Name: Kevina Wong
ID: 109179049

CSCI 3104, Algorithms **Profs. Chen & Grochow**
Problem Set 8 – Due Thurs Apr 2 11:55pm **Spring 2020, CU-Boulder**

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

Advice 2: Informal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

Instructions for submitting your solutions:

- All submissions must be typed.
- You should submit your work through the **class Canvas page** only.
- You may not need a full page for your solutions; pagebreaks are there to help Gradescope automatically find where each problem is. Even if you do not attempt every problem, please allot at least as many pages per problem (or subproblem) as are allotted in this template.

Quicklinks: 1a 1b 2a 2b

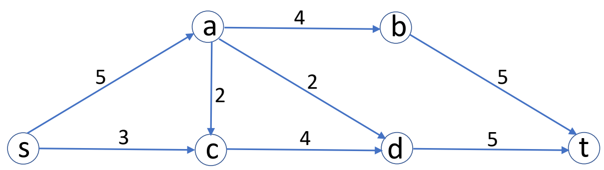
Name: Kevina Wong

ID: 109179049

CSCI 3104, Algorithms | **Profs. Chen & Grochow**

Problem Set 8 – Due Thurs Apr 2 11:55pm | **Spring 2020, CU-Boulder**

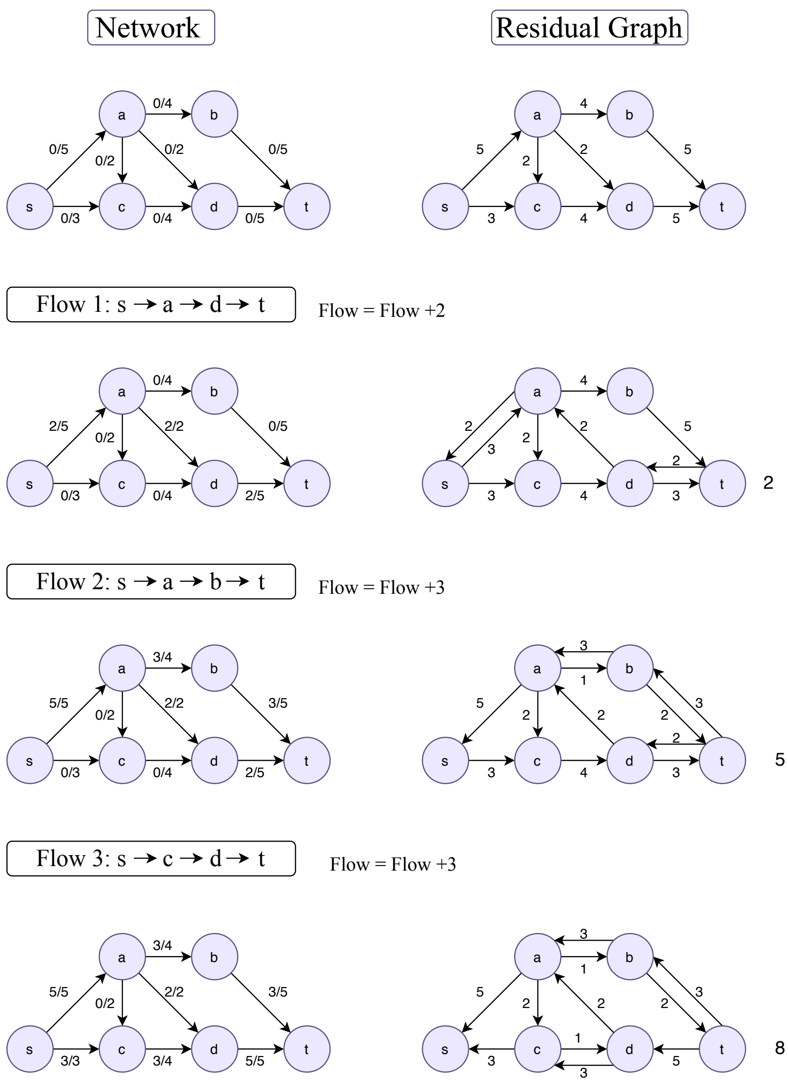
1. *Consider the following flow network, with each edge labeled by its capacity:*



(a) Using the Ford-Fulkerson algorithm, compute the maximum flow that can be pushed from $s \to t$. **You must use** $s \to a \to d \to t$ **as your first flow-augmenting path.**

In order to be eligible for full credit, you must include the following:

- The residual network for each iteration, including the residual capacity of each edge.
- The flow augmenting path for each iteration, including the amount of flow that is pushed through this path from $s \to t$.
- The updated flow network **after each iteration**, with flows for each directed edge clearly labeled.
- The maximum flow being pushed from $s \to t$ after the termination of the Ford-Fulkerson algorithm.



The algorithm, then, terminates with a maximum flow being pushed from $s \to t$ of 8.

(b) The Ford-Fulkerson algorithm will terminate when there is no longer an augmenting path on the residual network. At this point, you can find a minimum capacity cut. Indicate this cut and its capacity, and verify if max flow min cut theorem holds.

Answer:

A cut on a graph g will partition the vertices into two subsets of vertices. Our goal to find the minimum cut that will disconnect s and t while being the lowest capacity of all possible cuts. In this problem, our minimum cut s,t is the set $s, t = \{s \to a, s \to c\}$. The capacity of $\|s, t\| = 5 + 3 = 8$. The max flow min cut theorem says that the value of the maximum flow equals the capacity of the minimum cut. Based on the indicated cut s, t, capacity $\|s, t\|$, and our residual graph, we can verify that the theorem holds as the max flow is 8 and the min cut capacity is also 8.

Name: Kevina Wong
ID: 109179049

2. In this problem, we seek to generalize the max-flow problem from class to allow for multiple sources and sinks. Given a directed graph $G = (V, E)$ with capacity $c(u, v) > 0$ for each edge $(u, v) \in E$ and demand $r(v)$ at each vertex $v \in V$, a routing of flow is a function f such that

   - **(capacity contraint)** for all $(u, v) \in E$, $0 \le f(u, v) \le c(u, v)$, and
   - **(flow conservation)** for all $v \in V$,

   $$\sum_{u:(u,v)\in E} f(u, v) - \sum_{u:(v,u)\in E} f(v, u) = r(v),$$

   i.e., the total incoming flow minus the total outgoing flow at vertex v is equal to $r(v)$.

   We note that $r(v)$ can be positive, negative, or 0, just as in the max-flow setting from class. In particular, note that:

   - The vertex v is a **source** vertex precisely if $r(v) < 0$.
   - The vertex v is a **sink** vertex precisely if $r(v) > 0$.
   - The vertex v is neither a source nor a sink precisely if $r(v) = 0$.

   These conditions are the same as in the single-source, single-sink max-flow model from class.

   (a) Show how to find a routing or determine that one does not exist by reducing to a maximum flow problem. In other words, given a graph, c and r as above, construct a related graph H with capacities $c' \colon E(H) \to \mathbb{R}$ and $s, t \in V(H)$ such that you can use the maximum $s \to t$ flow on H to determine whether a routing exists in G. Hint: You may want to take a look at Chapter 7.5 of the recommended text by Kleinberg and Tardos to get some insight.

Answer:

When given a graph G = (V,E) with multiple sources and multiple sinks, it can be reduced to a single-source, single-sink graph G'=(V',E'). Thus, we can reduce this to a maximum flow problem. This can be done by adding two vertices: one being our new single source, $S \in V'$, and the other being our new single sink, $T \in V'$. Then, we must add edges. There must be edges, $e_{SA}, e_{SB}, e_{SC}... \in E'$ that connects our new source S to every one of our old sources $A, B, C, ... \in V$ and edges, $e_{TX}, e_{TY}, e_{TZ}, ... \in E'$ that connects our new sink, T to every one of our old sinks, $X, Y, Z, ... \in V$. Now, notice that each node in G has a demand, r, and each edge in G has a capacity c. Maximum flow problems do not have node demands, but they do have edge capacities. As specified from the flow conservation, we know that the demand of a vertex is equal to the incoming flow minus the outgoing flow. Therefore, we have to appropriately weight our new edges to account for the node demands, and the capacities of our new edges should be the equal to the absolute value of the demand of the vertex, $v \in V$ it is connected to. For the original multiple-source, multiple-sink problem, G, the way to know if a routing exists or not, the demands must be met exactly for a routing to exist. Similarly, to see if a routing exists for our new single-source, single-sink graph, G', the capacities of the new edges in E' must be filled exactly for a routing to exist. Otherwise, we can say that a routing does not exist. The way to find the routing is to find a maximum flow graph of G' that when applied to our original graph, G, it is still applicable.

(b) Suppose that additionally you are given a lower bound $l(u,v) > 0$ at each edge (u,v), and we are looking for a routing f satisfying

- **(lower bounds)** for all $(u,v) \in E$, $f(u,v) \geq l(u,v)$,

in addition to the **capacity constraint** and the **flow conservation**. Show how to find such a routing or determine that one does not exist by reducing to a maximum flow problem. Hint: You may want to solve this problem by reducing it to (a). Think about how to modify the graph and demands to equivalently enforce lower bounds $l(u,v)$ on the flows.

Answer:

Essentially, we would want to reduce the problem where we can have the same graph but with no lower bounds. Then, from there, we can apply the method from problem 2a and solve for a routing by turning it into a maximum flow problem. First, we know that we have an edge, e, in a graph $G = (V, E)$ with a lower bound of l_e. This means that at least l_e must be pushed. In addition, this e also has a capacity that we can call c_e. Let's say that we have an arbitrary flow, f_a, such that it fulfills both l_e and c_e. Further, recall that they specify that for every $v \in V$, there is a demand that must be fulfilled. This can be expressed as $f_{in}(v) - f_{out}(v) = d(v)$. We can create a hypothetical flow of $f_h = l_e$, so that it fulfils both the capacity and the lower bound. If $f_h^{in} - f_h^{out}$, let's call this f_h^{net}, is equal to the demand, then we can say that the condition is satisfied. Otherwise, we need to add two nodes and three edges creating a circulation in order to level out the imbalance of flow. Let's call this new graph $G' = (V', E')$ such that $V \in V'$ and $E \in E'$. However, in G', e does not have a lower bound. Now, since we had already pushed l_e on e, we know that the remaining flow we have left in e is $c_e - l_e$. Therefore, in G', the demand for our two new nodes is $-l_e$ and l_e. For our three added edges, the two side edges connecting the circulation would have the same capacity, c_e, and the edge connecting the two new nodes would have a capacity of $c_e - l_e$. With this, we are guaranteed that the lower bound is being pushed without the lower bound actually being there. This process will yield a multi-source, multi-sink graph, which we can then apply our process from 2a upon to further simplify it to a maximum flow problem and find whether a routing exists or not.