Christopher Verceles

Game Playing Agent

Heuristics Analysis

After several rounds of experimenting with custom evaluation functions and playing Isolation against some friends, I came up with 3 heuristics; the first that beat AB_Improved up to 90% of the time, the second heuristic that beats AB_Improved 40-60% of the time and the last heuristic that won 50-80% of games all out of 10 rounds each. I decided to limit these heuristics to simple calculations to reduce the possibility of timeouts, which was kept at 250ms.

Upon observing the AB_Improved player performance, it seemed to me that a more consistent win result could be achieved by combining the positive aspects of the open, center and improved heuristics. I implemented two versions of what I call the WeightedAverageScore. For this heuristic, I explored different levels of bias multipliers (1, 2 and 3) towards the Improved score discussed in class while still taking into account the center score and open score heuristics. However, after running 10 rounds each using a bias value of 1, 2 and 3, I noticed that a bias around the value of 2 resulted in better performance compared to AB_Improved and no bias at all or a bias greater than 2 seemingly decreased its performance;

$$WeightedAverageScore = (Bias * (OwnMoves - 2 * OppMoves) + CenterScore + OpenScore)/3$$

I arrived at the second heuristic, called OppositeImprovedScore by playing Isolation and running away from opponents who chased me and tried to limit my movements rather than the other way around, which is what AB_Improved would try to do. The OppositeImprovedScore simply returns an ImprovedScore negation and cause the agent to avoid ImprovedScore moves;

$$OppositeImproved = (OwnMoves - OppMoves) * -1$$

Finally, the last heuristic called ImprovedLessOpponentsScore was designed to choose moves where the opponents possible moves were less while still attempting to block the opponents moves and keep your own moves open at the same time. My attempt at implementing this behavior resulted in the average of LessenOpponentMovesScore and ImprovedScore;

$$ImprovedLessOpponentsScore = (ImprovedScore - OpponentsMoves)/2$$

The resulting tables on the next page shows the performance of these 3 custom heuristics running a total of 10 rounds with AB_Improved. Additional testing rounds were performed for each of the different bias levels. All testing rounds ran at a timeout value of 250ms on a 2.7 GHz Intel Core i5 processor (Macbook Pro 2015).

AB_Improved vs Custom: WeightedAverageScore Bias = 3

	AB_Improved Win %	AB_WeightedAverageScore Win %	AB_OppositeImproved Win %	AB_ImprovedLessOpponentsScore Win %
Round 1	45.7	54.3	48.6	48.6
Round 2	51.4	57.1	48.6	45.7
Round 3	32.9	40.0	50.0	55.7
Round 4	37.1	52.9	54.3	51.4
Round 5	45.7	57.1	48.6	51.4
Round 6	48.6	51.4	47.1	55.7
Round 7	45.7	67.1	44.3	50.0
Round 8	51.4	48.6	51.4	54.3
Round 9	41.4	40.0	45.7	40.0
Round 10	44.3	51.4	48.6	48.6
Round Wins vs AB_Improved		8/10	6/10	8/10

AB_Improved vs Custom: WeightedAverageScore Bias = 2

Round	AB_Improved	AB_WeightedAverageScore Win %	AB_OppositeImproved Win %	AB_ImprovedLessOpponentsScore Win %
1	47.1	48.6	34.3	48.6
2	51.4	54.3	45.7	47.1
3	47.1	57.1	50.0	50.0
4	58.6	48.6	45.7	48.6
5	47.1	57.1	48.6	47.1
6	50.0	51.4	45.7	51.4
7	51.4	54.3	47.1	41.4
8	47.1	47.1	51.4	50.0
9	45.7	51.4	52.9	42.9
10	50.0	58.6	48.6	48.6
Round Wins vs AB_Improved		9/10	4/10	5/10

AB_Improved vs Custom: WeightedAverageScore Bias = 1

Round	AB_Improved	AB_WeightedAverageScore Win %	AB_OppositeImproved Win %	AB_ImprovedLessOpponentsScore Win %
1	41.4	54.3	55.7	45.7
2	55.7	48.6	42.9	51.4
3	50	60	44.3	51.4
4	45.7	44.3	51.4	41.4
5	45.7	64.3	51.4	60
6	45.7	45.7	45.7	45.7
7	47.1	51.4	42.9	47.1
8	41.4	44.3	40	44.3
9	47.1	45.7	42.9	52.9
10	42.9	41.4	41.4	44.3
Round Wins vs AB_Improved		6/10	4/10	8/10

It can be reasoned that whenever 2*ImprovedScore returns a small enough evaluation result, it starts listening to any “recommendations” from the open score and center score heuristics to get a “2nd opinion”. Based on the data above, this combination of heuristics with a bias towards Improved score consistently returns a win against AB_Improved. It isn't very inventive, but it does work.

In conclusion, it seems that my WeightedAverageScore wins the most number of rounds against AB_Improved when its’ bias towards the improved score is set to 2. Setting the bias to 1, thus removing the bias, seems to decrease its performance against AB_Improved. On the other hand, it still wins 8/10 rounds against AB_Improved with a bias of 3, but we're aiming for the most consistent win result which happens at bias = 3. OppositeImproved and ImprovedLessOpponentsScore don't seem to be consistent enough to choose as my best heuristic.