

# COMP3608 Artificial Intelligence (Adv): Othello

Elizaveta Lisa Fedorenko: 309201470

Tianyu Pu: 310182212

April 21, 2012

## **Abstract**

## **1 Introduction and Background**

### **1.1 Othello: The Game**

Othello (also commonly known as Reversi), is a two-player strategy board game played on an 8x8 board. The game begins with four pieces in the centre, two for each player's colour. Both players take turns to place a piece in their colour onto the board, flipping over all the opponent's pieces that are caught or 'flanked' between the new piece and another of the player's colour. The move is only valid if at least one token was flanked in any direction. The game ends when the board is filled and/or when there are no more possible moves for either player.

### **1.2 Interpretation of the Rules**

There are a number of variations on the rules of Othello that are used in different situations such as tournaments. This section outlines the version of the rules that was used to create the game.

1. You can only place a piece on a tile that is empty and is in one of the eight squares immediately surrounding a piece of the opponent's colour.

2. You can flip the opponent's pieces that are directly flanked by your new piece (the piece just put down) and the nearest piece of your colour in any of the eight directions.
3. A move is only valid if you flip (or flank) at least one of your opponent's pieces.
4. A player may not pass unless they have no more valid moves: in this case they automatically pass.

### 1.3 Playing the game

Our version of the game begins by asking the player whether they wish to verse a human or else which of the three AIs they wish to play. Input is tested to be one of the letters A, B, C or H. If invalid input is given the prompt is given again. They are then asked for their name (or both their names). As the name can be any value this is not tested (It is only included to make the input more personal).

The player is then told which colour they are (the game is symmetric so this doesn't limit the game choices). The game defaults to human playing first. The rest of the user input requires the user to type in their move of the form 'x y' eg. 1 2. This is tested to be a valid move. It is not tested to be correct syntax. This is assumed (as per assignment spec). In order to make the game more user friendly, at each stage your valid moves are printed out as a list, so you can choose a move. And all the moves played so far are also printed. When the AI moves you also see their valid moves and which move they choose. This makes games more recordable and user-friendly.

In addition a score is given after every move. Also at the end game the final board, score and winner is displayed.

## 2 Implementation of Strategies

### 2.1 Heuristic

The game of Othello is complex Collins [2005]. We therefore designed two evaluation functions. Strategy A and B use a very simple intuitive evaluation strategy. The heuristic simply counts the number of their tokens on the board and aims to maximise it. However for strategy C we implemented a

far more complicated heuristic function that takes much more game strategy into consideration. There are three main methods to judge a position in Othello as follows et al. [1993]:

1. *Number of tokens* Obviously as the aim of the game is to have the most tokens, this is a good basic estimate of who is winning at any one point in time.
2. *Stability* Certain pieces are a lot easier to ‘flank’ than others. In particular corners are very stable and cannot be flipped. Pieces protruding from corners are also more stable.
3. *Mobility* Having more moves available means you are better positioned strategically and less constricted.

Each of these methods has its own shortcomings and advantages. We chose to implement two heuristics. The first is a basic one (`eval1()`) which simply uses the number of tokens to value its position. The second much ‘smarter’ AI uses a far more complex heuristic (`eval2()`), which can be represented thus:

$$h(\text{black}) = \text{number black pieces} * f(\text{pieces}) + \text{factor} * \text{number available moves} \quad (1)$$

where  $f()$  is a function of the stability of each piece.  $f()$  weights corner pieces and pieces joint to corner pieces higher than centre unprotected pieces. The factor is a constant which chooses the balance to be given between each of the strategy calculations. Clearly both heuristics are admissable. As it is optimal to have stable pieces, more moves and more pieces on the board.

## 2.2 Strategy A

Strategy A uses the minimax algorithm. This has been implemented recursively. We have allowed the depth explored by the algorithm to vary depending on the level (1, 2 or 3) chosen by the user. The algorithm can be summarised thus:

1. Perform a depth-first search either to the terminal state or to the value  $depth = level * 2 + 1$ , whichever comes first.
2. Evaluate the value of the heuristic at each terminal state.

3. Return the minimax value. Return MIN if it is the opponents move, and MAX heuristic at that level if it is your move.
4. Select the move with the maximum of the minimax values of the first level of children.

Note this assumes that min will play optimally in each turn.

## 2.3 Strategy B

Strategy B is very similar to strategy A. The exception is it avoids exploring some paths, or prunes some paths. This allows it to explore deeper. The basic algorithm can be summarised thus:

1. Traverse the tree in depth first search order.
2. At each non-leaf node store the 'best so far' value: max is the  $\alpha$  or the best maximum value so far, min is the  $\beta$  or the best minimum value found so far.
3. Don't explore below a node  $n$  if either:  $n$  is a MAX node and  $\alpha(n) \geq \beta(i)$  for some MIN node  $i$  that is an ancestor of  $n$ . This is the  $\beta$  cutoff. OR:  $n$  is a MIN node and  $\beta(n) \leq \alpha(i)$  for some MAX node  $i$  that is an ancestor of  $n$ . This is the  $\alpha$  cutoff.

Note this assumes that min will play optimally in each turn.

## 2.4 Strategy C

For strategy C we decided to implement a non-deterministic strategy. We ran the mini-max algorithm and selected the best three outcomes (ranked). We then returned one of the moves based on a random card draw where each solution was weighted based on its rank thus:

1. The best move had an 80% chance of being selected.
2. The second best move had a 15% chance of being selected.
3. The third best move had a 5% chance of being selected.

Although this AI does not always select the optimal move, it is non-deterministic and thus harder to predict. We felt that the other two strategies would be beatable by experienced players who could predict where the AI would move next and thus adjust their strategy. The element of chance added here doesn't allow such calculations. Further, sometimes taking a risky move may lead to a more optimal final state.

## **3 Empirical Evaluation of the Strategies**

### **3.1 Set Up**

We decided to compare the relative strength of the AIs by conducting test experiments. For the human vs AI games, we decided to let the human play randomly (pick a random valid move each turn) and have each respective AI strategy respond to that. The outcome was recorded (whether black or white won, with white being the computer opponent).

### **3.2 Findings**

#### **3.2.1 Expectation**

We expected strategy A to perform the worst as it didn't search as deeply and didn't have a great heuristic. We also expected strategy B to have shortcomings due to the predictability as well as the more basic heuristic function used. Strategy C we expected would have very few losses. This is because of a number of factors. Firstly it is random, and thus cannot be predicted by a clever opponent. Secondly and most importantly it uses a much more accurate heuristic function.

#### **3.2.2 Results**

With 10 runs per computer opponent, we noted these results:

- against strategy A (minimax), the randomly playing human won 4 times out of 10 games
- against strategy B (alpha-beta pruning), the randomly playing human won 4 times out of 10 games

- against strategy C (described above), the randomly playing human won 3 times out of 10 games

### 3.3 Conclusion

Had we run a greater number of tests, our results would have been more accurate. However, we were limited by the time we had to run these tests.

The results we observed were as we expected – both minimax and alpha-beta strategies were equivalent, since for both we limited the cut-off depth to the same number (3) and used the same heuristic function. Had we run more games, we would have expected slight discrepancies in the number of games that the AI won – but we would still expect a very similar outcome for both strategy A and B. We noted (but did not measure exactly) that the 10 runs for strategy B took less time to run than for strategy A, which is also what we would expect, knowing the properties of each algorithm.

Strategy C incorporated some level of unpredictability into the strategy, which showed no noticeable improvements against the other 2 strategies (with the results we observed), especially because the human was playing moves entirely at random. However we designed this strategy because it would be effective against experienced human players who could easily predict what the computer AI would play next – and therefore beat it. As can be seen from the results, there is no major difference when the games are played at random.

## 4 Reflection

### 4.1 Lisa's Reflection: 309201470

I thoroughly this assignment and feel that I learnt a lot from it. It was very satisfying to implement a working game and strategies as it highlighted the application of our studies to real-world scenarios. I particularly liked the open-ended-ness which meant that we could get creative with the third strategy. The most valuable thing I felt I learnt from this assignment is the complexity of AI. This complexity arises from user-input and makes it very difficult to create an ideal opponent. The human opponent will act very differently based on his/ her level of expertise and will often not choose the optimal strategy. Thus it makes the task of the AI much more complex in

moving optimally. Further it made me consider the importance of chance. Choosing the most strategically sound option each time is very predictable and is not an optimal strategy. In addition I learnt a lot about the game of Othello and the many complicated strategies involved.

## **4.2 Tian's Reflection: 310182212**

I feel that this has been one of the most interesting and educational code-writing tasks that I have done to-date. I found it challenging at times, but ultimately rewarding when everything came together. I think that the most important thing I learned from this assignment was the application of artificial intelligence to real-world, concrete examples (such as games). This assignment has allowed me to gain a deeper understanding of the course material covered in class, and has also taught me the value of perseverance. I also feel that writing this report, and having to communicate our findings to an audience, was also a very important skill to practice.

# **5 Extensions**

## **5.1 Multiplayer**

In addition to the assignment requirements we implemented a human-vs-human game option.

## **5.2 Possible moves**

We print out a list of possible moves for each player to make the game easier and more pleasant to play. If a move isn't possible for either player at any turn, we print out a message indicating this, and the turn passes immediately to the other player.

# **6 Appendix**

For each of the three strategies, we played each opening move to test whether we got the expected output, and to see if the AI responded correctly.

The output of the tests can be seen in the a, b and c files in the assignment folder.

## References

- E Collins. A minute to learn... a lifetime to master, July 2005. URL [boardgamegeek.com/thread/72572/a-minute-to-learn-a-lifetime-to-master](http://boardgamegeek.com/thread/72572/a-minute-to-learn-a-lifetime-to-master).
- Emmanuel Lazard et al. Strategy guide, 1993. URL <http://radagast.se/othello/Help/strategy.html>.