**Background**

Search strategies such as MiniMax which explore all possible game states may be impractical when time constraints are considered. Complex games and games with large boards leading to large numbers of possible moves result in game trees that exponentially increase both in size and in the time required to navigate through them. Heuristic evaluation functions provide a way to guess the utility of a game state at any level of the game tree without having to navigate to the final game states to determine who wins. Heuristic evaluation functions shoukd be based on properties of the game state that are strongly correlated to a winning position. The purpose of this exercise was to invent 3 possible heuristic evaluation functions and to test their performance against a player using an evaluation function which maximises the difference between the number of moves available to the Player and the number of moves available to the Opponent.

Isolation is an adversarial game so the heuristic evaluation strategies selected for testing employed strategies for making life difficult for the Opponent.

| **Heuristic** | **Description** |
| --- | --- |
| AB\_Improved | Rewards boards with the greatest difference between moves available to Player and moves available to Opponent  Mp - Mo  where Mp is number of moves available to Player and Mo is number of moves available to Opponent |
| AB\_Custom | Strongly rewards boards where Opponent has fewest available moves  -log(Mo)  where Mo is number of moves available to Opponent |
| AB\_Custom\_2 | Strongly rewards boards where Opponent's available moves are towards periphery of board  -log(Do)  where Do is the average distance from centre of all Opponent’s available moves |
| AB\_Custom\_3 | A combination of the other heuristics  10Mp – 6Mo -8Dp+ 2Do + Bp  where Mp is the number of moves available to Player, Mo is number of moves available to Opponent, Dp is the average distance from centre for Player’s available moves, Do is the average distance from centre of Opponent’s available moves and Bp is the number of blocking moves available to Player. |

**Table 1** – Heuristic evaluation functions tested

**Test**

The 3 heuristic functions under test and a base line heuristic function known as AB\_Improved (table 1) were each used by a test player who was pitted against a number of other players employing different strategies and different heuristic evaluation functions. The goal was to find a heuristic evaluation function that the test player could use that would be better than the AB\_Improved heuristic.

The test player employed Minimax with Alfabeta Pruning and Iterative Deepening.

The opponents employed a variety of strategies and heuristics (table 2).

| **Player** | **Description** |
| --- | --- |
| Random | Scores boards randomly with no correlation to game state |
| MM\_Open | Uses minimax with a heuristic maximising the utility of boards with the greatest number of moves available to the Player |
| MM\_Centre | Uses minimax with a heuristic maximising the utility of boards where the Player is currently furthest from the Centre of the board |
| MM\_Improved | Uses minimax with a heuristic maximising the utility of boards with the greatest difference in the number of moves available to the Player less the number of moves available to Opponent |
| AB\_Open | Uses minimax with alphabeta pruning and a heuristic maximising the utility of boards with the greatest number of moves available to the Player |
| AB\_Centre | Uses minimax with alphabeta pruning and a heuristic maximising the utility of boards where the Player is currently furthest from the Centre of the board |
| AB\_Improved | Uses minimax with alphabeta pruning and a heuristic maximising the utility of boards with the greatest difference in the number of moves available to Player less the number of moves available to Opponent |

**Table 2** – Players in competition

**Tournament Results**

The test player played 40 matches (NUM\_MATCHES was increased to 40 to make the results more statistically significant) against each opponent using each of the 4 heuristics. Table 3 shows the results.

Match # Opponent AB\_Improved AB\_Custom AB\_Custom\_2 AB\_Custom\_3

Won | Lost Won | Lost Won | Lost Won | Lost

1 Random 35 | 5 31 | 9 37 | 3 40 | 0

2 MM\_Open 25 | 15 23 | 17 26 | 14 30 | 10

3 MM\_Center 31 | 9 34 | 6 33 | 7 37 | 3

4 MM\_Improved 30 | 10 27 | 13 27 | 13 34 | 6

5 AB\_Open 19 | 21 21 | 19 20 | 20 25 | 15

6 AB\_Center 21 | 19 19 | 21 14 | 26 27 | 13

7 AB\_Improved 21 | 19 20 | 20 20 | 20 26 | 14

-------------------------------------------------------------------------------------------------------

Win Rate: 65.0% 62.5% 63.2% 78.2%

**Table 3** – Results showing how the 3 heuristics under test performed against AB\_Improved when competing against a variety of players employing different strategies .

Only one of the custom heuristics, AB\_Custom\_3, outperformed AB\_Improved both in each individual contest and over the course of the entire tournament. AB\_Custom\_3 achieved a win rate of 78.2% compared to AB\_Improved’s 65.0%.

This may be expected as AB\_Custom\_3 uses a heuristic that considers more aspects of the game state that may be considered beneficial to Player and detrimental to Opponent, than any of the other heuristics, including AB\_Improved which considers only the difference in the number of moves available to the both players.

The test player comfortably outperformed the Random player using all 4 heuristics as was expected.

Against MM\_Open AB\_Custom\_3 and AB\_Custom\_2 outperformed AB\_Improved with win rates of 75% and 65% compared to AB\_Improved with 62.5%. AB\_Custom slightly underperformed at 57.5%.

Against MM\_Centre all 3 test heuristics performed better than AB\_Improved with AB\_Custom\_3 again the top performer with a win rate of 92% versus AB\_Improved’s win rate of 75%.

The test player beat MM\_Improved player using all 4 heuristics. AB\_Custom\_3 was top performer with a win rate of 85% followed AB\_Improved on 75% then by AB\_custom\_2 and AB\_Custom both on 67.5%.

The contest with AB\_Open was the closest. AB\_Custom\_3 was best with win rate of 65% compared to AB\_Improved’s win rate of 47.5% .

The test player beat AB\_Centre player using only AB\_Custom\_3 and AB\_Improved. AB\_Custom\_3 scored a win rate of 67.5% compared to AB\_Improved’s 52.5%.

In the final match against player AB\_Improved the AB\_Custom\_3 heuristic again performed best with a win rate of 65%. The AB\_Improved heuristic scored just over 50%.

**Recommendation**

From the 4 heuristics under test AB\_Custom\_3 is the clear winner and it does perform significantly better than AB\_Improved. This could be due to the number of properties of the game state that could be considered to be beneficial to Player and detrimental to Opponent that have been included.

However, experimentation with weightings did not produce results as expected and often a simple increase in the weighting of terms thought to have a positive correlation to a good board position for Player, such as many available moves or centrality of available moves or having many blocking options on the Opponent, did not improve Player’s performance as expected.

Further experimentation would be required to optimise the weightings of each term.

# Success Criteria

| CRITERIA | MEETS SPECIFICATIONS |
| --- | --- |
| Have at least three (3) evaluation heuristics besides null\_score(), open\_move\_score(), and improved\_score() been implemented and analyzed? | At least three evaluation functions are implemented and analyzed. |
| Has the performance of agents against the testing agents been adequately described? | A brief report lists (using a table and any appropriate visualizations) and verbally describes the performance of agents using the implemented evaluation functions. Performance data includes results from tournament.py comparing (at a minimum) the best performing student heuristic against the ID\_Improved agent. |
| Does the report make a recommendation about the best evaluation function, and is this recommendation adequately justified? | The report makes a recommendation about which evaluation function should be used and justifies the recommendation with at le |