Introduction to AI
Practical Assignment 4

Uzi Friedman, Lior Siag

January 2021

1 Construction and Algorithm Explanation

We start off by creating all possible belief states. These include the current location of the agent, and its knowledge about blockages. Each possibly blocked edge can have a probability of either:

- 0 - Known to be free
- 1 - Known to be blocked
- $0 \leq x \leq 1$ - The edge is blocked with probability x.

Once we have created all states, we add transitions between them. A state [Loc, p1, p2, ..., pk] has a transition to state [Loc2, p1', p2', ..., pk'] if (Loc, Loc2) is an edge, and $\forall_{i \in [k]} pi = pi'$ or if $pi \neq pi'$ then $0 \leq pi \leq 1$ and $pi' = 0$ or $pi' = 1$ and the edge referred to by pi and pi' is adjacent to Loc2.
We then compute the probability of each such transition, based on the probabilities of blockages.
We find which states ares reachable by first adding a fake initial state, connected to all legal starting states. We run a BFS to find which states are reachable from this fake initial state, and mark all other states as unreachable. In order to compute the optimal policy we initialize the utility of the goal states to 0 and all other states to None. We then use value iteration on the set of reachable states until convergence, only updating a nodes value if we found an action which the expected utility from it is greater than the previously found one. The utility of an action (moving along an edge) is the negation of the weight of that edge. We only consider the expected utility of actions which all possible resulting states' utility has already been calculated (not None). We only look at reachable states and initialize the states' utilities to None to reduce computation time, but we could have initialized the utility to any value
Given the optimal policy, we generate graph instances and walk the agent through them using it. We start by creating a fake starting state and updating it to the actual starting state based on the agent's observations. This

is done because the starting space might be adjacent to edges which have possible blockages, so we need to update the state according to its observation. We then continuously move according to the policy's prescribed action for the current belief-state, updating the belief-state based on the agent's movements and observations.

2 How to run the code

Run main and enter the path to the graph file. The program will then print all the belief states, whether they are reachable or not, and their utility and optimal action if they are.

It will then create an instance of the graph according to the blockages probabilities, run the agent through it and ask the user if they want to run it again.

3 Example Graph

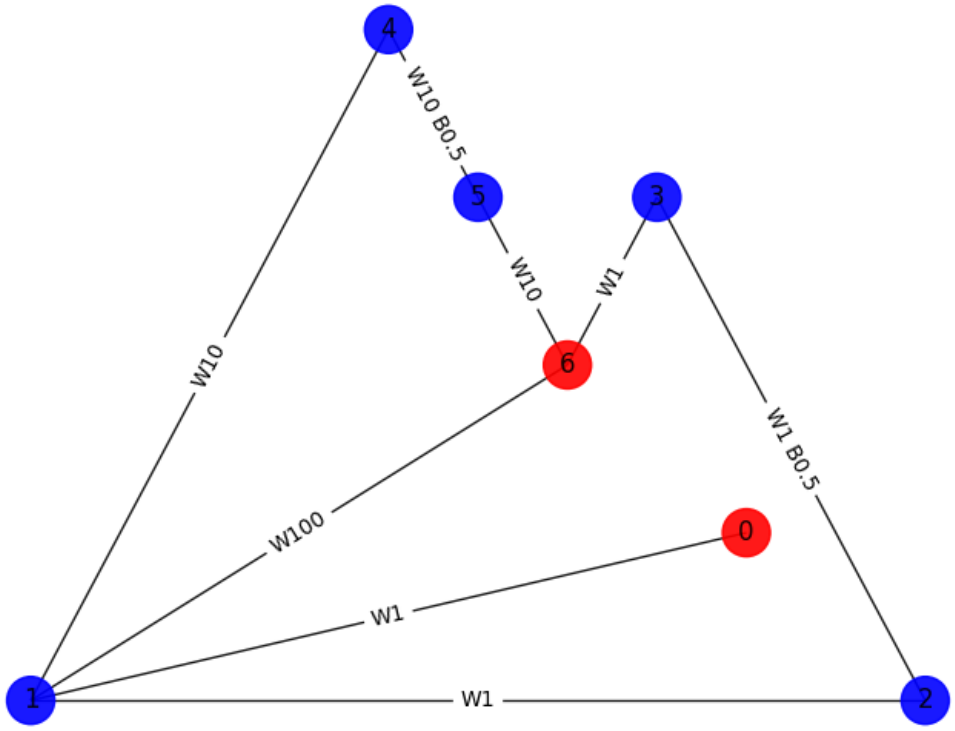

Figure 1: Example graph. Nodes are labeled with their respective ids and edges are labeled with their weight and blockage probability if given. The start and target are in red.