Some practice problems

CS218, Winter 2020

Flow
Problem 1 (flow)

Let $G = (V,E)$ be a flow network with source $s$, sink $t$ and integer capacities. Suppose that we are given a maximum flow $f$ in $G$.

- Suppose that the capacity of a single edge $(u,v) \in E$ is increased by 1. Give a $O(n+m)$-time algorithm to update the max flow.
- Suppose that the capacity of a single edge $(u,v) \in E$ is decreased by 1. Give a $O(n+m)$-time algorithm to update the max flow.

Problem 1 solution

**Answer:**

1. Execute one iteration of Ford-Fulkerson. If $(u,v)$ crosses a minimum cut, then the flow will change. In all cases one iteration is enough.
2. The flow $f$ might not be a legal flow (capacities are not satisfied). Look for an augmenting path from $u$ to $v$. If there is one, reroute the flow through the alternative path. If there is no augmenting path, reduce the flow using an augmenting path from $u$ to $s$, and from $t$ to $v$. 
Problem 2 (flow)

Which of the following claims are true and which are false. Justify your answer by giving either a (short) proof or a counterexample.

1. In any maximum flow there are no cycles that carry positive flow. (A cycle \( e_1, \ldots, e_k \) carries positive flow iff \( f(e_1) > 0, \ldots, f(e_k) > 0 \).)
2. There always exists a maximum flow without cycles carrying positive flow.
3. If all edges in a graph have distinct capacities, there is a unique maximum flow.
4. If we increase all edge capacities by a positive number \( c \), the minimum cut(s) remains unchanged.
5. If we multiply all edge capacities by a positive number \( c \), the minimum cut(s) remains unchanged.

Problem 2 Solution

1. False. Build a flow network with \( V = \{s, u, v, t\} \) and \( E = \{(s, u), (u, v), (v, t)\} \) with capacities \( c(s, u) = 2, c(u, v) = 1, c(v, t) = 1 \). A max flow is \( f(s, u) = 2, f(u, v) = 1, f(v, t) = 1 \) of value \( |f| = 1 \).

2. True. Let \( f \) be a maximum flow and let \( C \) be a cycle with positive flow. Let \( \delta = \min_{e \in C} f(e) \). Reducing the flow of each edge in \( C \) by \( \delta \) maintains the value of the flow and sets the flow \( f(e) \) of at least one of the edges \( e \in C \) to zero.

3. False. Build a flow network with \( V = \{s, u, v, t\} \) and \( E = \{(s, u), (u, v), (v, t)\} \) with capacities \( c(s, u) = 1, c(u, v) = 2, c(v, t) = 3 \). A max flow is \( f(s, u) = 1, f(u, v) = 0, f(v, t) = 1 \) of value \( |f| = 1 \). Also another flow is \( c(s, u) = 1, c(u, v) = 0, c(v, t) = 1 \).

4. False. Build a flow network with \( V = \{s, u, v_1, v_2, v_3, t\} \) and \( E = \{(s, u), (u, v_1), (u, v_2), (u, v_3), (v_1, t), (v_2, t), (v_3, t)\} \) with capacities \( c(s, u) = 4, c(u, v_1) = 2, c(u, v_2) = 2, c(u, v_3) = 2, c(v_1, t) = 1, c(v_2, t) = 1, c(v_3, t) = 1 \). The minimum cut is between \( v_1, v_2, v_3 \) and \( t \), if we increase all the capacities by one, the new min cut is between \( s \) and \( t \).

5. True. The capacity of every cut is multiplied by \( c \), thus the minimum cut(s) remains unchanged.
Problem 3 (flow)

You are given a flow network $G$ with source $s$, sink $t$ and unit-edge capacities, i.e., $c(u,v) = 1$ for any $(u,v) \in E$. You are also given a parameter $k$. We want to delete $k$ edges from $G$ so as to reduce the max flow in $G$ as much as possible. In other words, you should find a set of edges $F \subseteq E$ such that $|F| = k$ and the maximum flow in $G' = (V, E - F)$ is as small as possible. Give a polynomial time algorithm to solve this problem. Analyze the time complexity of your solution.

Algorithm: Compute the min s-t cut $(A,B)$ of network $G$ corresponding to a max flow $f$ of value $|f|$. Consider the set $F$ of directed edges from $A$ to $B$. If $|F| \geq k$, then we can just remove any $k$ edges from $F$, otherwise we just remove all edges in $F$ (and some other edges in $G$ to get to $k$). In any case, the cut $(A,B)$ is still a min cut in the new network, so by the max-flow min-cut theorem, the new flow is $|f| - |F|$. Deleting $k$ edges can’t reduce the value of the maximum flow more than that, so this is optimal. The value of the max flow is at most $n - 1$, since there are at most $n - 1$ edges out of $s$, each of which can carry at most 1 unit of flow. Each iteration takes $O(m)$ time, the total complexity of FF is $O(nm)$. 

Problem 3 Solution