1 Problem description

We have access to a minimum-cost (s,t)-flow algorithm. Reduce the shortest (s,t)-path problem to the minimum-cost (s,t) flow problem.

2 Solution

Our input to the (s,t) flow problem is a graph $G = (V,E)$ with edge lengths $l(e), e \in E$, and start and end vertices s and t.

The reduction is as follows:

Define a flow network $G' = (V', E')$.

Each edge and vertex in G will have a corresponding edge or vertex in G', where every edge capacity range is $[0, \infty]$ and the cost $\$(e)$ of each edge e in G' is the same as $l(e)$ in G. The balance function of G' will be -1 for s, 1 for t, and 0 for all other vertices. G' can be constructed in $O(|V| + |E|)$ time by brute force. The shortest path length in G will correspond exactly to the minimum-cost flow in G', or "Negative Cycle" if the minimum-cost flow returns the same.

3 Proof of correctness

Negative cycles: A negative cycle in G implies no feasible minimum-cost (s,t) flow in G', because an arbitrarily small-cost flow can be created by first finding some (s,t) flow and then adding an arbitrary amount of flow cycling through the corresponding negative-cost cycle in G'.

Note that this doesn't work if we are only trying to fail and detect negative cycles in G when the cycle is along a walk from s to t, since flow can be added to a cycle anywhere in the graph while all vertices remain balanced.

No negative cycles:

Let f be a minimum-cost (s,t) path, with cost k_f, and p be a shortest (s,t) path, with length k_p.

To prove $k_f = k_p$, we first need to prove that a minimum cost (s,t) flow in G' will never have an edge with flow value not equal to 1 or 0, provided there are no negative-cost cycles.

If (u,v) is a flow edge in f, where the flow value of (u,v) is greater than 1, there exists a path from s to u and from v to t. This is true because there are no negative cycles in the graph, and therefore if (u,v) was not along a path from s to t, it must be in a cycle due to the balance constraints, and a smaller cost flow could be found by removing that cycle.

Since these two paths exist, choose the two such paths such that minimize the total cost of the paths $s \to u$ and $v \to t$. We can then create an (s,t) flow of cost less than k_f by setting all edges along the path $s \to u \to v \to t$ to flow value 1. This new flow is a minimum cost flow of those containing (u,v). We know it is cheaper than f because the flow value of (u,v) is less than in f, and the total cost of the other edges is at least as small as the total cost of the other edges in f, because of the balance constraint, and the fact that it was selected using the cheapest paths. This is a contradiction which implies that no such edge can exist in f.

If (u, v) is a flow edge in f, where the flow value of (u, v) is in range $(0, 1)$, there must exist at least 2 flow paths from s to t in f, or in other words, some quantity of the 1 unit of outflow from s is traveling to t by a different path than some other quantity. Clearly, however, one of these routes is cheaper, and it would be cheaper to send all flow through it, which is feasible because all the upper bound capacities are ∞. This is a contradiction which implies no such edge can exist. Therefore all flow values must be either 1 or 0.

Case 1: $k_f < k_p$

In this case we can construct a shorter length (s, t) path. Simply take the non-zero flow edges in f and construct a path out of them. They are guaranteed to be a path because of balance rules plus the $0/1$ requirement we already proved, plus the non-use of cycles in f also previously proved. Because of the exact cost/length correspondence, the cost of this path is k_f, which is a contradiction.

Case 2: $k_f > k_p$

In this case we can construct a smaller-cost (s, t) flow by setting the flow value of all edges in p to 1, and all the others to 0. It is easy to see this is a valid flow with cost k_p, which contradicts f being the cheapest flow. Therefore $k_f = k_p$.