CSCI570 - Fall 2019 - HW8 Solution

1 Graded Questions

1.1 Question 1

Solution No.

Consider a counter-example graph G. It has three nodes s, t, v. s is the source node and t is the sink node. It has two edges $(s, v), (v, t)$ and each edge has a capacity of 1. The max-flow is 1 and increasing the capacity of any edge won't increase the max-flow.

1.2 Question 2

Solution First thing to notice is that a cycle p containing s and v exists in f. One can construct the other flow f' as follows: $\forall(v_1, v_2) \in p, f'(v_1, v_2) = f(v_1, v_2) - 1$. Obviously, $f'(v, s) = 0$. Since the incoming flow value of the source decrease by 1 and the outcoming flow value also decrease by 1, we have $|f'| = |f|$.

Now we need to show f' is valid. Since we decrease the flow value of each edge in the cycle p, all the nodes in the cycle p (except the source s) still have $\sum f(v, \cdot) = \sum f(\cdot, v), \forall v \in p, v \neq s$. Besides, since $f(v, s) = 1$, the new flow value of each edge in p will be at least 1. Thus, the new flow value of each edge in p will be larger or equal to 0. Since we are decreasing the flow value, so the new flow value of each edge will not exceed its capacity.

One can find a cycle starting at s in $O(|E|)$ time (using BFS/DFS). Once the loop is found, the rest can also be done in $O(|E|)$ time.

1.3 Question 3

Denote $c(e)$ to be the capacity of edge e in $G = (V, E)$ and $c'(e)$ be the capacity of the same edge e in the new constructed graph $G' = (V, E)$. To achieve the goal, one modification is that $c'(e) = (|E| + 1) \times c(e)$. This way, the min-cut in G won't be a min-cut in G' if it has more edges than the min-cut in G with the least number of edges.

1.4 Question 4

The problem is equivalent to solving max-flow in the following graph.

The graph is construct so that there are $n + k + 2$ nodes

- there's a source node s and a sink node t.
- the i-th client has a node, say u_i.
- the j-th station has a node, say v_j.
- for each client node u, there's an edge (s, u) with capacity 1.
- for each station node v, there's an edge (v, t) with capacity L.
- if the distance between i-th client and j-th station is within r, there's an edge (u_i, v_j) with capacity 1.

We then compute the max-flow of the constructed graph. If the value of the max-flow is n, then we can find the arrangement.

The graph has $O(n + k)$ nodes and at most $O(nk)$ edges. The running time will be polynomial.