Heuristic analysis of custom scoring function in 'Isolation' game

Sagarnil Das

10/30/2017

In the development of the Isolation game playing agent, I have developed 3 scoring functions. They fundamentally carried the same concept as the "improved" score function (#my_moves - #their_moves) but with some modifications.

1. **The Aggressive**: This one is inspired by the class lectures. Here the evaluation function puts a larger and larger multiplier on #their_moves in the improved score function. Later in the game, the agent tries to reduce the number of moves opponent has left in respect to the number of moves the agent has remaining.

2. **Move-steal**: The move-steal looks at whether the agent and the opponent has any overlap between the available moves to both of them. If there is an overlap and it's the agent's turn, it rewards the agent with a +1 point and if it is the opponent's turn, it penalizes the agent with a -1 point. The basic intention behind this algorithm is for the agent to steal any moves from the opponent player, thus reducing the number of moves available to the opponent player.

3. **Centralness**: This heuristic function gives a bonus to our agent if the agent is within the center region of the board and also gives a bonus if the opponent is outside the center region of the board.

Initially in the tournament .py, the num_matches was 5. But in my opinion, this was too small a sample size. So I made the num_matches to 50 and instead of playing against every opponent, I only kept the AB_improved opponent agent. This was the result I got:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	AB_Improved	59	41	50	50	58	42	58	42
	Win Rate:	59.0%		50.0%		58.0%		58.0%	

Where AB_Custom: **The Aggressive**

AB_Custom_2: **Move Steal**

AB_Custom_3: **Centralness**

We see that out of the 3 heuristic functions, the Move-steal and the Centralness performed the best with a winning rate of 58% and Aggressive performed the worst with a win rate of 50%. Both Move-steal and Centralness are almost similar to the performance of AB_Improved. Before running the game, I expected the Move-steal to perform the best and the results are as per my prediction. However I expected the Aggressive to perform better than random chance. Even though the other two algorithms are almost same as AB_improved, there might be some margin of error as AB_improved is playing against itself and it's a 50-50 chance for it. So I think, in order to beat AB_improved with much more precision, it would have to be a combination of stealing opponent's move, capturing the central position and being aggressive later in the game.

In my opinion, the Steal-Move is the most promising algorithm between these 3 because:

A) Strategically, being able to move to overlapping spaces to reduce the number of moves available to the opponent should really help the agent to win games.
B) The ability of this algorithm to find the overlapping moves is similar to the agent being able to look at a ply deeper, since it is considering where it and the opponent will be able to move in evaluated game state is sort of like looking ahead.