An Investigation, using Co-Evolution, to Evolve an Awari Player

James Edward Davis and Graham Kendall
Automated Scheduling, Optimisation and Planning Research Group
School of Computer Science and IT, Jubilee Campus, University of Nottingham, Nottingham, NG8 1BB, UK
gxk@cs.nott.ac.uk
http://www.cs.nott.ac.uk/~gxk

Abstract – Awari is a two-player game of perfect information, played using 12 "pits" and 48 seeds or stones. The aim is for one player to capture more than half the seeds. In this work we show how an awari player can be evolved using a co-evolutionary approach where computer players play against one another, with the strongest players surviving and being mutated using an evolutionary strategy (ES). The players are represented using a simple evaluation function, representing the current game state, with each term of the function having a weight which is evolved using the ES. The output of the evaluation function is used in a mini-max search. We play the best evolved player against one of the strongest shareware programs (Awale) and are able to defeat the program at three of its four levels of play.

1. Introduction

Game playing has a long history within AI research. Chess has received particular interest culminating in Deep Blue beating Kasparov in May 1997, albeit with specialised hardware [1] and brute force search, which managed to search up to 200 million positions per second. However, chess is still receiving research interest as scientists turn to learning techniques that allow a computer to 'learn' how to play chess, rather than being 'told' how it should play [2]. Learning techniques were being used for checkers (draughts) as far back as the 1950's with Samuel's seminal work ([3], re-produced in [4]). This would lead to Jonathan Schaeffer developing Chinook, which won the world checkers title in 1994 [5]. Like Deep Blue the question of whether Chinook used AI techniques or not is open to debate. Chinook had an opening and end game database and in certain games it was able to play the entire game from these two databases. If this could not be achieved, a form of mini-max search, with alpha-beta pruning was used. Despite Chinook becoming the world champion, the search has continued for a checkers player that is built using "true" AI techniques. Chellapilla and Fogel ([6],[7],[8]) developed Anaconda, so named, due to the strangle hold it places on its opponent. It is also called Blondie24 [8] which was a name given to the program late in its life in an experiment to see if the name affected the types of player it would attract when playing over the internet and if the other players would treat it differently to a program named something like 'David0203'. Anaconda (Blondie24) uses an artificial neural network (ANN), with 5046 weights, which are evolved via an evolutionary strategy. The inputs to the ANN are the current board state, presented in a variety of spatial forms. The output from the ANN is a value which is used in a mini-max search. During the training period the program is given no information other than a value which indicates how it performed in the last five games. It does not know which of those games it won or lost, nor does it know if it is better to achieve a higher or a lower score. Anaconda is certainly not given any strategy and contains no database of opening and ending game positions. The aim was to develop a game playing program that has no knowledge of the game, other than how to play legally, and to show that it can evolve its own strategies. Co-evolution is used to develop Anaconda, by playing games against itself. Anaconda has achieved expert levels of play (ratings of over 2000)

Both checkers and chess are games of perfect information, otherwise known as combinatorial games [9]. These games are classified as two-player games, with no hidden information, no chance moves, a restricted outcome (win, lose and draw) and with each player moving alternately. This is different to games such as poker [10], [11], [12], backgammon [13], [14], [15], [16], or bridge [17], [18], [19], [20] where there is hidden information, a chance element and, possibly, more than two players. A recent survey of computers and game playing [21], covers those games above, as well as others.

In this work we look at a combinatorial game called (amongst other things) Awari. We show how a co-evolutionary approach can produce a player that can play to a reasonably high level. There have been limited studies reported in literature that have looked at the game of Awari, although there are many Awari programs and competitions. The strongest shareware program is believed to be Awale by Myriad Software (http://www.myriad-online.com/awale.htm) and this is the program we use to test our developed player.

Bambam, from the research group at the University of Alberta led by Jonathan Schaeffer, was created in May 1999 to compete in the Awari tournament at the same university. It had a short term aim of becoming the strongest player on the planet and a long term aim of solving the game. The name, Bambam, was used due to its brute force play and the fact that it played with pebbles. One of the strongest players is Lithidion, produced in 1990 by Maarten van der Meulen and Victor Allis to compete in the Computer Olympiads. It won the gold medal in 1990, 1991, and 1992, at which time it was retired. Since that time there have not been any other serious computer tournaments so nothing has been able to challenge Lithidion. Thomas Lincke is working on solving the game and has produced an endgame database up to 37 seeds. His web site (http://www.shortestwin.com/awari/) allows you to interactively explore these positions.

2. Awari

The game of Awari is one of the oldest known strategy games. It is believed to have originated in Ethiopia about 3500 years ago and has since spread across Africa. The game is known by many different names, for example, Awale, Awele, Ajwa, Lela, Chisolo, Kalak, Oware, Coo, Coro Bawo, Nocholokoto, Dara, Congkak, Mancala, Bawo, Omweeso, Adita-ta, Kasonko, Layo, Gilberta, Schach, Wari and Walle; and this list is by no means complete. Awari is played with a hollowed out plank of wood and a number of stones or seeds.

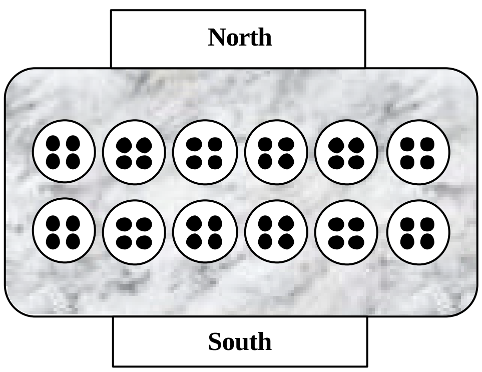

Figure 1 : Initial setup for Awari board

The plank has twelve hollows (pits), six designated as belonging to North and six belonging to South. At the start of the game, each pit contains four seeds (figure 1). As the game progresses each pit can contain any number of seeds, although the total number of seeds remains constant (48). The aim of the game is to capture the majority (>24) of the seeds. Once a seed is captured it is removed from the board and plays no further part, other than being used to evaluate the current game position. The game proceeds as follows.

South plays first and picks up all the stones from a non-empty pit on his/her side of the board. He/she deposits the stones, in an anti-clockwise direction, dropping one stone in each pit until all the stones have been deposited. If the number of seeds means that the player passes completely around the board, the starting pit is skipped. If the final stone is dropped in one of the opponents pits which has two or three stones in it (after the deposit), then those stones are captured. If the proceeding pit now has two or three stones in it, these stones are also captured. Capturing continues, in a clockwise direction, until either a pit does not have two or three stones in it, or the pit under consideration is one of your own pits. Figure 2 shows an example play, where both players can capture stones.

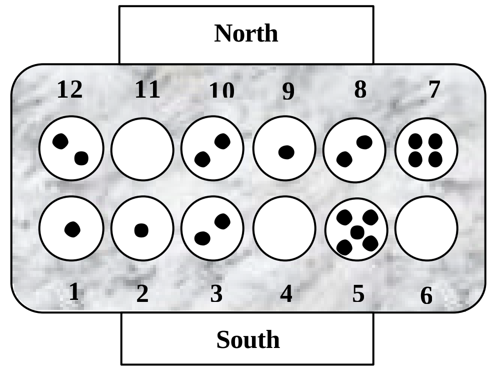

Figure 2 : Example play from Awari; south to play
If South plays pit five it will deposit seeds in pits 6, 7, 8, 9 and 10. It will capture the seeds in pits 10, 9 and 8 (total of 8 seeds captured). If North now plays pit 12 it will deposit seeds in pits 1 and 2 and will capture 4 seeds.

Figure 3 shows a typical position that players try to manoeuvre towards, as part of the strategy of the game. This is known as a Kroo, that is the accumulation of more than 12 seeds in one pit to allow for a complete revolution of the board. It also shows another strategy; the starvation of a players pits so that they are limited in the number of moves they can make. Figure 3 explains why a Kroo is such an important strategy.

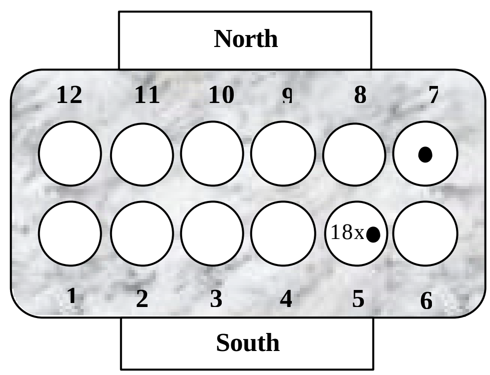

Figure 3 : Awari Strategy
When South plays pit 5 it will finish depositing in pit 12 (remember that the starting pit is skipped if play passes around the board). As play has passed around the board all the opponents pits will contain two stones, with the exception of pit 7 which will contain 3 stones. This results in all the seeds on the North side being captured (a total of 13 seeds).

Awari has as many rule variations as it has names. The rules we used in this study are given in appendix A.

3. Evolving an Awari Player

The aim of this study is to develop a player that, initially plays a poor (random) game of Awari but is able to evolve to play a better game as the player evolves using a co-evolutionary approach. The work of Fogel [8] used a neural network to determine a value that represented the current board position. This value was used in a mini-max search to decide which move the computer player should make. In effect the neural network represented a function that returned the current game state at a given time. In this work we are presenting the computer player with a simple evaluation function to ascertain if co-evolution is able to optimise the function to a level where the player can player Awari to a sufficiently high level.

The function we present to the evolving player is as follows

$$f = w_1 a_2 + w_2 a_3 + w_3 ß_2 + w_4 ß_3 + w_5 a_s + w_6 ß_s \qquad (1)$$

where :

$w_1..w_6$	The weightings for each term of f
a_2	The number of the opponents pits vulnerable to having 2 stones captured on the next move
a_3	The number of the opponents pits vulnerable to having 3 stones captured on the next move
$ß_2$	The number of the evolving players pits vulnerable to having 2 stones captured on the next move
$ß_3$	The number of the evolving players pits vulnerable to having 3 stones captured on the next move
a_s	The current score of the opponent
$ß_s$	The current score of the evolving player

This function was an initial attempt to devise a set of terms that seemed likely to capture the important elements of the game. Whether this function would allow an evolved player to improve its play over time was not known at the time the function was derived. Our aim was to see if, given a simple evaluation function, the player could evolve a good strategy. It is the weights for each term ($w_1..w_6$) that are evolved using a co-evolutionary strategy. The evolutionary process is conducted as follows.

- A population, P, of 20 players is created. Each member of the population, p_n (n=1..20), contains six real numbers which correspond to the weights, $w1..w6$. The weights are randomly initialised with values –1..+1.
- Each p_n plays every other p_n member twice, but they do not play themselves. They play once as north and once as south.
- For each move by the evolving player, a search tree is constructed. The depth of the search tree is determined by the available search time. In our experiments the search depth was 7. This was chosen after experimentation as a good trade off between the search needed by the player and the time taken to build the search tree. At this depth the search took about one minute. The evaluation function (1) assigns a value to each of the terminal nodes and these values are propagated up to the root of the search tree using the mini-max algorithm. The value at the root is used to decide which move to make.
- The winning player is awarded 3 points for a win, 1 point for a draw and zero points for a loss.
- If a game reaches move 250 the points are awarded on the state of the game at that point. There is an argument for simply awarding a draw but we decided against this and awarded a winning score for capturing more seeds.
- At the end of all the games the top m (for our experiments m=5) players were retained and the rest were discarded.
- Each retained player produces an equal number of children. If this would exceed the population size (as n=20 and m=5, this was not relevant to us) then the production is biased towards the fittest individuals. The production of a new player is produced by

$$p_n(w_i) = p_n(w_i) + N(0,1) \qquad (2)$$

  where the standard normal variable is sampled anew for each weight (see [22] for sample code to produce normal variables). This method of adaptation was used to mimic the successful approach by Fogel [8].
- We ran 250 generations in order to produce our evolved player.

4. Results

To compare our evolved player we used the game of Awale produced by Myriad Software. The shareware version of this package allows one level of play (initiation). If you register the software you are given access to three higher levels (beginner, amateur and grand master). Myriad were kind enough to supply us with the registered version for the purposes of this research, so that we could test our evolved player against all levels. Initially we tested a random set of weights against Awale at the lowest level (table 1). This player, not surprisingly was easily beaten by Awale. The game finished after 68 moves when Awale had captured 26 seeds. We also played the same random player at the grand master level (table 2) and the player was easily defeated after 50 moves when Awale had captured 29 seeds.

TABLE 1
PLAYING AWALE AT INITIATION LEVEL WITH RANDOM WEIGHTS

Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
68	0	26

TABLE 2
PLAYING AWALE AT GRAND MASTER LEVEL WITH RANDOM WIEGHTS

Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
50	0	29

The best evolved player from the algorithm presented above was played at each level of Awale over a series of five games. The first set of games is presented in table 3, when the evolved player competed against Awale as its lowest playing level (initiation). The results show that Awale was easily defeated by the evolved player. On average, the evolved player captured the majority of seeds in just over 47 moves. The numbers of seeds captured by Awale averaged just under 3.

TABLE 3
PLAYING AWALE AT INITIATION LEVEL

Game #	Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
1	29	30	3
2	39	32	0
3	39	32	7
4	67	27	2
5	63	28	2
	47.40	**29.80**	**2.80**

Table 4 shows the results when playing against the beginner level of Awale. Again, the evolved player was a relatively easily winner although the average number of moves has increased as has the average number of seeds captured by Awale. The fourth game of this series is interesting as it appears as if the evolved player has not captured the majority of the seeds. In fact, it won due to the fact that it captured all the stones due to a play it made (see rule 7, in appendix A). The sequence final sequence that led to this position is described in Appendix B.

TABLE 4
PLAYING AWALE AT BEGINNER LEVEL

Game #	Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
1	43	30	8
2	75	28	6
3	43	25	4
4	33	19	9
5	85	29	12
	55.80	**26.20**	**7.80**

Table 5 shows the five game series against Awale's amateur level. Things are getting a little tougher now. The average number of moves has risen dramatically and the evolved player suffered its first defeat (game 4). There was also a draw in this series (game 3) where the game was stalemated in that the same position kept repeating itself. In fairness to Awale, we did note that it could have made a winning play on a number of occasions but it did not exploit it. Not surprisingly, the average number of seeds captured by Awale has also risen dramatically in this five game series. One of the games from this series can be seen in appendices C. In particular, it shows that, although the evolved player won it does not does value winning as highly as capturing.

TABLE 5
PLAYING AWALE AT AMATEUR LEVEL

Game #	Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
1	59	35	6
2	103	32	9
3	158	15(d)	24(d)
4	116	15	26
5	105	24	19
	108.20	**24.20**	**16.80**

Table 6 shows the results from playing against Awale at its highest level (grand master) and unfortunately it shows that our evolved player is no match for Awale when playing at this level.

When playing at this level Awale asks you to set a maximum time allowed for each move and also an analysis depth. We set the move time 60 seconds (to match with our search time) and the depth at the lowest possible value, 12. However, we have no information as to what the analysis depth relates to. We cannot assume, for example, it is the search depth of a mini-max search. In addition, we have no information about the evaluation function used by Awale but, we suspect, it is more sophisticated that the one evolved by our player.

TABLE 6
PLAYING AWALE AT GRAND MASTER LEVEL

Game #	Moves in Game	Seeds captured by Evolved Player	Seeds Captured by Awale
1	86	4	25
2	76	4	28
3	86	4	25
4	76	5	28
5	76	5	28
	80.00	**4.40**	**26.80**

5. Discussion

The purpose of this study was to see if, presented with a simple, certainly not optimised, evaluation function a game could evolve that improved its play from an initial, random state. We have shown that this is possible and that the evolved player achieves reasonable level of play (certainly able to beat the authors). Future work will look at developing a player that is able to find its own objective function and its own strategies. The work of Fogel [8] already provides inspiration for this but there is also scope for using other machine learning techniques (such as genetic programming) to evolve a suitable evaluation function.

We would like to think that ultimately we could challenge the best human players.

6. Acknowledgements

The authors would like to express their thanks to Myriad Software, who supplied us with a licensed version of their Awale program and also to the reviewers for their helpful comments.

Appendix A : Rules of Awari

These are the rules that were followed for this study. There are many variations, many of which can be defined within the Awari program that we used (Awale).

1. The board comprises 12 pits (labeled 1..12), each with four seeds. Six of the pits belong to south (1..6). The other six belong to north (7..12).
2. The game starts with the players selecting who is North and who is South. South moves first.
3. On your turn, select a non-empty pit on your side of the board. "Sow" the seeds from that pit around the board, dropping one at a time, counter-clockwise into each pit.
4. If you choose a pit with enough seeds to go completely around the board (12 or more), the original pit is skipped and left empty.
5. If the last seed is dropped into a pit on your opponent's side, leaving that pit with 2 or 3 seeds, you capture all the seeds in that pit. The capture continues with consecutive ***previous*** pits on that side which also contain 2 or 3 seeds.
6. If all your opponent's pits are empty, you must make a move that will give him a move. If no such move can be made, you capture all the remaining seeds on the board, ending the game. If no move is possible, the winner is the person with the greater number of captured seeds.
7. If, by making a play, you can capture all the stones on your opponents side of the board, you win the game (as your opponent cannot make a play). Note, this overrides rule 6.
8. At the end of the game the seeds left on the board are not captured by any player.
9. The game is over when one player has captured 25 or more seeds, or both players have taken 24 seeds each (a draw).

Appendix B : End Play Against Beginner Level of Awale (Game 4)

When the evolved player played Awale at the beginner level the game ended when the evolved player captured all the stones on the opponents side (see rule 7 in appendix A). This is the final sequence that led to that position. Awale is playing north and is next to play. The scores at this time are north=9, south =10. North must play the pit with 15 seeds.

North					
15	**0**	**0**	**0**	**0**	**0**
0	**0**	**10**	**0**	**5**	**2**
South					

This leads to this position

North					
0	**1**	**1**	**1**	**1**	**1**
2	**2**	**12**	**2**	**6**	**3**
South					

When south plays it decides to play the pit with 6 seeds.

This leads to the position shown below, where south captures 10 stones but, more importantly, does not give north a move and thus wins the game.

North					
0	**0**	**0**	**0**	**0**	**0**
2	**2**	**12**	**2**	**0**	**4**
South					

Although the play above may seems obvious, the real credit goes to the evolved player in that it created this position in order to exploit it.

Appendix C : End Play Against Amateur Level of Awale (Game 2)

This is the final sequence of game 2 when the evolved player played Awale at the amateur level. Awale is playing north. South (the evolved player) is next to play. The scores at this time are north=9, south=19 and the current position is as follows.

North					
0	**1**	**0**	**0**	**0**	**0**
0	**0**	**0**	**2**	**17**	**0**
South					

South plays the pit with 2 seeds forcing north to play its pit with a single seed, leading to this position.

North					
1	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**18**	**1**
South					

The evolved player decides to play the pit with 18 seeds, leading to this position (before the capture).

North					
3(*)	**2**	**2**	**2**	**2**	**2**
1	**1**	**1**	**1**	**0**	**3**
South					

The capture starts at the pit marked with an asterix, leading to a capture of 13 seeds, giving south a total of 32 seeds, and the game.

Notice that from the initial position shown in this play south could have won immediately by playing the pit with 17 seeds. However, this would have only captured 11 seeds. This suggests that the evolved player gives more importance to capture than to winning the game. Whilst some human players may play like this (to inflict even more humiliation on their opponent) it would probably normally be better to secure the win as soon as possible.

References

[1] S. Hamilton, L. Garber, “Deep Blue's hardware-software synergy”. IEEE Computer, 30(10), pp 29-35, October 1997

[2] G. Kendall, G. Whitwell "An Evolutionary Approach for the Tuning of a Chess Evaluation Function using Population Dynamics". In proceedings of CEC2001 (Congress on Evolutionary Computation 2001), COEX Center, Seoul, Korea, May 27-29, pp 995-1002, 2001

[3] A. L. Samuel “Some Studies in Machine Learning using the Game of Checkers”. IBM Journal of Research and Development, 3(3), pp 210-229, 1959

[4] A. L. Samuel “Some Studies in Machine Learning using the Game of Checkers”. IBM Journal of Research and Development, Vol. 44 No. 1/2 January/March, pp 207-226, 2000

[5] J. Schaeffer, R. Lake, P. Lu “CHINOOK The World Man-Machine Checkers Champion”, AI Magazine, 17(1), pp 21-30, 1996

[6] K. Chellipilla, D. B. Fogel, “Evolution, Neural Networks, Games, and Intelligence.” Proc. IEEE, Vol 87, No. 9, pp 1471-1498, 1999

[7] K. Chellipilla, D. B. Fogel “Anaconda Defeats Hoyle 6-0: A Case Study Competing an Evolved Checkers Program against Commercially Available Software.” In Proceedings of Congress on Evolutionary Computation, July 16-19 2000, La Jolla Marriot Hotel, La Jolla, California, USA, pp 857-863, 2000

[8] D. B. Fogel, “Blondie24 : Playing at the Edge of AI”. Morgan Kaufmann, 2001, ISBN:1558607838

[9] A. S. Fraenkel, “Combinatorial Games : Selected Bibliography with a Succinct Gourmet Introduction”, Games of No Chance, Vol. 29, MSRI Publications, Cambridge University Press (ed. R. Nowakowski), pp 493-537, 1996

[10] G. Kendall, M. Willdig, "An Investigation of an Adaptive Poker player". Accepted for The 14th Australian Joint Conference on Artificial Intelligence (AI'01), Adelaide, Australia, 10 - 14 December 2001

[11] L. Barone, L. While, “Adaptive Learning for Poker”. In proceedings of the Genetic and Evolutionary Computation Conference, pp 566-573, 2000

[12] J. Schaeffer, D. Billings L. Peña, D. Szafron, “Learning to Play Strong Poker.” In proceedings of the Sixteenth International Conference on Machine Learning (ICML-99), J. Stefan Institute, Slovenia (Invited Paper), 1999

[13] P. J. Darwen, “Why Co-Evolution beats Temporal Difference Learning at Backgammon for a Linear Architecture, but not a Non-Linear Architecture.” In proceedings of CEC2001 (Congress on Evolutionary Computation 2001), COEX Center, Seoul, Korea, May 27-29, pp 1003-1010, 2001

[14] B. Pollack, A. D. Blair, “Co-evolution in the Successful Learning of Backgammon Strategy.” Machine Learning, 32(3), pp 225-240, 1998

[15] G. Tesauro, “Temporal Difference Learning and TD-Gammon.” Communications of the ACM, 38(3), pp 58-68, 1995

[16] G. Tesauro, “Comments on Co-Evolution in the Successful Learning of Backgammon Strategy.” Machine Learning, 32(3), pp 241-243, 1998

[17] L. Frank, “Search and Planning under Incomplete Information: A Study Using Bridge Card Play.” Springer Verlag, 1998

[18] M. Ginsberg, “GIB: Steps Toward an Expert-level Bridge-Playing Program.” In proceedings of International Joint Conference on Artificial Intelligence, pp 584-589, 1999

[19] S. Smith, D. Nau, T. Throop, “Computer Bridge: A Big Win for AI Planning.” AI Magazine, 19(2), pp 93-105, 1998

[20] S. Smith, D. Nau, T. Throop, “Success in Spades: Using AI Planning Techniques to Win the World Championship of Computer Bridge.” In proceedings of AAAI National Conference, pp 1079-1086, 1998

[21] J. Schaeffer, “The Games Computers (and People) Play.” Academic Press, Vol. 50 (ed Zelkowitz M.V.), pp 189-266, 2000

[22] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1996). Numerical Recipes in C, 2nd ed., Cambridge University Press, ISBN 0-521-43108-5