Fitness Value Optimization for Disc Set in Board Game through Evolutionary Learning

Dharm Singh
College of Technology and Engineering, MPUAT, Udaipur, India

Chirag S Thaker
Faculty of Engineering
Suresh Gyan Vihar University
Jaipur, India

Sanjay M Shah
Faculty of Engineering
Suresh Gyan Vihar University
Jaipur, India

ABSTRACT

The Artificial Intelligence research field since ages has incorporated a series of novel and trend setting distinct approaches including neural networks, fuzzy logic and genetic algorithms to apply them to various problem-solving domains. Machine learning techniques such as evolutionary learning, neural networks and reinforcement learning alone are difficult to apply to board games because they need an extremely large number of computations which are having tendency to increase exponentially in numbers as the search depth increases to find better move(s). Many board game researchers find that machine learning approach through evolutionary learning using some optimization methods like genetic algorithm gives better results to build robust and better artificially intelligent game playing programs.

In case of board game, board squares plays vital role in terms of exploring the game based topographies to assign relative weight to board squares as per their positions. These weight assignments in game-playing programs are derived through quality search and rules acquaintance and game playing experience. When the move search reaches the end of a game tree structure, attained optimized evaluation function values are used to assess board position "goodness".

The paper takes Game of Reversi as its object game and exploits its symmetric phenomenon to develop genetically evolutionary game playing program to learn its impact on the evolution of weight values for a particular disc sets through weight value land scape. The collected results for the said disc sets endorse the earnest efficacy of genetic algorithm as an evolutionary optimization instrument.

The first two sections is about game introduction and game search space. The next section discusses history of game program development and game playing phases. Section five and six aims game of Reversi implementation and collected results respectively. The last two sections are about conclusion and references.

General Terms

Fitness value optimization, checkers,reversi et. al.

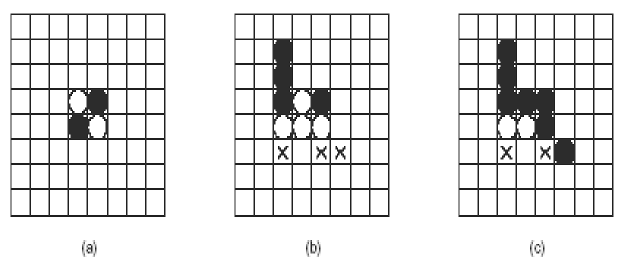
Keywords

Artificial Intelligence, Board Game, Genetic Algorithm, Game of Reversi, Board Square Weight.

1. INTRODUCTION

Game of Reversi is one of the prominent and traditionally very popular in Japan played since centuries. It was brought to European countries and America in the mid 1970's and soon attained international popularity for its simplicity and fascinating in the name of Game of Othello. A line for the game talks about it very perfectly as it says "A Minute to Learn A Lifetime to Master". So beginners and experts enjoy it initially for its simple playing rules, but complex strategies must be mastered to play the game well [1].

Game of Reversi is a two-player game played on an 8 X 8 board. It is a two-player, deterministic, zero sum, alternative move and perfect information game. While studying the game board characteristics it is found that the game is having symmetry phenomenon which helps in dividing the 8X8 board (which are sixty four disc blocks) into ten disc sets. The goodness is that all board-discs (or pieces) are identical with one white side and one black side. The initial board setup is very well presented in Figure 1(a). Each player takes turns placing discs on the board with his color face up.



(a) The initial setup. (b) After four moves (the legal moves for black are marked with X's). (c) After black has moved to the rightmost X.

Fig. 1: The Game of Reversi Board:

A player is only allowed to move in an open space that causes an opponent's disc or discs to be ranked by the new disc and another one of the player's own discs. Discs may be captured vertically, horizontally, or diagonally. Figure 1(b) shows the legal moves for black for the given board pattern. Once a move is made, the captured discs are flipped. Figure 1(c) shows the board layout resulting from a move by black in the sixth row of the sixth column. The game is continued until there are no legal moves available for either player. If a player has no legal move, he must pass. The winner is the player with the most on board discs in the final board configuration [2][3].

2. REVERSI SEARCH SPACE

In the Game of Reversi, a well versed study shows there are 364（about 3.4*1030）possible states, even if the symmetric and erase part of the illegal states (which are not possible to appear) are considered, there are still a huge number of states which require an extremely high space complexity to store them. Therefore, before the winning strategy has been proved, the Computer Reversi (Othello) still have room to be improved.

In Artificial Intelligence research domain, there are still many games waiting to be explored from an evolutionary perspective [4]. This research domain, despite its many decade long history, has a very renewed and pioneering feel to it at the moment among researchers across worldwide [5].

As per the Reversi playing rules and expert domain knowledge it is found that the game allows us to turn over the adversary's pieces only after a legal move. For each of the turn (hand), there are around 1 (including pass) to 15 (seldom more than 15) conceivable legal moves. The average of move(s) is set to eight legal moves. Comparing Game of Reversi's possible moves to other such board game , it is found that for each turn the average / maximum possible moves are still much less than other board games (compare to Game of Go game which has more than 200 possible moves for each hand or Chinese Chess which has more than 50 possible moves for each hand). Even if the evaluation function design is not precise, due to availability of high speed computing competency, a fair "better" move can be reached in computer Reversi playing programs after examining several layers of min-max search and alpha-beta pruning[6][7].

3. PREVIOUS WORK

In initial years of Reversi (Othello) playing program development, one of the first master-level Game of Reversi (Othello) programs was created and named called IAGO. [8] The program was based on alpha-beta search techniques with kill tables. Second program was re-developed as a successor to Iago named BILL. [9] It was based on similar search techniques, but instilled Bayesian learning to optimize its evaluation function. While Bill is majorly focusing on usage of more game position knowledge to evaluate board positions, its main strength is still the alpha beta search.

Then in a significant advancement in this field took place in October 1993 when LOGISTELLO made its tournament debut. It has been one of the top Reversi (Othello) programs ever since. Its main emphasis is on deep searches and reasonably good evaluation functions, which are the latest and required program improvements. For the first time, a new table estimation technique is presented which significantly improved the evaluation function. Quality is not compromised at any additional run time cost. It means the selective search procedure PRoBCuT is generalized to enable the program to cut off even more variations in advance that probably have no potential impact on the move decision process [10].

LOGISTELLO's evaluation features fall into two classes, namely mobility measures and board square capture patterns. It focuses points like motivated to have stable discs (once captured they can't be flipped back – for Game of Reversi corner discs and horizontal and vertical last edge nearer to corner falls in this category), maximizing the number of potential moves and parity. IAGO and BILL programs introduced novel approach which is table based evaluation scheme, in which values of all edge configurations were pre computed by (probabilistic) minimal algorithms and stored in a table for a quick evaluation of the edge structure.

The second feature subset dealt with mobility and potential mobility. In these game program versions the humblest approach is to count legal or potential moves which unfortunately are relatively time consuming compared to the time needed for all other features and making/undoing moves during the game tree search. In order to speed up the computation, the globally defined mobility measures were approximated by the sum of mobility local to the lines of the board, i.e. the horizontals, verticals, and diagonals. [11]

4. REVERSI GAME PLAYING PHASES

Over a period of very long experiences it is an acquired "knowledge" that for an intelligent "player", A game of Reversi (Othello) can be broken down into three main phases: the beginning game, the middle game, and the end game. An initial book can effectively handle the beginning game. The end game is simply played with an modest aim to maximize one's pieces while minimizing opponent's discs numbers. And a good strategy for the middle game is much more elusive. The goal of the middle game is to strategically position one's pieces on the board such that they give the much needed room in terms of potential mobility so that latter stage moves can be converted into a large number of permanent discs placements during the end game [12]. The mid game strategies are further divided into two basic segments in the Game of Reversi.

A positional strategy stresses the importance of specific positions and piece configurations on the board. Places such as corners and edges are considered valuable, while other interior "regions" need to be avoided for letter phase of the games. Corners are specifically valuable because once taken, they can never be recaptured. Normally in a board game, a player using a positional strategy tries to maximize his valuable pieces while minimizing his opponent's. Positional strategies with their motives can be easily understood and implemented; beginners studying the game often develop them independently [13].

Mobility is another powerful mid game strategies exist with an powerful set of fitness values. Very good and sophisticated positional strategy can give its advantages if and only if the game does have an essentially strong understanding of “mobility” notions. In the game of Reversi, corner capturing is still considered an important mid-term goal, while seizing edges and other specific advantageous disc pattern formations is still not on the book of good game move strategies. But here Mobility strategies are constructed around the impression that the easiest way to capture a corner is to force one’s opponent to make moves that surrender that corner.

Mobility strategies often comprise short-term basic goals like keeping an initial low piece count and clustering pieces up to the final moves of the middle game. Mobility is one of the core and concrete ideas that form the foundation of all modern tournament play. Mobility has been shown to be much stiffer to envisage, learn and implement than a positional strategy. To independently discover a mobility strategy through evolutionary set of algorithms would therefore be a substantial demonstration of better game “move” optimization which leads to better board game evolution.

The Game of Reversi program was implemented using classical Genetic algorithm in which first set of population members are randomly generated and simple Evaluation Function serves as a "fitness discoverer" for the entire population. These successive populations are non-overlapping populations in nature. A simple genetic algorithm specifies either an individual or a population of individuals[14].

The simple genetic algorithm creates an initial population by cloning the individual or population of individuals being passed on their creation. In each newly derived generation, the algorithm creates an entirely new population of individuals by selecting from the previous population then mating to produce the new offspring for the new population using basic genetic operators. This process continues for a finite number of times till good fitness values are achieved or specific stopping criteria are met which are based on the kind of board game and nature of population constituents [15].

In genetic mating, population members do play each and every member from the population and the system will assign a fitness function to this lastly evolved member with respect to its performance in this evolutionary game with a certain crossover probability and some mutation probability. This game loop is performed for each set of population members to have sizable amount of members to be carried to of next generation pool.

Then the next generation is built according to these fitness values of the evolved pool constituents. The "fitness” can also be a string value for each generation. So the most important parameter of selection here is the "weight" value of evaluation function. If the weights are set correctly, our computer program will choose right move and play well. If the weights are not fixed correctly, the program may choose less important move and play very badly. Genetic algorithm comes here into picture which helps in learning good weights through genetic optimization. For the Reversi program the weights as "genes", and they are evolved to get a good genetic value set [16].

But, like the biological evolution, the weight evolution which ultimately results in better move selection happens slowly over a large population of individual. Reversi program weights are evolved with different weights for different board discs with every successive generation [17]. Instead of having just one weight value of the generation at a time, many weight values of generations co-exists and evolved. All of them are to be tested and then keep a few of the best ones. This gives better and better individual weight sets each time.

5. GAME OF REVERSI IMPLEMENTATION

For any board game constructing an Evaluation function is very important and critical. The construction of evaluation function usually works just by calculating simple mathematical features of the game position (In Reversi Game, aim is to put the player is better place with respect to stability and mobility.) The Reversi evaluation function is a number; which is got by computing a linear function for the positional features. Positional features are based on how the evaluation function uses six features of the position, it calculates a number for each one, and then multiplies each numbers by its own "weight" value [18]. As some of the features are more important:

The Evaluation function is based on following game feature metrics constituents:

- Number of stable discs (aka stable disc count)
- Number of playable squares (aka mobility)
- Potential mobility (means frontier that is the number of empty squares aside opponent stones)
- Parity (Who will put the last stone if no pass occurs)
- Edge pattern (pre-computed evaluation for each 10 squares edge+2X configuration in 8 phases of game)
- Corner pattern (10 squares on a triangle pined at corner plus one more diagonal square as shown in Figure 2)

The end search evaluation function majorly focuses on the usage of disc patterns which has small edge or parity feature associated with it. All these components are incrementally updated to achieve a good depth of search. The Reversi board is represented as a vector of length 64. Black disc is represented as 1 and white disc as -1. Empty board space has value of 0.

The Reversi board is represented as a vector of 64 elements where each vector element tells the weight value of one square in Reversi board. [19] The cumulative fitness value of all the board squares is calculated using dot product of two vectors (weights represented by Wi which are disc weight and functional features shown by Fi which are collective feature values of all feature metrics constituents mentioned in previous paragraph) as follows:

Fitness Weight Value:

$$f = (W1 \times F1) + (W2 \times F2) + \ldots + \quad (Wn \times Fn)$$

Where the function value f is called the static evaluation of a game board configuration and is the fitness function value for a given board configuration [20].

For the initial set of weights of each position are initialized as a value between -1 to1 as their opening values. These opening

values are used for the first set of generation. After the first pass is over, the weight values are evaluated as per the new discs positioning of the board which helps in choosing-deciding the next move. For each of the possible move the fitness weight value f is calculated for each of the possible depth values.

After all "f" results are calculated for all possible positions. To make the next move selection, the one with the highest fitness weight value (f) is selected. After the move has been made for the next move the whole above mentioned process for the all possible move for all depth variant types are explored and calculated and this loop is repeated for the entire game of Reversi to find the winner of the game.

The Reversi playing program demonstrates "human-like" approach to artificial game playing by evolving game playing parameters using genetic approach. The program took ten board square positions to make its genetic string as the Reversi board is symmetric and these ten disc families (shown in Figure 2), which cover entire categories of board discs.

Each set of board squares was assigned a multiplying co efficient based on the importance of that particular set. This group co efficient of ten functions sets underwent coefficient (weight) change as genetic operators were applied on them. These Weight results are collected and analyzes for two specific discs sets - corner discs and adjacent discs to corner discs.These Weight results are collected and analyzes for two specific discs sets - corner discs and adjacent discs to corner discs.

H	C	F	F	F	F	C	H
C	X	I	J	J	I	X	C
F	–	Y	K	K	Y	I	F
F	J	K	M	M	K	J	F
F	J	K	M	M	K	J	F
F	I	Y	K	K	Y		F
C	X	I	J	J	I	X	C
H	C	F	F	F	F	C	H

Fig 2: Othello Board Discs Set

6. RESULTS

The Game of Reversi program is performed on a Pentium machine with the RAM size of 2 GB. The results are collected for two discs sets. The population size was set to 200. Evolutionary weight behavior was visible even with this relatively small population size. The program employed fitness-proportional selection method.

Each member of a certain population underwent one game against each of the specified generation member. The program is iterated for a considerable amount of generations with a ply depth of one, two and three. The program took a considerably reasonable simulation time for a simulation time for a handful set of generations. For the Reversi program, the genetic operators are being used in each population formation of all generations. The crossover, survivor and mutation percentage rate was kept 90, 10 and 0.01 for all generations respectively. Mutation as it flips the genetic string value is set very low because of its impact on the fitness of mutated string.

Sixteen-generation file for each generation consisting of two hundred population members has been collected for corner discs (denoted by H in Fig. 2). The collection of maximum weight (fitness) values attained by corner discs during the entire generation for a set of corner discs are represented in the below mentioned Fig 3.

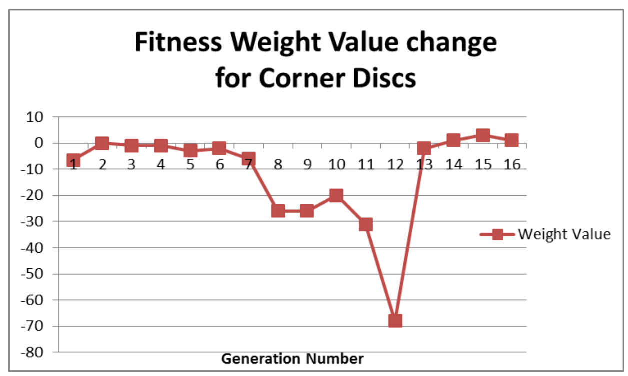


Fig 3: Weight changes in corner discs

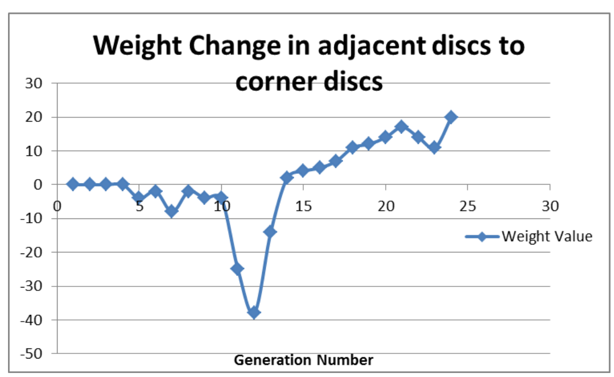


Fig 4: Weight changes in discs adjacent to corner discs

The collection of maximum weight (fitness) values attained by discs adjacent to corner discs during the entire generation for a set of corner discs (denoted by C in Fig. 2) are represented in the below mentioned Fig 4.

7. CONCLUSIONS

The weight changes in corner discs in Fig 3 indicates that the weight range spectrum attains the value from the initial phase of the game which shows the stability feature of the corner disc as it proves that once the corner is occupied by any player that cannot be taken back by its opponent. For generation 8 to 12 the low values are indicative that the corner discs during these games were captured by the opponent. For generation 13 to 16 the steady growth speaks about positive solidarity of the corner discs and its contribution in these generations.

Weight value graph shown in figure 4 indicate that firmness support of discs sets which are adjacent to corner discs. It also shows that these discs also supports "firmness" of disc capturing but is not severe enough compared to set of corner discs. The weight values for generations 14 to twenty four shows that the

discs adjacent to corner disc are contributing positively to the winning side of the Reversi player program which derived evolutionary learning for the disc set.

Based on the results of the experiment and collected–analyzed results for disc sets, the paper concludes that GAs enriches the authority of the board game-playing computer program by increasing the potentiality of better move selection. This results in providing a reasonable chance to play the game of Reversi more competently and meritoriously.

After this real-world test on a specific board game, it proves that evolutionary learning through genetic algorithm projected in this paper paves a new way to use it for a group of problem spheres which requires optimization as its one of the vital area. It is possible to enhance the efficiency of learning greatly. The optimization of the genetic algorithm can improvise fitness functions in order to calculate the board state accurately and make significant progress to improvise the computer game of Reversi.

8. REFERENCES

[1]. S. Schiffel and M. Thielscher. A multiagent semantics for the game description language. In Proc. of the Int.'l Conf. on Agents and Artificial Intelligence, Porto 2009. Springer LNCS.

[2]. T. Srinivasan, P.J.S. Srikanth, K. Praveen and L. Harish Subramaniam, "AI Game Playing Approach for Fast Processor Allocation in Hypercube Systems using Veitch diagram (AIPA)", IADIS International Conference on Applied Computing 2005, vol. 1, Feb. 2005, pp. 65-72.

[3]. Thomas P. Runarsson and Simon M. Lucas. Co-evolution versus self-play temporal difference learning for acquiring position evaluation in small-board go. IEEE Transactions on Evolutionary Computation, 9:628 – 640, 2005.

[4]. Yannakakis, G., Levine, J., and Hallam, J. (2004). An evolutionary approach for interactive computer games. In Evolutionary Computation, 2004. CEC2004. Congress on Evolutionary Computation, volume 1, pages 986–993, Piscataway, NJ. IEEE.

[5]. Hauptman and M. Sipper. Evolution of an efficient search algorithm for the Mate-in-N problem in chess. In Proceedings of the 2007 European Conference on Genetic Programming, pages 78–89. Springer, Valencia, Spain, 2007.

[6]. P. Aksenov. Genetic algorithms for optimising chess position scoring. Master's Thesis, University of Joensuu, Finland, 2004. Y. Bjornsson and T.A. Marsland. Multi-cut alpha-beta-pruning in game-tree search. Theoretical Computer Science, 252(1-2):177–196, 2001.

[7]. O. David-Tabibi, A. Felner, and N.S. Netanyahu. Blockage detection in pawn endings. Computers and Games CG 2004, eds. H.J. van den Herik, Y. Bjornsson, and N.S. Netanyahu, pages 187–201. Springer-Verlag, 2006.

[8]. Rosenbloom, P. (1982). A world championship level Othello program. Artificial Intelligence, 19:279-320.

[9]. Lee, K. -F., and Mahajan, S. (1990). The development of a world class Othello program. Artificial Intelligence, 43:21-36.

[10]. Hong, J.-H. and Cho, S.-B. (2004). Evolution of emergent behaviors for shooting game characters in robocode. In Evolutionary Computation, 2004. CEC2004. Congress on Evolutionary Computation, volume 1, pages 634–638, Piscataway, NJ. IEEE.

[11]. Billman, D., and Shaman, D. (1990). Strategy knowledge and strategy change in skilled performance: A study of the game Othello. American Journal of Psychology, 103:145-166.

[12]. Matt Gilgenbach. Fun game AI design for beginners. In Steve Rabin, editor, AI Game Programming Wisdom 3, 2006.

[13]. J. Clune. Heuristic evaluation functions for general game playing. In Proc. of AAAI, 1134–1139, 2007.

[14]. Holland, J. H. (1975). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. Ann Arbor, MI: University of Michigan Press.

[15]. Goldberg, D. E. (1989). Genetic Algorithms in Search,Optimization and and Machine Learning. Reading, MA: Addison-Wesley.

[16]. R.M. Axelrod, The evolution of cooperation (BASIC Books, New York, 1984).

[17]. J¨org Denzinger, Kevin Loose, Darryl Gates, and John Buchanan. Dealing with parameterized actions in behavior testing of commercial computer games. In Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG), pages 37–43, 2005.

[18]. D. Fogel, "Using evolutionary programming to create networks that are capable of playing tic-tac-toe," in Proceedings of IEEE International Conference on Neural Networks, San Francisco: IEEE, pp. 875–880,1993.

[19]. M. Muller, "Computer Go," Artificial Intelligence, vol. 134, pp. 145–179, 2002.

[20]. Singh Dharm, Thaker Chirag S and Shah Sanjay M. Quality of State Improvisation Through Evaluation Function Optimization In Genetic Application Learning IEEE xPlore identifier 10.1109/ETNCC.2011.5958494 page(s) 93-97.