Exercise 9.11 *Book of Kleinberg and Tardos [10] - Exercise 14 in Chapter 6*

A large collection of mobile wireless devices can naturally form a network in which the devices are the nodes, and two devices x and y are connected by an edge if they are able to directly communicate with each other (e.g., by a short range radio link). Such a network of wireless devices is a highly dynamic object, in which edges can appear and disappear over time as the devices move around. For instance, an edge (x, y) might disappear as x and y move far apart from each other and lose the ability to communicate directly.

In a network that changes over time, it is natural to look for efficient ways of maintaining a path between certain designated nodes. There are two opposing concerns in maintaining such a path: we want paths that are short, but we also do not want to have to change the path frequently as the network structure changes. That is, we would like a single path to continue working, if possible, even as the network gains and loses edges. Here is a way we might model this problem.

Suppose we have a set of mobile nodes V, and a particular point in time there is a set E_0 of edges among these nodes. As the nodes move, the set of edges changes from E_0 to E_1, then to E_2, then to E_3, and so on, to an edge set E_b. For $i = 0, 1, 2, \ldots, b$, let G_i denote the graph (V, E_i). So, if we were to watch the structure of the network on the nodes V as a "time lapse," it would look precisely like the sequence of graphs $G_0, G_1, G_2, \ldots, G_{b-1}, G_b$. We will assume that each of these graphs G_i is connected.

Now consider two particular nodes $s, t \in V$. For an $s - t$ path P in one of the graphs G_i, we define the length of P to be simply the number of edges in P, and we denote this $\ell(P)$. Our goal is to produce a sequence of paths $P_0, P_1, \ldots, P_b$ so that for each i, P_i is an $s - t$ path in G_i. We want the paths to be relatively short. We also do not want that there is too many changes - points at which the identity of the paths switches. Formally, we define $\textsc{changes}(P_0, P_1, \ldots, P_b)$ to be the number of indices i $(0 \leq i \leq b - 1)$ for which $P_i \neq P_{i+1}$.

Fix a constant $K > 0$. We define the cost of the sequence of paths $P_0, P_1, \ldots, P_b$ to be

$$\textsc{cost}(P_0, P_1, \ldots, P_b) = \sum \ell(P_i) + K \times \textsc{changes}(P_0, P_1, \ldots, P_b).$$

(i) *Suppose it is possible to choose a single path P that is an s, t-path in each of the graphs $G_0, G_1, \ldots, G_b$. Give a polynomial time algorithm to find the shortest such path.*

(ii) *Give a polynomial-time algorithm to find a sequence of paths $P_0, P_1, \ldots, P_b$ of minimum cost, where P_i is an $s-t$ path in G_i for $i = 0, 1, \ldots, b$.*

Solution

Consider two particular nodes $s, t \in V$.

(i) Assuming it is possible to choose a single path P that is an s,t-path in each of the graphs $G_0, G_1, \ldots, G_b$, we first build a graph $G = (V, E)$ such that $E = \bigcap_{i=0}^{b} E_i$ as the paths that are in each of the graphs $G_0, G_1, \ldots, G_b$ must only use edges that are present in all the graphs.

Constructing G can be done in $O(mb)$ using a $2 \times m$ array A where $m = \min_{i=0,1,\ldots,b} |E_i|$. First store in A the edges of the graph $G_{i^\star}$ such that $i^\star = \arg \min_{i=0,1,\ldots,b} |E_i|$ in lexicographic order, ordering each edge $\{v_k, v_\ell\}$ with $k < \ell$, and then considering the edges in order of increasing k first, and then of increasing ℓ for a given k. Once A is filed with the ordered edges of $G_{i^\star}$, examine in turn each set E_i for $i \neq i^\star$. If E_i does not contain an edge of $E_{i^\star}$, remove that edge from $E_{i^\star}$ (this can be done by setting $A[1,j] = A[2,j] = 0$ if that edge is the jth edge of $E_{i^\star}$). The resulting array contains the set of G. Computational complexity: $O(mb)$ assuming the edges of each graph are ordered in lexicographical order. If the edges are not ordered, one can use an array B of size $n(n-1)/2$ (all possible edges), where $B[j] = 1$ if E_i contains edge $\{k, \ell\}$ where

$$j = \frac{n(n-1)}{2} - \frac{(n-k)(n-k+1)}{2} + \ell - k,$$

and 0 otherwise. Then, when going through the edges of the other set of edges, set $B[j] = 0$ if the edge (k, ℓ) associated with j does not belong to one of these sets.

Then, we compute the shortest path between s and t in G using, e.g., Dijkstra's algorithm that computes in $\theta((n+m)\log n)$ all the shortest paths from s to all the nodes of E.

(ii) If there exists an optimal solution for graphs $G_0, G_1, \ldots, G_b$, then the restricted solution to $G_0, G_1, \ldots, G_{b-1}$ should be optimal as well. We can therefore use dynamic programming.

Consider the graph $G_{ij} = (V, E_{ij})$ such that $E_{ij} = E_i \cap E_{i+1} \cap \cdots \cap E_j$. Using the previous question *(i)*, we can compute the shortest paths for all pairs of nodes in G_{ij}. If, for a pair of nodes v_k, v_ℓ, there exists no path, we set of the length $\ell(k,\ell)$ to $+\infty$. Otherwise, $\ell(k,\ell)$ is set to the length of the shortest path between v_k and v_ℓ in G_{ij}. $\ell(j)$ be the shortest path between s and t in G_j.

Let $C(i,j) = \textsc{cost}(P_i, P_{i+1}, \ldots, P_j)$.

$$C(i,j) = \min \begin{cases} \min\{C(i,j-1) + \ell(j)\,; C(i-1,j) + \ell(i)\} + K & \text{if } \ell(i,j) = +\infty \\ (j-i+1) \times \ell(i,j) & \text{if } \ell(i,j) \neq +\infty \end{cases}$$

Complexity.

- $O(|E_j| \log n)$ for computing ℓ_j
- $O(m \log n + mb)$ for computing $\ell(i,j)$ using *(i)*
- $O(n^2)$ for computing $c(i,j)$ assuming $\ell(j)$ and $\ell(i,j)$ are available for all i, j.
- $O(n^2 + m \log n + mb + \max\limits_{j=0,1,\ldots,b} |E_j| \log n) = O(n^2 + mb + \max\limits_{j=0,1,\ldots,b} |E_j| \log n)$ for the overall complexity.