John McAvenue Analysis of Heuristic Evaluation Functions

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