Heuristic Analysis for Isolation AI agent

Summary

Below I describe the three heuristics that were tried in building an effective AI agent for playing the game of isolation. The goal of the heuristics was to beat ID_Improved agent by winning as many games as possible against the test agents in tournament.py. In addition, each heuristic was compared against ID_Improved in head to head matches to make sure it outperforms it in addition of the test agents.

Heuristics

ID_Improved uses quite an effective heuristic where a board position is given a score by calculating the open moves left for the player and subtracting the open moves left for the opponent. It is a very good heuristic which I decided to augment for my submission and came up with three variants. Each variant calculates a score that is then added to the ID_Improved used heuristic to make sure that given equal choices, we prefer the more advantageous positions. Here are the heuristics that were analysed:

“Stay away from the walls” - calculates a score from 0-1 which is a normalized distance from the center. I normalized it to be between 0 and 1 so that it would not overemphasize the distance to the center but instead will return a small enough number that can be used as a tiebreaker. The score is added to the open moves heuristic. Thus, the open moves still dominates the score, but this heuristic makes AI agent to prefer center positions from where the knight has the most space to operate.

“Stay close to open fields” - as the board gets taken up by the moves there are less and less space for making the moves. This heuristic makes sure that the knight does not trap itself in positions with less space around it. This heuristic makes the agent stay around to the clusters of open squares. I do this by calculating the distance of the player’s position to all other open squares on the board (sum(sqrt(dr^2 - dc^2)) - where dr = difference between rows of open square and the player’s row, dc = difference between columns of open square and the player’s column, all summed together). Then that distance is divided by the max distance of a single board from all the pieces (to make sure it is a number < 1) and subtracted from 1 to make sure that the smallest distances are preferred. Then this gets added to the open moves heuristic to come up with a final number.

Combination of stay away from the walls / stay close to open fields - combine the above into a single heuristic where one will avoid the edges and stick close to open fields.

Heuristic Function Evaluation and Comparison

To evaluate the performance of each heuristic function, 400 game simulations were run. We want to run a larger number than the standard 20 games in tournament.py to make sure that there is less variation in the results due to the randomness.

Stay away from the walls heuristic evaluation

Fig 1 shows how ID_Imroved and Student agent fared against all the other test agents in 400 game tournament.

	ID_Improved	Student
Random	388	382
MM_Null	364	376
MM_Open	270	298
MM_Improved	264	271
AB_Null	346	338
AB_Open	250	252
AB_Improved	226	232
winning %	**75.29**	**76.75**

Fig 1. Number of wins for ID_Improvied and Student agent using “stay away from the walls” heuristic against test agents out of 400 games.

Pretty close to ID_Improved level, seems like an improvements but not by much, >1% improvement over 400 games.

When this heuristic was placed to run against ID_Improved (instead of just test agents) using the same tournament.py with some minor code tweaks to run just these two agents against each other. Fig 2 shows the results of 400 and 600 game trials.

	ID_Improved	Student	% Student wins
400 game tourney wins	195	205	51.25
600 game tourney wins	277	323	53.83

Fig 2. Number of ID_Improved and Student with “stay away from the walls” heuristic wins head to head against each other. Student wins slightly.

Running two tournaments seemed to hold up the advantage of the student agent’s heuristic over the ID_Improved.

Stay close to open fields heuristic evaluation

Fig 3 shows the performance of this heuristic against other agents, together with ID_Improved runs for comparison.

	ID_Improved	Student
Random	385	389
MM_Null	371	368
MM_Open	272	309
MM_Improved	264	298
AB_Null	329	340
AB_Open	240	265
AB_Improved	230	266
winning %	**74.68**	**79.82**

Fig 3. Student heuristic that prefers staying close to open fields

This heuristic performed noticeably better than ID_Improved. Only 2 less wins against MM_Null and everywhere else it either slightly outperformed ID_Improved, or outperformed by a much wider margin (e.g. 9% better against AB_Improved). Overall winning percentage for ID_Improved stayed around 75% as before, but Student agent moved up to almost 80%.

Stay close to open fields was pitted against ID_Improved in a sequence of matches to see how well it performs there. The results are in Fig 4.

	ID_Improved	Student	% Student wins
400 game tourney wins	162	238	59.50%

Fig 4. ID_Improved head to head against stay close to open fields heuristic.

Stay close to open fields and stay away from the walls combined heuristic

Lastly we check our 3rd heuristic, which is a combination of the two above. Fig 5. Shows the results of the runs against the test agents.

	ID_Improved	Student
Random	379	387
MM_Null	371	365
MM_Open	280	302
MM_Improved	272	297
AB_Null	332	354
AB_Open	256	259
AB_Improved	232	243
winning %	**75.79**	**78.82**

Fig 5. ID_Improved vs student with combined heuristic

Again Student agent seems to have done better than ID_Improved, but none of the results stand out when compared to the “stay close to open fields” heuristic. It is better than “stay away from the walls” by a few percentage points. Also good to see ID_Improved hanging around 75% consistently between the simulations.

Fig 6. Shows the results of running ID_Improved against the combined heuristic.

	ID_Improved	Student	% Student wins
400 game tourney wins	192	207	51.75%

Fig 6. Combined heuristic and its slight advantage over ID_Improved

The advantage seems to be there but not a significantly high number like it was with “stay close to open fields”.

Conclusion

From all the heuristics measured, “stay close to open fields” appears to be a clear winner. It has the best scores against the test agents, consistently outperforming them >66% of the time. It also did the best when compared against ID_Improved. It seems like the cost to calculate that heuristic is not prohibitive in that it does not decrease the depth of iterations when measured. When doing a more verbose logging and outputting the depths reached on opening moves by various heuristics, they seemed to stay at the same level for all. Math operations it employes are pretty simple squares and square roots. As the game progress, the number of such calculations to be done is decreasing as well (less open squares). “Stay close to open fields” is my recommended heuristic out of all the options considered.