CS390-ATA: Homework 3

Jayden Lee, Saad Sharief

Due Date: **February 06, 2025**

Contents

Exercise 7.3

This problem is meant to model the following scenario. Suppose you drive to campus but you are running late for the CS580 midterm, and in this extreme circumstance you are willing to speed up to 25% over the speed limit. However, you don't want to run too high a risk of getting a ticket, so you decide you are willing to speed for a limited number k of streets. The goal is to get to the exam as fast as possible, going 25% over the speed limit for up to k streets, and obeying the speed limit everywhere else.

Formally, the input consists of a directed graph $G = (V, E)$, positive edge weights $l : E \to R_{>0}$, vertices $s, t \in V$, and an integer $k \in N$. The k-speeding distance from s to t is defined as the minimum total distance of any (s, t)-walk, except there the k largest edges are decreased by a factor of $\frac{4}{5}$. For example, if $k = 3$, and you have an (s, t)-walk with edge lengths 6, 3, 4, 6, 9, 3, 6, 9, 7, 1, then the total distance including the k "speedups" becomes

$$6 + 3 + 4 + 6 + \frac{4}{5} \times 9 + 4 + 6 + \frac{4}{5} \times 9 + \frac{4}{5} \times 7 + 1 = 50 \qquad [1]$$

Design and analyze an algorithm computing the k-speeding distance from s to t. Your running time may depend on k. (You may assume k < n since any shortest walk should be a path anyway).

Solution

We solve this problem by constructing an expanded graph G' that uses multiple level to track how many edges have been "sped up" so far. Each level represents a state where a certain number of speedups have been used. For each vertex $v \in V$ in the original graph, we create $k + 1$ copies, forming nodes v_i for $i \in \{0, 1, ..., k\}$. The index i in node v_i represents vertex v with i speedups used so far.

The edges in G' are constructed as follows. For each edge $(u, v) \in E$ in the original graph, we add two types of connections. First, we add edges between nodes in the same level: (u_i, v_i) with weight $\ell(u, v)$ for all $i \in \{0, ..., k\}$. These edges represent going at normal speed. Second, we add edges that represent using the speedup option: if an edge (u, v) is used as a speedup, it connects different level through edges (u_i, v_{i+1}) with weight $\left(\frac{4}{5}\right)\ell(u, v)$ for all $i \in \{0, ..., k - 1\}$.

To find the optimal solution, we run Dijkstra's algorithm starting from s_0 (representing the start vertex s with 0 speedups used) and find the shortest path to any of the vertices $t_0, t_1, ..., t_k$. The minimum distance among these paths gives us the k-speeding distance from s to t.

The complexity analysis is as follows. The new graph G' has $O(nk)$ vertices and $O(mk)$ edges, where n and m are the number of vertices and edges in the original graph. Constructing G' takes $O((m + n)k)$ time. Running Dijkstra's algorithm on G' takes $O(mk \log(nk))$ time. The total running time is therefore $O((m + n)k + mk \log(nk))$.

Proof of Correctness

We prove the correctness of our algorithm by showing that the minimum distance computed in G' exactly corresponds to the k-speeding distance in the original graph G. We demonstrate this through first showing that our solution is at least as good as the optimal solution, and then showing it cannot be better.

Consider any walk w' from s_0 to any $t_0 ... t_k$ in our expanded graph G' with total distance D. We can construct a corresponding walk w in the original graph G as follows: For each edge (u_i, v_j) in w', where u_i and v_j are copies of vertices $u, v \in V$ at levels i and j, we include edge (u, v) in w. When we apply speedups to edges in w that correspond to cross-level edges in w' (those of the form (u_i, v_{i+1})), the total distance becomes exactly D.

Now consider any walk w in the original graph G with k-speeding distance D. Let $h = \min\{s, k\}$ represent the number of edges that are sped up, where s is the actual number of speedups used in w. We can construct a corresponding walk w' in G' as follows: Start from vertex s_0 at level 0, and for each edge in w, if the edge is not sped up, use the corresponding edge within the current level, and if the edge is sped up, use the corresponding cross-level edge to advance to the next level

This walk w' will end at vertex t_h and have total distance D. When we apply this construction to the walk in G with minimum k-speeding distance, we prove that the k-speeding distance in G is at least the minimum distance from s_0 to any of $t_0, ..., t_k$ in G'.