Assignment 1 - Intro to AI

Jackson Lee (jcl286), Serena Zeng (sz572), Varsini Dhinakaran (vd299)

February 2023

1 Understanding the methods

1.1

The first move of the agent is to the east rather than the north because, with the known information, there are no blocks due east. This means that the f value of the cell to the east will be smaller than the f value of the cell to the north, as that is the shortest known path between the start state and the goal state calculated by the Manhattan distance. This is due to the fact that the g value is the same for both directions, as they are the same distance from the start cell. However, the h value for the east cell will be smaller than that of the north cell, causing the overall f value to be smaller.

1.2

Call n the number of unblocked cells in the grid world, and m the number of cells that are *reachable* from the start position. Clearly, $m \leq n$, as reachable cells must also be unblocked.

On the ith iteration of $A*$ during repeated $A*$, the number of moves must be $\leq m$. This is because A* finds optimal paths, and will therefore not repeat cells on a path, and there are only m reachable cells. If this path traversal reaches the target, then the search concludes. However, if this path traversal does not reach the target, the path must end at a *dead end cell* - call this cell c_i.

We also know that each iteration of $A*$ must end at a different dead-end cell - that is, c_{i+1} cannot equal any of $c_1, c_2...c_i$. This is because $c_1, c_2...c_i$ have already been explored at this point, and the agent now knows all the blocked cells surrounding $c_1, c_2...c_i$. Therefore, A* will not dead end at one of those cells again.

As there are only m reachable cells, and A* must dead-end at a different cell every time, A* only runs m times at most before dead-ending at every reachable cell. Therefore, after m iterations of A*, every reachable cell will have been explored. If the target is reachable, the target will have therefore been found at this point. If the target is unreachable, the agent will know every blocked cell adjacent to every reachable cell. In this case, the agent will know

that the target is unreachable. As every A* iteration consists of at most m moves, the number of total moves is:

$$m * m \leq n * n = n^2$$

A finite number.

2 The Effects of Ties

50 trials on 50 grid worlds of size 101x101 using repeated forward A*:

	Avg Runtime	Avg Nodes Expanded
Favor Larger g	0.2627	3327.2
Favor Smaller g	1.4324	30828.56

When running the A* algorithm, ties in f value can either be broken in favor of larger g values or smaller g values. If they are broken in favor of smaller g values, a path with a high h value is being pursued, meaning that there is more possibilities for blocks in the path, as the h value is only a heuristic value. In contrast, if the tie is broken in favor of the greater g value, then a path with a smaller h value is being pursued, meaning there is a smaller chance of encountering blocks compared to a larger h value path. This reasoning is supported by the observations drawn from testing both comparisons, where the number of expanded cells and the run time for the program when running the branch in favor of the smaller g value is larger than the one in favor for the larger g value. On average, the tie-breaking in favor of the larger g value is more efficient than the tie-breaking in favor of the smaller g value.

3 Forward vs. Backward

	Avg Runtime	Avg Nodes Expanded
Repeated Forward	0.2627	3327.2
Repeated Backward	1.8700	18773.26

Running repeated backward A* is significantly slower than running repeated forward A*, both in terms of runtime and number of nodes expanded. This can be understood by the fact that much of the time, most of the known obstacles will lie near the start node. This is due to the agent only knowing about obstacles next to nodes it has explored. Therefore, when searching with forward A*, the search tree is quickly pruned by the obstacles. However, when searching with backward A*, the search tree is not pruned until it is close to the target (the target and start nodes having been switched). This explains the dramatic increase in expanded nodes for backward A*.

4 Heuristics in the Adaptive A*

4.1

The Manhattan distance from a node $n = (n_x, n_y)$ to a target $t = (t_x, t_y)$ is:

$$h(n) = |n_x - t_x| + |n_y - t_y|$$

Since the agent is limited to moving in the four cardinal directions, the cost of moving from n to node n' via the action a is given by the Manhattan distance:

$$c(n, a, n') = |n_x - n'_x| + |n_y - n'_y|$$

We can use the triangle inequality, $|x| + |y| \geq |x + y|$, to show that:

$$\begin{aligned} h(n') + c(n, a, n') &= |n'_x - t_x| + |n'_y - t_y| + |n_x - n'_x| + |n_y - n'_y| \\ &\geq |n'_x - t_x + (n_x - n'_x)| + |n'_y - t_y + (n_y - n'_y)| \\ &= |n_x - t_x| + |n_y - t_y| = h(n) \end{aligned}$$

which satisfies consistency.

4.2

We begin with initial h values that are consistent under initial actions costs:

$$h(n) \leq c(n, a, n') + h(n')$$

Let's say the action costs then increase to c_{new}, or stay the same. Then:

$$h(n) \leq c(n, a, n') + h(n') \leq c_{new}(n, a, n') + h(n')$$

As the costs increase or stay the same. Thus, h is still consistent even with costs c_{new}.

We then run one instance of adaptive A*, obtaining values of g and f. As h is consistent, $g(n)$ represents the distance from the start to node n, for n that is expanded during the search. If $dist(n, n')$ is the distance between n and n', by the triangle inequality for distances, we have:

$$g(n') \leq g(n) + dist(n, n') \leq g(n) + c_{new}(n, a, n')$$

As the distance between n and n' is at most $c_{new}(n, a, n')$. Negating both sides of the inequality and adding $g(n_{goal})$ gives:

$$\begin{aligned} g(n_{goal}) - g(n') &\geq g(n_{goal}) - g(n) - c_{new}(n, a, n') \\ g(n_{goal}) - g(n') + c_{new}(n, a, n') &\geq g(n_{goal}) - g(n) \\ h_{new}(n') + c_{new}(n, a, n') &\geq h_{new}(n) \end{aligned}$$

Where we use the definition of the updated heuristic in the last line. Thus, the heuristic remains consistent after each instance of adaptive A*, even if the action costs increase.

5 Heuristics in the Adaptive A*

50 trials on 50 grid worlds of size 101x101:

	Avg Runtime	Avg Nodes Expanded
Repeated Forward	0.2627	3327.2
Adaptive Repeated Forward	0.1991	3220.88

Adaptive Repeated Forward A* on average expands fewer nodes than Repeated Forward A*. For non-adaptive A*, the h values of the nodes are the Manhattan distances between that node and the target, and thus, never change. On the other hand, after each iteration of adaptive A*, the h values of all the nodes expanded in that iteration are updated. Therefore, this observation is expected because Adaptive Repeated Forward A* utilizes information about the nodes gained from previous iterations of A*.

In this trial, the average run time of Adaptive Repeated Forward A* is faster than the average run time of Repeated Forward A*. However, we did observe that in other trials, despite expanding fewer nodes, the average run time of Adaptive Repeated Forward A* is sometimes greater. This is likely because of the overhead time to implement the adaptive algorithm (continuously updating the h-values of expanded nodes).

6 Statistical Significance

We describe the statistical hypothesis test for the problem described in part 3: we use a hypothesis test to determine if the difference in the number of expanded nodes between repeated forward A* and repeated backward A* is systematic. To do so, we will use a paired t-test.

We first run repeated forward A* on our 50 generated grids of size 101, counting the number of expanded nodes for each run. We repeat the same procedure for repeated backward A*. As we run the algorithms on the same grids, we can take the difference in expanded node count between the algorithms - call this difference d_i for the ith grid. We obtain an average expanded node count difference of $\bar{d}$, and a sample standard deviation of differences S_d.

We will then test to see whether the population means are different, or equivalently, whether the average difference μ_d is not 0, across every possible 101x101 grid.

The null hypothesis H_0: $\mu_d = 0$

The alternative hypothesis H_a: $\mu_d \neq 0$.

We then assume the null hypothesis. We calculate the t statistic as:

$$t = \frac{\bar{d}}{S_d/\sqrt{50}}$$

We take the number of degrees of freedom to be $50 - 1 = 49$. Using a calculator, we then calculate the area under the t-distribution from $-\infty$ to

$-|t|$, with 49 degrees of freedom. Multiplying this value by 2 (for a two-tailed distribution) gives the p-value, p. We then choose the statistical significance level α to be 0.05 and compare to see if $p < \alpha$. If this is true, we reject the null hypothesis, and this suggests that there is a systematic difference between the number of nodes expanded by repeated forward A* and repeated backward A*. Otherwise, we do not have enough evidence to suggest that there is such a systematic difference.