

Artificial Intelligence Nanodegree

Project II – Isolation Game-playing Strategies

Following the main Goal of the Game of Isolation, mainly to maximize one own possibilities to move while trying to quickly minimize the opponents moves, we can categorize the Heuristics into two groups.

Aggressive Strategies focus on minimizing the opponents moves, while **Defensive Strategies** focus on maintaining one's own capacity to move across the board.

In this Implementation I explored three heuristics, and measured their outcome on game outcomes.

Heuristic 1: "Number of myMoves"

This Heuristic calculates the difference in number of possible moves available to me, against the number of possible moves available to the opponent, both at state S , and tries to **maximize my advantage**. This metric will favor a greedy gameplay, that tries to maximize possible moves and will try to limit the amount of freedom to the opponent.

It is a balanced, slightly aggressive gameplay strategy.

$$\#myMoves = \# \text{ of own moves } (s) - \# \text{ opponent moves } (s)$$

Heuristic 2: "Number of myMoves - minimizing"

This Heuristic calculates the difference in number of possible moves available to me, against the number of possible moves available to the opponent, both at state S , and tries to **minimize the opponents moves**. This metric will favor a quick and aggressive Gameplay and therefore falls into the category of Aggressive Heuristics.

This is achieved by up-weighting the number of opponents moves against my own, wherein the higher weight forces ever more aggressive Gameplay.

$$\#myMoves = \# \text{ of own moves } (s) - \text{weight} * \# \text{ opponent moves } (s)$$

Heuristic 3: "Centrality index"

This second Heuristic tries to minimize the distance to the central region of the board in choosing next moves, while adhering to an overall strategy of keeping on the topside of possible number of moves overall.

Centrality Index informs the Agent about moves to maximize the number of available moves overall, and will force a more defensive Gameplay.

#centrality_weighted_myMoves = # of own moves (s) * # distance from center – # opponent moves (s)