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,

$$traverse\_time = W \times (1 + K \times P), \tag{1}$$

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,

$$P = f * S + N, \tag{2}$$

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

procedure H(currentVertex, Simulator)
  $DoomedPeople \longleftarrow$ Sum of people in all vertices + $PeopleInVehicle[Simulator]$
  Find the ClosestShelter, d_s
  $t_{now} \longleftarrow CurrentTime[Simulator]$
  if $t_{now} + d_s > Deadline[Simulator]$ **then**
    return $DoomedPeople \times LARGE\_VALUE + Cost[currentVertex]$
  else
    $DoomedPeople \longleftarrow DoomedPeople - PeopleInVehicle[Simulator]$
  end if
  $V_p \longleftarrow$ Find path to all Vertices with people
  for $v \in V_p$ **do**
    Find the ClosestShelter, d_s
    if $t_{now} + cost[v] + (1 + People[v] * K) * d_s < Deadline[Simulator]$ **then**
      $DoomedPeople \longleftarrow DoomedPeople - People[v]$
    end if
  end for
  return $DoomedPeople \times LARGE\_VALUE + cost[currentVertex]$
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 $O(VlogV + E)$.

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 (official proof in the following section) that the heuristic is admissible as it is must to be more optimistic than reality. The heuristics is summarized below:

2.1 proof of admissibility

Assume a perfect heuristic function, h^* exists.

Assume that there is only a single vertex with people, with cost to get to, from some vertex v: $D_{p_1}(v)$, and the closest shelter to that vertex is $D_s(p_1)$. In this case the heuristic suggested above provides a perfect heuristic of how many people will be saved, i.e.

$$h = h^* \tag{3}$$

Now assume that there are more vertices with people than a single one. We denote one of these vertices as p_2, and the cost to get to it from some vertex v as $D_s(p_2)$. Also, we denote the distance to the closest shelter from some vertex v as: $D_s(v)$. From the triangle inequality, the cost of reaching both vertices and going to the closest shelter is obviously larger than reaching a single town and going to the closest shelter., i.e.,

$$D_{p_1}(v) + D_s(p_1) \leq D_{p_1}(v) + D_{p_2}(p_1) + D_s(p_2) \tag{4}$$

or,

$$D_{p_2}(v) + D_s(p_2) \leq D_{p_1}(v) + D_{p_2}(p_1) + D_s(p_2) \tag{5}$$

The same arguments apply on paths with more than one additional vertex.

Thus, the calculated heuristic is more optimistic than the optimal heuristic,

$$h \leq h^*, \tag{6}$$

for any of the cases.