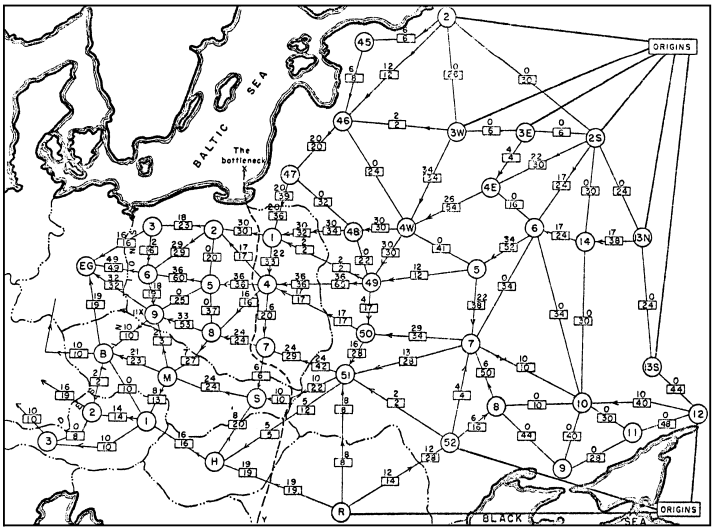
Col. Hogan: *One of these wires disconnects the fuse,*
the other one fires the bomb.
Which one would you cut, Shultz?
Sgt. Schultz: *Don't ask me, this is a decision for an officer.*
Col. Hogan: *All right. Which wire, Colonel Klink?*
Col. Klink: *This one. [points to the white wire]*
Col. Hogan: *You're sure?*
Col. Klink: *Yes.*
[Hogan cuts the black wire; the bomb stops ticking]
Col. Klink: *If you knew which wire it was, why did you ask me?*
Col. Hogan: *I wasn't sure which was the right one,*
but I was certain you'd pick the wrong one.

— "A Klink, a Bomb, and a Short Fuse", *Hogan's Heroes* (1966)



Harris and Ross's map of the Warsaw Pact rail network

21 Maximum Flows and Minimum Cuts

In the mid-1950s, Air Force researchers T. E. Harris and F. S. Ross published a classified report studying the rail network that linked the Soviet Union to its satellite countries in Eastern Europe. The network was modeled as a graph with 44 vertices, representing geographic regions, and 105 edges, representing links between those regions in the rail network. Each edge was given a weight, representing the rate at which material could be shipped from one region to the next. Essentially by trial and error, they determined both the maximum amount of stuff that could be moved from Russia into Europe, as well as the cheapest way to disrupt the network by removing links (or in less abstract terms, blowing up train tracks), which they called 'the bottleneck'. Their results (including the figure at the top of the page) were only declassified in 1999.[1]

This one of the first recorded applications of the *maximum flow* and *minimum cut* problems. For both problems, the input is a directed graph $G = (V, E)$, along with special vertices s and t called the *source* and *target*. As in the previous lectures, I will use $u \to v$ to denote the directed edge from vertex u

[1]Both the map and the story were taken from Alexander Schrijver's fascinating survey 'On the history of combinatorial optimization (till 1960)'.

to vertex v. Intuitively, the maximum flow problem asks for the largest amount of material that can be transported from one vertex to another; the minimum cut problem asks for the minimum damage needed to separate two vertices.

21.1 Flows

An ***(s, t)-flow*** (or just a ***flow*** if the source and target are clear from context) is a function $f : E \to \mathbb{R}_{\geq 0}$ that satisfies the following ***conservation constraint*** at every vertex v except possibly s and t:

$$\sum_u f(u \to v) = \sum_w f(v \to w).$$

In English, the total flow into v is equal to the total flow out of v. To keep the notation simple, we define $f(u \to v) = 0$ if there is no edge $u \to v$ in the graph. The ***value*** of the flow f, denoted $|f|$, is the total net flow out of the source vertex s:

$$|f| := \sum_w f(s \to w) - \sum_u f(u \to s).$$

It's not hard to prove that $|f|$ is also equal to the total net flow *into* the target vertex t, as follows. To simplify notation, let $\partial f(v)$ denote the total net flow out of any vertex v:

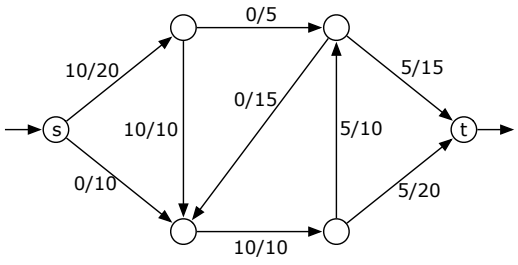
$$\partial f(v) := \sum_u f(u \to v) - \sum_w f(v \to w).$$

The conservation constraint implies that $\partial f(v) = 0$ or every vertex v except s and t, so

$$\sum_v \partial f(v) = \partial f(s) + \partial f(t).$$

On the other hand, any flow that leaves one vertex must enter another vertex, so we must have $\sum_v \partial f(v) = 0$. It follows immediately that $|f| = \partial f(s) = -\partial f(t)$.

Now suppose we have another function $c : E \to \mathbb{R}_{\geq 0}$ that assigns a non-negative ***capacity*** $c(e)$ to each edge e. We say that a flow f is ***feasible*** (with respect to c) if $f(e) \leq c(e)$ for every edge e. Most of the time we will consider only flows that are feasible with respect to some fixed capacity function c. We say that a flow f ***saturates*** edge e if $f(e) = c(e)$, and ***avoids*** edge e if $f(e) = 0$. The ***maximum flow problem*** is to compute a feasible (s, t)-flow in a given directed graph, with a given capacity function, whose value is as large as possible.



An (s, t)-flow with value 10. Each edge is labeled with its flow/capacity.

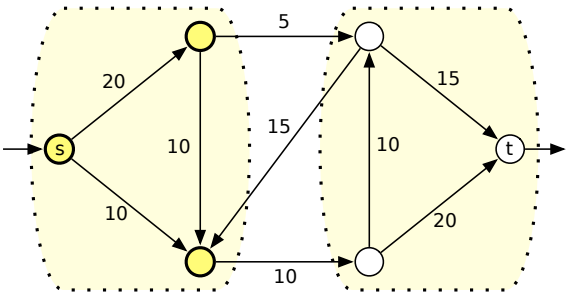
21.2 Cuts

An ***(s, t)-cut*** (or just ***cut*** if the source and target are clear from context) is a partition of the vertices into disjoint subsets S and T—meaning $S \cup T = V$ and $S \cap T = \emptyset$—where $s \in S$ and $t \in T$.

If we have a capacity function $c : E \to \mathbb{R}_{\geq 0}$, the ***capacity*** of a cut is the sum of the capacities of the edges that start in S and end in T:

$$\|S, T\| := \sum_{v \in S} \sum_{w \in T} c(v \to w).$$

(Again, if $v \to w$ is not an edge in the graph, we assume $c(v \to w) = 0$.) Notice that the definition is asymmetric; edges that start in T and end in S are unimportant. The ***minimum cut problem*** is to compute an (s, t)-cut whose capacity is as large as possible.



An (s, t)-cut with capacity 15. Each edge is labeled with its capacity.

Intuitively, the minimum cut is the cheapest way to disrupt all flow from s to t. Indeed, it is not hard to show that **the value of *any* feasible (s, t)-flow is at most the capacity of *any* (s, t)-cut.** Choose your favorite flow f and your favorite cut (S, T), and then follow the bouncing inequalities:

$$\begin{aligned}
|f| &= \sum_w f(s \to w) - \sum_u f(u \to s) && \text{by definition} \\
&= \sum_{v \in S} \left(\sum_w f(v \to w) - \sum_u f(u \to v) \right) && \text{by the convervation constraint} \\
&= \sum_{v \in S} \left(\sum_{w \in T} f(v \to w) - \sum_{u \in T} f(u \to v) \right) && \text{removing duplicate edges} \\
&\leq \sum_{v \in S} \sum_{w \in T} f(v \to w) && \text{since } f(u \to v) \geq 0 \\
&\leq \sum_{v \in S} \sum_{w \in T} c(v \to w) && \text{since } f(u \to v) \leq c(v \to w) \\
&= \|S, T\| && \text{by definition}
\end{aligned}$$

Our derivation actually implies the following stronger observation: ***$|f| = \|S, T\|$ if and only if f saturates every edge from S to T and avoids every edge from T to S.*** Moreover, if we have a flow f and a cut (S, T) that satisfies this equality condition, f must be a maximum flow, and (S, T) must be a minimum cut.

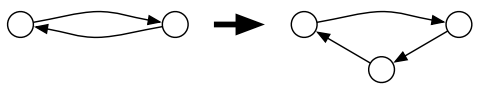
21.3 The Max-Flow Min-Cut Theorem

Surprisingly, for any weighted directed graph, there is always a flow f and a cut (S, T) that satisfy the equality condition. This is the famous *max-flow min-cut theorem*:

The value of the maximum flow is* equal *to the capacity of the minimum cut.

The rest of this section gives a proof of this theorem; we will eventually turn this proof into an algorithm.

Fix a graph G, vertices s and t, and a capacity function $c : E \to \mathbb{R}_{\geq 0}$. The proof will be easier if we assume that the capacity function is ***reduced***: For any vertices u and v, either $c(u{\to}v) = 0$ or $c(v{\to}u) = 0$, or equivalently, if an edge appears in G, then its reversal does not. This assumption is easy to enforce. Whenever an edge $u{\to}v$ and its reversal $v{\to}u$ are both the graph, replace the edge $u{\to}v$ with a path $u{\to}x{\to}v$ of length two, where x is a new vertex and $c(u{\to}x) = c(x{\to}v) = c(u{\to}v)$. The modified graph has the same maximum flow value and minimum cut capacity as the original graph.

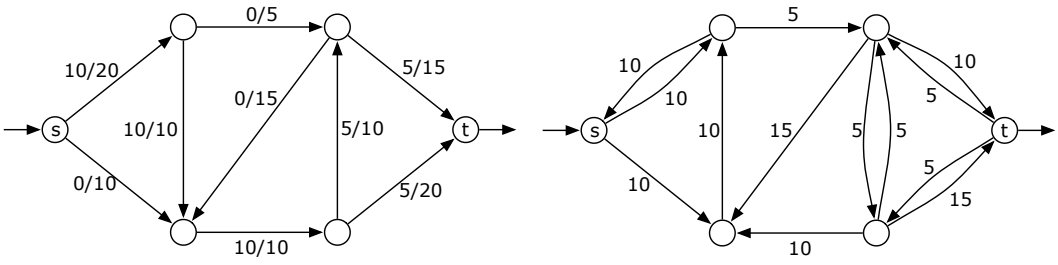


Enforcing the one-direction assumption.

Let f be a feasible flow. We define a new capacity function $c_f : V \times V \to \mathbb{R}$, called the ***residual capacity***, as follows:

$$c_f(u{\to}v) = \begin{cases} c(u{\to}v) - f(u{\to}v) & \text{if } u{\to}v \in E \\ f(v{\to}u) & \text{if } v{\to}u \in E \\ 0 & \text{otherwise} \end{cases}.$$

Since $f \geq 0$ and $f \leq c$, the residual capacities are always non-negative. It is possible to have $c_f(u{\to}v) > 0$ even if $u{\to}v$ is not an edge in the original graph G. Thus, we define the ***residual graph*** $G_f = (V, E_f)$, where E_f is the set of edges whose residual capacity is positive. Notice that the residual capacities are *not* necessarily reduced; it is quite possible to have both $c_f(u{\to}v) > 0$ and $c_f(v{\to}u) > 0$.



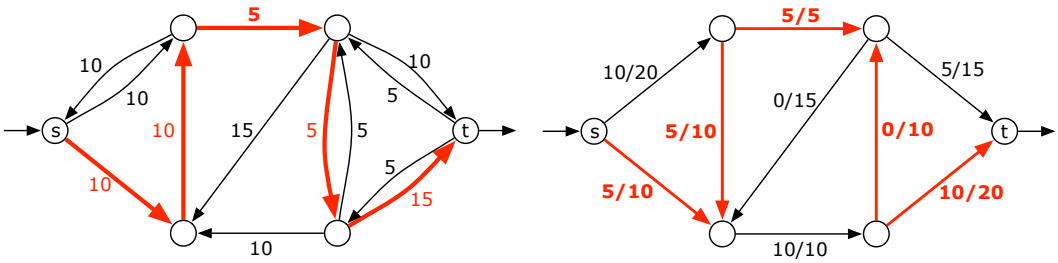
A flow f in a weighted graph G and the corresponding residual graph G_f.

Suppose there is no path from the source s to the target t in the residual graph G_f. Let S be the set of vertices that are reachable from s in G_f, and let $T = V \setminus S$. The partition (S, T) is clearly an (s, t)-cut. For every vertex $u \in S$ and $v \in T$, we have

$$c_f(u{\to}v) = (c(u{\to}v) - f(u{\to}v)) + f(v{\to}u) = 0,$$

which implies that $c(u{\to}v) - f(u{\to}v) = 0$ and $f(v{\to}u) = 0$. In other words, our flow f saturates every edge from S to T and avoids every edge from T to S. It follows that $|f| = \|S, T\|$. Moreover, f is a maximum flow and (S, T) is a minimum cut.

On the other hand, suppose there is a path $s = v_0{\to}v_1{\to}\cdots{\to}v_r = t$ in G_f. We refer to $v_0{\to}v_1{\to}\cdots{\to}v_r$ as an ***augmenting path***. Let $F = \min_i c_f(v_i{\to}v_{i+1})$ denote the maximum amount of flow that we can



An augmenting path in G_f with value $F = 5$ and the augmented flow f'.

push through the augmenting path in G_f. We define a new flow function $f' : E \to \mathbb{R}$ as follows:

$$f'(u \to v) = \begin{cases} f(u \to v) + F & \text{if } u \to v \text{ is in the augmenting path} \\ f(u \to v) - F & \text{if } v \to u \text{ is in the augmenting path} \\ f(u \to v) & \text{otherwise} \end{cases}$$

To prove that the flow f' is feasible with respect to the original capacities c, we need to verify that $f' \geq 0$ and $f' \leq c$. Consider an edge $u \to v$ in G. If $u \to v$ is in the augmenting path, then $f'(u \to v) > f(u \to v) \geq 0$ and

$$\begin{aligned} f'(u \to v) &= f(u \to v) + F && \text{by definition of } f' \\ &\leq f(u \to v) + c_f(u \to v) && \text{by definition of } F \\ &= f(u \to v) + c(u \to v) - f(u \to v) && \text{by definition of } c_f \\ &= c(u \to v) && \text{Duh.} \end{aligned}$$

On the other hand, if the reversal $v \to u$ is in the augmenting path, then $f'(u \to v) < f(u \to v) \leq c(u \to v)$, which implies that

$$\begin{aligned} f'(u \to v) &= f(u \to v) - F && \text{by definition of } f' \\ &\geq f(u \to v) - c_f(v \to u) && \text{by definition of } F \\ &= f(u \to v) - f(u \to v) && \text{by definition of } c_f \\ &= 0 && \text{Duh.} \end{aligned}$$

Finally, we observe that (without loss of generality) only the first edge in the augmenting path leaves s, so $|f'| = |f| + F > 0$. In other words, f is *not* a maximum flow.

This completes the proof!

Exercises

1. Let (S, T) and (S', T') be minimum (s, t)-cuts in some flow network G. Prove that $(S \cap S', T \cup T')$ and $(S \cup S', T \cap T')$ are also minimum (s, t)-cuts in G.

2. Suppose (S, T) is the *unique* minimum (s, t)-cut in some flow network. Prove that (S, T) is also a minimum (x, y)-cut for all vertices $x \in S$ and $y \in T$.

3. Cuts are sometimes defined as subsets of the edges of the graph, instead of as partitions of its vertices. In this problem, you will prove that these two definitions are *almost* equivalent.

   We say that a subset X of (directed) edges *separates s and t* if every directed path from s to t contains at least one (directed) edge in X. For any subset S of *vertices*, let δS denote the set of directed edges leaving S; that is, $\delta S := \{u \to v \mid u \in S, v \notin S\}$.

   (a) Prove that if (S, T) is an (s, t)-cut, then δS separates s and t.

   (b) Let X be an arbitrary subset of edges that separates s and t. Prove that there is an (s, t)-cut (S, T) such that $\delta S \subseteq X$.

   (c) Let X be a *minimal* subset of edges that separates s and t. Prove that there is an (s, t)-cut (S, T) such that $\delta S = X$.

4. A flow f is ***acyclic*** if the subgraph of directed edges with positive flow contains no directed cycles.

   (a) Prove that for any flow f, there is an acyclic flow with the same value as f. (In particular, this implies that some maximum flow is acyclic.)

   (b) A *path flow* assigns positive values only to the edges of one simple directed path from s to t. Prove that every acyclic flow can be written as the sum of $O(E)$ path flows.

   (c) Describe a flow in a directed graph that *cannot* be written as the sum of path flows.

   (d) A *cycle flow* assigns positive values only to the edges of one simple directed cycle. Prove that every flow can be written as the sum of $O(E)$ path flows and cycle flows.

   (e) Prove that every flow with value 0 can be written as the sum of $O(E)$ cycle flows. (Zero-value flows are also called *circulations*.)

5. Suppose instead of capacities, we consider networks where each edge $u \to v$ has a non-negative ***demand*** $d(u \to v)$. Now an (s, t)-flow f is *feasible* if and only if $f(u \to v) \geq d(u \to v)$ for every edge $u \to v$. (Feasible flow values can now be arbitrarily large.) A natural problem in this setting is to find a feasible (s, t)-flow of *minimum* value.

   (a) Describe an efficient algorithm to compute a feasible (s, t)-flow, given the graph, the demand function, and the vertices s and t as input. *[Hint: Find a flow that is non-zero everywhere, and then scale it up to make it feasible.]*

   (b) Suppose you have access to a subroutine MAXFLOW that computes *maximum* flows in networks with edge capacities. Describe an efficient algorithm to compute a *minimum* flow in a given network with edge demands; your algorithm should call MAXFLOW exactly once.

   (c) State and prove an analogue of the max-flow min-cut theorem for this setting. (Do minimum flows correspond to maximum cuts?)