Selection of Suitable Evaluation Function Based on Win/Draw Parameter in Othello

Basit Shahzad
College of Computer & Information Science, King Saud University, Riyadh
basit.shahzad@gmail.com

Lolowah R. Alssum
College of Computer & Information Science, King Saud University, Riyadh
l.alssum@yahoo.com

Yousef Al-Ohali
College of Computer & Information Science, King Saud University, Riyadh
yousef@ksu.edu.sa

***Abstract*---Computer games have made their presence vocal by making themselves present in the homes and industry. Games have emerged to provide a simulated experience of the outdoor games with ease and customization. Another class of games come into play when the indoor games are played without any physical opponent. In such case computer itself takes the responsibility of being an opponent and tests the human intelligence. Board games are especially very popular to be played on computer with computer as an opponent. This paper discusses on of the board games: Othello. The game of Othello has proved its prominence by being an active research area since long time now and has been successful to grab extensive focus of researchers, knowledge engineers and game developers. Othello is not as simple as Checkers and not as complex as Chess: both in its execution time and complexity, therefore it is an appropriate choice to be considered as a benchmark in the games development. Finding a better evaluation function to implement Othello has been an open question of research since long. In this paper we have compared different available strategies at length. Extensive experimentation (approaching to 144,000 experiments collectively) has been done to measure the effectiveness of each evaluation function. After thorough experimentation it is proved that Multi Layer Perceptron Neural Network (MLPNN) is the best strategy among available with respect to its win/draw comparisons. As winning a game in slightly more time is considered to be effective instead of losing it quickly.**

I. INTRODUCTION

The game Othello can be perceived to have three phases: the beginning phase of game, the middle phase of game and the ending phase of game [1]. Although there is no standard definition of where the beginning, middle and the end game begins yet the first 10, 15 or at most 20 moves can be considered as beginning of the game. Brian Rose thinks that, the beginning game is over when one of the edge squares is taken [2]. The beginning and end game can be handled easily, the first by using an open book, which is the moves that are prepared and memorized before the game begins (collection of openings), while the second by maximizing the player's pieces and minimizing the opponent's pieces. The middle part of the game is more complex as compared to the beginning phase and ending phase of the game. There are two main strategies that can be used in the middle game. First is the positional strategy, in this strategy the position of the pieces in the board is very important, for example the corners and the edges of the board are more valuable than other board positions, especially the corners because once taken, the opponent can never recapture [1]. In this strategy the player can uses two evaluation functions; the first is used during the start and the middle of the game to maximize the player's valuable position while minimizing opponent's valuable position.

The second strategy is the mobility strategy, the goal of this strategy is to maximize the number of moves that can be chosen from and minimizes the opponent's moves, therefore this strategy can force an opponent to make a bad move by limiting its chance to make a move and the opponent is somehow forced to make bad move as the choice for adapting a move is limited. This strategy is the same as positional strategy and uses two evaluation function but the different is in the first evaluation function. The first evaluation function is used during the begin and the middle of the game to maximize the player's mobility and the number of corner squares that are

978-0-7695-4654-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ITNG.2012.175

occupied by player's pieces while minimizing the opponent's mobility and the number of corner squares that are occupied by opponent's pieces.

II. IMPLEMENTATION STRATEGIES

Artificial Neural Networks are powerful computational systems consisting of many simple processing units "neurons, nodes" connected together (by weighted connections) according to a specific network architecture to perform a specific task. Each unit receives inputs, processes inputs (by applying Activation Function) to form an output. ANNs can learn and generalize from training data by adjusting the weights of the connections between units according to some learning rule [3].

Weighted Piece Counter (WPC) is a simple representation of strategy (linear evaluation function) which gives each board position a weight. The linear evaluation function is computed by summing the product of the inputs value and their weights.

Multilayer Perception Networks (MLP) is a feed-forward neural network, which means that the data flow in one direction from input layer to output layer. It could contain more than one hidden layer between input and output layer. Each layer in the network contains a set of units which receive the inputs from units in the previous layer and send the outputs to units in the next layer. There are no connections between units in the same layer. Also, each layer is fully connected to the next layer [4].

The Temporal Difference Learning (TDL) is a learning method that is used to solve the prediction problems by estimating the next action of the system using the past experiences. Learning happens when there is a change in the system's state and it is based on the error between temporally successive predictions. The goal of learning is to make the previous prediction more closer to the current prediction [5].

Monte Carlo Methods is a learning method that learns directly from online or simulated experience. The basic idea is to run the program (game) from each possible move in the current board position randomly to the end of the game a fixed number of times. Then the average result is returned for each possible move and the move that has the higher value is selected [5] [6].

Tournament Play Technique was used in Monte Carlo algorithm to improve its accuracy. Each candidate move will be played number of times before analyzing its performance. The move with the lowest average result will be eliminated. The process is repeated until two candidate moves remain and the best of them will be choosing [7].

III. EXPERIMENTATION

Experiments and observations in comparative evaluations of functions can be formulated in terms of scientific investigations. In this section, we present a comparative investigation of selected evaluation functions. The objective is to compare different evaluation functions and to discover the strengths and weaknesses of each function. The comparisons are made after the simulation for the game engine was developed. The evaluation functions are evaluated on the basis of the time per game and time per move, the complexity and the wins and draws by each player in each evaluation function at different levels to conduct this experimentation.

The evaluation environment was tested on a system that uses Windows Vista Service Pack 2 operating system. The computer specification included the: Processor: Intel Core 2 Due CPU P8400 speeds of 2.26 GHz, Memory (RAM): 4 GB, Hard Disk: 320 GB and System type: 32- bit Operating System.

A. Experimental Design

In this evaluation, we used the criteria of the number of wins/draws/losses and time consumption to examine the six evaluation functions at four different levels of search (depth of game tree).

Criteria of space consumption has not been used in these experiments as it will yield variable results depending on the platform used for experimentation and there would be no significant difference in the results because of the type of search algorithm used in this research. In order to get more realistic results 144,000 experiments were run. The extensive experimentation helped ensuring that the results are realistic and uninfluenced as the game is random in nature. When there is more than one legal move that has the same evaluation value, the choice of the best move would be random.

At the end of each game we have identified the following information about the winning, as winning a game is the ultimate objective of playing it.

IV. RESULTS

This section presents the results of experiments that we have conducted on the evaluation environment and explain the variables used in these results. Two experiments are run: first experiment manages and governs the number of win/loss and draws while second experiment covers the time take to complete the moves and game ultimately to measure the efficiency of the evaluation function.

The experiments have been performed by considering the player 'Black', and then 'White'. 144,000 experiments have been rum collectively, which includes 3000 experiments for each evaluation function on each level. Out of the 144,000 experiments performed collectively, half of this are for the player 'Black' and rest half are for player 'White'. Table 1 describes the results for 72,000 experiments run for the player Black and 'White'.

Table 1 describe the results of 144,000 experiments to determine the win, lose and draws for player 'Black' and player 'White'. In order to quantify the results we use a statistical yardstick to measure the relative performance of each evaluation function at each level. We assumed it realistic to assign a value 1 to each win and 0.5 to each draw to measure the strength of each evaluation function.

Table 2 gives the cumulative number of points (1 point for each win and 0.5 points for each draw) obtained by each evaluation function at all levels from Table 1.

Based on the facts detailed in table 3 it can be identified that MLPNN performs better as compared to other algorithms as for as the winning of games is concerned. Lolo Engine is the nearest competitor in this regard. The evaluation functions are also to be evaluated based on the efficiency they observe. For this measure to be considered, 144,000 experiments were conducted. The results of these experiments are given in table 1:

V. CONCLUSION

The paper has discussed the parameters for measuring the effectiveness of the evaluation function used to implement the Othello environment. Six specific evaluation functions were considered for this purpose and were evaluated for the possible number of 'win/draw/lose' and the time efficiency. MLPNN has proved itself to be very effective as its winning ratio is better as compared to the other evaluation functions under consideration. It can be concluded that MLPNN should be obvious choice if the decision is to be made based on the 'win/lose'. It can also be observed that although MLPNN takes slightly higher time to complete the game yet its results as compared to 'Ensemble' are remarkable while 'Ensemble' is only slightly better in efficiency as compared to MPLNN.

REFERENCES

[1] D. Moriarty and R. Miikkulainen, "Evolving Complex Othello Strategies Using Marker-Based Genetic Encoding Of Neural Networks", 1993.

[2] B. Rose Othello: A Minute to Learn... A Lifetime to Master, http://othellogateway.com/rose/book.pdf.

[3] K. Gurney, *An Introduction to Neural Networks*, Routledge, 1997.

[4] D. Michie, D. Spiegelhalter and C. Taylor, *Machine Learning, Neural and Statistical Classification*, 1994.

[5] R.Sutton and A.Barto, *Introduction to Reinforcement Learning*, MIT Press, 1998.

[6] P. Hingston and M. Masek, "Experiments with Monte Carlo Othello", *IEEE Congress on Evolutionary Computation*, 2007, pp. 4059-4064.

[7] J. Nijssen, "Playing Othello Using Monte Carlo", 2007.

TABLE 1: EVALUATION FUNCTION-WIN/DRAW STATISTICS

Level #	Function Name	Total Played	# of Wins (B)	# of loses (B)	# of Draws (B)	# of Wins (W)	# of loses (W)	# of Draws (W)
Level 1	Standard WPC	3000	1425	1500	75	1662	1329	9
	CTDL	3000	1678	1315	7	1740	1190	70
	Ensemble	3000	655	2314	31	1136	1829	35
	Mobility	3000	744	2203	53	1292	1695	13
	MLPNN	3000	2439	561	0	2129	822	49
	Lolo Engine	3000	745	2207	48	2141	821	38
Level 2	Standard WPC	3000	1169	1746	85	1196	1732	72
	CTDL	3000	1385	1528	87	1544	1350	106
	Ensemble	3000	495	2478	27	968	1999	33
	Mobility	3000	1461	1494	45	1600	1342	58
	MLPNN	3000	1514	1465	21	2437	533	30
	Lolo Engine	3000	1421	1510	69	2476	489	35
Level 3	Standard WPC	3000	1058	1886	56	1119	1817	64
	CTDL	3000	1559	1339	102	1388	1563	49
	Ensemble	3000	890	2087	23	422	2519	59
	Mobility	3000	1504	1439	57	1482	1447	71
	MLPNN	3000	2625	313	62	2038	892	70
	Lolo Engine	3000	1510	1419	71	2034	908	58
Level 4	Standard WPC	3000	1122	1800	78	1280	1634	86
	CTDL	3000	1382	1545	73	1584	1331	85
	Ensemble	3000	1297	1659	44	294	2659	47
	Mobility	3000	1464	1462	74	1450	1485	65
	MLPNN	3000	1673	1307	20	2347	624	29
	Lolo Engine	3000	1405	1535	60	2353	610	37

TABLE 2: EVALUATION FUNCTION-TOTAL POINTS FOR WHITE AND BLACK PLAYER

Level #	Function Name	Black	White	Total Points
Level 1	Standard WPC	1462.5	1666.5	1564.5
	CTDL	1681.5	1744.5	1713
	Ensemble	670.5	1140.5	905.5
	Mobility	770.5	1296.5	1033.5
	MLPNN	2439	2133.5	2286.25
	Lolo Engine	769	2145.5	1457.25
Level 2	Standard WPC	1211.5	1200.5	1206
	CTDL	1428.5	1548.5	1488.5
	Ensemble	508.5	972.5	740.5
	Mobility	1483.5	1604.5	1544
	MLPNN	1524.5	2441.5	1983
	Lolo Engine	1455.5	2480.5	1968
Level 3	Standard WPC	1086	1123.5	1104.75
	CTDL	1610	1392.5	1501.25
	Ensemble	901.5	426.5	664
	Mobility	1532.5	1486.5	1509.5
	MLPNN	2656	2042.5	2349.25
	Lolo Engine	1545.5	2038.5	1792
Level 4	Standard WPC	1161	1284.5	1222.75
	CTDL	1418.5	1588.5	1503.5
	Ensemble	1319	298.5	808.75
	Mobility	1501	1454.5	1477.75
	MLPNN	1683	2351.5	2017.25
	Lolo Engine	1435	2357.5	1896.25

FIG 1: CUMMULATIVE POINTS FOR EVALUATION FUNCTIONS

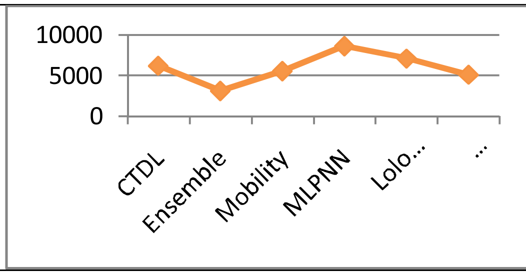

TABLE 3: CUMMULATIVE POINTS FOR EVALUATION FUNCTIONS

Function name	Points
Standard WPC	5098
CTDL	6206.25
Ensemble	3118.75
Mobility	5564.75
MLPNN	8635.75
Lolo Engine	7113.5