

Evaluation Function Heuristic Report

Introduction

The goal of the project is to build an agent that plays the adversarial game 'Isolation'. In this project, we incorporated the Minimax algorithm to determine next move for each level. And to meet the time constraints and make the calculation more efficient, Alpha-beta Pruning and Iterative Deepening methods are applied. Along with these techniques, I involved several different heuristics to evaluate each minimax search level. This report includes description of each heuristics and analysis on their test results.

Heuristics

`custom_score()` – In this heuristic, the player will favor moves that is closer to the center of the board. That is, every time the position is at the center or around center, there will be an extra bonus score added to the evaluation.

`custom_score_2()` – In this heuristic, the player plays defensively. In each move, the player will avoid the opponents by choosing a spot that is more distant from the opponents' last move. The value returned is the distance between player move and opponent's move.

`custom_score_3()` – In this strategy, the player will play aggressively by approaching opponent's move and force opponents to go into corners. The value returned is $(\text{own_moves} - 3 * \text{oppn_moves})$

Results

This script evaluates the performance of the `custom_score` evaluation function against a baseline agent using alpha-beta search and iterative deepening (ID) called 'AB_Improved'. The three 'AB_Custom' agents use ID and alpha-beta search with the `custom_score` functions defined in `game_agent.py`.

Playing Matches									

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	10	0	9	1	8	2
2	MM_Open	6	4	6	4	7	3	4	6
3	MM_Center	9	1	9	1	8	2	6	4
4	MM_Improved	8	2	5	5	6	4	6	4
5	AB_Open	5	5	6	4	6	4	5	5
6	AB_Center	6	4	4	6	5	5	4	6
7	AB_Improved	5	5	4	6	6	4	7	3

Win Rate:		68.6%		62.9%		67.1%		57.1%	

Among all three heuristics, `Custom_2()`, the defensive play, has the highest win percentage. At the same time, `custom_score()`, which is the central biased strategy, also have good performance.