

Heuristic Analysis

1. Introduction

Advance Game Playing Agent project is to create game agent to play Isolation game by implementing two types of agents:

Minimax Agent - use minimax algorithm to apply depth-first search for possible moves in each isolation game state.

Alpha-Beta Agent - apply alpha-beta pruning and treatise deepening to improve performance of minimax agent.

Agents goal is to win the game. To do so, they need evaluation function which indicates score of players in each game state and assigns value according to situation at the time of evaluation. Score in isolation is defined according to game's rules. Players take turn and move in L-shaped from current position. Players with no legal move left lose the game.

2. Evaluation Functions

1. Available Moves Difference (Custom Score1)

Available Moves Difference gets number of legal moves of each player then find the different between legal moves of active player and inactive player. Return that value as game score.

This function returns higher value when the player has more legal moves than opponent's. Output of this function is simple and well represent states that player win the game. Range of this function is $[-49, 49]$

2. Ratio of Available Moves (Custom Score2)

Ratio of Available Moves gets number of legal moves of each player then divides legal moves of active player with legal moves of inactive player. Return that value as game score.

Ration of Available Moves return more drastic change comparing with Available Moves Difference. Range of this function is $[0, inf)$.

Since denominator could be zero and could lead to error, if-loop is implemented before calculate score to check if opponent's move is zero, player is the winner.

3. Natural Log of Available Moves Ratio (Custom Score3)

Natural Log of Available Move Ratios gets number of legal moves of each player then divides legal moves of active player with legal moves of inactive player then take $\ln()$ to the result. Return that value as game score.

Range of this function is $(-1.69, 1.69)$

Similar to ration of Available Moves we need checker if function return definite state.

3. Performance Comparison and Analysis

***** 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	8	2	8	2
2	MM_Open	5	5	5	5	3	7	3	7
3	MM_Center	8	2	9	1	7	3	6	2
4	MM_Improved	7	3	8	2	5	5	5	5
5	AB_Open	4	6	5	5	5	5	3	7
6	AB_Center	7	3	4	6	4	6	5	5
7	AB_Improved	4	6	5	5	5	5	3	7
Win Rate:		62.9%		65.7%		52.9%		50.0%	

The result shows that Number of Available Moves Different performed best. None liner function in custom_score2, and custom_score3 does not perform well.

I would suggest Different of Available Moves is best evaluation function.

This also could be a result of single pre-defined move I used as opening book in the get_move() function. Improve opening book could improve agent's performance.