Game Playing Agent Project Report

This project involved implementing a basic game playing agent to compete against a computer controlled player at Knight’s Isolation, played on a 9 x 11 board. The computer controlled player was used a basic Minimax algorithm to decide its next move, and this was played against with an agent using Minimax with Iterative Deepening and Alpha-Beta Pruning to set a bench mark score. This benchmark was then used to evaluate a set of custom heuristics for the Minimax algorithm, which it was aimed would improve on the standard, Player Moves – Opponent Moves, used to evaluate a game state in case of not having sufficient time to search each branch to every possible endgame.

1. Results

Shown below in fig. 1 are the results for each heuristic tested against the benchmark, which was a simple combination of the number of available player and opponent moves. Most of the heuristics tested, including the benchmark model, used an Iterative Depth of 4 with a time limit of 150ms for each move. The full table of results can be found in the appendix section. The heuristics tested were as follows.

Offensive to Defensive (OTD):

$H = P - 2 * O$ if $m <= 0.5$ else $2 * P - O$

Defensive to Offensive (DTO):

$H = 2 * P - O$ if $m <= 0.5$ else $P - 2 * O$

Offensive to Defensive Graded (OTD-G):

$H = P * 2m - O * 2(1 - m)$

Ultra Offensive to Defensive Graded (UOTD-G):

$H = P * 4m - O * 4(1 - m)$

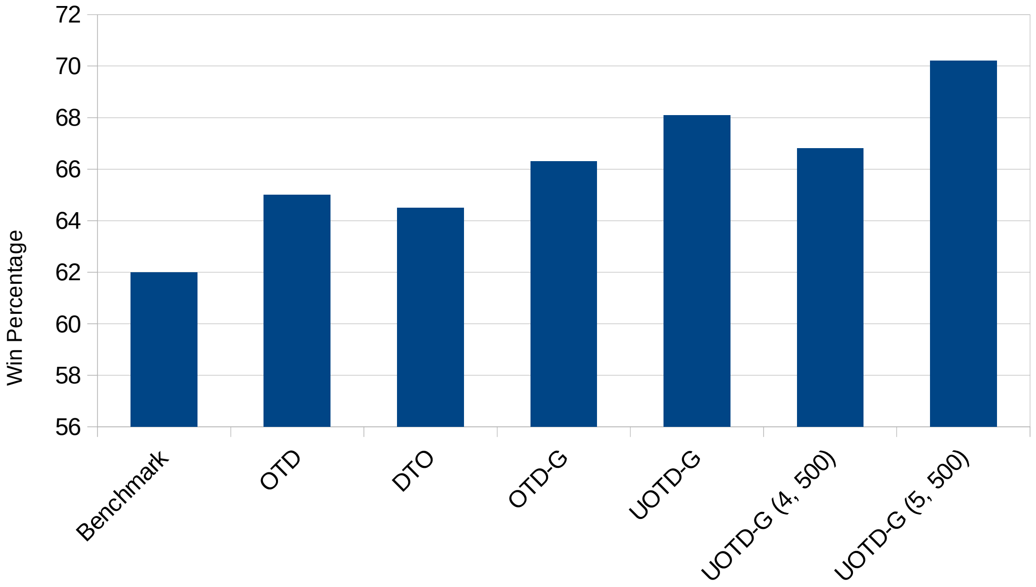

Figure 1. Bar chart showing percentage win rates for different heuristics against a computer controlled Minimax player

The final, best performing heuristic (UOTD-G) was experimented with to query the relative importance of accuracy and speed for performance. The time limit was increased to 500 ms for two further trial runs, with one of these using an iterative search depth of 5 instead of 4.

2. Discussion

What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during search?

The features used in this heuristic are the number of available moves for each opponent, as well as the proportion of used squares on the board. This feature is intended to indicate the level of development of the game from start to finish, which can be helpful in deciding what level of aggression with which to play.

Analyze the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?

Using a search depth of 4, 150 ms provides a better performance than 500ms, meaning that the extra time must be more of an advantage to the computer than the player in this scenario. However, this changes when the extra time is used to increase the player's search depth to 5, a configuration which gives the best performance of all those tested.

3. Appendix

Run #	Benchmark	OTD	DTO	OTD-G	UOTD-G	UOTD-G (4, 500)	UOTD-G (5, 500)
1	53	60	67	61	68	66	74
2	59	66	61	78	70	62	66
3	60	57	58	70	71	71	72
4	65	63	66	68	64	59	68
5	72	67	64	58	70	67	73
6	64	63	68	75	73	65	75
7	61	65	56	63	64	68	60
8	58	70	68	63	70	65	70
9	66	67	68	62	68	73	72
10	62	72	69	65	63	72	72
Average	62	65	64.5	66.3	68.1	66.8	70.2