CMSC 451: Lecture 14
Network Flows: Basic Definitions
Tuesday, Oct 31, 2017

Reading: Sect. 7.1 in KT.

Network Flow: "Network flow" is the name of a variety of related graph optimization problems, which are of fundamental value. We are given a *flow network*, which is essentially a directed graph with nonnegative edge weights. We think of the edges as "pipes" that are capable of carrying some sort of "stuff." In applications, this stuff can be any measurable quantity, such as fluid, megabytes of network traffic, commodities, currency, and so on. Each edge of the network has a given *capacity*, that limits the amount of stuff it is able to carry. The idea is to find out how much flow we can push from a designated source node to a designated sink node.

Although the network flow problem is defined in terms of the metaphor of pushing fluids, this problem and its many variations find remarkably diverse applications. These are often studied in the area of operations research. The network flow problem is also of interest because it is a restricted version of a more general optimization problem, called *linear programming*. A good understanding of network flows is helpful in obtaining a deeper understanding of linear programming.

Flow Networks: A *flow network* is a directed graph $G = (V, E)$ in which each edge $(u, v) \in E$ has a nonegative *capacity* $c(u, v) \geq 0$. (In our book, the capacity of edge e is denoted by c_e.) If $(u, v) \notin E$ we model this by setting $c(u, v) = 0$. There are two special vertices: a *source* s, and a *sink* t (see Fig. 1).

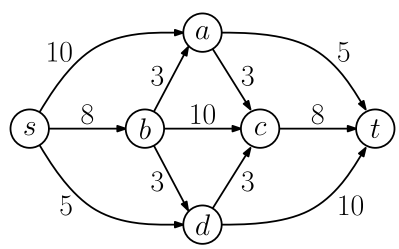


Fig. 1: A flow network.

We assume that there is no edge entering s and no edge leaving t. Such a network is sometimes called an *s-t network*. We also assume that every vertex lies on some path from the source to the sink.[1] This implies that $m \geq n-1$, where $n = |V|$ and $m = |E|$. It will also be convenient to assume that all capacities are integers. (We can assume more generally that the capacities are rational numbers, since we can convert them to integers by multiplying them by the least common multiple of the denominators.)

[1]Neither of these is an essential requirement. Given a network that fails to satisfy these assumptions, we can easily generate an equivalent one that satisfies both.

Flows, Capacities, and Conservation: Given an s-t network, a *flow* (also called an s-t *flow*) is a function f that maps each edge to a nonnegative real number and satisfies the following properties:

Capacity Constraint: For all $(u, v) \in E$, $f(u, v) \leq c(u, v)$.

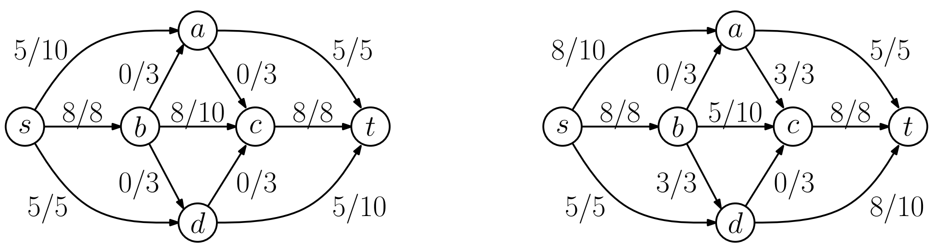
Flow conservation (or flow balance): For all $v \in V \setminus \{s, t\}$, the sum of flow along edges into v equals the sum of flows along edges out of v.

We can state flow conservation more formally as follows. First off, let us make the assumption that if (u, v) is *not* an edge of E, then $f(u, v) = 0$. We then define the total flow into v and total flow out of v as:

$$f^{\text{in}}(v) \;=\; \sum_{(u,v)\in E} f(u, v) \qquad \text{and} \qquad f^{\text{out}}(v) \;=\; \sum_{(v,w)\in V} f(v, w).$$

Then flow conservation states that $f^{\text{in}}(v) = f^{\text{out}}(v)$, for all $v \in V \setminus \{s, t\}$. Note that flow conservation *does not* apply to the source and sink, since we think of ourselves as pumping flow from s to t.

Two examples are shown in Fig. 2, where we use the notation f/c on each edge to denote the flow f and capacity c for this edge.



A valid flow ($|f| = 18$) A maximum flow ($|f| = 21$)

(a) (b)

Fig. 2: A valid flow and a maximum flow.

The quantity $f(u, v)$ is called the *flow* along edge (u, v). We are interested in defining the total flow, that is, the total amount of fluid flowing from s to t. The *value* of a flow f, denoted $|f|$, is defined as the sum of flows out of s, that is,

$$|f| \;=\; f^{\text{out}}(s) \;=\; \sum_{w\in V} f(s, w),$$

(For example, the value of the flow shown in Fig. 2(a) is 5+8+5 = 18.) From flow conservation, it follows easily that this is also equal to the flow into t, that is, $f^{\text{in}}(t)$. We will prove this later.

Maximum Flow: Given an s-t network, an obvious optimization problem is to determine a flow of maximum value. More formally, the *maximum-flow problem* is, given a flow network

$G = (V, E)$, and source and sink vertices s and t, find the flow of maximum value from s to t. (For example, in Fig. 2(b) we show flow of value $8 + 8 + 5 = 21$, which can be shown to be the maximum flow for this network.) Note that, although the value of the maximum flow is unique, there may generally be many different flow functions that achieve this value.

Path-Based Flows: The definition of flow we gave above is sometimes call the *edge-based* definition of flows. An alternative, but mathematically equivalent, definition is called the *path-based* definition of flows. Define an *s-t path* to be any simple path from s to t. For example, in Fig. 1, $\langle s, a, t\rangle$, $\langle s, b, a, c, t\rangle$ and $\langle s, d, c, t\rangle$ are all examples of s-t paths. There may generally be an exponential number of such paths (but that is alright, since this just a mathematical definition).

A *path-based flow* is a function that assigns each s-t path a nonnegative real number such that, for every edge $(u, v) \in E$, the sum of the flows on all the paths containing this edge is at most $c(u, v)$. Note that there is no need to provide a flow conservation constraint, because each path that carries a flow into a vertex (excluding s and t), carries an equivalent amount of flow out of that vertex. For example, in Fig. 3(b) we show a path-based flow that is equivalent to the edge-based flow of Fig. 3(a). The paths carrying zero flow are not shown.

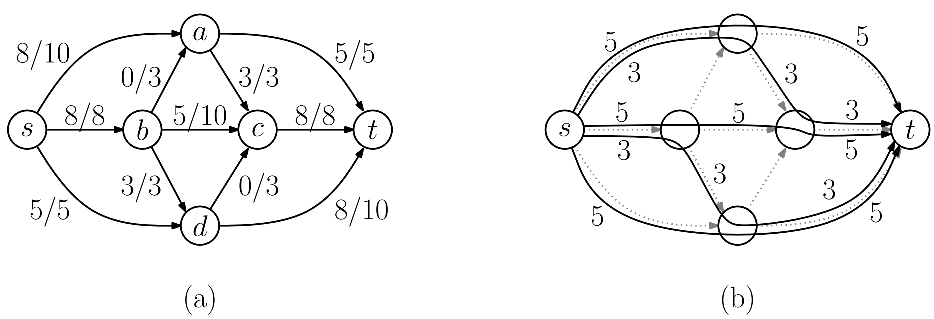


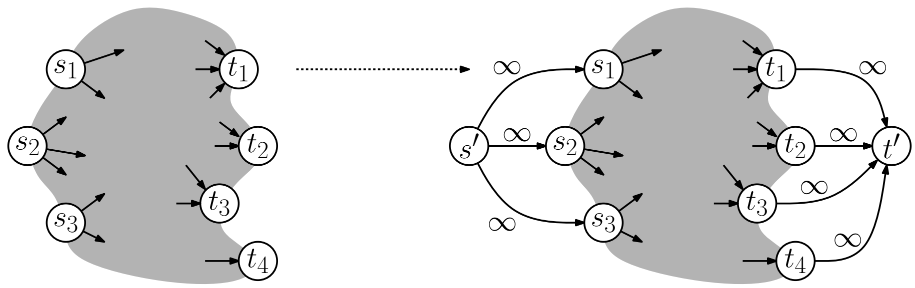
Fig. 3: (a) An edge-based flow and (b) its path-based equivalent.

The *value* of a path-based flow is defined to be the total sum of all the flows on all the s-t paths of the network. Although we will not prove it, the following claim is an easy consequence of the above definitions.

Claim: Given an s-t network G, under the assumption that there are no edges entering s or leaving t, G has an edge-based flow of value x if and only if G has a path-based flow of value x.

Multi-source, multi-sink networks: It may seem overly restrictive to require that there is only a single source and a single sink vertex. Many flow problems have situations in which many source vertices $s_1, \ldots, s_k$ and many sink vertices $t_1, \ldots, t_l$. This can easily be modeled by just adding a special *super-source* s' and a *super-sink* t', and attaching s' to all the s_i and attach all the t_j to t'. We let these edges have infinite capacity (see Fig. 4). Now by pushing the maximum flow from s' to t' we are effectively producing the maximum flow from all the s_i's to all the t_j's.

Note that we don't assume any correspondence between flows leaving source s_i and entering t_j. Flows from one source may flow into *any* sink vertex. In some cases, you would like to



(a) (b)

Fig. 4: Reduction from (a) multi-source/multi-sink to (b) single-source/single-sink.

specify the flow from a certain source must arrive at a designated sink vertex. For example, imagine that the sources are manufacturing production centers and sinks are retail outlets, and you are told the amount of commodity from s_i to arrive at t_j. This variant of the flow problem, called the *multi-commodity flow problem*, is a much harder problem to solve (in fact, some formulations are NP-hard).