

Introduction to AI - assignment 1

Oded Yechiel

Matan Rusanovsky

13/11/18

1 Introduction

In this assignment we had to provide a hurricane simulator and 6 agents to roam the simulator and rescue people. The rescue is performed by loading the people from towns to shelter towns.

Roads are connecting towns and are weighted with values. The vehicle traverse time between towns is calculated using the following formula,

$$traverse_time = W \times (1 + K \times P), \quad (1)$$

where, W is the weight between the towns, K is a slowdown parameter that is multiplied by the number of people in the vehicle.

1.1 Agents

The first three agents are Human agent, mainly for debugging purposes, a greedy agent, also for debugging purposes, and a Vandal agent, used to complicate the given problem.

The last three agents are Greedy agents, based on Heuristics which will be explained in the next section, an A* agent and a real-time A* agent, which are based on the same Heuristics.

1.2 Performance

In order to assess the agents, each of the later agents will record the number of expansions they perform. The overall performance, P is calculated as,

$$P = f * S + N, \quad (2)$$

where, S is the number of people saved, and N , is the number of expands the agent performs, and $f < 0$ is a factor to multiply S . Usually $|f|$ will be a large number. Lower performance value, P , indicates better performance.

In this report we shall explain the Heuristics used in this assignment. Please refer to our git repository for simulations and results: <https://github.com/odedyec/IntroToAI>

2 Heuristics

The chosen heuristics attempts to assess the number of people that there is no way to save. Since the performance is mainly based on the amount of people we save, each unsaved person is multiplied by a large value in the heuristic.

Algorithm 1 Heuristic calculation

```
1: procedure H(currentVertex, Simulator)
2:   DoomedPeople  $\leftarrow$  Sum of people in all vertices + PeopleInVehicle[Simulator]
3:   Find the ClosestShelter,  $d_s$ 
4:    $t_{now} \leftarrow$  CurrentTime[Simulator]
5:   if  $t_{now} + d_s > \text{Deadline}[\text{Simulator}]$  then
6:     return DoomedPeople  $\times$  LARGE\_VALUE + Cost[currentVertex]
7:   else
8:     DoomedPeople  $\leftarrow$  DoomedPeople - PeopleInVehicle[Simulator]
9:   end if
10:   $V_p \leftarrow$  Find path to all Vertices with people
11:  for  $v \in V_p$  do
12:    Find the ClosestShelter,  $d_s$ 
13:    if  $t_{now} + \text{cost}[v] + (1 + \text{People}[v] * K) * d_s < \text{Deadline}[\text{Simulator}]$  then
14:      DoomedPeople  $\leftarrow$  DoomedPeople - People[ $v$ ]
15:    end if
16:  end for
17:  return DoomedPeople  $\times$  LARGE\_VALUE + Cost[currentVertex]
18: end procedure
```

It is possible to find the shortest path between two vertices in polynomial values. Since all the weights are bound to be positive, the Dijkstra shortest path algorithm provides a solution in $O(V \log V + E)$.

The calculation of the Heuristics assumes starts by summing all of the people that are yet to be saved. Then by calculating the shortest path to the closest shelter will suggests if the people in the vehicle are bound to be saved, and if not all of the people are doomed. Then for each vertex with people a closest shelter vertex is found and assuming that this is the only vertex in the world, we test whether it is possible to save these people. This proves that the heuristic is admissible as it is must to be more optimistic than reality. The heuristics is summarized below: