

On the Performance Evaluation of Refinement-Based Heuristics Strategy to Evolve Ayo Game

¹O. Akinyemi, Ibidapo, ²Longe, Harrison O. Dele and ³O. Olugbara, Oludayo

¹Department of Computer and Information Sciences, Covenant University, Ogun State, Ota, Nigeria

²Department of Computer Sciences, University of Lagos, Lagos, Nigeria

³Department of Computer Sciences, Durban University of Science and Technology, Durban, South Africa

Abstract: The volume of research on Artificial Intelligence (AI) is a motivating factor for researchers in AI to solve a number of difficult problems in real life situations. One of such problems is game playing which has become one of the most interesting AI applications to the public. Ayo game belongs to the family of the board game called mancala and it is one of the oldest games of strategy known among the Yorubas in Nigeria. Some attempts have been made to solve the game but due to its irregular pattern as the game progresses, the exact solution had hitherto not been found. We have used refinement-based heuristic approach which instead of exploring the entire search space of Ayo, pruned part of the space that are not relevant and thus reduce the amount of computation to evolve Ayo game player with a view to suggesting move strategies. In this research we evolve Ayo game player through simulation which was tested against Awale shareware. For the purpose of evaluation we designed a paper-based questionnaire to harvest users' impression about the performance of our prototype simulation through a series of game playing experiment with Awale shareware and found that the prototype simulation is computationally efficient and a better user satisfaction results were obtained.

Key words: Ayo game, move refinement, minimax search, heuristics, evaluation

INTRODUCTION

Most game problems are difficult to solve since they are either PSpace or ExpTime in nature. To solve this problem, researchers have resorted to explore heuristics from AI since exact solution is unrealistic (Daoud, *et al.*, 2004). The problem of finding heuristic to play mancala family of board games is still an open issue in AI research. Heuristic search is particularly beneficial to problems with well-defined initial conditions such as single start state distribution. Instead of exploring the entire search space, an appropriate heuristic function helps to prune parts of the spaces that are not relevant and this reduces the computation. We have developed a prototype simulation of an Ayo player using Refinement-Based Heuristic (RBH) strategy that followed minimax search method to build game tree, heuristically evaluates the leaves, backup the heuristic values to the root and use them to select a move. Although, practice have shown that the deeper the search programs search the better they are. However, mathematical models intended to explain this formally paradoxically showed that deeper search result in worse play (Beal, 1980; Nau, 1983). Since the score of a position exactly foretell the result at the end of the game when

both players played optimally, our RBH algorithm evaluates the score of each game position and suggest the best position to move from and as well take cognizance of the opponent's action relative to its own play at any instant of play which does not require any game databases as opposed the work of (Romein and Bal, 2002) that used Retrograde Analysis (RA) to create a number of databases knowing fully well that both game databases and retrograde approaches can be very expensive to implement. The goal of this study is to evaluate the performance of refinement-based heuristic approach to evolve Ayo game player with a view to suggesting move strategies in Ayo game.

Brief description of Ayo: Around the world, various versions of mancala games (to which Ayo belongs) have been observed dating back to the Empire Age of ancient Egypt (Murray, 1952). Players take turns harvesting seeds by moving around the board according to various rules. Ayo game is the most popular board game among the Yorubas who occupy roughly the south-western states of Nigeria and parts of Republic of Benin. Ayo game is one of the oldest games of strategy known. It is a game of perfect information known as combinatorial. It is a two

player game with no hidden information, no chance move, a restricted outcome (win, lose and draw) and with each player moving across the board. Ayo is a game of strategy which has been shown to be of great use in solving human psychological related problems due to its attributes (Ayeni and Longe, 1985). Two persons play Ayo at a time with the board put in between the players (Akinyemi *et al.*, 2009) as shown in fig 1 below. The board is a hollow out plank of wood consisting of two rows of six pits belonging to either row and each pit contains four seeds of the plant “*Caesalpinia crista*” (Odeleye, 1997) such that a total of forty-eight seeds are contained in a board at the start of the game. Often, there are two extra hollow normally placed centrally at the end of each row of the board. These are called “seed bags” that are used to store the captured seeds by each of the players. As the game progresses, each pit can contain any number of seeds or no seed at all. Just like any other game, the ultimate objective of the game is to capture more seeds than the opponent to emerge as winner or have equal number of seeds as a drawn game. As a seed is captured, it is removed from the board and put in the seed bag and plays no further part, other than being used to evaluate the current game position (Fig. 1).

The objective of the players is to capture their opponent’s seeds (as many as possible). A move consists of a player choosing a non-empty pit on his side of the board and removing all of the seeds contained in that pit. The seeds are redistributed (sown) one seed per pit. The seeds are sown in counter clockwise direction from the pit that has been just emptied. A pit containing 12–17 seeds is called an Odu while the one that contains 23–27 seeds is regarded as Ikare (Adebiyi, 1999). If the chosen pit is an Odu or Ikare, the same redistribution continues but the emptied pit is skipped on each circuit of the board (Odeleye, 1997). A capture is made when the last pit sown is on the opponent’s side and contains after the addition of the sowing seed either two or three seeds. Thus, the seeds in the pit are captured and removed from the game. Also captured are the immediately preceding pits which meet the same conditions. One important feature of this game is that each player has to make a move such that his opponent has a legal move to play. If this does not happen, then the opponent is rewarded with all the remaining seeds on the board. This rule is referred to as the golden rule (Adewoye, 1990). If during the game, it is found that there are not enough seeds to make a capture but both players can always proceed with a legal move, the game is stopped and the players are awarded seeds that reside on their respective side of the board. The initial game is rapid and much more interesting, where both the players capture seeds in quick succession. To determine



Fig. 1: Player of Ayo game

the optimal strategy during the initial play is hard and thus has not yet been studied. It involves planning at least 2–3 moves in advance and remembering the number of seeds in every pit (Odeleye, 1997; Broline and Loeb, 1995).

Literature review: Over the years, a number of games have been reported to have utilized Minimax Search Algorithm (MSA) but Ayo was rarely mentioned until recently. The strongest Ayo program today is believed to be Awale (Myriad software) but the technique upon which this shareware was built has not been widely publicized. However, endgame databases (Lincke and Marzetta, 2000) have been offered for evolving Awari player. In (Allis *et al.*, 1994) Lithindion was used which is an artificial Awari player that uses a combination of alpha-beta search and endgame database. Similarly, Ayeni and Longe (1985) opined that solving Ayo game as a linear programming problem could be expensive but most suitable for myopic decision. It is therefore, inadequate for futuristic (or hyper-myopic) decision which is of paramount interest in Move-and-Capture Games (MCG) such as Ayo. Retrograde Analysis (RA) has been used to solve Awari (Romein and Bal, 2002) by searching the entire state space on a parallel computer with 144 processors. As reported by the researchers, the state space contains 889,063,398,406 positions and was searched using RA. It was admitted by the authors that no single computer has enough processing power and memory to search the state space but even on a modern, parallel computer the problem was extremely challenging.

This is a major drawback alluded to RA because such a method could not be easily implemented on a small memory device like wireless handset for playing Ayo game. Furthermore, both endgame database and retrograde approaches can be very expensive to implement. In contrast, a Genetic Algorithm (GA)

technique was proposed by Rijswijk to mine endgame databases for relevant features, useful for design of a static evaluation function, simply called evaluator. The hybrid of co-evolution and minimax has been investigated for evolving Awari player (Davis and Kendall, 2002) but such method cannot suggest a best move until after learning. Consequently, it has poor starting ability and suffers overlay delay to suggest a move. The first problem posed by minimax search is how an evaluation function is developed and applied to the game tree, since computer game programs are differentiated by the quality of the evaluation. The research of Davis and Kendall (2002) uses evolutionary strategy to evolve a simple evaluation function and the output of the function is then used in a minimax search. They reported that their implementation took about one minute to suggest a move for a search depth of seven but this is not too efficient for a computer that is expected to play at a faster speed. The proposed evaluator was a linear combination of six 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. An attempt was made by Olugbara (2006) to solve the game of Ayo by using a refinement procedure in the investigation of minimax search technique for playing Ayo game. Equally, Akinyemi *et al.* (2008) described a method of using nearest neighbor strategy to evolve Ayo game playing. Similarly, Akinyemi *et al.* (2009) presents a heuristic method for evolving Ayo game playing with an appealing result.

MATERIALS AND METHODS

Evolving Ayo game player:

Formal representation of Ayo game: In Ayo game a position consists of a certain distribution of seeds in the pits and it includes the captured seeds in the stores or kept by the players if there are no stores. In addition, a position includes the knowledge which player is to move next. The number of possible positions is a function of the number of pits (and stores) and the number of seeds. Generally, for mancala family, this number has been derived from basic combinatorics and is given by Donkers as:

$$p = k \binom{n+m-1}{m} \tag{1}$$

Where:

- k = The number of players
- m = The total number of seeds and
- n = Either the total number of pits and stores together or the total number of pits incremented with the number of players if no stores are present

The process of seed sowing in Ayo describe a linear motion that can repeat θ times, called θ cycles, usually $\theta = 0, 1, 2$ and so on. When $\theta = 0$, the distance travelled is less than $n - k$ but when $\theta > 0$, the distance travelled is greater than $n - k$ and this is called kroo (Daoud *et al.*, 2004) or Odu (Broline and Loeb, 1995; Odeleye, 1997). Let (i, j) denote a pair of positions that both players (North and South) can scoop or drop seeds during the sowing assignment. This coordinate denotes player's pit i and opponent's pit j . the value p_x is the number of seeds in x and $P_x \in S$ for which:

$$i = 0 (1) n / k - 2$$

and:

$$j = (n/k - 1) (1)(n - k - 1).$$

An equivalence relationship exists between the features p_x , i and j according to the following Eq. 2:

$$\left\langle P_x \binom{n}{k-2} \right\rangle = \left\{ \begin{array}{l} \{(i, j): (p_x(n-k-1)\theta) \sim (i, j)\}, x = i \\ \{(i, j): (p_x n - k + (n-k-1)\theta) \sim (i, j)\}, x = j \end{array} \right\} \tag{2}$$

By equivalence relation we mean a measure of likeness or similarity of objects. A relation \sim on a set S is called equivalence if has the following three essential properties:

- Reflexive; for each a in S , $a \sim a$
- Symmetric; if $a \sim b$, then $b \sim a$
- Transitive; if $a \sim b$ and $b \sim c$, then $a \sim c$

If \sim is an equivalence relation on set S , the equivalence class of each element a in S , denoted by $\langle a \rangle$ is the set of elements to which a is related and it is given as:

$$\langle a \rangle = \{x | a \sim x\}$$

Using the symmetric property of \sim and for a given coordinate (i, j) , it follows that from Eq (2) that:

$$P_x = \begin{cases} j - i + (n - k - 1)\theta, x = i \\ i - j + n - k + (n - k - 1)\theta, x = j \end{cases} \tag{3}$$

The value p_x is equivalent to the linear distance travelled during the sowing assignment.

Refinement-based heuristic strategy: The Refinement Based Heuristic (RBH) strategy utilized the minimax concept as it was considered to enhance minimax algorithm to evolve an Ayo game player. The idea of minimax algorithm is such that for every Two-Person

Zero-Sum (TPZS) game like Ayo, two players (Max and Min) choose a legal move turn-by-turn and each tends to maximize his advantage to the detriment of the opponent. Max player tries as much as possible to increase the minimum value of the game, while Min tends to decrease its maximum value at a node as both players play towards optimality. This process is achieved using the stockman equality by Bruin *et al.* (1994):

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

All reachable positions form a directed cyclic graph called state space where the nodes represent positions and the arcs represent moves. Specifically, a node n in Ayo game tree G with game value $f(G)$ is called feasible if $f(n) = f(G)$ and n is the immediate child of root node r . In additions, if n gives the player the best reward, it is called a best node. The number in the nodes is the scores and the scores of the leaves follow from the rule of Ayo game which could be negative, suggesting highest payoff for Max, zero for equal payoff or positive for highest payoff for Min. by payoff we mean the value that would be attributed to the action of making a move from a particular pit. Example of a best node represented by bold dashes line and circle for the game configuration in Fig. 2 is shown in Fig. 3 by using Eq. 4. The RBH algorithm has three main components:

- Build game tree
- Compute game value and
- Refine feasible moves

The “buildTree” procedure constructs a game tree in top-down fashion using breath-first traversal and nodes are evaluated as fan out is made to all nodes adjacent to their immediate parents. The “computeValue” procedure computes the game value bottom-up using Eq (4) above and “predictMove” uses Move Refinement Procedure (MRP) to predict the best move. A refinement is a mapping that accepts a set of moves and then evaluates each move and returns a move with the best advantage. The MRP used a 4-ply look-ahead that is represented in a myopic rule as: Given a game state, let move $[k] = \{m_1, m_2, \dots, m_k\}$ be a set of k feasible moves. We call m_k the head and m_1 the tail. A move is protected if it is not vulnerable to being forfeited when the opponent plays.

- If $k=1$ Then select the only available move and stop
- If tail/head is not protected for South/North player respectively, then select it, else select a move with the highest mobility strength

North					
N ₆	N ₅	N ₄	N ₃	N ₂	N ₁
0	0	1	0	1	1
0	4	2	1	3	1
S ₁	S ₂	S ₃	S ₄	S ₅	S ₆
South					

Fig. 2: A typical Ayo game configuration

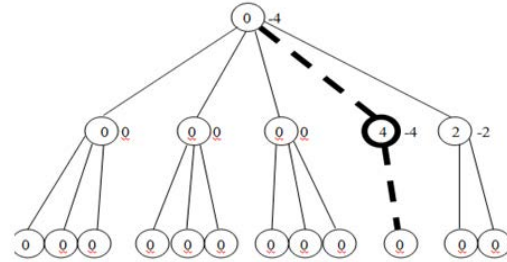


Fig. 3: Game tree for typical Ayo game configuration

In order to allow for mobility, bluffing and measure the correctness of move selection we simply regarded game strategies as episodes. The similarity between the i th target episode x_i and the source episodes y_j of the j th class is computed and the largest similarity measure is returned. The target episode with the maximum similarity greater than or equal to a given threshold value (called bluffing threshold) and a game value less or equal to the tradeoff seed is selected. Mobility is a process that allow for further game move no matter the pressure from the opponent while bluffing is the ability to tradeoff some seeds so as to gain an advantage of capturing more seeds in the nearest future. The similarity between two episodes x_i and y_j is calculated using the product-moment metric for linear correlation coefficient:

$$\text{Corr}(x_i, y_j) = 1 - \frac{\sum_{k=1}^m (x_k - \bar{x}_1)(y_k - \bar{y}_1)}{\sqrt{\sum_{k=1}^m (x_k - \bar{x}_1)^2 \sum_{k=1}^m (y_k - \bar{y}_1)^2}} \quad (5)$$

Other metrics used were the Canberra and angular metrics. But while we did this we observed that none of these metrics could be used in isolation. Nevertheless we preferred the correlation metric because it suggests best moves faster and allow for bluffing better than the other two metrics (i. e. angular and Canberra) as could be seen in our experimental results.

Functionalities provided by the prototype simulation: The prototype simulation provides the following functionalities for users to be able to play the game:

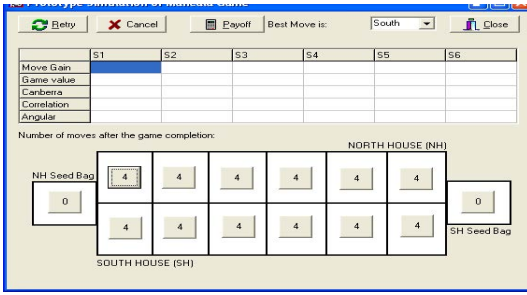


Fig. 4: A screenshot the Ayo game simulation

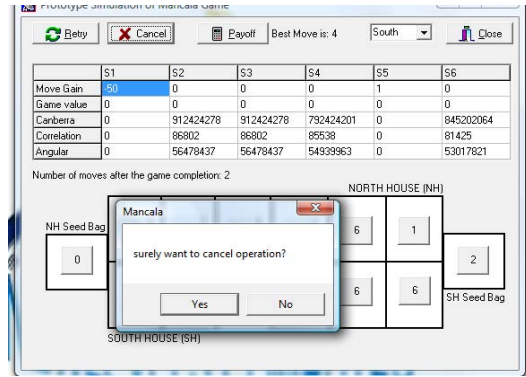


Fig 6: Simulation screenshot showing cancellation of operation

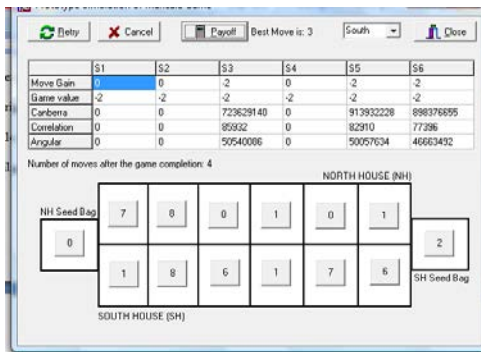


Fig. 5: A typical screenshot of Ayo simulation showing best move

- Payoff
- Cancel
- Retry

A sample screenshot of the prototype simulation for playing Ayo game is shown in fig 4 below. To play the game, the player would have to choose a pit (South or North) but by default, South is chosen. Any of the players (South or North) could start the game first. We have experimented with several cases whereby the prototype simulation started the game first while playing with an expert human player and as well played second. When playing the game as soon as it is the turn of the developed application to play, payoff menu (button) on the interface is clicked, this spawns the game evaluation section and the respective game values are computed. Suggestion is then made on the best pit to move from. This is indicated in front of the payoff button as shown in Fig 5 below. As seen in Fig. 4, the payoff has suggested that move should be made from pit three (3) that has six (6) seeds; this gives rise to the capture of 2 seeds. Apparently, there could be a situation when the player makes mistake in playing from a wrong pit as suggested when this happens, the cancel button is simply clicked

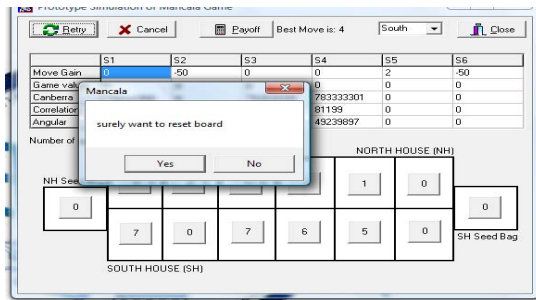


Fig. 7: Simulation screenshot showing board reset

and a dialogue box appears which requests if the operation is to be canceled. As soon as 'yes' is clicked on, the former action is reversed hence the player could play the right suggestion. A typical screenshot for cancellation of an operation is shown in Fig 6. In between the game evaluation part and the game play interface is a mechanism that registers the number of moves taken for a particular game play. As soon as the game is over and the player is still interested in another play, the retry button is clicked. A dialogue box appears that request if the person would want to reset the board (Fig 7). When the 'yes' button is selected, the initial game configuration of Ayo is displayed with four seeds on each pit.

RESULTS AND DISCUSSION

We implemented the RBH to evolve *Ayo* player using C++ on a PC with Microsoft Windows XP Professional operating system and Pentium (R) 4, 3.00GHz, 80GB hard disk with 1GB RAM. The performance of the RBH was evaluated by playing series of games with Awale shareware (simply referred to as Awale). The Awale is still at the moment the benchmark program generally accepted

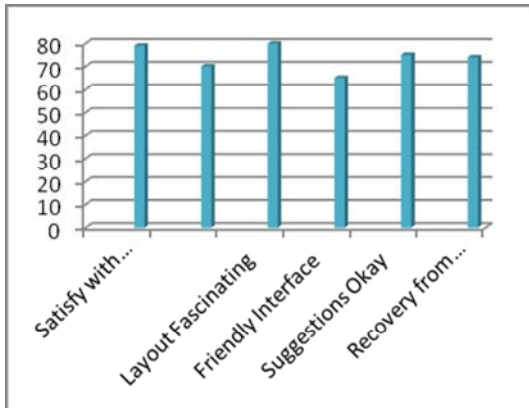


Fig. 8: Summary of user satisfaction with the prototype application

within academic community. In order to test the prototype application we registered to play with Awale at its various available levels, that is, Initiation, Beginner, Amateur and Ground master. Subsequently, some human expert players of Ayo game were contacted such that they could use the developed application (RBH) so as to adjudge a better user satisfaction among RBH and Awale. In order to achieve this arduous task, a paper-based questionnaire was designed and administered to the Ayo game players which were used for the assessments of players' perceptions of the prototype simulation. The players interacted with the prototype simulation by performing a regular game playing on Ayo board. For the purpose of objectivity, the players in the game tournaments were made to use the prototype simulation against the opponent to suggest the pit to move from thereby validating the correctness and fastness of move suggestions by the prototype simulation as opposed to Awale and that of human mental reasoning.

Data analysis: Statistical Package for Social Sciences (SPSS 15.0 for Windows) was used to analyse the feedback gotten from the administered questionnaire so as to generate the frequency distribution, mean score, standard deviation and variance for all the ratings for the prototype application (Fig. 8).

System evaluation: The null hypothesis is that the mean difference between RBH and Awale user satisfaction is zero. The alternative hypothesis is that there is a mean difference between the two applications:

- $H_0: \mu_d = 0$
- $H_1: \mu_d \neq 0$

Satisfaction ratings for RBH were significantly higher ($M=34.21$) than for Awale ($M=30.61$) as indicated by a significant t-test, $t(49)=-1.46$, $t_{crit}=2.0$, $p<0.05$. The null hypothesis is thus rejected and the finding indicates that RBH was more satisfying than Awale.

CONCLUSION

We have been able to simulate the playing of Ayo game using simple heuristic approach though enhanced with an endgame strategy which can make optimal or nearly optimal decisions like human. The algorithm employed to evolve our Ayo player is computationally effective and can improve AI performance and make computer players more adaptable and responsive. Similarly, it has the tendency to incorporate new play strategies in form of expert instruction and thus become sensitive to its mistakes/weaknesses.

REFERENCES

- Adebiyi, E.F., 1999. A step further toward improving ayo game system. *AMSE.*, 2: 53-69.
- Adewoye, T.O., 1990. On certain combinatorial number theoretic aspects of the African Game of Ayo. *AMSE. Rev.*, 14: 41-63.
- Akinyemi, I.O., E.F. Adebiyi and H.O. Longe, 2009. Critical analysis of decision making experience with a machine learning approach in playing ayo game. *World Acad. Sci. Eng. Technol.*, Vol. 56.
- Akinyemi, I.O., O.O. Olugbara, H.O.D. Longe and E.F. Adebiyi, 2008. On nearest neighbour refinement strategy for evolving ayo game player. *J. Comput. Sci. Appl.*, 15: 48-58.
- Allis, L.V., V.D.M. Meulen and V.D.H.J. Herik, 1994. Proof-number search. *Artif. Intell.*, 66: 91-124.
- Ayeni, J.O.A. and H.O.D. Longe, 1985. Game people play: Ayo. *Int. J. Game Theor.*, 14: 207-218.
- Beal, D.F., 1980. An Analysis of Minimax Search. In: *Advances in Computer Chess*. Clarke, M.R.B. (Ed.). Edinburgh University Press, Edinburgh, Scotland, pp: 103-109.
- Broline, D.M. and D.E. Loeb, 1995. The combinatorics of mancala-type games: Ayo. *Tchoukaillon UMAP. J.*, 1: 1-15.
- Bruin, D.A., W. Pijls and A. Plaat, 1994. Solution trees as a basis for game tree search. *ICCA. J.*, 17: 207-219.
- Daoud, M., N. Kharma, A. Haidar and J. Popoola, 2004. Ayo, the Awari player, or how better representation trumps deeper search. *Proceedings of the Congress on Evolutionary Computation CEC2004*, June 19-23, 2004, IEEE, Montreal, Quebec, Canada, ISBN: 0-7803-8515-2, pp: 1001-1006.

- Davis, J.E. and G. Kendall, 2002. An investigation, using co-evolution, to evolve an awari player. Proceedings of the 2002 Congress on Evolutionary Computation CEC'02, May 12-17, 2002, IEEE, Nottingham, England, ISBN:0-7803-7282-4, pp: 1408-1413.
- Lincke, T.R. and A. Marzetta, 2000. Large endgame databases with limited memory space. ICGA. J., 23: 131-138.
- Murray, H.J.R., 1952. A History of Board Games other than Chess. Clarendon Press, Oxford, England, Pages: 367.
- Nau, D.S., 1983. Pathology on game trees revisited and an alternative to minimaxing. Artif. Intell., 21: 221-244.
- Odeleye, A.O., 1977. Ayo: A Popular Yoruba Game. Oxford University Press, Ibadan, Nigeria.
- Romein, J.W. and H.E. Bal, 2002. Awari is solved. ICGA. J., 25: 162-165.