Introduction To AI - Assignment 1

Nimrod Berman, Yossi Iram

November 2022

1 The heuristic

As told in class, in the second part of the work we ignore broken vertices. In addition, we construct a Heuristic Graph. This is a full graph that contains only vertices with people on them. An edge (a, b) weight is the shortest path from a to b in the original graph. Finally, our heuristic values are the distance to each unvisited vertex in the new graph.

The reason we pick this heuristic is as follow. First, assuming that the graph cannot be broken, will always drive a better solution then if the graph could be broken along the game. Second, changing the graph structure helps to choose the fastest way from vertex to vertex. Possibly avoiding redounded visits in vertices without people. Last, taking the fastest way will enable to pick all people in the a fast route. Combining this reasoning, produce an admissible function that we can use to solve the Evacuation Problem.

2 Agent Performance Comparison

In our work, we have implemented sevral agent to solve the hurricane evacuation problem. The agents are as follow: 1. Greedy agent without heuristic, 2. Greedy agent with heuristic, 3. A star agent, 4. Real time A star agent. The heuristic, as told in class, is to ignore brittle blocks and just find that fastest way to save all people in the graph. In addition, the heuristic values, are the shortest paths to unvisited vertex.

2.1 Comparison on small graphs

In the example given in the assignment, we observe that all agents perform pretty much the same and finished the game. solving the problem with 3-4 steps (see env.txt).

2.2 Comparison on 20 Vertices graph

To evaluate and understand empirically the differences between the agents, we built a complex world with 20 vertices and 26 edges(see env2.txt) and run a simulation with each agent type. In what follows we report the results we observed:

Greedy agent without heuristic. Without heuristic the agent is exposed to broken vertices. Thus, it can be trapped in an island vertex which will finish its run. Indeed, when running the simulation, the agents greedly choose to move the colsest vertex and 'get stuck' there.
Final Score: 1998

Greedy agent with heuristic. The agent chooses the next unvisited vertex where people on it. It assume there are not broken vertices.
Final Score: 20970

A star agent. Given the huge amount of possible expansions and the limitation of expansions, the agents fails to found a solution and fails.
Final Score: 0

Real time A star agent. Given a limitation of 10 expansions, the agent solves the problem with 2 searches.
Final Score: 20938, T=0.000001
Final Score: 10938, T=0.01

2.3 Discussion

As mentioned in the assignement description, sometimes searching takes more time the just picking greedly. We observe this on our experiment results. The greedy heuristic agent was the fasteset to finish. When expansion is a computationally heavy action, the A star algorithms can be maybe optimal in sense of solution but sub optimal in sense of overall score when other factor are in play(e.g. time).