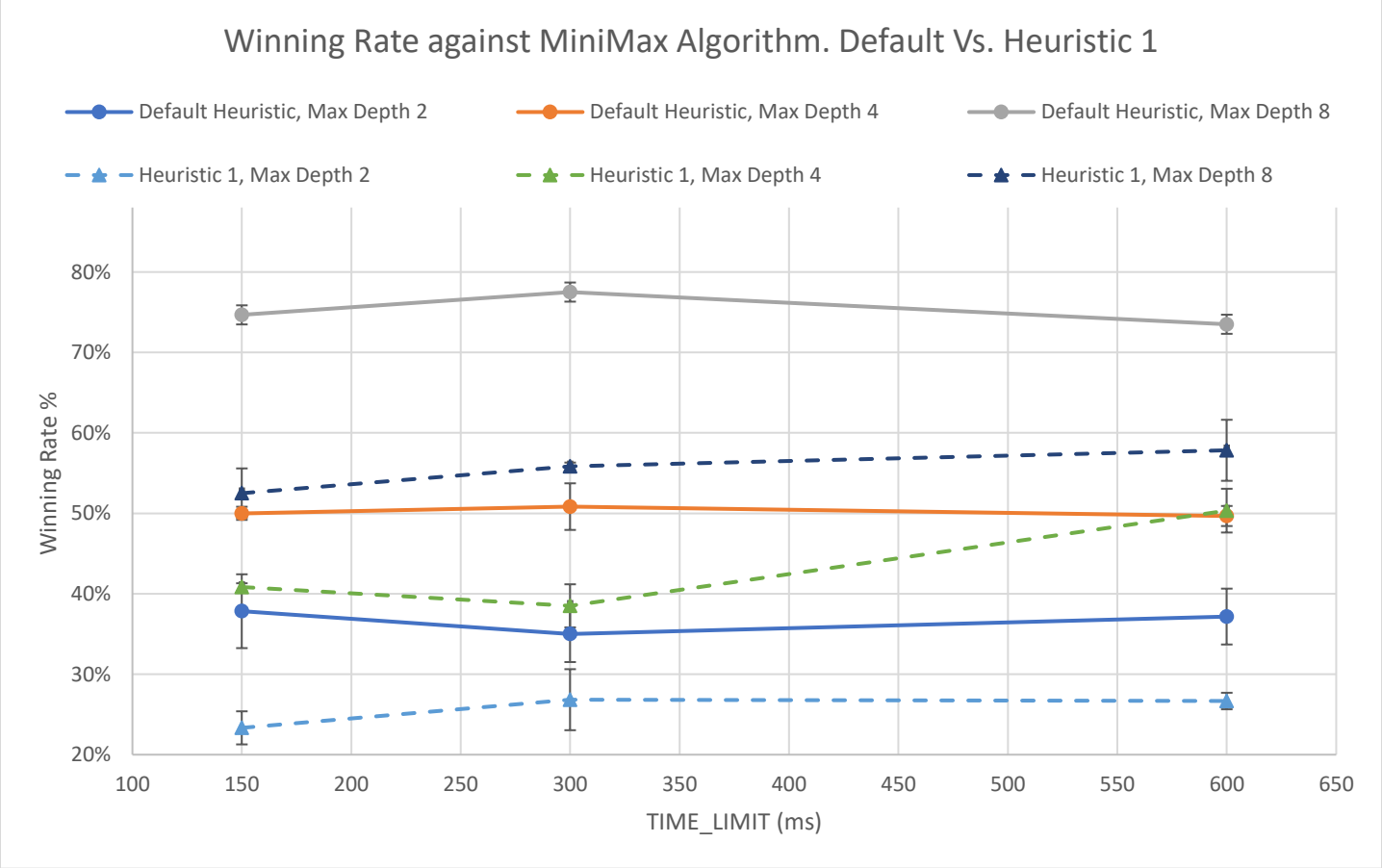
Report: Build an Adversarial Game Playing Agent

Juan Kikushima

For this project I chose the Option 1 experiment: Develop a custom heuristic. The heuristic I came up is a space heuristic which calculates the Manhattan distance from each of the players' locations to the center of the board, which is located on cell 57. The positions closest to the center have a positional advantage, as being closer to the walls decreases the options for moves. The heuristic returns the difference between the calculated Manhattan distance of the opponent and our player. If the result is positive the Manhattan distance from the opponent's positions is larger than our player, which means our player's position has a positional advantage.

I have created a performance baseline for the implemented alpha-beta pruning with iterative deepening algorithm using the default score which consists of the difference between the opponent's and our player's liberties. I made several tests with varying maximum depths and time limits, for 50 rounds, which means 200 matches in total per test all of them with the "fair_matches" flag. As the tests results fluctuate I decided to run 3 tests per case and averaging the resulting winning rate. The complete results are shown in table 1, where the Heuristic described above is labeled as "Heuristic 1" and the performance baseline is labeled as "Default". The chart below shows the comparison between Heuristic 1 (dashed lines) and the performance baseline (solid line).

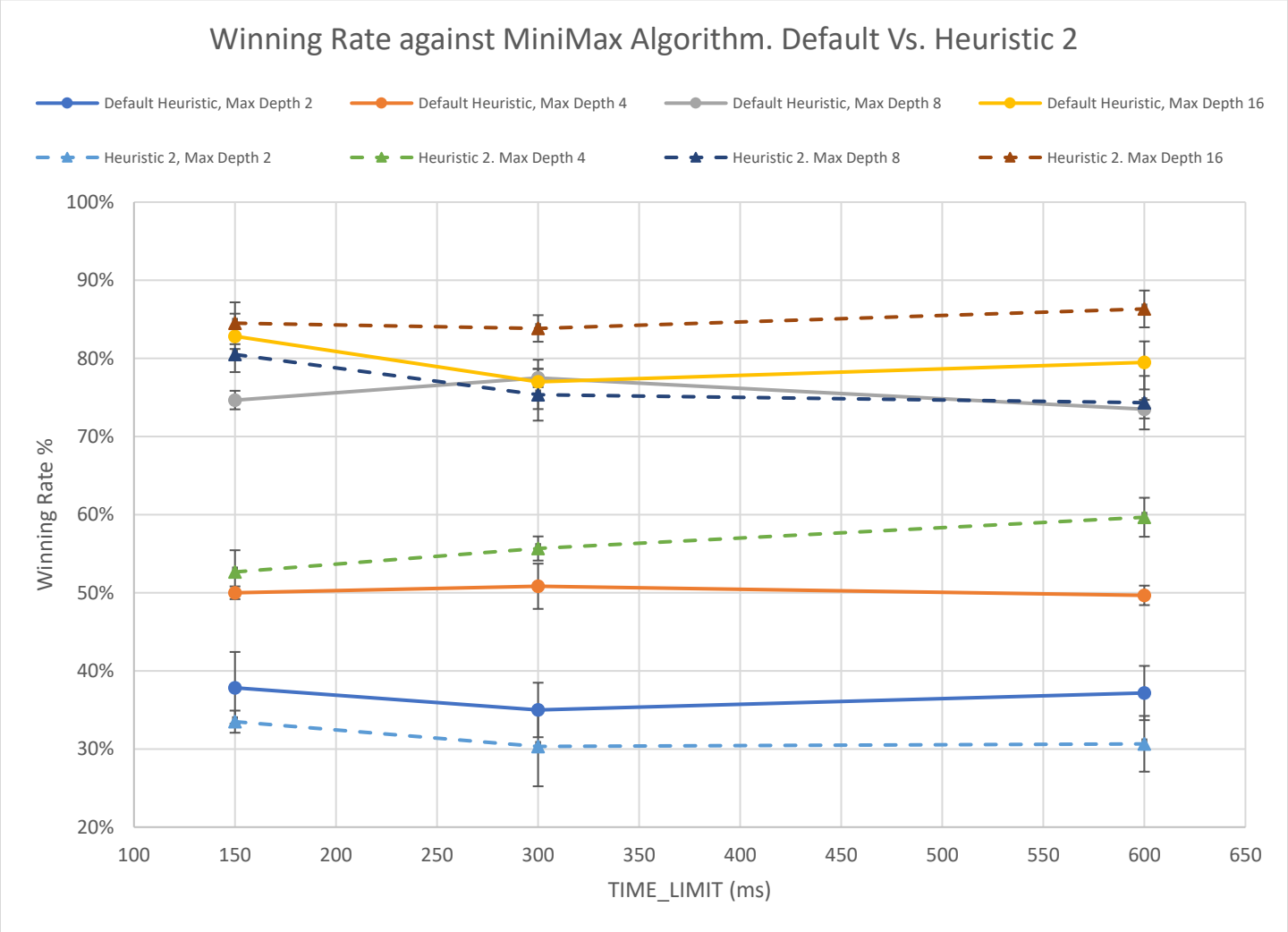


On the chart, we can observe that the Heuristic 1 is not performing better than the default score as all the tests return a lower winning rate than the performance baseline. However, it is evident that increasing both the time limit and the maximum depth for the algorithm increases the winning rate of both algorithms. The maximum depth plays a major role on the increase of the winning rate, which means that given the time, the more nodes the algorithm gets to visit, the better the result it will achieve.

I decided to modify the heuristic to get better results. Heuristic 2 first considers the default score which returns the difference between the number of liberties of our player and the opponent. If the number of Liberties is the same for both

players, the heuristic then calculates the Manhattan distance from each of the players' locations to the center of the board just like in Heuristic 1. If the difference in liberties is not the same, the heuristic calculates a score based on the difference of liberties multiplying the opponents score by 1.5 in order to make the heuristic more aggressive.

The next chart shows positive results for the modifications, as the Heuristic 2 almost always beats the performance baseline. This shows that a good heuristic needs to be creative and mix different characteristics of the game. In the same way as in the previous chart, the winning rate increases when the time limit and more significantly the maximum depth of the algorithm is increased, which again shows that having the algorithm examine more nodes in the allowed time gives better results. For Heuristic 2 I compared up to a maximum depth of 32, and we can see that between a maximum depth of 16 and 32 the increase in winning rate is less significant as the comparison between maximum depth of 8 and 16.



To prove the Heuristic 2 is statistically doing better than the custom score I run a 500 round test for both the baseline and the Heuristic 2, with a maximum depth of 8 and the default time limit of 150 ms. The performance baseline obtained a 68% winning rate, while the Heuristic 2 improved significantly getting a 79% winning rate. I would have liked to make more tests like this, but they took a long time, and they seemed to be consistent with my previous results.

Table 1. Experiment results.

Heuristic	Max Depth	TIME_LIMIT	test 1	test 2	test 3	**Average Winning %**	Standard Deviation
Default	2	150	44%	33%	37%	**38%**	5%
Default	2	300	35%	31%	40%	**35%**	3%
Default	2	600	42%	33%	37%	**37%**	3%
Default	4	150	50%	51%	49%	**50%**	1%
Default	4	300	54%	47%	52%	**51%**	3%
Default	4	600	50%	48%	51%	**50%**	1%
Default	8	150	76%	75%	73%	**75%**	1%
Default	8	300	77%	75%	81%	**78%**	2%
Default	8	600	71%	73%	78%	**74%**	3%
Default	16	150	84%	86%	79%	**83%**	3%
Default	16	300	81%	75%	75%	**77%**	3%
Default	16	600	80%	83%	76%	**80%**	3%
Heuristic 1	2	150	26%	23%	21%	**23%**	2%
Heuristic 1	2	300	32%	26%	23%	**27%**	4%
Heuristic 1	2	600	26%	27%	28%	**27%**	1%
Heuristic 1	4	150	42%	41%	41%	**41%**	0%
Heuristic 1	4	300	39%	35%	42%	**39%**	3%
Heuristic 1	4	600	50%	54%	48%	**50%**	3%
Heuristic 1	8	150	57%	49%	52%	**53%**	3%
Heuristic 1	8	300	56%	56%	57%	**56%**	0%
Heuristic 1	8	600	53%	61%	60%	**58%**	4%
Heuristic 2	2	150	33%	36%	33%	**34%**	1%
Heuristic 2	2	300	31%	37%	24%	**30%**	5%
Heuristic 2	2	600	30%	36%	27%	**31%**	4%
Heuristic 2	4	150	57%	52%	50%	**53%**	3%
Heuristic 2	4	300	57%	54%	57%	**56%**	2%
Heuristic 2	4	600	59%	57%	63%	**60%**	2%
Heuristic 2	8	150	80%	81%	81%	**81%**	1%
Heuristic 2	8	300	71%	76%	79%	**75%**	3%
Heuristic 2	8	600	70%	77%	77%	**74%**	3%
Heuristic 2	16	150	84%	88%	82%	**85%**	3%
Heuristic 2	16	300	82%	86%	85%	**84%**	2%
Heuristic 2	16	600	88%	83%	88%	**86%**	2%