

Heuristic Analysis

In this project, we created an agent for playing a variant of the 7x7 isolation game where each move is time-limited and restricted to L-shaped movements like a knight in chess. We wrote three different scoring functions to test three different heuristics and compared them to a baseline to determine the best heuristic. Our agent played against seven different types of opponents that used different playing strategies. The following is my analysis of my three heuristics as compared to the baseline.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	8	2	6	4	6	4
2	MM_Open	9	1	7	3	5	5	7	3
3	MM_Center	9	1	8	2	8	2	9	1
4	MM_Improved	5	5	5	5	5	5	5	5
5	AB_Open	6	4	7	3	4	6	5	5
6	AB_Center	8	2	5	5	4	6	6	4
7	AB_Improved	4	6	3	7	4	6	5	5
<hr/>									
Win Rate:		70.0%		61.4%		51.4%		61.4%	

Heuristic 1

For the first heuristic, I used the heuristic presented in the lecture that two times the number of available opponent moves from the number of moves for the player. According to the lecture this heuristic will *“cause the computer player to chase after the opponent”*. This heuristic yielded the best result although not by a very large margin. In the above chart, there is a tie between heuristic 1 and 3, but in the majority of my test runs, it was the winner.

Heuristic 2

For the second heuristic, I chose a heuristic that favors moves closer to the center of the board. The premise is that there are more available moves when you play closer to the center. This heuristic did not perform any better than the baseline.

Heuristic 3

For the third heuristic, I chose a heuristic that calculate the ratio of my moves compared to my opponent's moves.

Conclusion

Heuristic number 1 yielded the best results because it rewards the player for choosing a move that results in a larger number of available moves but more importantly, it gives an even larger reward for limiting the number of available moves for the opponent. The tradeoff is that this heuristic is more complex and takes more CPU time to calculate. Since each turn is CPU time-limited, it means that a simpler heuristic such as “number my moves” is able to go deeper.