Isolation Game Agent Heuristic Analysis

The game of Isolation, but with players moving as chess knights, is solved by developing an agent incorporating Minimax, Alpha-Beta pruning and Iterative Deepening algorithms. Three custom heuristics are implemented. Each heuristic is tested against other agents/bots with their own heuristics and ID_Improved is used as the benchmark agent.

Note: All pictures of test results can be found in this associated zip.

Table of Contents

Introduction

The heuristics implemented for this version of Isolation are reflecting the opponents moves, attacking the opponent and moving to an area where there are the most open spaces.

Heuristics Overview

Reflection Heuristic

The reflection heuristic was implemented as it was suggested in one of the lesson videos and is strategically a sound heuristic. If the opponent agent is stronger, the reflection agent will make the same moves, essentially making them equal. The heuristic may also force each agent to implement a contingency plan when there are not enough moves to implement a strong heuristic, making the game one of survival.

The reflection heuristic finds the opponent player's position. If the opponent's position reflected by 180 degrees is a valid move for the current player, return $+\infty$ as the score to absolutely prioritize this move. If not, return the current utility.

Note: Further investigation is required to find the correct solution when reflecting the opponent is not possible.

Attack Opponent Heuristic

The attack heuristic was implemented as strategically the agent is completely limiting the opponent's moves, while using a minimal amount of its own. Theoretically, this would diminish the number of moves for the opponent at a greater rate than that of the agent's own moves.

The attack heuristic finds the opponent player's moves. For every move in the opponent's moves that also exists in the current player's moves, increase the score to be returned. The base score is the current utility.

Note: Further investigation is required to find the correct base score.

Survival Heuristic

The survival heuristic was implemented as the agent will always move to an area with the most open spaces, increasing the number of future moves the agent will have. Theoretically, this would allow the agent to always have more moves than the opponent.

The survival heuristic finds the area with the most open spaces. For each side of the board (segregated by the board mid-point), the number of taken spaces is calculated. The function returns the minimum taken spaces count.

Note: Further investigation is needed to determine the correct logic for returning the side with maximum open spaces (minimum taken spaces).

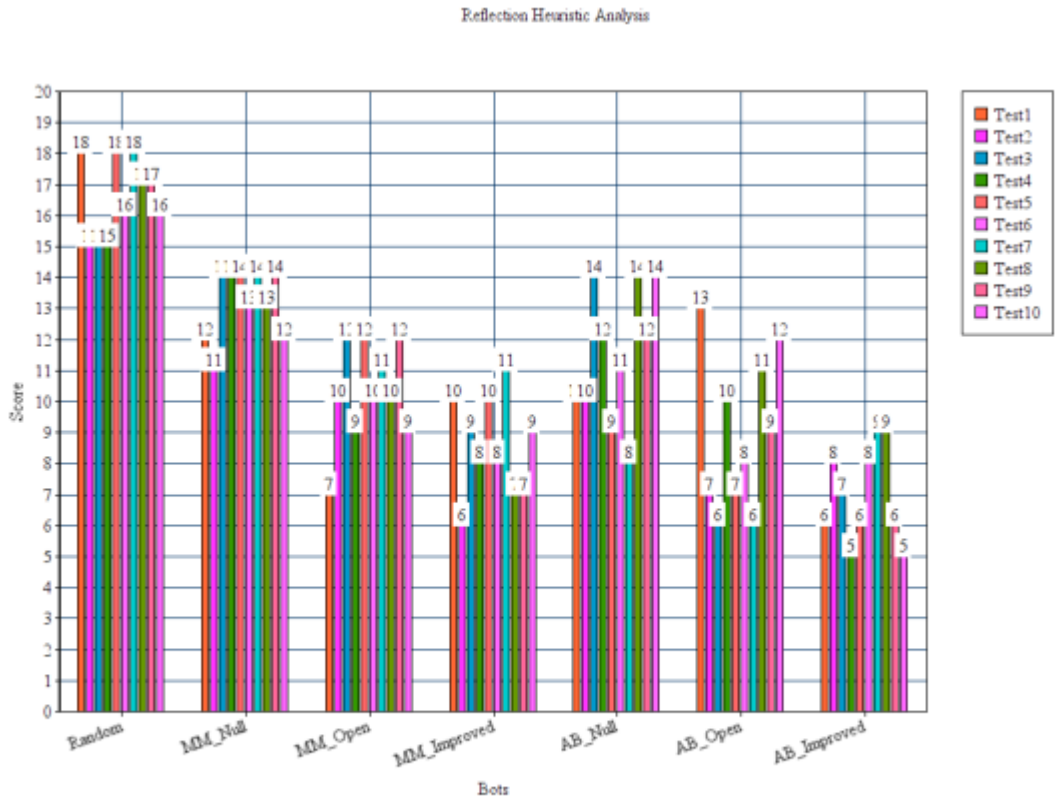
Analysis of Heuristics

Reflection Heuristic

Results

Table of Winning percentage for Reflection Heuristic against ID_Improved

Test	ID_Improved (%)	Reflection Heuristic (%)
1	64,29	54,29
2	61,43	47,86
3	62,14	55,00
4	60,71	52,14
5	69,29	54,29
6	58,57	52,86
7	59,29	55,00
8	60,71	57,86
9	63,57	55,00
10	55,71	55,00
Average	61,57	53,93
Standard Deviation	3,67	2,61



Analysis

It is clear that the reflection heuristic is not as strong as ID_Improved, but still wins more than half the games. If the difference in averages is less than 8%, then a few improvements could easily allow this heuristic to beat ID_Improved. The Standard deviation of the reflection heuristic is statistically significantly lower, by more than 1%. This suggests the reflection heuristic is more reliable than ID_Improved. If further investigation could slightly improve the reflection heuristic, it may be able to consistently beat ID_Improved.

The reflection heuristic consistently beats most algorithms majority of the time, but starts to match MM_Improved and more often loses against AB_Improved. It can be safely stated that, with a few improvements, the reflection heuristic could beat the Improved heuristics.

It is recommended this heuristic be investigated further.

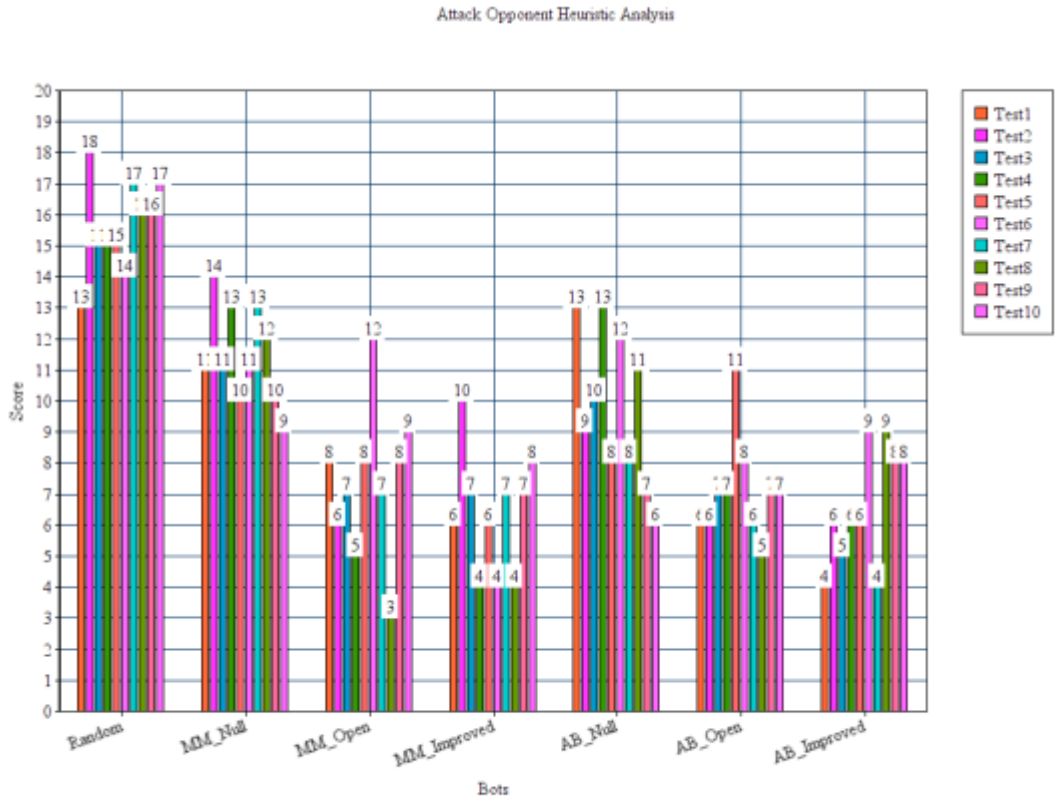
Attack Heuristic

Results

Table of Winning percentage for Attack Heuristic against ID_Improved

Test	ID_Improved (%)	Attack Heuristic (%)
1	53,57	43,57
2	61,43	49,29
3	57,86	44,29

4	57,86	45,00
5	57,14	45,71
6	66,43	50,00
7	60,00	44,29
8	54,29	42,86
9	57,14	45,00
10	62,14	45,71
Average	58,79	45,57
Standard Deviation	3,84	2,33



Analysis

The Attack heuristic performs poorly against ID_Improved. The heuristic will need many improvements to match ID_Improved. Most likely a contingency plan is required when the player has many more moves than the opponent. However, the standard deviation is statistically far lower than ID_Improved, thus this heuristic is more reliable than ID_Improved and is worth investigating further as it has the potential to beat ID_Improved.

As can be seen in the graph above, the attack heuristic starts to lose, majority of the tests, against the Open heuristic bots and performs poorly against the Improved heuristic bots. There are also a few outliers for most of the tests. This suggests that much improvement is required for this heuristic to compete with the better heuristics.

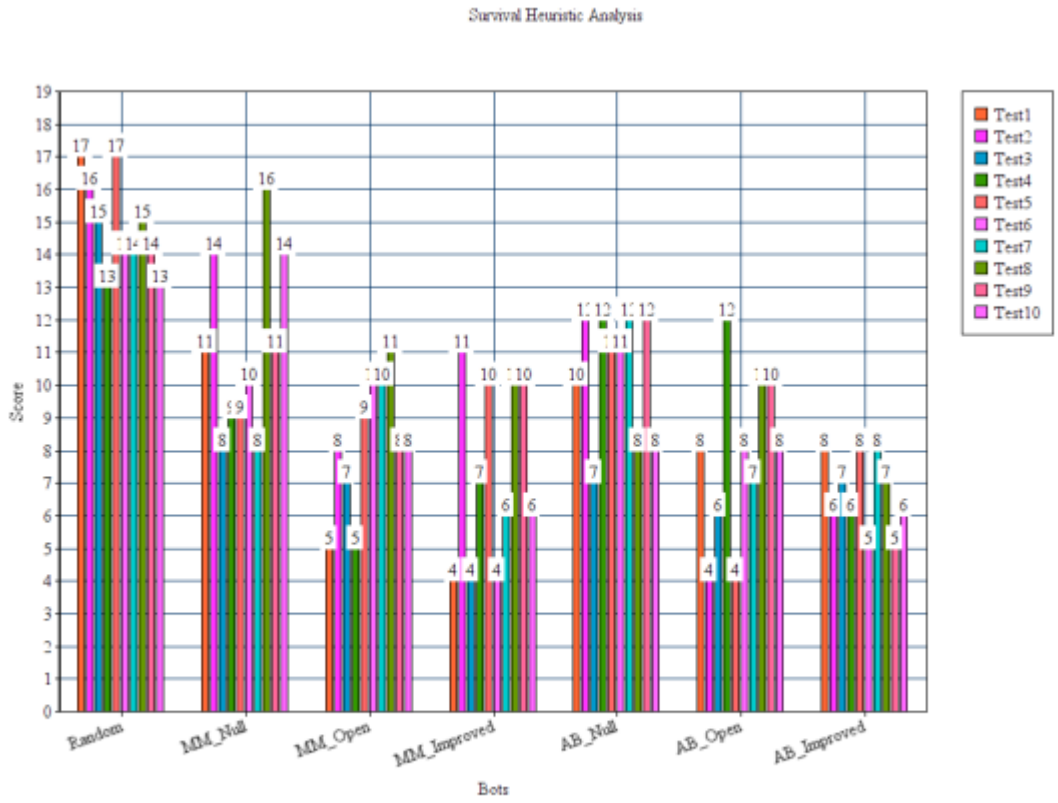
It is suggested this heuristic be improved after the reflection heuristic.

Survival Heuristic Analysis

Results

Table of Winning percentage for Survival Heuristic against ID_Improved

Test	ID_Improved (%)	Survival Heuristic (%)
1	61,43	45,00
2	65,71	50,71
3	60,71	38,57
4	56,43	45,71
5	58,57	48,57
6	55,71	44,29
7	53,57	46,43
8	57,86	55,00
9	54,29	50,00
10	61,43	45,00
Average	58,57	46,93
Standard Deviation	3,79	4,44



Analysis

The Survival heuristic performs poorly against ID_Improved. As noted above, changing the logic as to which section is chosen may improve this heuristic. However, the standard deviation is greater than that of ID_Improved, suggesting it is less reliable than ID_Improved and the other two heuristics. This suggests that this type of heuristic is unreliable and requires a fallback. This heuristic should be investigated last if the other two do not show improvements. It may be beneficial to consider a different heuristic entirely.

Similar to the attack heuristic, the survival heuristic starts to perform poorly against the Open heuristic bots. However, the results are also scattered, suggesting the heuristic is more unreliable than the attack heuristic. Vast improvements are needed to increase the winning rate of this heuristic.

It is suggested that this heuristic have a low priority for improvement or be replaced with an entirely different heuristic.

Conclusion

The above results and analysis conclude that, in the heuristics' current states, the reflection heuristic should be used, thus the reflection heuristic is implemented in the current solution. However, improvements must be made to the heuristic and in the case there are no improvements in the results, the attack heuristic should be reconsidered.

Note: All pictures of test results can be found in this associated zip.