Flows in Graphs	Instructor: Padraic Bartlett

Lecture 1: The Max-Flow Min-Cut Theorem

Week 2 | *Mathcamp 2011*

Graphs are often used as ways to visualize **networks**: i.e. a collection of pipes carrying water, or wires carrying current, or cables carrying data. How can we model this?

1 Basic Definitions

Well: if we're modelling some notion of **flow**, we're intuitively thinking of a way of assigning weights to all of our edges, modelling the amount of (fluid/current/data) travelling through each edge, along with a direction that this flow is travelling. In other words, if we have a directed graph G containing an edge $(x, y) \in E(G)$, we want to assign a value $k = f(x, y)$ to the ordered pair (x, y) to indicate that a flow of k units of current is travelling from x to y on this edge.

What other rules should we ask our flows to obey? Well: often, we like to study flows where we don't have any "buildup" of fluids or current at most nodes: i.e. at most of the nodes in our graph, the total flow into that vertex is the flow out of that vertex. This property is called **Kirchoff's law**, and can be mathematically stated for any vertex v as

$$\sum_{w \in N^+(v)} f(v, w) - \sum_{w \in N^-(v)} f(w, v) = 0.$$

We define a **flow** as any map on a graph G that obeys this property at every node other than two special nodes, the **source** and **sink**:

Definition. A **real-valued flow** on a graph G with a pair of distinguished vertices s, t is a map $f : E \to \mathbb{R}$ that satisfies the following property: for any vertex $v \neq s, t$, we have

$$\sum_{w \in N^+(v)} f(v, w) - \sum_{w \in N^-(v)} f(w, v) = 0.$$

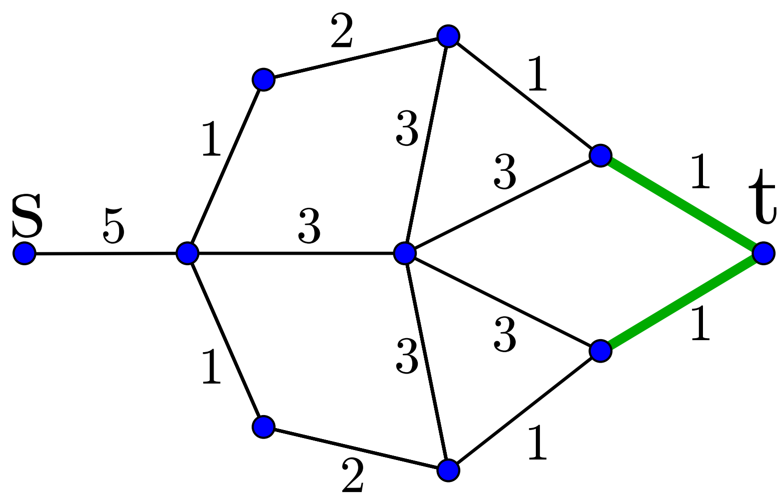
We call such directed graphs G with distinguished vertices s, t **networks**. Often, we will ask that our flows take on only positive values.

Given this idea of **flow**, there are a number of natural extensions we can make. For example, we often want to study graphs where the edges are representing pipes with some fixed **capacity**: in other words, you might want to find out what flows are **feasible** on a graph where all of the edges can carry at most one gallon per minute (say.) This motivates the following definitions:

Definition. A **capacity function** on a graph G is a function $c : E(G) \to [0, \infty]$, that assigns capacity values to every edge. Given a graph G and capacity function c, we say that a flow f on G is **feasible** if $0 \leq f(x, y) \leq c(x, y)$, for every $\{x, y\} \in E(G)$.

So: given a graph G with some given capacity function c, how "big" of a feasible flow can we define on G? In other words, if we're thinking of G as a collection of pipes with certain capacities, what's the most current we can pump out of the source vertex?

At first, answering this question seems difficult. Consider the graph below, with capacities written on all of the edges:



It appears that any flow on this graph cannot pump any more than ≤ 2 units of flow out of the source node, as the two "bridge" edges (bolded here in green) seem like they're acting as a "bottleneck" for our flows – any flow that goes from s to t will have to cross this bridge, and because it has total capacity 2 we clearly can only have at most 2 units of flow leaving s as well.

Is this the only constraint we have? Are there other constraints? Also, how can we find these "bridges" in general?

2 The Max-Flow Min-Cut Theorem

First, we want to generalize this idea of "bridges" and "maximal flow" that we discussed above. We do this with the following definitions:

Definition. For a connected graph G with a flow f, the **value** of f, $|f|$, is the total amount of current leaving the source node s : i.e. $\sum_{y\in N^+(s)} f(s,y) - \sum_{y\in N^-(s)} f(s,y)$.

More generally, given a subset U of $V(G)$, we say that the **total flow** out of U is just the sum

$$\sum_{x\in U, y\notin U} f(x,y) - \sum_{x\in U, y\notin U} f(y,x).$$

We denote this as $f^+(U) - f^-(U)$. In this notation, the value of f is just $f^+(s) - f^-(s)$; as well, we can reformulate Kirchoff's law as demanding that $f^+(v) - f^-(v) = 0$, at every vertex that's neither the source nor the sink.

For a connected graph G, a **cut** is a collection of vertices S such that $s \in S$ and $t \notin S$. The **capacity** of any cut S, $c(S)$, is the sum

$$\sum_{x \in S, y \notin S} c(x, y).$$

We mentioned above that these cuts gave us bounds on how big the value of any feasible flow can be. We prove this here with two propositions:

Proposition 1 *If G is a graph with source and sink nodes s, t, capacity function c, and feasible flow f, and S is any cut on G containing s, then the value of f is given by $f(S)$.*

Proof. First, notice that the net flow out of S, $f(S)$, is given by the sum of the net flows out of the nodes of S: i.e.

$$f^+(S) - f^-(S) = \sum_{v \in S} f^+(v) - f^-(v).$$

Why is this? Well, for any edge (x, y), we either have

- $x, y \notin S$, in which case it shows up in neither the left or right sums,
- $x, y \in S$, in which case it doesn't show up on the left and shows up twice on the right, once positively in $f^+(x)$ and once negatively in $f^-(y)$,
- $x \in S, y \notin S$, in which case it shows up exactly once positively on both sides, or
- $x \notin S, y \in S$, in which case it shows up exactly once negatively on both sides.

Applying Kirchoff's law, we can see that all of these pairs $f^+(v) - f^-(v)$ are 0 except for when $v = s$, and thus that we have $f(S) = f^+(s) - f^-(S) = |f|$.

Corollary 2 *The value of any flow f is bounded above by the capacity of any cut S.*

Proof. By the above, we have

$$|f| = f^+(S) - f^-(S) \leq f^+(S) \leq c(S).$$

Surprisingly, it turns out that this bound is tight! In other words, the maximum value amongst all feasible flows is given by the minimal capacity amongst all cuts C. This is the **Ford-Fulkerson Max-Flow Min-Cut** theorem, which we state and prove here:

Theorem 3 *Suppose that G is a graph with source and sink nodes s, t, and a rational capacity function c. Then the maximum value of a flow is equal to the minimum value of a cut.*

Proof. We prove our claim here via the Ford-Fulkerson algorithm, defined as follows:

1. Input: a feasible flow f. Let R be the set currently given by the single source vertex $\{s\}$, and S be the subset of R currently defined as $\emptyset$.

2. Iteration: Pick any vertex $v \in R$ that's not in S. For every vertex $w \notin R$ with $f(v,w) < c(v,w)$, add w to R. (Mentally, think of these as the edges in our graph that "could have more flow on them.") Also, add any vertex $w \notin R$ where $f(w,v) > 0$ (mentally, think of these as the edges in our graph that are "flowing backwards" into R.)

   Once you've done this search process over all of v's neighbors, add v to the set S (the "searched" set.) Repeat this step until $R = S$.

3. If the sink vertex t is not in R at this time, terminate. Otherwise, if it is in R, then there is a path $P = s, e_1, v_1, \ldots, e_n, t$ from s to t, along (unoriented) edges where we either have $f(e_i) < c(e_i)$, if we're traversing this edge in the oriented direction, or $f(e_i) > 0$, if we're traversing it backwards. Call such a path an **augmenting path** for f, and let $\epsilon = \min\{\min_{\text{oriented}} |c(e_i) - f(e_i)|, \min_{\text{reversed}} |f(e_i)|\}$. Now, (as the name suggests,) **augment** f along this path: i.e. define f', our new flow, by

   - $f'(y,z) = f(y,z) + \epsilon$, for (y,z) any edge involved in our path P that we traversed in that order.
   - $f'(z,y) = f(z,y) - \epsilon$, for (y,z) any edge involved in our path P that we traversed in reverse order.
   - $f'(e) = f(e)$, for all other edges e.

   This new flow has value ϵ-greater than our old flow. Go to (1) with our new flow f', and repeat this process.

Because ϵ is at least $1/a$, where a is the least-common denominator of the capacities of the edges, we know that this process will terminate in finitely many steps.

If we input the feasible flow f_0 that's identically 0 on all edges, this process will return two things for us: a flow f, and a set S with the properties that

1. S contains s and not t, and thus is a cut, and
2. S is specifically the collection of vertices v in $V(G)$ for which there is a path P from s to v, on which $f(e) < c(e)$.

This means that for any pair $x \in S, y \notin S$, we have $f(x,y) = c(x,y)$ (otherwise, we would have added y to R and thus to S:) similarly, for any pair $x \notin S, y \in S$ we have $f(x,y) = 0$ (as otherwise we would have added this vertex.) But this means that

$$f(S) = f^+(S) - f^-(S) = f^+(S) = c(S).$$

Therefore, by our earlier corollary, we know that (1) this flow f has maximum value, and (2) this cut S has minimum capacity.

Two important things to recognize in the above theorem are the following:

- If our capacity function c is finite and integer-valued, then the maximum flow we create is integer-valued! Again, this is because ϵ is guaranteed to be integral if the capacity function is integral.

- We could in fact scale this result to work for "negative" flows and capacity functions, by just adding and subtracting off constants at the start/end of applying our theorem. This is kind of a useful thing to do, in many situations.

These properties – specifically, the second one – are quite remarkable, and are in fact what we'll focus on for the next two lectures. Specifically: using this idea, we will go on to prove a number of (otherwise-tricky!) combinatorial theorems, like Hall's marriage theorem, Menger's theorems, and Dilworth's theorem, with almost no effort at all!