Graph Reduction Problem

Frank Liu

Suppose Bob is travelling through a jungle (Represented by a graph $G = (V, E)$, Bob starts from his hut, vertex h.

There are 2 types of paths/edges: river paths ($R \subset E$) and dirt paths ($L \subset E$) of different lengths (weights) to areas (vertices v in V) that he can take. He wants to get the rare Amazonian cotton candy located at the temple (vertex t), and store it in any of the special dropoff vertices ($D \subset V$). However, once he gets this candy he can no longer take the rivers because the candy will melt :(

Bob wants to find the shortest path he needs to get the candy and store it at a dropoff zone.

1) How do we calculate the shortest path from his hut h to the temple t?
2) How do we calculate the shortest path from the temple t to all the drop off zones d?
3) How do we combine parts 1 and 2 to find the solution to the problem?
4) What if we now have an algorithm $P(G, s, f)$, which finds the shortest path between from vertice s to vertice f in a graph G, but are only allowed to run it once? The runtime of your solution should be no slower than $O(|E| \log |V|)$.
5) Prove the correctness of your solution to 4

Solution:

1) Run Dijkstra's on G with h as the source vertex, this will give us the shortest path from h to all vertices which includes t.
2) Run Dijkstra's on a graph with the same vertices as G but only dirt paths, $G' = (V, L)$, with t as the source vertex this will give us the shortest path from t to all vertices which includes all the drop-off zones $d \in D \subset V$.
3) We can take the shortest path from h to t from 1) and then combine it with the shortest path out of the shortest paths from t to d from 2).
   In other words:

   Shortest Valid Path = (Shortest path from h to t)+(Shortest path from t to d')

   such that $len(d') = min_{d \in D}\{len(d)\}$

4) Let us first create a copy of the graph $G_1 = (V_1, E_1)$ with the exact same vertices and edges. This will represent Bob before he reaches the temple. Let us then create a copy of the graph $G_2 = (V_2, L_2)$ with the same vertices but only dirt paths. We then want to add an edge $e = (t_1, t_2)$ with weight 0 which connects the temple vertex in the first graph $t_1 \in G_1$ to the temple vertex in the second graph $t_2 \in G_2$. We then want to create a new vertex d' and connect all of drop off vertices $d \in D_2 \subset V_2$ to it with edges of weight 0, $C = \{(d, d'), \forall d \in D_2\}$. Let G' be this new graph we have created:

$$G' = (V_1 \cup V_2 \cup \{d'\}, E_1 \cup L_2 \cup \{e\} \cup C)$$

We would then run $P(G', h_1, d')$ where $h_1 \in G_1$ is the starting vertex/hut in the first graph. I would then iterate through the path and remove one of the duplicate temples and the edge e as well as d' and the edge connecting to it to end up with a path in our original graph.

Note that creating the two copies of the graph takes $O(|V| + |E|)$ time each, and creating the new vertice d' and adding the necessary edges is $O(|D|)$. Iterating through the path takes $O(|E|)$ time Thus our runtime is :

$$O(|V|+|E|)+O(|D|)+O(|E|) = O(|V|+|E|)+O(|V|)+O(|E|) = O(|E| \log |V|)$$

=======Shorter Version=======
We will create 2 copies of the graph G, one with all the edges G_1 and the other without any river edges G_2. We will add an edge with 0 weight from the temple in G_1 to the temple in G_2. We will then connect all the dropoff points in G_2 to a new vertice d' with edges of weight 0. Let G' be this combined graph. We then run $P(G', h_1, d')$. Creating G' takes $O(|V| + |E|) = O(|E| \log |V|)$

5) To prove the correctness I will start by proving that the paths considered by my solution are only valid paths (visit the temple before arriving at a dropoff vertex) and that it considers all of them.

Proof. Let P be the paths we consider and V be the valid paths.

$\subseteq$) Let $p \in P$ be a path considered in our solution, this means it starts at $h_1 \in G_1$ and ends at d'. The endpoint d' is only connected to the dropoff points $d \in D_2$ of the layer G_2, thus our solution must reach a dropoff point right before the end. The only way from G_1 which we start to G_2 is through $e = (t_1, t_2)$ thus we must pass through the temple. Thus p starts at s visits the temple t and aftwards only takes dirt paths to end at a dropoff zone $d \in D$, and is valid. $p \in V$ and thus $P \subset V$.

$\supseteq$) Let $v \in V$ be a valid path. We know that at some point it visits t and afterwards it only takes dirt paths. Note that we can reconstruct v in our graph, if $v = h, v_2, v_3, \ldots, t, v_k, v_{k+1}, \ldots, d$ we can create:

$$v' = h_1, v_{2,1}, v_{3,1}, \ldots, t_1, t_2, v_{k,2}, v_{k+1,2}, \ldots, d, d'$$

Where v_i, j is the corresponding $v_i \in G_j$. Note that v' is a path from h_1 to d' in our graph G' thus $v \in P$ and therefore $V \subset P$.

Thus we conclude that $P = V$ $\square$

Let p be a valid path in G, and p' be the corresponding path that we consider in G'. Note that p' has the same edges except for two additional ones $e = (t_1, t_2)$ and an edge from (d, d') for some $d \in D$. These two edges have no weight so the weight of the total weight of the edges in p and p' are the same and order is maintained between paths. Thus by the correctness of the algorithm P our solution finds the shortest path out of all valid paths.

=======Shorter Version=======
Our algorithm considers all valid paths as any path that starts from h_1 must go from t_1 to t_2 and end up at d' which is only connected to all the dropoff points. Note that the extra 2 edges the path must include are both weight 0 so the order is not affected and the correctness of our algorithm follows from the correctness of P.