15-887 Solutions Homework 2

1 Moving Target

Optimal Planner

The problem was too large for a distance-based heuristic (for example *Manhattan distance*) to work efficiently. It was simply not informative enough.

A typical approach consisted in computing a more informative heuristic by solving a relaxed version of the problem, for example, ignoring the time constraints and simply solving in x, y. Dijkstra algorithm can efficiently solve the 1000x1000 grid, and provide us an heuristic that takes into account the costs of the cells. This heuristic could then be used with A* to solve the problem in x, y, t.

Note however, that in this approach, it was important to show that the heuristic was consistent. That is necessary to guarantee the planner is optimal.

Suboptimal Planner

Many possible solutions here. A simple solution consists in computing an heuristic as above, but then using weighted A*. Another alternative would be adopting an RRT-like approach.

2 Planning with heuristics

2.1

a)

We start by proving that $h(G) = 0$. This comes from $h(G) = \min\left(h_1\left(G\right), h_2\left(G\right)\right) = \min\left(0, 0\right) = 0$.

Then, we prove that $h(s) \leq c(s, s') + h(s')$:

$$\begin{aligned} h(s) &= \min(h_1(s), h_2(s)) \\ &\leq \min(c(s, s') + h_1(s'), c(s, s') + h_2(s')) \\ &= c(s, s') + \min(h_1(s'), h_2(s')) \\ &= c(s, s') + h(s') \end{aligned}$$

b)

Similar to above, but replace min by max.

2.2

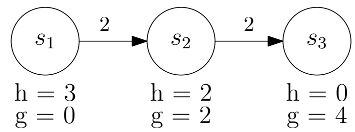
Correct answer: b.

Remember that A* algorithm always takes the state with the smallest f from the *OPEN* list.

2.3

Correct answer: f.

There are no guarantees regarding the monotonicity of the f values of the nodes expanded. Simple counter-example (adapted from the answer by Shen Li):



$\epsilon = 1.1 \implies f(s_1) = 3.3 < f(s_2) = 4.2 > f(s_3) = 4$
$\epsilon = 10 \implies f(s_1) = 30 > f(s_2) = 22 < f(s_3) = 4$
It is easy to prove that the heuristic is consistent.

2.4

Correct answer: e.

a) is false. Suppose the inadmissible heuristics always have values much larger than h_0. In that case, MHA* may always expand nodes from $OPEN_0$, thus expanding as many nodes as A*.

b) is false. Suppose the inadmissible heuristics are not very informative, and happen to have values much smaller than h_0 on the states relevant to the search. In that case, MHA* may expand many more node.

c) is false. Same as b)

d) is false. Same as b)