

A STRATEGIC HYBRID TECHNIQUE TO DEVELOP A GAME PLAYER

Randle O.A

¹Department of Computer Science, Tshwane University of Technology, Soshanguve,
South Africa

rotran86@yahoo.com

ABSTRACT

The game of Awale is a member of the mancala family and there have been reoccurring issues about the heuristics used for playing the game and therefore it is an open issue. This study introduces the concept of a new idea (heuristic) to evolve the game player by combining minimax with ADMF. The results show the combination is viable and interesting.

KEYWORDS

Game tree, Minimax search, Endgame databases, Awale

1. INTRODUCTION

Game playing has been very successful especially in the aspect of Ayo(Awale or Awele) but one problem still eludes the game, which is what heuristic can be used for playing Mancala games by a computer [1]. Two Person Zero Sum (TPZS) games which provided sufficient information can be solved but these methods of computing these solutions were only realised after the initiation of advanced digital systems [2, 3]. There have been similarities or equalities found between TPZS games and Linear Programming (LP) and thus has been implemented frequently in solving game problems [4,5]. The limitation of linear programming are the fact that they are inadequate for hyper-myopic decisions which is needed in Awale board games and secondly LP are expensive [5]. The best choice for hyper-myopic forecast is Minimax Search Algorithm(MSA) and has been successfully implemented and evolved for game playing such as chess where massive parallel hardware-orientated alpha-beta search(an improved version of minimax search) was implemented [7,8,9].

The MSA constructs a game tree in ascending order from the root and then employs backward induction to predict the game value as it descends the tree from the leaves. The Principal Variation (PV) (principal continuation or critical path) is the optimum path that suggests best moves for players of TPZS game and most programs try to determine this path. In the past, several games were reported utilizing MSA, but Ayo was rarely mentioned until recently. The strongest Ayo program today is believed to be Awale, developed by Didier and livier Guillion of Myriad software [10], but the technique upon which this shareware was built has not been widely made public.

Endgame databases [11] have been offered for evolving Ayo players, while a combination of alpha-beta search algorithm and endgame databases was used to evolve lithindion [12]. Drop-out expansion which is a hybrid method was used to develop an Awale player called Marvin [13]. Softwari [14] is a player that constructs large endgame databases. All these methods are unique with one common characteristic which is the fact that they focus on searching and database utilization and did not pay attention to evaluation function.

Finding maximum play for all possible board positions has been the aim of Retrograde Analysis (RA) [15,16]. This technique is applicable to search spaces which can be enumerated completely within the memory of a computer. Genetic algorithm has also been used to evolve a player [17] due to its ability to mine endgame databases for relevant characteristics. Another technique was the hybrid technique which is a combination of minimax search and co-evolution [18] and was further enhanced by [19].

The major difficulty posed by minimax search is how an evaluator is developed and applied to a game tree. Evaluator design is an important consideration in games and computer game programs are differentiated by the quality of their evaluators. The fundamental problems that can arise in using minimax search are: (1) how to design a suitable evaluator and (2) how to select a correct move without the rationality assumption. In reality, a player can be irrational in move selection as a play strategy like bluffing/trap setting as applicable to Ayo and in general, we would not base move selection on rationality assumption. Bluffing is a powerful play strategy in Ayo and we define it as the ability to tradeoff an invaluable number of seeds so as to gain advantage. Two important factors that must be taken into consideration when bluffing are: when to bluff and the number of seeds (here called tradeoff seeds) to tradeoff.

2. MINIMAX SEARCH AND ADMF

Generally, the value of a leaf is estimated by the evaluator and represents the number in proportion to the chance of winning the game. The evaluator can be extended to the minimax function, which determines the value for each player in a node and is formally given in (1) as follows [20, 21]:

$$f(n) = \begin{cases} eval(n), & \text{if } n \text{ is a leaf node} \\ \max\{f(c) \mid c \text{ is a child node of } n\}, & \text{if } n \text{ is a max node} \\ \min\{f(c) \mid c \text{ is a child node of } n\}, & \text{if } n \text{ is a min node} \end{cases} \quad (1)$$

The function eval(n) scores the resulting board position at each leaf node n. The standard method of scoring is in terms of a linear polynomial [22]. It has been shown that every game tree algorithm constructs a superposition of a max (T^+) and a min(T^-) solution tree. The equivalent evaluator is the following Stockman equality [23]:

$$f(n) = \begin{cases} \max\{g(T^-) \mid T^- \text{ is a min tree rooted in } n\} \\ \min\{g(T^+) \mid T^+ \text{ is a max tree rooted in } n\} \end{cases} \quad (2)$$

Where the function g is defined by [23]:

$$\begin{aligned} g(T^+) &= \max\{f(c) \mid c \text{ is a terminal in } T^+\} \\ g(T^-) &= \min\{f(c) \mid c \text{ is a terminal in } T^-\} \end{aligned} \quad (3)$$

AMDF does not construct the PV directly, but tries to predict it using a technique that accepts a set of moves and then evaluates each move and returns a single move with the best advantage. The problem for the current work is as follows: Given a computer (C) and a human player (H), if H can defeat C, is it possible for C to use the strategies of H to defeat H and any other player that H can defeat? This is the objective of the AMDF algorithm and machine learning techniques like neural network, Refinement Assisted Minimax(RAM), nearest neighbour search and case-based reasoning are very important refinement tools that we have investigated for accomplishing this arduous prediction task.

Conventionally, the basic idea of minimax algorithm is synonymously related to the following optimization procedure. Max player tries as much as possible to increase the minimum value of the game, while Min tends to decrease its maximum value at node n as both players play towards optimality. The entire process can be formally described by the following extended Stockman formula (4) below:

$$f(n) = \begin{cases} \max\{f(c) \mid c \text{ is a child node of } n\} - f(n), & \text{if } n \text{ is a min node} \\ \min\{f(c) \mid c \text{ is a child node of } n\} + f(n), & \text{if } n \text{ is a max node} \end{cases} \quad (4)$$

3. Implementation of Aggregate Malanobis Distance Function

The algorithm is designed in such a way that the moves (strategies) are classified into two (2) classes c_1 and c_2 which represent good and bad strategies respectively-. Using the Aggregate malanobis distance function (AMDF) which finds the malanobis distance [24, 25] of each strategy on the current board state for both classes of strategies. The result of the bad strategy is then divided by the sum of both the good and bad strategies. Thereby selecting the highest possible score as the best .The AMDF algorithm is described more compactly by the following flow chart:

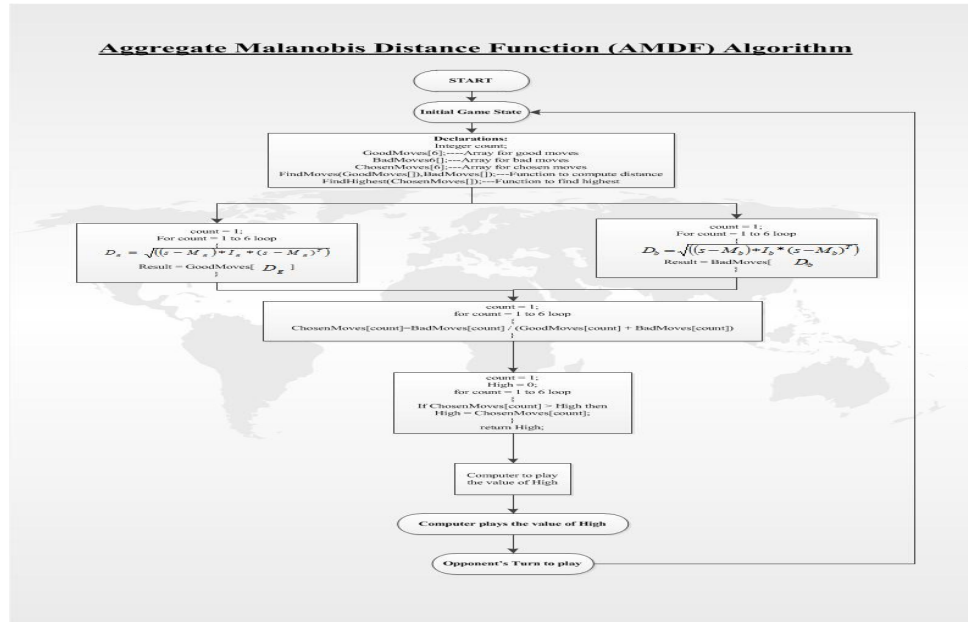


Figure 3.1 Flowchart of our algorithm

Furthermore we provide the mathematical explanation for our evolved player, where given data points are given as vectors $x = (x_1, x_2, x_3, \dots, x_n) \in \mathfrak{X}^n$ and let a dataset D consists of N data points $\{x_1, x_2, x_3, \dots, x_n\}$. The general problem of data clustering is to partition a dataset into m clusters of similar data points. The pd-clustering technique relates probability and distance using a simple inverse principle. For each $x \in D$ and cluster centroid c_k , the probability $p_k(x)$ that x belongs to D is given as [26].

$$\frac{p_k(x)d_k(x)}{q_k} = T \tag{5}$$

where T is a constant.

This result can be interpreted as meaning that cluster membership is more probable the closer the data point is to the cluster centroid and the larger the cluster. [26] have shown that Equation (5) is the solution of the following extremal problem:

$$\min \left\{ \sum_{i=1}^N \left(\frac{d_1(x)p_1^2(x)}{q_1} + \frac{d_2(x)p_2^2(x)}{q_2} \right) \middle| p_1(x) + p_2(x) = 1, p_1(x), p_2(x) \geq 0 \right\} \tag{6}$$

Where $d_1(x)$ and $d_2(x)$ are distances of the data point x to the cluster of size q_1 and q_2 and $p_1(x)$ and $p_2(x)$ are the cluster probabilities. To solve Equation (6), the Lagrangian of the problem is defined as:

$$L(p_1(x), p_2(x), \lambda) = \sum_{i=1}^N \left(\frac{d_1(x) p_1^2(x)}{q_1} + \frac{d_2(x) p_2^2(x)}{q_2} \right) - \lambda (p_1(x) + p_2(x) - N) \quad (7)$$

By zeroing the partial derivatives $\frac{\partial L}{\partial p_1}$ gives the solution to Equation (7) as follows:

$$p_k(x) = \frac{\prod_{j \neq k} d_j(x) / q_j}{\sum_{i=1}^K \prod_{j \neq i} d_j(x) / q_j} \quad (8)$$

where k is the number of clusters. The distance function $d(x, y)$ that measures the closeness of the vectors x and y is usually given as:

$$d(x, y) = \|x - y\|, \forall x, y \in \mathfrak{R}^n \quad (9)$$

where $\|\cdot\|$ is a norm. There are several norms for distance computation and examples include Chebycheu, Proscrute, Euclidean and Mahalanobis, which is preferred than the Euclidean because it is consistent across conditions and it pays equal attention to all components.

$$d(x, c_k) = \left\{ (x - c_k)^T \sum_k^{-1} (x - c_k) \right\}^{1/2} \quad (10)$$

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

$$\sum_k = \frac{\sum_i^N u_k(x_i) (x_i - c_k) (x_i - c_k)^T}{\sum_i^N u_k(x_i)} \quad (11)$$

Where

$$u_k(x_i) = \left[\left(\frac{d_1(x_i, c_1)}{q_1} \right)^2 \frac{1}{d_2(x_i, c_2)} \right] \left(\frac{d_1(x_i, c_1)}{q_1} + \frac{d_2(x_i, c_2)}{q_2} \right)^{-2} \quad (12)$$

4. Experiment and Result

To evaluate the performance of our evolved player, the player played a series of games with Awale in tournaments. First, we registered for the Awale program before a direct access is given to play at all levels, but the initial level is free. The results obtained from the series of 10 games at each level of the game (each player starts 5 times) played are recorded in the table below.

We record moves up to the level where a player has captured the maximum number of seeds required to win or draw. However, in practice the game continues until fewer seeds (say three) remain on board. The four levels are;

- (1) Playing Awale at Initiation level
- (2) Playing Awale at beginner level
- (3) Playing Awale at Amateur level
- (4) Playing Awale at Grandmaster level

Table 1: Results of the experiment

Level	Noof moves(average)	No of seeds won by evolved player (average)	No of seeds won by Awale(average)
Initiation	23	26	5
Beginner	33	26	7.6
Amateur	39.6	27	18.6
Grand master	61.6	13.4	25

The results in the Table 1 shows that the evolved player performed very well against awale at the 3 initial stages and also competed strongly against Awale at the grand master stage but lost to the Grandmaster.

5. Conclusion

The results of the experiment has shown the performance of the evolved player has been good. ADMF algorithm is computationally efficient and predicts the best move within a very short time. ADMF can improve AI performance and makes computer players more adaptable and responsive. We have realised that it is unnecessary to search large game postions to evolve a player that performs pretty well also the deeper the search the less effective the method becomes and the longer it takes to suggest moves.

ACKNOWLEDGEMENTS

I would like to thank my mentor Prof O.O Olugbara for his contributions to this paper.

REFERENCES

- [1] Donkers, H.H.L.M., Uiterwijk, J.W.H.M. and Voogt, A.J.D.V.(2002) Mancala Games- Topics in Artificial Intelligence and Mathematics. Step by Step Proceedings of the 4th Colloquium Board Games in Academia.
- [2] Nash, J.(1953) Two-Person Cooperative Games. *Econometrics* 21, pp. 128-140
- [3] Washburn, A.(2001) A New Kind of Fictitious Play, <http://www.nps.Navy.mil/papers/>
- [4] Dantzig, G.(1951) A proof of the equivalence of the programming problem and game problem, *Activity analysis of production and allocation*, Koopmans (ed.) Wiley, New York, pp. 339-347.
- [5] Ayeni, J. O. A and Longe, H. O. D.(1985) Game People Play: Ayo. *International Journal of Game Theory*, Vol. 14 issue 4, pp. 207-218.
- [6] Knuth, D.E. and Moore, R.W(1975) An Analysis of Alpha-beta Pruning, *Artificial Intelligence* 6, 4, 293-326.
- [7] Thompson, K.(1982) Computer Chess Strength, in *Advances in Computer Chess* 3, M.Clarke (ed.), Pergamon Press, Oxford, pp. 55-56.
- [8] Thomson, K.(1996) 6-piece Endgames, *ICCA Journal*, vol. 19, no. 4, pp. 215-226.
- [9] Hamilton, S. and Garber, L.(1997) Deep Blue's hardware-software synergy, *IEEE Computer*, 30, 10, pp. 29-35.
- [10] Myriad software, <http://www.myriad-online.com/awale.htm>
- [11] Lincke, T.R and Marzetta, A.(2000) Large endgame databases with Limited Memory Space. *ICGA Journal*, 23, 3, pp. 131-138.
- [12] Allis, V., Muellen, M. V.D. and Herik, J.V.D(1994) Proof-number Search, *Artificial Intelligence*, vol. 66, pp. 91-124.
- [13] Lincke, T.R.(2000) Strategies for the Automatic Construction of Opening Books: Computers and Games, pp. 74-86.
- [14] The University of Alberta Computer Awari Group, <http://www.cs.ualberta.ca/~awari/>
- [15] Bal, H.E. and Allis, L.V.(1995) Parallel Retrograde Analysis on a Distributed System.Supercomputing.
- [16] Thomson, K. (1986) Retrograde Analysis of Certain Endgames, *ICCA Journal*, vol. 9, no. 3, pp. 131-139.
- [17] Rijswijk, J.V.D.(2001) 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.
- [18] Davis, J.E. and Kendall, G. (2002) An Investigation, using co-evolution, to evolve an Awari Player. In proceedings of Congress on Evolutionary Computation (CEC 2002), pp. 1408-1413.
- [19] Daoud, M., Kharm, N., Haidar, A. and Popoola, J.(2004) Ayo, the Awari Player, or How Better Representation Trumps Deeper Search, *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pp. 1001-1006.
- [20] Bruin, A.D., Pijls, W. and Plaat, A. (1994) Solution Trees as a Basic for Game Tree Search, *ICCA Journal*, 17, 4, pp. 207-219.
- [21] Bruin, A.D. and Pijls, W. (1996) Trends in Game Tree Search. *SOFSEM*, pp. 255-274.
- [22] Samuel, A.L.(1959) Some Studies in Machine Learning Using the Game of Checkers, *IBM J. of Res. And Dev.* 3, 210-229.
- [23] Pijls, W. and Bruin, A.D. (2001) Game Tree Algorithms and Solution Trees, *theor.Compt. Sci.*, 252 (1-20): pp. 197-215.
- [24] Mahalanobis, P.C. (1936) On the generalized distance in statistics. *Proceedings of the national Institute of Science of India*, pp. 49- 55.
- [25] Maeschalck, De .R., Jouan-Rimbaud, D., massart, D.L.(2000) The Malanobis distance. *Chemometrics and Intelligent Laboratory Systems*, pp. 1-18.
- [26] Iyegun, C. and Ben-Isreal, A. (2008) Probabilistic Distance Clustering Adjusted for Cluster Size. *Probability in the Engineering and Informational Sciences*, 22, 603-621.