Homework 1 Fast Trajectory Replanning

Yulin Ni and Gloria Salas Paucar

Part 1

a. The goal of this program is to find the shortest track for the agent. The agent at its starting point E2 sees that all of its adjacent blocks are unblocked, including the one to the east. Since the target is to its east, the target has a field of view limited to only its adjacent blocks, and the block immediately to its east is unblocked, so the agent assumes that its shortest presumed unblocked path towards the target goes straight from E2 to E5, where the target is. Therefore, the agent's first move is to the east.
b. One way to prove that the number of moves of the agent until it reaches its goal or discovers that it is impossible is bounded by the number of unblocked cells squared is that the number of blocks is finite, so the agent is bound to be able to either reach the goal or determine that it is impossible to reach the goal within a finite amount of time. The worst case scenario that could occur here would be that the agent has to traverse the entire map. Also, A* has a close list so the agent will never expand to the cells that are already previously expanded. This would solve the problem of having an infinite loop. In order to solve the problem of the goal being blocked, the program would have to stop executing once every cell has already been expanded to. Thus, if the goal is blocked by other cells then the program will just have to stop executing.

Part 2

Part 3

Part 4

The Manhattan distance is the distance between two points in a maze and is strictly based on both or either the horizontal and vertical path. The formula

$$h(s) = |X_s - X_{goal}| + |Y_s - Y_{goal}| \tag{1}$$

In the maze we generated, the agent only has four directions to move: up, down, left and right. The Manhattan distance is defined as the fastest distance. In order for it to be heuristic, the following formula must be true

$$h(s) <= h(s') + c(s, a, s') \tag{2}$$

Let us note that a heuristic is consistent if, for every node and every node's successor is generated by any action. Now we can substitute all places were h(s) and h(s') are with the corresponding Manhattan distance we defined previously.

$$|X_s - X_{goal}| + |Y_s - Y_{goal}| <= |X'_s - X_{goal}| + |Y'_s - Y_{goal}| + c(s, a, s') \tag{3}$$

Then simplify

$$|X_s - X_{goal}| + |Y_s - Y_{goal}| - |X'_s - X_{goal}| + |Y'_s - Y_{goal}| <= c(s, a, s') \tag{4}$$

Let us say that s and s' are neighboring cells, s' will be the instant successor to this initial state. Afterwards, say that s' will be vertically adjacent to s. Thus, the following proves true.

$$|X_s - X_{goal}| - |X'_s - X_{goal}| = 0 \tag{5}$$

$$|Y_s - Y_{goal}| - |Y'_s - Y_{goal}| = 1 \tag{6}$$

$$|Y_s - Y'_s| = 1 \tag{7}$$

$$|Y_s - Y_{goal}| - |Y'_s - Y_{goal}| = c(s, a, s') \tag{8}$$

$$h(s) - h(s') = |Y_s - Y_{goal}| - |Y'_s - Y_{goal}| \tag{9}$$

If we have the following inequality

$$|A| - |B| <= |A - B| \tag{10}$$

then

$$h(s) - h(s') <= |(Y_s - Y_{goal}) - (Y'_s - Y_{goal})| = 1 = c(s, a, s') \tag{11}$$

Therefore, the Manhattan distances are considered consistent heuristics.

Adaptive A*

We can prove Adaptive A* leaves heuristics consistent over iterations, we have to prove it's validity for the following cases, where s denotes a parent node, and s', its successor.

0.1 In the case both s and s' are expanded

The new heuristics for both of the nodes are

$$h_{new}(s) = g(s_{goal}) - g(s) \tag{12}$$

$$h_{new}(s') = g(s_{goal}) - g(s') \tag{13}$$

Now, we substitute the nodes into the consistency inequality

$$h_{new}(s) <= h_{new}(s') + c(s, a, s') \tag{14}$$

$$g(s_{goal}) - g(s) <= g(s_{goal}) - g(s') + c(s, a, s') \tag{15}$$

$$g(s') <= g(s) + c(s, a, s') \tag{16}$$

Using the above equation, we can conclude that the heuristic remains consistent because the algorithm calculates the g value of the successor state based off of the g value of its predecessor taken into account the cost to transition that was added, so the inequality is never false.

0.2 In the case s, but not s' is expanded

The consistency inequality we prove here in this case is below

$$h_{new}(s) <= h(s') + c(s, a, s') \tag{17}$$

First, considering that case 1 holds true, we can deduce that case 2 holds true as well, because the inequality would still be valid if the node were to be expanded too. Nothing will change in regards to the situation, the only thing we changed was the amount of information. Additionally,

$$f(s) <= f(s') \tag{18}$$

because the f value of a visited node in the previous search (s') cannot be any smaller than an expanded node in our previous search (s). The algorithm prioritizes expansion between the nodes of the smallest f values. We substitute into the equation above what we have causing the split between the f-value into h+g values,

$$h_{new}(s) + g(s) <= h(s') + c(s, a, s') + g(s) \tag{19}$$

Notice

$$h_{new}(s) <= h(s') + c(s, a, s') \tag{20}$$

proves the inequality true.

0.3 In the case neither are expanded

Unlike the other cases, the adaptive A* algorithm does not modify either s or s', so considering that the consistency inequality would hold true in normal A*, our conclusive result is that this case holds true too.

Part 5

Part 6

To determine whether there is a significant difference between the performances of Repeated Forward A* versus Repeated Backward A*, one can conduct a Student's t-test. To perform a Student's t-test on the performance speed data between these two search algorithms, one must first calculate the t-value by using the following formula:

$$\frac{|\overline{x_1} - \overline{x_2}|}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{21}$$

on samples of performance data from running Repeated Forward A* and Repeated Backward A*. $\overline{x_1}$ and $\overline{x_2}$ refer to the mean of Forward A* and Backward A*'s sample data, respectively. s_1 and s_2 are the standard deviations of the two data sets, and n_1 and n_2 refer to the amount of data from each respective set. After obtaining our t-value, we then use that value to test our null hypothesis H_0, which posits that there is no statistically significant difference between the two samples. To do so, we must identify a critical value based on the degrees of freedom of our data sets (df = $n_1 + n_2 - 2$) from a t-table, and compare our t-value to said critical value. If the t-value is lower than the corresponding critical value, then we don't reject our null hypothesis: in other words, we say that there is no significant statistical difference between the two algorithms' performances. Otherwise, if the t-value is higher, then we reject our null hypothesis: in other words, we say that there is in fact a significant statistical difference between the two algorithms' performances.

Source: https://www.youtube.com/watch?v=pTmLQvMM-1M