Artificial Intelligence Nanodegree, Udacity

12 February 2017

I approached the heuristics from the hypothesis that depth trumps an intricate heuristics scoring function early in the game when there are many more possible game states that can be reached by the two agents playing against each other.

The following heuristics were created and evaluated:

1. *Action mobility*: compute the ratio of available moves for an agent one step ahead compared to the current game state. Larger is better.
2. *Action focus*: 1 - action mobility. Tries to minimize opposing agent's open choices one game state ahead.
3. *Open moves ahead*: Compute an agent's legal move advantage one game state ahead compared to opposing agent.
4. *Super duper Hugo Heuristic*: A variant on the difference in number of legal moves between agents in the current game state. The difference is that it weighs the opponent's legal moves higher than the agent's own moves. Finally, it is scaled with the number of open spaces remaining on the board to account for how many spaces are closed off following a specific move.

Please see heuristics.py for the implementations.

To evaluate the hypothesis, I compared the results in the tournament between a game agent that only had access to the difference in moves between agents versus one that used the same heuristic early in the game, and switched to a more sophisticated heuristic later in the game.

***Table 1**. Match comparison. Left: agent that switches heuristic later in the game (improved to open moves ahead). Switches at 50% filled spaces. Right: agent that stays with the same heuristic throughout the entire game (improved).*

Opponent type	Student Wins	Student Losses
Random	32	8
MM_Null	28	12
MM_Open	24	16
MM_Improved	22	18
AB_Null	26	14
AB_Open	28	12
AB_Improved	23	17
Total	65.36% Student wins	

Opponent type	Student Wins	Student Losses
Random	32	8
MM_Null	26	14
MM_Open	26	14
MM_Improved	17	23
AB_Null	23	17
AB_Open	29	11
AB_Improved	26	14
Total	63.93 % Student wins	

I increased the number of matches played between the opposing agents to remove some of the random bias that will otherwise be more apparent if we play few games, but had to keep the total to 40 matches per opponent type due to the old hardware that I'm running. Table 1 indicates that the agent that switches heuristic (from improved to the open moves ahead heuristic) performs on average equally well to the agent that doesn't switch heuristic when the heuristic is switches early in the game (when 25% of the board is filled).

I decided to see if my hypothesis could be more strongly validated by moving the point at which the agent switches heuristic, as well as which end-game heuristic to use. The results are shown in Table 2, and indicate that the choice of end-game heuristic, and when to switch heuristic, has a positive impact on the ability of an agent to win games of isolation. We can also see that moving the point at which the agent switches heuristic improves the number of wins by 5 percentage points (comparing the left sides of tables 1 and 2).

Table 2. *Match comparison. Left: agent that switches heuristic later in the game (improved to super duper Hugo heuristic). Switches at 50% filled spaces. Right: agent that stays with the same heuristic throughout the entire game (improved).*

Opponent type	Student Wins	Student Losses
Random	34	6
MM_Null	34	6
MM_Open	26	14
MM_Improved	20	20
AB_Null	31	9
AB_Open	28	12
AB_Improved	24	16
Total	70.36% Student wins	

Opponent type	Student Wins	Student Losses
Random	31	9
MM_Null	28	12
MM_Open	18	22
MM_Improved	20	20
AB_Null	30	10
AB_Open	26	14
AB_Improved	23	17
Total	62.86% Student wins	

When comparing the heuristic-switching game agent from Table 1 (Left) to the ID_Improved agent in the tournament, it consistently outperforms the ID_Improved agent on my hardware by an average of 6.43 percentage points. This can be viewed in table 3.

Table 3. *Match comparison. Left: agent that switches heuristic later in the game (improved to super duper Hugo heuristic). Right: agent that stays with the same heuristic throughout the entire game (improved).*

Opponent type	Student Wins	Student Losses
Random	34	6
MM_Null	34	6
MM_Open	26	14
MM_Improved	20	20
AB_Null	31	9
AB_Open	28	12
AB_Improved	24	16
Total	70.36% Student wins	

Opponent type	ID Wins	ID Losses
Random	32	8
MM_Null	29	11
MM_Open	23	17
MM_Improved	19	21
AB_Null	31	9
AB_Open	22	18
AB_Improved	23	17
Total	63.93% ID_Improved wins	

It is clear that the ability of the agent to evaluate as many game states as possible early in the game is advantageous to simply having a more elaborate heuristic function, and that penalizing the student agent by weighing the opposing agent's number of legal moves higher later in the game forces the student agent to move away from game states where the weighted difference leads to a very small or negative score.

I experimented somewhat with a number of other heuristics, such as the ratio of available moves on step ahead (given that the agent picks the move that leads to the highest number of available moves in the next game state) to the number of legal moves in the current game state, but found very little correlation with the agent's ability to win against the other opponents in the tournament. I also found that trying to minimize the opposing agent's choices, rather than maximizing the student agent's own available moves, also performed worse compared to the ID_Improved heuristic and the strategy to switch heuristic at mid-game.

Therefore, the heuristic scoring function submission to Udacity will be one that consists of a combination of two scoring functions, one for the first half of the game, and another for the second half, in order to find a good balance between the necessity of searching deep early in the game, and 'wisely' evaluating the right move later in the game (as one misstep is relatively more costly in late-game compared to early in the game).