Sampada Sakpal Username: sakpals StudentID: 759512

Assignment 2: Experimentation
AI for 2048 game

Table 1.1: Max-depths tested for Average and Maximum propagation

	Max Depth	Mean score	Score standard deviation	Mean max-tile	Max-tile standard deviation	Average execution time (seconds)
Average propagation	0	1265.2	431.7	128	49.6	0.000
	1	1556	484.3	140.8	38.4	0.000
	2	5877.2	2586.5	486.4	212.6	0.003
	3	8414.8	3545.9	665.6	234.6	0.027
	4	9244	3862.9	691.2	281.6	0.080
	5	10635	3363.5	819.2	250.8	0.296
	6	12090.8	4477.1	819.2	250.8	1.039
Maximum propagation	0	1288.8	461.81	128	49.6	0.000
	1	1347.2	449.3	147.2	57.6	0.000
	2	7077.6	1791.4	537.6	179.2	0.009
	3	8832.4	4183.1	640	262.3	0.023
	4	9934.4	3596	768	256	0.083
	5	11814.4	3962.6	819.2	250.8	0.328
	6	17354.4	6311.8	1280	524.6	1.358

Note: Results are derived from a sample of 10 runs.

Figure 1.1: Average Propagation mean scores for various depths

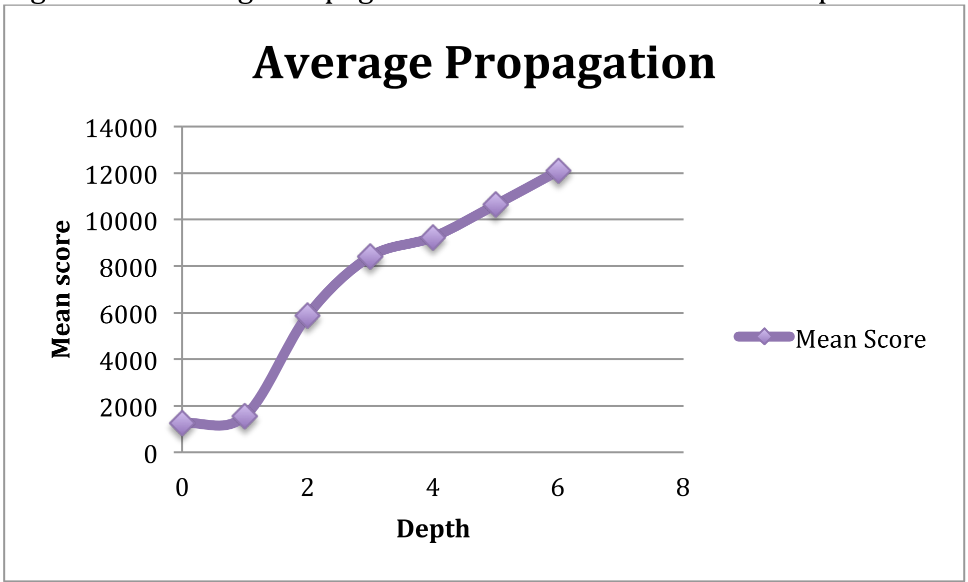
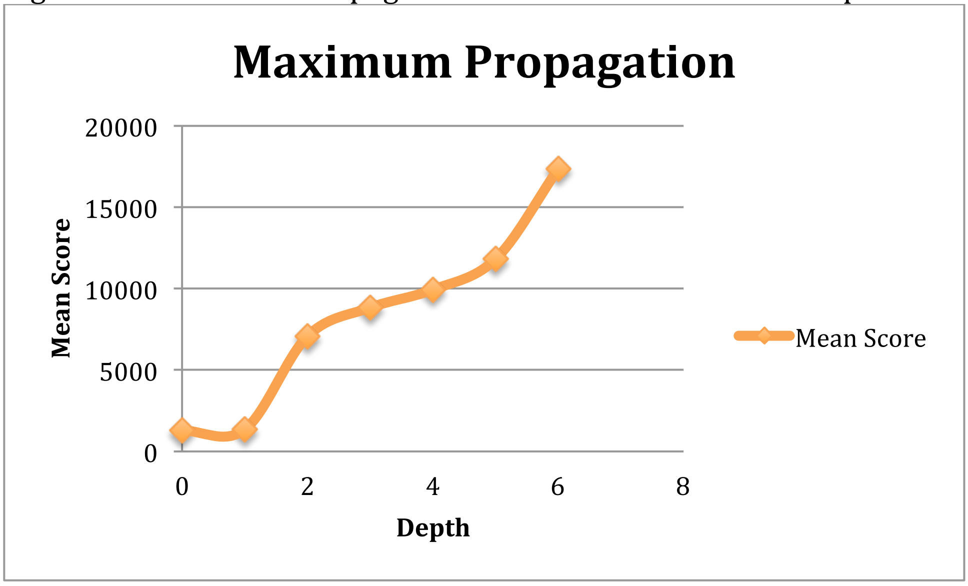


Figure 1.2: Maximum Propagation mean scores for various depths



According to Table 1.1, as the max-depth is increased the mean score also increases for both cases of average and maximum propagation, so it is better to use a max-depth of 6. As the depth increases, the number of nodes generated increases exponentially, allowing the graph to simulate various paths and increase the chances of finding the most rewarding pathway, however the depth should only be extended to a certain limit, taking into account this algorithm's time complexity.
Looking at Table 1.1, the average execution time for both instances of a particular depth begins to increase more dramatically from depth 4 onwards, as the number of generated nodes increases exponentially. For a node that can generate a maximum of four action nodes, as the depth, n, increases, its time complexity can be modelled according to:

$$\frac{4^{n+1}-1}{4-1} \approx O(4^n)$$

Looking at the trends in both Figures 1.1 and 1.2, there is a more significant increase in the mean scores from depth 1 onwards, and only a slight difference in mean scores between depths 0 and 1. This is due to the reason that at depth 0, there are no nodes generated for the four possible actions that can be taken, and so no outcome is simulated to predict the most rewarding graph path. A depth 0 search is as effective as having a random action generated, and as such scores greater than 5000 are difficult to achieve.
At depth 1, though actions are simulated, it is inadequate in predicting the long term yields of scores, since *only* a maximum of four action nodes are generated. It is not much more effective than depth 0.

The standard deviations for the scores are quite large for all cases (ranging from 431.7 to 6311.8). This indicates that the scores are dispersed broadly from the mean, which could be the result of the random nature of generating a tile value at a vacant tile position after each action.

The standard deviation for the max-tile also becomes large as the depth increases indicating that the max tile value varies greatly from the mean. Since the next largest tile can only be created if two tiles with the same value join, the frequency of this occurring increase as the depth of the graph increases, as more nodes are generated and expanded.

This algorithm is a variant of Djikstra's. Deleting nodes from the heap and increasing the scores stored in the auxiliary array, the total complexity of the algorithm can be resolved to $O\big((v+e)\log(v)\big)$ upon reaching the max depth. $(v = number\ of\ game\ states\ generated, and\ e = number\ of\ edges\ (moves))$

Overall, looking at the mean max tile values and scores from Table 1.1, maximum propagation produces a larger score than compared to average propagation and is a more effective implementation of the AI used to solve the 2048 game. As the depths get larger, the difference becomes more apparent, however solving for depth greater than 6, will reduce the efficiency of the solver as the time complexity increases exponentially. So a max depth of 6 is ideal.