Heuristic Analysis for Game-Playing Agent

Introduction

The objective of the project is to build an AI agent that can learn to play the game of Isolation. The agent uses heuristic function to search for the next move. Three heuristic functions are tried and the best one is used as the final heuristic function.

Heuristics

Three heuristic functions are used in the tournament to produce score of winning and losing by the agent. The table below shows the performance of agent using each heuristic function

Heuristic	Highest Performance
Heuristic 1	72.9%
Heuristic 2	74.3%
Heuristic 3	64.3%

Highest performance achieved by AB_Improved agent is 68.4%

Heuristic 1 (custom_score_2): my_moves - 2 * opponent_moves + 2 * common_moves

The first heuristic function is created by subtracting twice the moves left by the opponent from number of moves left by the agent and adding twice the number of common moves as the heuristic function, so the agent always uses the move that leads to maximum number of moves left and tries to minimize number of opponent moves.

Heuristic 2 (custom_score_3): my_moves - 4 * opponent_moves + 4 * common_moves

The second heuristic function takes the number of moves left for the agent and subtracts four times number of moves left for the opponent and adds four times the common moves. This function increases the penalty applied to opponent moves left.

Heuristic 3 (custome_score): my_moves - 3 * (opponent_moves) + 3 * common_moves

The third and final heuristic function subtracts three times the opponent moves from the number of moves available to the agent. This leads to the agent maximizing number of his own moves, while putting more emphasis on reducing opponent moves available.

Final Heuristic Selection

Highest performance is achieved by Heuristic function 2, which is higher than baseline AB_Improved performance. This function penalizes more for higher opponent moves, making the agent win more often. Other heuristic functions that used lesser weight to penalize opponent moves, seemed to make performance worse. So, from our experiment heuristic function 2 seems like the best function to use.