The heuristic function

We used the abstraction of a graph G whose vertices are the vertices with people in the original graph+initial state, and where the edges between every 2 vertices represent the shortest path between them.

The heuristic is the following:

We used the spanning tree heuristic, and so h(x) works like that:

First, we create a spanning tree SPT over G, then h(A->B) returns the following values:

Case 1) If A->B is not on the spanning tree, return infinity.

Case 2) If B contains people(meaning we didn't pass through it and it is not the initial state) then h(A->B) returns 0(best possible value).

Otherwise, it returns AVG/BTime, where AVG is the average weight of an edge in G, and BTime is the internal time which the agent wasn't to B in the internal graph(Each state contains how much time we weren't in node V for every node V in the internal graph).

Justification

The justification for that heuristic function is that the sum of the travel cost in a minimum spanning tree is bounded upwards by 2*the optimal solution. As for the other details of the h function, the values themselves were tested empirically and based on intuition for what we thought would work.