Post-Processing Step - Student Example

Based on a Piazza post from Spring '17

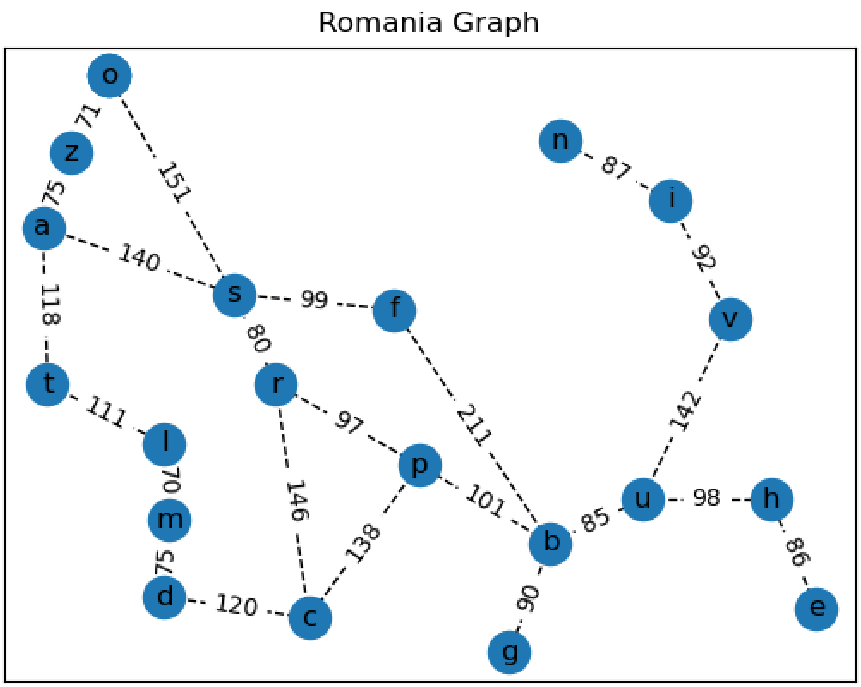

Figure 1: Romania Network Graph

So with the path from start `c` to goal `r`, I get to the following stage:

- Frontier from Start (PQ): [(138, `p`), (146, `r`), (195, `m`)]
- Frontier from Goal (PQ): [(97, `p`), (146, `c`), (179, `f`), (220, `a`), (231, `o`)]
- Explored States from Start: {`d`: 120, `c`: 0}
- Explored States from Goal: {`s`: 80, `r`: 0}

I expand `p` in "Frontier from Start", and since this value isn't in the explored states from goal, I add it to the explored states from start. So I now have

Explored States from Start {`p`: 138, `d`: 120, `c`: 0}

I then expand `p` in "Frontier from Goal" and sure enough, there is a `p` in "Explored States from Start" and so I update μ from infinity to $138 + 97 = 235$. Now I check if $top_f(146, \mathtt{r}) + top_r(146, \mathtt{c}) \geq \mu(256)$, and it is $(146 + 146 \geq 235)$, so I exit.

One Solution

At this point:

- Frontier from Start (PQ): [(146, `r`), (195, `m`)]
- Frontier from Goal (PQ): [(97, `p`), (146, `c`), (179, `f`), (220, `a`), (231, `o`)]
- Explored States from Start: {`p`: 138, `d`: 120, `c`: 0}
- Explored States from Goal: {`s`: 80, `r`: 0}

The goal search is to be expanded next. (97, `p`) is popped off the Goal PQ and is added to the goal explored set. It is found in the explored set of the start search.

- Frontier from Start (PQ): [(146, `r`), (195, `m`)]
- Frontier from Goal (PQ): [(146, `c`), (179, `f`), (220, `a`), (231, `o`)]
- Explored States from Start: {`p`: 138, `d`: 120, `c`: 0}
- Explored States from Goal: {`p`: 97, `s`: 80, `r`: 0}

A list of potential crossover points can be made from all start_explored set nodes that are also found in the union of the goal_frontier and the goal_explored nodes.

Crossover Points: {`p`, `c`}

Now, for each crossover point, we can check to see if the path cost from start to that point + the path cost from goal to that point exceeds the lowest cost path we have found so far (μ, which is initialized to infinity). If it is, then that point becomes the lowest cost point.

start_cost(`p`) + goal_cost(`p`) = 97 + 138 = 235

start_cost(`c`) + goal_cost(`c`) = 0 + 146 = 146

Whichever crossover point comes out as the best crossover belongs to the shortest path. You would then need to stitch the start to goal path together through this path.

Best crossover point: `c`
Final path: [`c`, `r`]

NOTE: The node (97, `p`) was never actually explored in the sense of finding its neighbors. It was still added to the explored set, but PRIOR to iterating through its neighbors, the algorithm checked if this node was in the explored set, therefore not unnecessarily exploring more neighbors.