

Heuristic Analysis

This analysis presents four heuristics for getting better performance than the ID_Improved agent. Each one increases the complexity in terms of computation and implementation.

Let's analyze each one:

heuristic_one **More Moves**

Intuition

The more legal moves a player has, the better the outcome of the game will be.

Implementation

This function return the number of legal moves a player has.

heuristic_two **Difference of Moves**

Intuition

The more moves a player has than his opponent, the better the outcome of the game will be.

Implementation

This function returns the difference of moves a player and his opponent has.

heuristic_three **Uses the players moves and the depth to calculate the score.**

Intuition

The chance of winning the game is high, if the player has more moves than the sum of his opponent's moves and blank spaces at it's current depth.

Implementation

This function returns the difference of moves a player and the sum of his opponent's moves and blank spaces at it's current depth.

heuristic_four **Uses the players moves and the depth to calculate the score**

while forcing the player to choose center positions if they're available.

Intuition

This assumes that the outcome will be better if the player chooses the center positions if they are available to the player.

Implementation

This function returns the difference of moves a player with the center moves and the sum of his opponent's moves with blank spaces at a given depth.

Evaluation

The `tournament.py` script was run for each individual heuristic. The results are presented in the following table:

Heuristic	ID win %	Student win %
heuristic_one	71.43	63.57
heuristic_two	70.00	68.57
heuristic_three	73.57	72.14
heuristic_four	67.86	70.00

Conclusion

As per the results, the `heuristic_four` performs better than the `ID_improvement` agent and all other heuristics. I recommend choosing it when building an agent.

Further tweaking with tuning parameters might lead to better results.