

Reversi Game Implementation

Kyle Alexander

registration: 100082709

1 Introduction

Reversi is a game played on an 8 x 8 Board by 2 people with 64 pieces that have both light and dark sides. Players alternate turns and in each turn a player places a single piece or "Disc" on to the board capturing all pieces between the placed piece and all pieces of the the players colour that are either horizontal, vertical or diagonal to the placed piece. Players win by having the most pieces flipped to their colour on the board at the point when neither player can legally play a piece. A move is only legal when it captures an opponents piece.

The computer opponent for an implementation of Reversi has been studied since very early in the field of computer ai and decision making research with the earliest computer implementation being released in 1977 (Wright, 1977) written in FORTRAN.

Section 3 looks at heuristic or evaluation functions for Reversi while Section 4 compares search algorithms and game trees. Section 5 examines when existing programs have prioritised different functions and changes in the early and late game.

2 History

The history of computer Reversi started in 1977 when the first version written in Fortran and published by Creative Computing. The idea was picked up quickly and soon after Nintendo released an arcade version of the game exclusively in Japan (Picard, 2013) which offered both 1 and 2 player modes of play.

Achievements in the computers strategy have made leaps since the late 1970's with the early Moor program beating the Othello world champion in a six game match in as early as 1980. Milestones in AI strategy include the Logistello which beat the world champion in a six game match winning all six games.

3 Evaluation Functions

Evaluation functions are at the heart of any board game playing computer and are used to determine the strength of a move given a particular state of the board. In Reversi there are three common Evaluation functions stated which are coin parity, mobility and how many corners are captured. Both Clune (2007) and Sannidhanam and Annamalai (2015) agree that the coin parity function, which attempts to maximise the amount of pieces on the board, despite being a seemingly good strategy often employed by new players result in very poor performance as large numbers of pieces can be captured in a single move.

Rosenbloom (1982) also states that attempting to maximise ones own discs as a strategy to play the game results in loss in most scenarios as it does not account for mobility

and towards the end game can leave the player with majority pieces but these pieces are easily captured and the player has very few moves possible to recapture any of the opponents pieces.

3.1 Edge Control and Weighted Board Strategies

One strategy that seems to always be considered if not implemented with some weighting to how the final move is scored is capturing the edges and corners of the board. Clune (2007) mentions that a brief analysis of Othello produces a piece differential pay off function that exclusively scores a board position based on what edges the player controls. Interestingly the weightings per corner or edge are not equal and the pay off prioritises

1. bottom right corner
2. top right corner
3. top edge
4. bottom left corner
5. bottom edge
6. top left corner
7. right edge
8. left edge

in descending order. Other papers are less specific with Sannidhanam and Annamalai (2015, sec 5.1.3) talking about how assigning weights to corners based on how plausible it will be for the player to capture that corner in the future. It does not talk about how it works the values out but does talk about the importance of corners as they cant be recaptured. It can be assumed that this extends to the numerical analysis in (Clune, 2007, see Othello) as edges also have less chance to be recaptured but are not completely invulnerable. Lee and Mahajan (1986) Also mentions Edge Control in a small section however it states that playing a strategy entirely based on edge capturing can lead to reduced mobility and Edge traps which are moved specifically designed to punish edge greedy players.

Rosenbloom (1982) extends the edge strategy with a more general weighted square strategy that doesn't just apply values to the corners and edges but to the entire board. It specifically references Rosenbloom (1982, fig. 2.1) which denotes several weak board positions with an x as the positions can easily give away corners to the opponent.

While aiming to occupy the edges can be used as part of strategy it has pitfalls such as mobility restriction and can be countered by edge traps if it is the only strategy used. The weighted square works as a more complete strategy and was used by an implementation of reversi called IAGO. Its performance and losses to the human entrants proved how by itself the weighted square strategy was still quite poor compared to others methods at the time.

3.2 Mobility

Maximising mobility is a method used that attempt to maximise the amount of moves the player can make on a given turn. Mobility is discussed in Lee and Mahajan (1986) and Sannidhanam and Annamalai (2015) while Rosenbloom (1982) mentions a similar strategy which is the minimum disc strategy.

Both of these strategies at there base rely on keeping the players options open through the game so there is little chance of the opponent being able to assert strategic control. This strategy also simultaneously attempts to limit the opponents moves on their turns. An example algorithm sourced from Sannidhanam and Annamalai (2015) would be

```
if((Max Player Actual Mobility Value
+ Min Player Actual Mobility Value) !=0)
    Actual Mobility Heuristic Value =
    100* (Max Player Actual Mobility Value
    - Min Player Actual Mobility Value)/
    (Max Player Actual Mobility Value
    + Min Player Actual Mobility Value)
else
    Actual Mobility Heuristic Value = 0
```

Lee and Mahajan (1986) talks about while attempting to maximise mobility is necessary to form a good strategy extra steps must be taken to make the method competitive. It goes on to further talk about as well as maximising the players own moves that the moves made by the user should minimise the amount pieces captured to limit opponents moves and in particular to limit the directions in which pieces are captured to further limit mobility.

3.3 Stability

Stability is topic discussed in both Rosenbloom (1982) and Sannidhanam and Annamalai (2015). Both pieces agree that it is an important factor when determining the strength of a players positioning on the board and Sannidhanam and Annamalai (2015) uses the stability of each board position to build a similar image as used in the weighted square strategy mentioned above 3.1.

Lee and Mahajan (1986) explains how disc stability is used as a basis for all further evaluation functions and goes on to list 2 reasons why disc stability is the most important factor to win which are

- Stable discs can be used to create more stable discs.
- Stable discs are the deciding factor at the end of the game, when all discs are, by definition, stable.

Sannidhanam and Annamalai (2015) groups discs into three types of stability which are stable, unstable and semi stable. Stable discs are discs that cannot be recaptured at any point in the future of the game; unstable discs are pieces that can be immediately captured and semi stable pieces are pieces that can be captured in the future but not immediately.

4 Game Tree

The Game tree is used to consider moves past the current one and since considering all possible future moves costs time and computing power at an exponential rate algorithms for the game tree are mostly based on keeping the search of the game tree as efficiently as possible. Both Lee and Mahajan (1986) and Rosenbloom (1982) are papers on the creation of a Reversi implementation and both use a minimax tree with alpha-beta pruning and iterative deepening.

Lee and Mahajan (1986) Also mentions how it performed tree pruning based on individual moves as well as the resultant board positions, moves that placed pieces on unstable positions early game were pruned off before being considered as even though the evaluation function may have been preferable at the depth that was searched these positions could easily compromise the players position later on in the game. This pruning was done only until a certain point e.g a disc count was exceeded in this case 35. The search function used by BILL in the above paper also used a lightly modified version of its alpha beta pruning known as a zero window search. This search establishes earlier and narrower alpha beta values than the typical \pm infinity. This is done to increase alpha beta cut off's which increases the speed of the search.

Conversely the IAGO implementation analysed in Rosenbloom (1982) does not implement a zero window search but diverts time to properly ordering nodes to improve the alpha beta pruning. This implementation also uses a Killer table so that it can record moves that produced cut off's in previous parts of the tree.

both Olivito et al. (2010) and Rosenbloom (1982) also use the information gained for the iterative deepening which involves executing the alpha beta searches at increasing depths successively so as to get the largest amount of information in a fixed time period to order nodes which will also improve the cut off's gained from alpha beta pruning.

5 Early, Mid and Late Game

Most implementations of Reversi employ varying algorithms over the course of a game. A common early game tool is an opening book that picks an opening from a set of pre recorded opening that have been proved to be strategically sound. Lee and Mahajan (1986) Uses an opening book for at least 2 moves on every opening and sometimes more depending on starting colour. IAGO and the implementation noted in Olivito et al. (2010) however do not use opening books and play early game with an evaluation function that heavily prioritises mobility and gaining corners. They both stay very similar in the mid game prioritising the same aspects which is different to the implementation of BILL in Lee and Mahajan (1986) which seems to place a lot more importance on gaining control of edges as well as corners.

All implementations mentioned in the papers above share the same late game strategy which is to completely solve the game from the current boards position. Evaluation functions for the end game heavily favour the coin parity strategy (Lee and Mahajan, 1986, sec. 3.4.1.3)

References

- Clune, J. (2007). Heuristic evaluation functions for general game playing. In *AAAI*, volume 7, pages 1134–1139.
- Lee, K.-F. and Mahajan, S. (1986). Bill: a table-based, knowledge-intensive othello program.
- Olivito, J., González, C., and Resano, J. (2010). Fpga implementation of a strong reversi player. In *Field-Programmable Technology (FPT), 2010 International Conference on*, pages 507–510. IEEE.
- Picard, M. (2013). The foundation of geemu: A brief history of early japanese video games. *Game Studies*, 13(2).
- Rosenbloom, P. S. (1982). A world-championship-level othello program. *Artificial Intelligence*, 19(3):279–320.
- Sannidhanam, V. and Annamalai, M. (2015). An analysis of heuristics in othello.
- Wright, E. (1977). Othello. <http://www.atariarchives.org/bcc3/showpage.php?page=258>. (Accessed on 10/18/2016).

Literature review

Introduction: brief description of project, areas of knowledge required, roadmap	First	2.1	2.2	3	Fail
Discovery of suitable quantity and quality of material	First	2.1	2.2	3	Fail
Description of key issues and themes relevant to the project	First	2.1	2.2	3	Fail
Evaluation, analysis and critical review	First	2.1	2.2	3	Fail

Quality of writing

Clarity, structure and correctness of writing	First	2.1	2.2	3	Fail
Presentation conforms to style (criteria similar to conference paper reviews)	First	2.1	2.2	3	Fail
References correctly presented, complete adequate (but no excessive) citations	First	2.1	2.2	3	Fail

Revised Workplan (if applicable)

Measurable objectives : appropriate, realistic, timely	First	2.1	2.2	3	Fail
--	-------	-----	-----	---	------

Comments

<div style="border: 1px solid black; height: 480px; width: 100%;"></div>
--

Supervisor: supervisor

Markers should circle the appropriate level of performance in each section. Report and evaluation sheet should be collected by the student from the supervisor.