Introduction to Artificial Inteligence

Programming Assignment 4

Programming assignment - Decision-making under uncertainty

Hurricane Evacuation Problem

Goals

Sequential decision making under uncertainty using belief-state MDP for decision-making: the Hurricane Evacuation problem. (This is actually the Canadian traveler problem.)

Hurricane Evacuation Decision Problem - Domain Description

The domain description is similar to that described in assignment 1, except that again we do not know the locations of the blocakges, and there is only one known location with evacuees. For simplicity, however, we will assume that the blockages occur independently, with a known given probability, and persistence is perfect: once blocked (or unblocked), an edge remains blocked (resp. unblocked) forever. Blockages are revealed with certainty when the agent reaches a neigbouring vertex.

Thus, in the current problem to be solved, we are given a weighted undirected graph, where each edge has a known probability of being blocked. These distributions are jointly independent. The agent's only actions are traveling between vertices and terminating. Traversal times are the weight of the edges. Also for simplicity, we will assume only one agent, starting at Start and only one vertex with evacuees (also called a **target** vertex). The problem is to find a policy that saves (in expectation) as many people as possible.

The graph can be provided in a manner similar to previous assignments, for example:

```
#V5    ; number of vertices n in graph (from 1 to n)

#E1 1 2 W3       ; Edge from vertex 1 to vertex 2, weight 3
#E2 2 3 W2       ; Edge from vertex 2 to vertex 3, weight 2
#E3 3 4 W3 B0.3  ; Edge from vertex 3 to vertex 4, weight 3, probability of blockage 0.3
#E4 4 5 W1       ; Edge from vertex 4 to vertex 5, weight 1
#E5 2 4 W4       ; Edge from vertex 2 to vertex 4, weight 4
#Start 1
#Target 5
```

The start and target vertices, should be determined via some form of user input (in the file like the above example, or by querying the user). For example, in the above graph the start vertex is 1, and the target vertex is 5.

Solution method

The Canadian traveller problem is known to be PSPACE-complete, and is in fact the same as the Hurricane Evacuation problem scenario presented in this assignment. So you will be required to solve only very small instances. We will require that the entire belief space be stored in memory explicitly, and thus impose a limit of at most 10 vertices with at most 10 possible blockages. Your program should initialize belief space value functions and use a form of value iteration (discussed in class) to compute the value function for the belief

states. Maintain the optimal action during the value iteration for each belief state, so that you have the optimal policy at convergence.

Requirements

Your program should read the data, including the parameters (start and target vertices). You should construct the policy, and present it in some way. Provide at least the following types of output:

1. A full printout of the value of each belief-state, and the optimal action in that belief state, if it exists. (Print something that indicates so if this state is irregular, e.g. if it is unreachable).
2. Run a sequence of simulations. That is, generate a graph instance (blockage locations) according to the distributions, and run the agent through this graph based on the (optimal) policy computed by your algorithm. Display the graph instance and sequence of actions. Allow the user to run additional simulations, for a newly generated instance in each case.

Deliverables

1. Source code and executable files of programs.
2. Explanation of the method employed in your algorithm.
3. Non-trivial example runs on at least 2 scenarios, including the input and output.
4. Submit makefile and short description on how to run your program.

Official deadline: TBA 2021