CS4234: Optimization Algorithms

Problem Set 5

Solution Sketches

S-1.

Problem 1.a. In Figure 1, there is a flow network with three flows missing. That is, it is a directed graph, where each edge has a capacity and a flow, except for three edges where the flow is missing. Determine the proper value for the missing flows x, y, and z.

Solution: The value of the missing variables is:

$$\begin{aligned} x &= 2 \\ y &= 5 \\ z &= 13 \end{aligned}$$

This can be determined by making sure the flow balances at every node.

Problem 1.b. What is the value of the flow in Figure 1?

Solution: The value of the flow is 13. This can be easily seen by looking at the flow outgoing from the source.

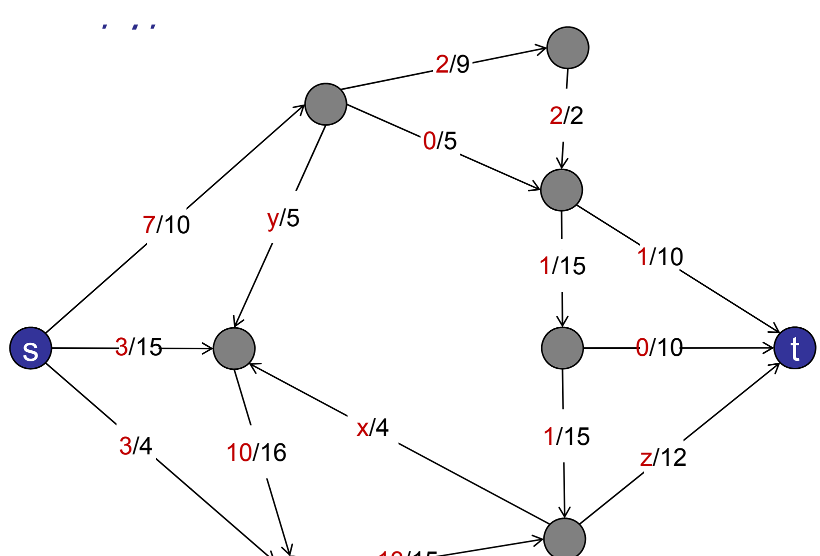

Figure 1: Flow network with three flows (x, y, z) missing.

Problem 1.c. Prove or disprove the following statement: If every edge e in a flow network has a unique value for its capacity (i.e., no two edges have the same capacity), then there is one unique maximum flow.

Solution: This statement is false. Imagine a network with two disjoint paths from s to node A each with cumulative capacity 10, and a path from A to node t with capacity 5. (Notice that the paths from s to A may consist of multiple edges, e.g., a path with edges of capacity 3 and 7 and a path with edges of capacity 4 and 6. Please draw a picture.) There are > 1 possible maximum flows, notably, the 5 units of flow from s to A could travel either of the two different paths (or both).

Problem 1.d. Prove or disprove the following statement: Given a flow network G, if we replace a directed edge e with two directed edges e_1 and e_2 in opposite directions, and with the same capacity, then the value of the maximum flow remains unchanged.

Solution: This is false. A simple example is the case where the graph G contains only a single directed edge from t to s with capacity 10. In graph G the maximum flow is zero, as there is no path from s to t. By replacing the directed edge with two directed edge, the maximum flow increases to 10.

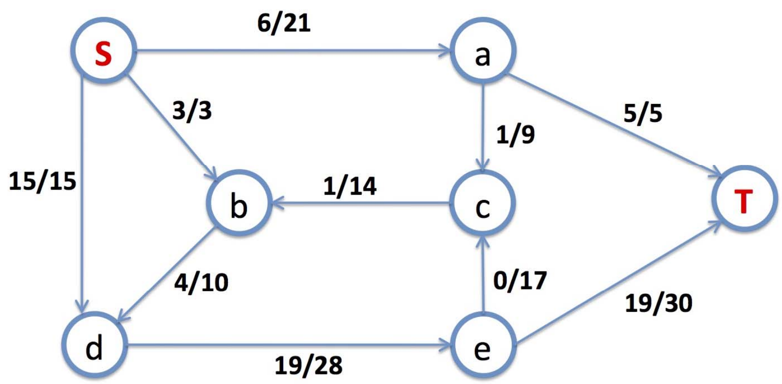

Figure 2: Flow network for Problem S-2 with non-maximum flow.

S-2. See the flow network in Figure 2. Execute one iteration of Ford-Fulkerson on this flow network, and identify: (a) the augmenting path; (b) the bottleneck edge; and (c) the new flow assignment after the iteration is complete.

Solution: In our execution of Ford-Fulkerson, we find an augmenting path from $s \to a \to c \to b \to d \to e \to T$. The bottleneck edge on this path is from $(b \to a)$ and has residual capacity 6. The new flow assignment is:

$$\begin{aligned}
(S, a) &= 12/21 \\
(a, c) &= 7/9 \\
(c, b) &= 7/14 \\
(b, d) &= 10/10 \\
(d, e) &= 25/28 \\
(e, T) &= 25/30
\end{aligned}$$

The resulting flow has value 30. That is, in fact, the maximum flow in this graph. (Consider the cut with nodes $\{S, a, b, c\}$ on one side; the edges crossing the cut have capacity 30.)

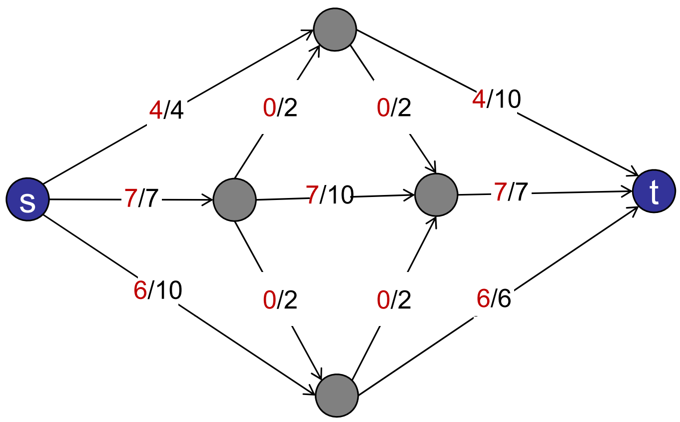

Figure 3: Flow network for Problem S-3.

S-3.
Problem 3.a. What is the value of the flow in Figure 3?

Solution: The value of the flow is 17.

Problem 3.b. Is this a maximum flow? If not, find a maximum flow. (*Hint:* draw the residual graph.)

Solution: This is not the maximum flow. The maximum flow has value 19. (Draw a picture. Note the augmenting path in the residual graph with 2 units of capacity.) This is the maximum flow: consider the cut where one set of nodes contains s and the bottom node of the diamond. This cut has capacity 19.

Problem 3.c. Find an st-cut with minimum capacity in the graph in Figure 3. Draw the cut, indicating which vertices are in each half of the cut. Prove that your cut is, in fact, the minimum cut. *(Note: This was not fully covered at the end of class. See if you can figure it out. If not, just skip it.)*

Solution: The minimum capacity cut consists of the source and the bottom node of the diamond. The cut has capacity 19, and the flow found in the previous part has capacity 19. Hence this is the minimum capacity cut.

S-4. Assume you are given a flow network that you want to disrupt. In this flow network, every edge has capacity exactly 1. That is, you are given a directed graph $G = (V, E)$ with a specified source s and target t where for every edge e, $c(e) = 1$. You are also given as input an integer k.

Find a set of exactly k edges that, when deleted, reduce the maximum flow in G as much as possible. That is, find a set E' of k edges that minimize the maximum flow in the graph $G = (V, E \setminus E')$. Give an algorithm for finding these k edges and prove that it is correct.

Solution: The goal here is to reduce the maximum flow, and hence the minimum cut, as much as possible. First, find the minimum capacity cut. Let (A, B) be the *st*-cut. If the original flow is F, then this cut must have F edges crossing from A to B. Delete any k of those edges. Now, the minimum cut is at most $F - k$, and hence the maximum flow is at most $F - k$.

Notice that it is impossible to reduce the maximum flow by more than k. Assume in some alternate solution that the minimum capacity cut in G' is $< F - k$. Now, add back the k edges there were deleted—this can increase the capacity of the cut by at most k, and the resulting minimum cut as $< F$, which is a contradiction.

As an exercise, what goes wrong with this algorithm/analysis if edges may have arbitrary capacity, i.e., the capacity of an edge may be > 1? Why does this approach fail?