CSI535 – Introduction to A.I. Assignment #1): Direction Finding Using A*

Due: Friday 03/05/04
Worth: 20% of Final Grade
Late Policy: You lose one full grade for each week (including partial weeks) you are late.

Read the instructions carefully, ask questions if you have any doubts.

The A* algorithm is a very useful A.I. technique for direction finding in applications such as robot navigation and computer game character movement (many advancement to A* have been created by the gaming industry). Direction finding involves making the optimal series of actions to move from a start to goal point.

You will be using an environment that consists of a 20 x 20 cell universe where your agent takes up exactly one cell. Your agent can make only one of four movements (back, forward, right, left one block). The actions your agent performs may be one of these movements or combinations of multiple movements. You will begin by implementing A* to navigate this universe in its most basic form and then make changes to handle progressively difficult versions of the environment. Email your code to ke@cs.albany.edu. Note that the better your admissible your heuristic the more points you get.

Question 1). Basic version of environment (20 points)

Assume that your agent can see the entire map. Implement a program to determine the optimum series of actions (shortest distance) for the following map using A*. The heart represents the goal state, the smiley face the start state, the black blocks are borders that cannot be crossed, all non-filled blocks can be passed through.

![Map](image)

Define the state space, constraints, actions and goal test. Carefully define your admissible heuristic under all conditions and your $f$ and $g$ functions. Draw the first two levels of your data structure (nominate if you intend to use a graph or tree) showing $f$, $g$ and $h$ values. Present your results by showing the $f(n)=g(n)+h(n)$ values for each node in the map your implementation evaluates. State the number of nodes your agent expands. Present the pseudo code to your algorithm.

Question 2). Environment with elevation (25 points)

Now imagine the environment has a topography to it. In this question you will be using the map below. Some cells are “higher” than others, your agent can climb from one elevation to another at a fixed cost dependent only on the elevation of the destination cell but not on the elevation of the cell the agent is
moving from (this is a simplification). As before, the black cells are impossible to go through, the dark gray cells can be traversed through at cost 10, the light gray cells the cost is 5, traversing through all remaining cells cost 1.

Define the changes (if any) to your space state space, constraints, actions and goal test. Carefully define your admissible heuristic under all conditions and your f and g functions. Present your psuedo code. Present your results by showing the f(n)=g(n)+h(n) values for each entry in the following map that your agent investigates. Note the non-uniform cost aspect of A* can be used to handle direction planning to avoid oppositions, pits, collect as much reward as possible etc.

Question 3. Environment with topography and partial access to the environment (35 points)

Now imagine that your agent does not have a map, but does have 360 degree vision but the cells with non-zero height block his/her entire view beyond that cell. Furthermore, the agent has a transmitter that can provide the Euclidean to the goal. The agent perceives two predicates for each cell: isBlocked() and isGoal(). Discuss changes to your basic algorithm that allow handling an environment with topography and partial access. Define the state space, constraints, actions and goal test. Carefully define your admissible heuristic under all conditions and your f and g functions. Present your psuedo code. Present your results by showing the f(n)=g(n)+h(n) values for each entry in the above map that your agent investigates. Does your agent find the same trail as before?

Question 4. Is close enough good enough? (20 points)

Much has been made of the need for A* to use an admissible heuristic. It is known that if h(n) underestimates the value f(OPTIMAL_GOAL)-f(n) for all n then A* returns the optimal route. However, what if h(n) occasionally (δ percent of the time) overestimates the correct value by at most some small number ε. Clearly the proof of optimality for A* no longer holds. But does the algorithm now return hopelessly sub-optimal results? Empirically (Ph.D. students can tackle this formally) investigate the additional cost above optimality if your heuristic overestimates the correct cost occasionally. You can introduce over-estimation by adding a stochastic (RNG) aspect to the calculation of h(n).

For the interested student

Question 3 and Question 4 introduce two simplifications. Firstly, that the cost to go to a cell of a specific height only depends on that cell. If the cost of going between two cells is |CellA-CellB| what would be an admissible heuristic? Secondly, if blocks of non-zero height only block out what can be seen behind according to the distance the agent is from the block and the height of the cell, what is an admissible heuristic?