COL751 - Lecture 21

1 Min-cost Reachability tree in Digraphs

Let $G = (V, E, wt)$ be a directed weighted graph on n vertices and m edges, with a root node s. We consider the problem of computing directed reachability tree rooted at vertex s having least weight.

For a vertex $v \in V$ in G, let $\mathcal{E}(v) := \arg\min_{e \in \mathrm{IN}(v)} wt(e)$ be an arbitrarily chosen minimum weight edge entering v in G. Further, let $H = \big(V, \cup_{v \neq s} \mathcal{E}(v)\big)$. Observe that if H is a tree rooted at s, then we can simply return H. If not, then it must contain a cycle, say C.

In the following lemma we prove that there exists an optimal reachability tree for our input instance (G, s) that contains all but one edge of cycle C.

Lemma 1. *There exists an optimal solution, say T_0, to instance (G, s) that contains a sub-path from cycle C of length $|C| - 1$.*

Proof: Let $C = (v_1, \ldots, v_\ell, v_1)$, and let T be any optimal solution to (G, s). Without loss of generality assume that none of the internal vertices of TREEPATH(s, v_1, T) lies in cycle C. We compute a new tree T_0 from T by changing parent of each $v_i,\ i \geqslant 2$, to v_{i-1}. It is each to verify that T_0 is a tree rooted at s, and $wt(T_0) \leqslant wt(T)$. This proves the claim. □

Motivated by Lemma 1 we have the following greedy algorithm for computing min-cost reachability tree, wherein, we contract an arbitrary cycle lying in H into a super-node and recurse on the new graph.

foreach $v \neq s$ **do**
    $\mathcal{E}(v) \leftarrow$ an arbitrarily chosen min-cost edge entering v.
    $wt^*(x, v) := wt(x, v) - wt(\mathcal{E}(v))$, for each edge (x, v) entering v.
end
Let $H = \big(V, \cup_{v \neq s} \mathcal{E}(v)\big)$.
if H *is acyclic* **then**
    Return H.
else
    $C \leftarrow$ directed cycle in H.
    Contract C to a single supernode, yielding $G^* = (V^*, E^*, wt^*)$.
    $T^* \leftarrow$ Min-Cost-Reachability-tree(G^*, s).
    Extend T^* to tree T in G by adding all but one edge of cycle C.
    Return T.
end

Algorithm 1: Min-Cost-Reachability-tree(G, s)

Homework: Prove that Algorithm 1 correctly computes Min-Cost Reachability-tree for (G, s). Also formally argue why new weight function wt^* was needed.

Theorem 1. *For any n vertices, m edges digraph $G = (V, E, wt)$ with a designated source node s we can compute a Min-Cost Reachability-tree for (G, s) in $O(mn)$ time.*