6 integrated development environment running on Microsoft Windows XP Professional Version 2002 operating system installed on Intel® core ™i7 CPU 870"293 GHz, 3.18GB RAM. In

order to test the effectiveness of the CBPD-clustering system, we organized a tournament between the CBPD-clustering system, and Awale shareware (simply put as Awale). The results obtained from a series of six games played at each level, for which each player was allowed to start thrice are recorded in Table I. Sample screenshot of game played between Awale and CBPd-Clustering is shown in Figures 3 and 4.

The result of the tournaments shows that the CBPD-clustering remains stronger at all levels. Note that the difference in the percentage score account for the remaining seeds on the board when the game is over at each round of play.

As shown in Table 1, the evolved CBPd-Clustering technique performs very strongly across the game tournament. It comprehensively defeats the Awale shareware in all game play in the tournament with a very strong starting strategy.

5.1 Discussion

The result in table 1 shows the percentage of wins in a tournament between the CBPD-clustering system, and Awale. In the tournaments, there were four (4) levels of play namely Initiation, Beginner, Amateur and Grandmaster. At each levels of play there are six (6) games starting from Game 1 to Game 6 in two (2) different rounds. In round one (1), CBPD- Clustering starts the game first while Awale plays second while in round two (2) Awale starts the game first and CBPD-Clustering plays second.

At each level of play, the CBPD-Clustering has the highest percentage of wins ranging from 77.6% to 99.6%, except once at grand master level in game 1 when CBPd-clustering starts first and capture 58.1% of the seeds while Awale had 41.7 % (see figure 3); and once when Awale starts the game play first where CBPD-Clustering capture 87.1% of the seeds while Awale had 12.8 %; as seen on the screenshot in Figure 4. More importantly, one could see that the average game move increases as the tournament progresses to higher levels (that is, from initiation to grandmaster level) with the CBPD-Clustering having the highest playing percentage in the tournaments.

VI. CONCLUDING REMARK

In this paper, we have evolved an Ayo game player using CBPD-clustering method with an improved endgame database. The developed system has been tested and the results show a reasonably good performance of the evolved system over Awale shareware.

Effort would be made in future to do a thorough comparative analysis of game play strategy between CBPd-Clustering technique and Awale shareware and make a display of every tournament in a number of snapshots in order to showcase the strength of the CBPd-Clustering technique.

REFERENCES

[1]. A. O. Odeleye. Ayo: A popular Yoruba game, Oxford University Press, Ibadan, Nigeria, 1977.

[2]. I. O. Akinyemi; H. O. Dele Longe, and O. O. Olugbara. On the Performance Evaluation of Refinement-Based Heuristics Strategy to Evolve Ayo Game Player. Asian Journal of Information Technology.15 (19): 3624 – 3630, 2016.

[3]. R. J. Van. "Learning from Perfection: A Data Mining Approach to Evaluation Function Learning in Awari". *In proceedings* of the 2nd International Conference (CG-00), Hamamatsu, vol. 2063, pp.115-132, 2001.

[4]. J. W. Romein and H. E. Bal. "Awari is Solved". *Journal of the International Computer Games Association (ICGA)*, vol. 25, pp.162-165, 2002.

[5]. I. O. Akinyemi; E. F. Adebiyi and H. O. D. Longe. Critical Analysis of Decision Making Experience with a Machine Learning Approach in Playing *Ayo* Game. World Academy of Science, Engineering and Technology (32), pp 49 – 54, 2009.

[6].Myriad software, http://www.myriad-online.com/awale.htm Accessed 14/04/2013.

[7]. T. R. Lincke and A. Marzetta. Large endgame databases with Limited Memory Space ICGA Journal, 23, 3, pp. 131-138, 2000.

[8]. V. Allis; M. V. D. Muellen and J. V. Herik. Proof-number Search, Artificial Intelligence, vol. 66, pp. 91-124, 1994.

[9]. T. R. Lincke. Strategies for the Automatic Construction of Opening Books: Computers and Games, pp. 74-86, 2000.

[10]. H. E. Bal and L. V. Allis. Parallel Retrograde Analysis on a Distributed System. *Proceedings of the 1995 ACM/IEEE Conference on Supercomputing.* pp. 1.-15, 1995.

[11]. K. Thomson. Retrograde Analysis of Certain Endgames, ICCA Journal, vol. 9, no. 3, pp. 131-139, 1986.

[12]. J. W. Romein and H. E. Bal. Notes Awari is Solved, Journal of the ICGA, vol. 25, pp. 162-165, 2002.

[13]. J. V. D. Rijswijck. Learning from Perfection: A Data Mining Approach to Evaluation Function in Awari. In proceedings of the 2nd International Conference (CG-00), vol. 2063, pp. 115-132, 2001.

[14]. J. E. Davis and G. Kendall. An Investigation, using co-evolution, to evolve an Awari Player. In proceedings of Congress on Evolutionary Computation (CEC 2002), pp. 1408-1413, 2002.

[15]. M. Daoud; N. Kharma; A. Haidar and J. Popoola. Ayo, the Awari Player, or How Better Representation Trumps Deeper Search, Proceedings of the 2004 IEEE Congress on Evolutionary Computation, pp. 1001-1006, 2004.

[16]. I. O. Akinyemi; O. O. Olugbara and H. O. D. Longe. An Endgame Procedure for Generating Integer Sequence. African Journal of Computing & ICT. Vol. 6. No. 4, pp 143– 150, 2013.

[17]. A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications, 7(1) 39-59, 1994.

[18]. J. Hu; J. Qi and Peng Y. New CBR adaptation method combining with problem- solution relational analysis for mechanical design. Computers in Industry, 66(1), 41-51, 2015.

[19]. A. Yan and D. Wang. Trustworthiness evaluation and retrieval-based revision method for case-based reasoning classifiers. Expert Systems with Applications, 42(1), 8006-8013, 2015.

[20]. Z. Fan; Y. Li; X. Wang and Y. Liu. Hybrid similarity measure for case retrieval in CBR and its application to emergency response towards gas explosion. Expert Systems with Applications, 41(5) 2526-2534, 2014.

[21]. A. Yan; H. Shao and P. Wang, A soft-sensing method of dissolved oxygen concentration by group genetic case-based reasoning with integrating group decision making. Neurocomputing, 169 422-429, 2015.

[22]. M. Han and Z. Cao. An improved case-based reasoning method and its application in endpoint prediction of basic oxygen furnace. Neurocomputing, 149(C), 1245-1252, 2015.

[23]. G. Xing; J. Ding; T. Chai; P. Afshar and H. Wang. Hybrid intelligent parameter estimation based on grey case-based reasoning for laminar cooling process. Engineering Applications of Artificial Intelligence, 25(2), 418-429, 2012.

[24]. C. Wang and H. Yang. A recommender mechanism based on case-based reasoning. Expert Systems with Applications, 39(4) 4335-4343, 2012.

[25]. Z. Fan; Y. Li and Y. Zhang. Generating project risk response strategies based on CBR: A case study. Expert Systems with Applications, 42(6) 2870-2883, 2015.

[26]. S. Zhong; X. Xie and L. Lin. Two-layer random forests model for case reuse in case-based reasoning. Expert Systems with Applications, 42(24), 9412-9425, 2015.

[27]. O. O. Olugbara; M. O. Adigun; S. O. Ojo and T. O. Adewoye. An efficient heuristic for evolving an agent in the strategy game of Ayo, ICGA Journal, 30, 92-96, 2007.

[28]. O. O. Olugbara; T. O. Adewoye and I. O. Akinyemi. An Investigation of Minimax Search technique for Evolving Ayo/Awari Player. Proceedings of IEEE-ICICT 4th International Conference on Information and Communication Technology, Cairo, Egypt, 2006.

[29]. A. K. Jain; M. N. Murty and P. J. Flynn. Data clustering: a review. ACM Comput. Surveys 31 (3), 264–323, 1999.

[30] P. Bradley; U. Fayyad and C. Reina. Scaling clustering algorithms to large databases. In: The Fourth International Conference on Knowledge Discovery and Data Mining, AAAI, NY, 1998.

[31]. C. Iyegun and A. Ben-Isreal. Probabilistic Distance Clustering Adjusted for Cluster Size. Probability in the Engineering and Informational Sciences, 22, 603-621, 2008.

[32]. P. C. Mahalanobis (1936). On the generalized distance in statistics. Proceedings of the national Institute of Science of India, pp. 49- 55, 1936.

[33]. D. R. Maesschalck; D. Jouan-Rimbaud and D. L. Massart. The Mahalanobis distance. Chemometrics and Intelligent Laboratory Systems, 50, pp. 1-18, 2000.

[34]. D. M. Broline and D. E. Loeb. The Combinatorics of Mancala-type, 1995. Available online: http:/www.arxiv.org/ps_cache/math/.pdf/9502/95022225.pdf.

[35]. T. O. Adewoye. On Certain Combinatorial Number Theoretic Aspects of the African Game of Ayo. AMSE REVIEW, 14(2), pp. 41-63, 1990.

Figure 1: Players of *Ayo G*ame [2]

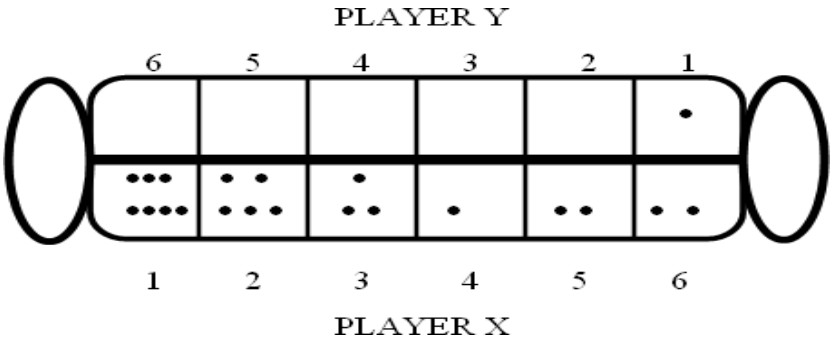

Figure 2: CDG Initial Configuration in Ayo Game

Start

Input n (the size of the board = 6)
Output $x = (x_n, x_{n-1}, \ldots, x_2, x_1)$ (x is the pit for seeds on the board)

Step 0: $x_n = n + 1$, $d_n = 2$, $j = n - 1$

If j<1 stop otherwise go to step 1

Step 1: $x_j = (j+1)\left\lfloor \frac{|(j+1) - (x_{j+1} - d_{j+1})|}{j+1} \right\rfloor + (x_{j+1} - d_{j+1})$

If $j = 1$, stop otherwise, go to step 2

Step 2: $d_j = d_{j+1} + 2\left\lfloor \frac{|(j+1) - (x_{j+1} - d_{j+1})|}{(j+1)} \right\rfloor$

$j = j - 1$

Go to step 1

End

Figure 2.1: Algorithm for the Implementation of the maximal CDG-Vectors

APPENDIX

Extrema Problem

$$\min \sum_{i=1}^{n} \left(\frac{d_1(x_i)P_1(x_i)^2}{q_1} + \frac{d_2(x_i)P_2(x_i)^2}{q_2} \right) : q_1 + q_2 = N, q_1, q_2 \geq 0 \quad (3)$$

where d_1x and d_2x are distances of the data point *x* to the clusters of sizes q1 and q2 and p_1x and p_2x are the cluster probabilities. In order to solve Equation (3), the Lagrangian of the problem is defined as:

$$L(q_1, q_2, \lambda) = \sum_{i=1}^{n} \left(\frac{d_1(x_i)P_1(x_i)^2}{q_1} + \frac{d_2(x_i)P_2(x_i)^2}{q_2} \right) + \lambda(q_1 + q_2 - N) \quad (4)$$

Distance Metric

$$d(x,y) = \| x - y \|, \forall\, x, y \in \mathbb{R}^n \quad (5)$$

where ||.|| is a norm that can be Chebychev, Proscrute, Euclidean or Mahalanobis. The Mahalanobis metric is generally preferred to the Euclidean because it is consistent across conditions and it pays equal attention to all components. The Mahalanobis distance function is written as [31, 32, 33]:

$$d(x, c_k) = \{x - c_k^T (\sum_k^{-1} x - c_k)\}^{\frac{1}{2}} \quad (6)$$

Where A^T means transpose vector of A and $\sum_k^{-1}$ is the inverse matrix of the covariance matrix $\sum_k$ given by

$$\sum_k^{-1} = \frac{\sum_i^N u_k(x_i) x_i - (c_k\, x_i - c_k)^T}{\sum_{i=1}^{N} u_k(x_i)} \quad (7)$$

Aggregate Mahalanobis distance

$$D_x^m = \frac{\{(x-c_2)^T \sum_2^{-1} (x-c_2)\}^{1/2}}{\{(x-c_2)^T \sum_2^{-1} (x-c_2)\}^{1/2} + \{(x-c_1)^T \sum_1^{-1} (x-c_1)\}^{1/2}} \quad (8)$$

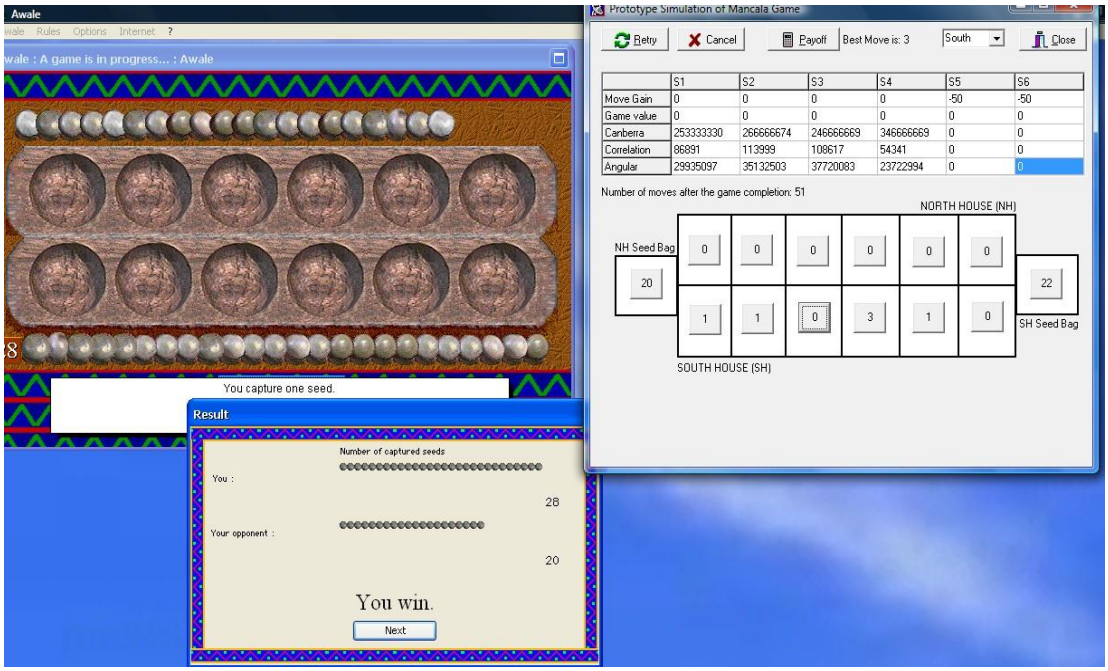

Figure 3: Screenshot Showing Complete Game Play With Total Seeds Capture and Number of Moves When CBPd-Clustering Starts First.

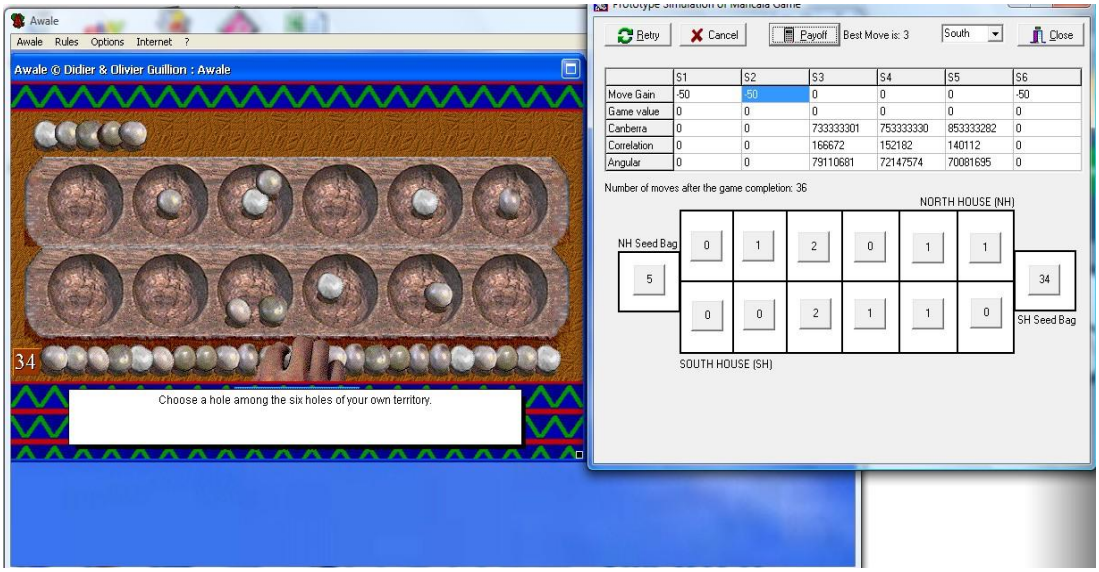

Figure 4: Screenshot Showing Complete Game Play With Total Seeds Capture and Number of Moves When Awale Starts First.

Table 1: Result of Game Tournament Between CBPD-Clustering and Awale

Level of Play	Average Game Moves	**Round 1** CBPd-Clustering (Start First)		**Round 2** Awale (Start First)	
		% Wins by CBPd-Clustering	% Wins by Awale	% Wins by CBPd-Cluster	% Wins by Awale
Initiation					
Game 1	19	81.5	16.5	77.6	20.0
Game 2	18	91	7.0	88.8	10.0
Game 3	27	99.6	0.1	97.7	2.1
Game 4	23	97.7	2.0	98.9	0.1
Game 5	19	99.5	0.5	99.0	0.9
Game 6	21	98.5	1.1	96.4	2.4
Beginner					
Game 1	37	84.9	13.7	85.7	12.6
Game 2	38	89.2	10.0	90.0	9.1
Game 3	36	99.1	0.5	99.8	0.1
Game 4	33	96.2	3.1	97.1	2.5
Game 5	36	81.2	18.8.	95.3	4.5
Game 6	34	95.4	4.1	99.8	0.2
Amateur					
Game 1	44	84.1	14.9	77.6	20.0
Game 2	47	89.2	10.2	88.8	10.1
Game 3	41	99.0	0.9	97.7	2.1
pGame 4	48	95.1	4.0	99.1	0.1
Game 5	52	93.1	6.9	96.4	0.9
Game 6	48	97.3	2.7	87.9	1.8
Grandmaster					
Game 1	36	58.1	41.7	75.8	22.0
Game 2	43	87.2	11.6	87.1	12.8
Game 3	41	89.1	10.1	86.8	11.1
Game 4	47	99.1	0.6	86.9	11.9
Game 5	44	97.2	2.5	98.9	0.7
Game 6	48	99.3	0.1	98.7	1.1