Karol Misiarz

11.01.2018

Heuristic Analysis Isolation AIND

This document describes heuristic for Isolation game developed for Udacity course “Artificial Intelligence Nanodegree”

First Custom Score (H2)

The winning move of Isolation game is selection of the unoccupied field when the opponent hasn’t got any moves left. It seems important to track the difference between amount of moves our player has in relation to our opponents.

First custom scores that is proposed has calculates that difference. By subtracting number of possible moves of our opponent from number of our possible moves we have a way to get the first heuristic. We can use different linear combinations of both of those values by changing beta multiplier:

```
H1 = number of our moves – beta * number of opponent’s moves
```

Second Custom Score (H3)

Like chess, isolation game is about position. Amount of possible moves are drastically different in the middle of the board from near the wall and close to the edges.

The second heuristic is based on position on the board. Function that calculates this value should give low score for the edges and walls, but

should be rather flat in the centre area. Function proposed is the z value of the half sphere that is placed on the board. Diameter of this sphere is the same as the dimension of the board. Function graph is presented in Figure 1.

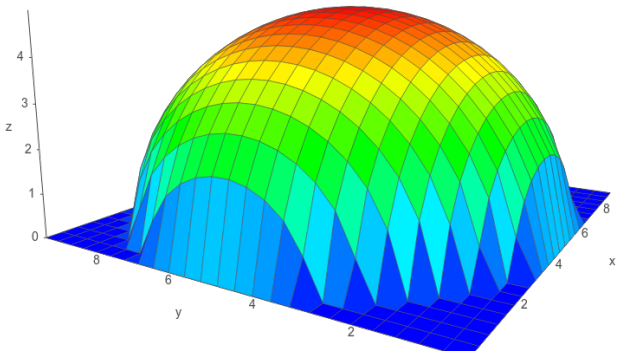

Figure 1: Plot of custom score 2

This way, the values for the middle positions have similar score value, and values in in the corners and close to the walls are much smaller. Gamma multiplier is used to change the radius of the sphere.

```
r = (height + width) / 4
distance from centre = sqrt((loc x - r)^2 + (loc y - r)^2)
if distance from centre <= gamma * r:
     H2 = sqrt((gamma * r)^2 - distance from centre^2)
else:
     H2 = 0
```

Custom Score Combined (H1)

In the beginning of the game, it's important to occupy the middle positions, as it's very hard to block the opponent. As the game progresses, we want to keep track of the difference between the moves more than about the static position on the board. Progress is in range (0-1) and the ration between available spaces to number of all the spaces on the board. Alpha value gives additional linear combination between H2 and H3.

```
progress = blank spaces left / number of all spaces
H1 = progress * H2 + alpha * (1 - progress) * H3
```

Retults

For alpha (Custom 1) = 1, beta (Custom 2)= 1, gamma (Custom 3)= 1:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	10	0	9	1	10	0	10	0
2	MM_Open	8	2	9	1	9	1	7	3
3	MM_Center	8	2	10	0	8	2	7	3
4	MM_Improved	7	3	8	2	9	1	7	3
5	AB_Open	7	3	7	3	6	4	5	5
6	AB_Center	7	3	6	4	5	5	7	3
7	AB_Improved	6	4	5	5	6	4	5	5
	Win Rate:	71.7%		75.0%		71.7%		63.3%	

For alpha = 1, beta = 0.8, gamma = 2:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	9	1	10	0	10	0
2	MM_Open	7	3	8	2	8	2	7	3
3	MM_Center	10	0	10	0	9	1	10	0
4	MM_Improved	8	2	7	3	4	6	9	1
5	AB_Open	6	4	3	7	6	4	6	4
6	AB_Center	5	5	8	2	5	5	6	4
7	AB_Improved	5	5	6	4	4	6	5	5
	Win Rate:	71.4%		72.9%		65.7%		75.7%	

For alpha = 1, beta = 1.2, gamma = 2:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	10	0	9	1	8	2	10	0
2	MM_Open	8	2	8	2	8	2	8	2
3	MM_Center	9	1	10	0	10	0	9	1
4	MM_Improved	8	2	9	1	7	3	8	2
5	AB_Open	6	4	6	4	6	4	5	5
6	AB_Center	7	3	5	5	5	5	7	3
7	AB_Improved	4	6	6	4	5	5	4	6
	Win Rate:	74.3%		75.7%		70.0%		72.9%	

Summary

The best results have been obtained by the combined score (H1). It has outperformed two other scores. It is easy to see that some kind of linear combination between both H2 and H3 should bring the best results as both H2 and H3 are bringing positive results winning against other algorithms.

In the isolation game it is important to control the difference in the number legal moves of our and our opponent's piece. It is also important to control the position of our knight on the board. By combining those two functions we can get more universal method.

As the game progresses the heuristic changes, giving bigger weight to the difference between available moves and less for the position on the board. It's important to notice that at the late game stage, we should consider agressively blocking our opponent instead defensively choosing positions with more spaces available.

The heuristic proposed is a good start for the position evaluation, it is easy to understand and compute. Custom score 1 includes both of the players in the calculation. Custom score 2 gives a static field of values, that can be initialized within IsolationPlayer parent class. However,

because the chosen constants are not optimised, more optimisation or more complex strategies should be invented either with the use of master human characteristic of the winning strategy or by using neural network and reinforcement learning in self-play to learn weights. Those weights can then be used as a custom score. The other way would be to visualise strategies ‘invented’ by reinforcement learning and using them to create better heuristics.