

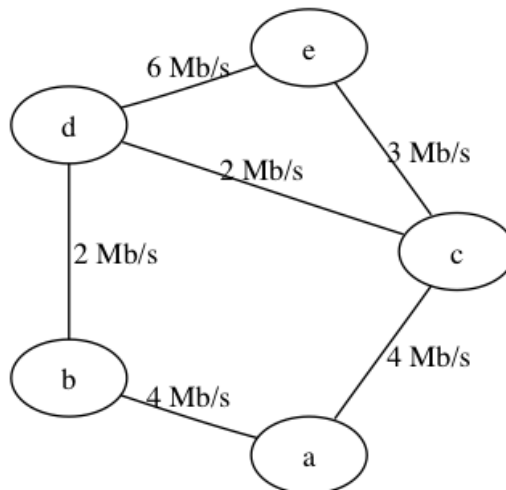
Lecture 9

In which we introduce the maximum flow problem.

1 Flows in Networks

Today we start talking about the *Maximum Flow* problem. As a motivating example, suppose that we have a communication network, in which certain pairs of nodes are linked by connections; each connection has a limit to the rate at which data can be sent. Given two nodes on the network, what is the maximum rate at which one can send data to the other, assuming no other pair of nodes are attempting to communicate?

For example, consider the following network, and suppose that a needs to send data at a rate of $6Mb/s$ to e . Is this possible?



The answer is yes: a can split its stream of data, and send $4Mb/s$ to c and $2Mb/s$ to b . The node b relays the $2Mb/s$ stream of data that it receives from a to d , while node c splits the $4Mb/s$ stream that it receives into two parts: it relays $3Mb/s$ of

data to e , and send the remaining $1Mb/s$ to d . Overall, d receives $3Mb/s$ of data, which it relays to e .

Can a send more data, such as, say, $7Mb/s$ to e ? The answer is no. Consider the two links (b, d) and (a, c) in the network, and suppose you removed them from the network. Then there would be no way reach e from a , and no communication would be possible. This means that all the data that a sends to e must pass through one of those links. Between the two of them, the two links (b, d) and (a, c) support at most the transmission of $6Mb/s$ of data, and so that is the maximum rate at which a can send data to e .

The networks that we will work with could model other settings in which “stuff” has to go from one place to another, subject to capacity constraints in links. For example the network could model a public transit system, or a highway system. In some cases, we will be interested in instances of the problem which are constructed to model other combinatorial problems.

In some such applications, it makes sense to consider networks in which the capacity of a link depends on the direction, so that the capacity of the link $u \rightarrow v$ could be different from the capacity of the link $v \rightarrow u$. Formally, an instance of the maximum flow problem is defined as follows.

Definition 1 *A network is a directed graph $G(V, E)$, in which*

- *a vertex $s \in V$ and a vertex $t \in V$ are specified as being the source node and the sink node, respectively.*
- *every directed edge $(u, v) \in E$ has a positive capacity $c(u, v) > 0$ associated to it. If both the edges (u, v) and (v, u) belong to E , we allow their capacities to be different.*

Sometimes we will write expressions that include capacities between pairs of vertices that are not connected by an edge. In that case the convention is that the capacity is zero.

A *flow* in a network is a specification of how to route “stuff” from s to t so that no link is used beyond its capacity, and so that every link, except the sender s and the receiver t , relays out “stuff” at exactly the same rate at which it receives from other vertices. In the motivating example of a communication network, if nodes were sending out less data than they receive, there would be data loss, and they cannot send out more data than they receive because they are simply forwarding data. Formally, we have the following definition.

Definition 2 *A flow in an network (G, s, t, c) is an assignment of a non-negative number $f(u, v)$ to every edge $(u, v) \in E$ such that*

- For every edge $(u, v) \in E$, $f(u, v) \leq c(u, v)$;
- For every vertex $v \in V$, $\sum_{u \in V} f(u, v) = \sum_{w \in V} f(v, w)$

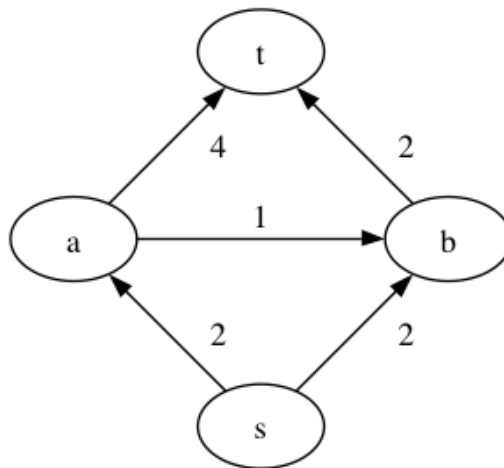
where we follow the convention that $f(u, v) = 0$ if $(u, v) \notin E$. (This convention is useful otherwise the second condition would have to be something like $\sum_{u:(u,v) \in E} f(u, v) = \sum_{w:(v,w) \in E} f(v, w)$)

The *cost* of the flow (the throughput of the communication, in the communication network example) is

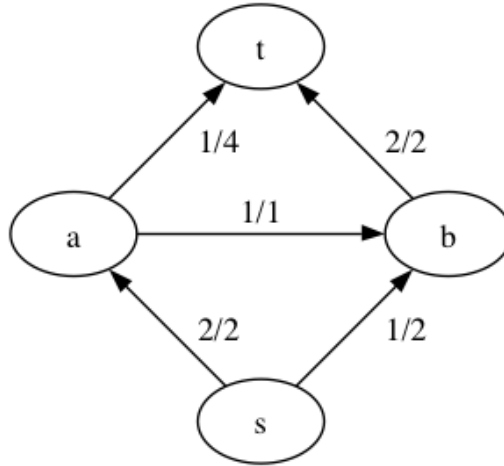
$$\sum_v f(s, v)$$

In the *maximum flow* problem, given a network we want to find a flow of maximum cost.

For example, here is an example of a network:



And the following is a flow in the network (a label x/y on an edge (u, v) means that the flow $f(u, v)$ is x and the capacity $c(u, v)$ is y).



Is the flow optimal? We are only sending 3 units of flow from s to t , while we see that we can send 2 units along the $s \rightarrow a \rightarrow t$ path, and another 2 units along the $s \rightarrow b \rightarrow t$ path, for a total of 4, so the above solution is not optimal. (Is the solution we just discussed, of cost 4, optimal?)

In general, how do we reason about the optimality of a flow? This is an important question, because when we talked about approximation algorithms for minimization problems we noted that, in each case, we were able to reason about the approximation quality of an algorithm only because we had ways to prove lower bounds to the optimum. Here that we are dealing with a maximization problem, if we are going to reason about the quality of solutions provided by our algorithms, we need ways to prove upper bounds to the optimum.

When we considered the first example of the lecture, we noticed that if we look at any set of edges whose removal disconnects the receiver from the sender, then all the flow from the sender to the receiver must pass through those edges, and so their total capacity is an upper bound to the cost of any flow, including the optimal flow. This motivates the following definition.

Definition 3 (Cut) A cut in a network $(G = (V, E), s, t, c)$, is partition $(A, V - A)$ of the set of vertices V into two sets, such that $s \in A$ and $t \in V - A$. We will usually identify a cut with the set A that contains s . The capacity of a cut is the quantity

$$c(A) := \sum_{u \in A, v \notin A} c(u, v)$$

The motivation for the definition is the following: let A be a cut in a network, that is, a set of vertices that contains s and does not contain t . Consider the set of edges $\{(u, v) \in E : u \in A \wedge v \notin A\}$. If we remove those edges from the network, then it is

not possible to go from any vertex in A to any vertex outside A and, in particular, it is not possible to go from s to t . This means that all the flow from s to t must pass through those edges, and so the total capacity of those edges (that, is $c(A)$) is an upper bound to the cost of any feasible flow.

Even though what we just said is rather self-evident, let us give a rigorous proof, because this will help familiarize ourselves with the techniques used to prove rigorous results about flows. (Later, we will need to prove statements that are far from obvious.)

Lemma 4 *For every network (G, s, t, c) , any flow f in the network, and any cut A ,*

$$\sum_{v \in V} f(s, v) \leq \sum_{a \in A, b \notin A} c(a, b)$$

That is, the cost of the flow is at most the capacity of the cut.

We will derive the lemma from Fact 5 below.

If (G, s, t, c) is a network, f is a flow, and A is a cut, define the *net flow out of A* to be

$$f(A) := \sum_{a \in A, b \notin A} f(a, b) - \sum_{b \notin A, a \in A} f(b, a)$$

that is, the total flow from A to $V - A$ minus the total flow from $V - A$ to A . Then we have:

Fact 5 *For every network (G, s, t, c) , every flow f , and every cut A , the net flow out of A is the same as the cost of the flow:*

$$f(A) = \sum_{v \in V} f(s, v)$$

PROOF: Consider the expression

$$S := \sum_{v \in V} f(s, v) - \sum_{a \in A} \left(\sum_{v \in V} f(v, a) - \sum_{w \in V} f(a, w) \right) + \sum_{a \in A, b \notin A} f(a, b) - \sum_{b \in B, a \in A} f(b, a)$$

on the one hand, we have

$$S = 0$$

because for every edge (u, v) such that at least one of u, v is in A we see that $f(u, v)$ appears twice in the expression for S , once with a $+$ sign and once with a $-$ sign, so all terms cancel. On the other hand, we have

$$\sum_{a \in A} \left(\sum_{v \in V} f(v, a) - \sum_{w \in V} f(a, w) \right) = 0$$

because of the definition of flow, and so

$$\sum_{v \in V} f(s, v) = \sum_{a \in A, b \notin A} f(a, b) - \sum_{b \in B, a \in A} f(b, a) = f(A)$$

□

To prove Lemma 4, consider any network (G, s, t, c) , any flow f and any cut A . We have

$$\text{cost}(f) = f(A) \leq \sum_{a \in A, b \notin A} f(u, v) \leq \sum_{a \in A, b \notin A} c(u, v) = c(A)$$

So we have proved Lemma 4, and we have a way to “certify” the optimality of a flow, if we are able to find a flow and a cut such that the cost of the flow is equal to the capacity of the cut.

Consider now the complementary question: how do we see if a given flow in a network can be improved? That is, what is a clear sign that a given flow is not optimal? If we see a path from s to t such that all the edges are used at less than full capacity, then it is clear that we can push extra flow along that path and that the solution is not optimal. Can there be other cases in which a given solution is not optimal? Indeed there can. Going back to the last example that we considered, we had a flow of cost 3, which was not optimal (because a flow of cost 4 existed), but if we look at the three possible paths from s to t , that is, $s \rightarrow a \rightarrow t$, $s \rightarrow a \rightarrow b \rightarrow t$, and $s \rightarrow b \rightarrow t$, they each involve an edge used at full capacity.

However, let us reason in the following way: suppose that, in addition to the edges shown in the last picture, there were also an edge of capacity 1 from b to a . Then we would have had the path $s \rightarrow b \rightarrow a \rightarrow t$ in which every edge has one unit of residual capacity, and we could have pushed an extra unit of flow along that path. In the resulting solution, a sends one unit flow to b , and b sends one unit of flow to a , a situation that is perfectly equivalent to a and b not sending each other anything, so that the extra edge from b to a is not needed after all. In general, if we are sending $f(u, v)$ units of flow from u to v , then we are effectively increasing the capacity of the edge from v to u , because we can “simulate” the effect of sending flow from v to u by simply sending less flow from u to v . These observations motivate the following definition:

Definition 6 (Residual Network) Let $N = (G, s, t, c)$ be a network, and f be a flow. Then the residual network of N with respect to f is a network in which the edge u, v has capacity

$$c(u, v) - f(u, v) + f(v, u)$$

The idea is that, in the residual network, the capacity of an edge measures how much *more* flow can be pushed along that edge in addition to the flow f , without violating the capacity constraints. The edge (u, v) starts with capacity $c(u, v)$, and $f(u, v)$ units of that capacity are taken by the current flow; in addition, we have $f(v, u)$ additional units of “virtual capacity” that come from the fact that we can reduce the flow from v to u .

An *augmenting path* in a network is a path from s to t in which every edge has positive capacity in the residual network. For example, in our last picture, the path $s \rightarrow b \rightarrow a \rightarrow t$ is an augmenting path.

The *Ford-Fulkerson* algorithm is a simple greedy algorithm that starts from an empty flow and, as long as it can find an augmenting path, improves the current solution using the path.

Algorithm *Ford-Fulkerson*

- Input: network $(G = (V, E), s, t, c)$
- $\forall u, v. f(u, v) := 0$
- compute the capacities $c'(\cdot, \cdot)$ of the residual network
- while there is a path p from s to t such that all edges in the path have positive residual capacity
 - let c'_{\min} be the smallest of the residual capacities of the edges of p
 - let f' be a flow that pushes c'_{\min} units of flow along p , that is, $f'(u, v) = c'_{\min}$ if $(u, v) \in p$, and $f'(u, v) = 0$ otherwise
 - $f := f + f'$, that is, $\forall u, v. f(u, v) := f(u, v) + f'(u, v)$
 - for every pair of vertices such that $f(u, v)$ and $f(v, u)$ are both positive, let $f_{\min} := \min\{f(u, v), f(v, u)\}$ and let $f(u, v) := f(u, v) - f_{\min}$ and $f(v, u) := f(v, u) - f_{\min}$
 - recompute the capacities $c'(\cdot, \cdot)$ of the residual network according to the new flow
- return f

At every step, the algorithm increases the cost of the current solution by a positive amount c'_{\min} and, the algorithm converges in finite time to a solution that cannot be improved via an augmenting path. Note the “clean-up” step after the flow is increased, which makes sure that flow pushed along a “virtual edge” in the residual network is realized by reducing the actual flow in the opposite direction. The following theorem shows that the Ford-Fulkerson algorithm is optimal, and it proves the important fact that whenever a cut is optimal, its optimality can always be proved by exhibiting a cut whose capacity is equal to the cost of the flow.

Theorem 7 (Max Flow-Min Cut) *The following three conditions are equivalent for a flow f in a network:*

1. *There is a cut whose capacity is equal to the cost of f*
2. *The flow f is optimal*
3. *There is no augmenting path for the flow f*

PROOF: We have already proved that (1) implies (2), and it is clear that (2) implies (3), so the point of the theorem is to prove that (3) implies (1).

Let f be a flow such that there is no augmenting path in the residual network. Take A to be the set of vertices reachable from s (including s) via edges that have positive capacity in the residual network. Then:

- $s \in A$ by definition
- $t \notin A$ otherwise we would have an augmenting path.

So A is a cut. Now observe that for every two vertices $a \in A$ and $b \notin A$, the capacity of the edge (a, b) in the residual network must be zero, otherwise we would be able to reach b from s in the residual network via positive-capacity edges, but $b \notin A$ means that no such path can exist.

Recall that the residual capacity of the edge (a, b) is defined as

$$c(a, b) - f(a, b) + f(b, a)$$

and that $f(a, b) \leq c(a, b)$ and that $f(b, a) \geq 0$, so that the only way for the residual capacity to be zero is to have

- $f(a, b) = c(a, b)$
- $f(b, a) = 0$

Now just observe that the cost of the flow is

$$\text{cost}(f) = f(A) = \sum_{a \in A, b \notin A} f(a, b) - \sum_{b \notin A, a \in A} f(b, a) = \sum_{a \in A, b \notin A} c(a, b) = c(A)$$

and so the capacity of the cut is indeed equal to the cost of the flow. \square

Remark 8 *Suppose that we had defined the residual network as a network in which the capacity of the edge (u, v) is $c(u, v) - f(u, v)$, without the extra “virtual capacity” coming from the flow from v to u , and suppose that we defined an augmenting path to be a path from s to t in which each capacity in the residual network (according to this definition) is positive. Then we have seen before an example of a flow that has no augmenting path according to this definition, but that is not optimal. Where does the proof of the Max-Flow Min-Cut theorem break down if we use the $c(u, v) - f(u, v)$ definition of residual capacity?*