Lecture 7: The max-flow min-cut theorem · 1MA020

Vilhelm Agdur[1]

[1] vilhelm.agdur@math.uu.se

14 November 2023

Continuing our exploration of things that can be done with weighted graphs, we move on to considering *flows* in weighted directed graphs, which model things like traffic flows. We see the connection between these and edge cuts of graphs, and prove the Ford-Fulkerson theorem.

In our exercises, we saw this figure:

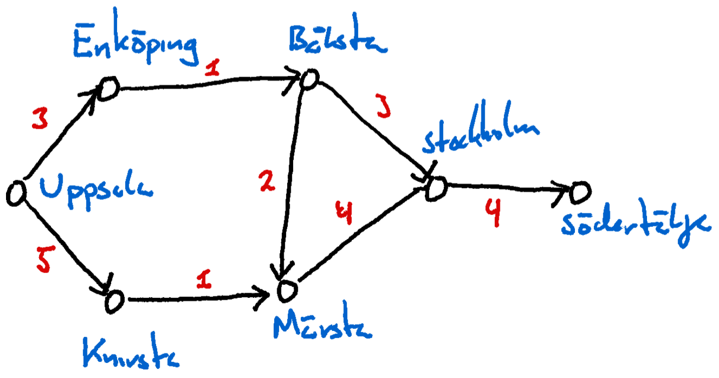

Figure 1: A hypothetical graph modelling public transport flow from Uppsala to Södertälje.

If we interpret the edge-weights as how many thousands of passengers can travel the route per hour, it makes sense to ask the question of how many can travel from Uppsala to Södertälje per hour. Intuitively, it is clear that the answer must be two thousand, because the traffic is bottlenecked by the Enköping-Bålsta and Knivsta-Märsta connections – the fact that five thousand per hour can get from Uppsala to Knivsta doesn't help at all, and upgrading the Bålsta-Stockholm route wouldn't improve things.

Let us now turn these intuitive considerations into actual rigorous mathematics. First, let us define the graphs we are working on:

Definition 1. A *weighted directed graph* G consists of a directed graph (V, E) and a weight function $w : E \to \mathbb{R}$. For it to be a *flow network* we additionally require that all weights be positive, that there be a distinguished *source* vertex s and *sink* vertex t, and that whenever $u \to v$ is an edge, $v \to u$ is not also an edge.[2]

We define a *capacity function* $c : V \times V \to [0, \infty)$ by that $c(v, v') = w(v \to v')$ whenever $v \to v'$ is an edge of G, and $c(v, v') = 0$ otherwise.

[2] Recall that in general, it is allowed in a simple directed graph to have both the edge $u \to v$ and the edge $v \to u$ – what is not allowed is multiples of the same edge, or loops. In this case, the restriction of not having such back-and-forth edges is not a genuine restriction, however: If we have such a pair of edges, we can get an equivalent flow network by introducing a "dummy vertex" in the middle of one of the edges, as in Figure 2.

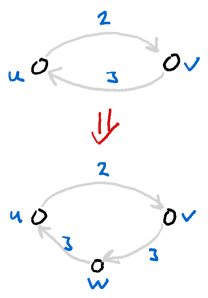

Figure 2: Subdividing an edge using a dummy vertex, to get around the restriction that there be no back-and-forth edges.

These flow graphs define the capacity of the network to handle traffic. Next, let us define the actual *flows*, which are the possible actual traffic situations. There are two natural constraints these should satisfy:

1. The flow through an edge can't be greater than the actual capacity of the edge, so we aren't putting more cars on the road than will actually fit.
2. Other than the source and the sink nodes, where we imagine vehicles are entering and exiting the graph, the flow into a vertex must equal the flow out of it. Trains do not magically vanish, nor do they appear out of nowhere or teleport.

Definition 2. A *flow* on the flow network G with capacity function c is a function $f : V \times V \to [0, \infty)$ which satisfies

1. the *capacity constraint* that
$$f(v, v') \leq c(v, v') \qquad \forall v, v' \in V,$$
2. and the *conservation constraint*
$$\sum_{w \in V} f(w, v) = \sum_{u \in V} f(v, u)$$
whenever $v \in V \setminus \{s, t\}$.

The *value* of a flow, denoted by $|f|$, is the net out-flow at the source, that is
$$|f| = \sum_{v \in V} f(s, v) - \sum_{w \in V} f(w, s).$$

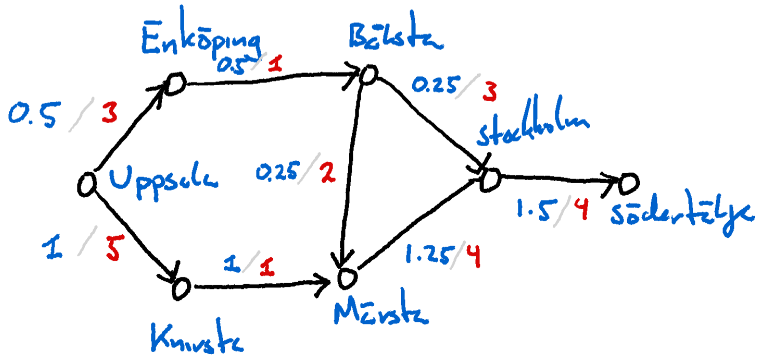

Figure 3: An example of a flow in the graph from Figure 1. The weights of the edges are given in red, and the flows through them in blue. This flow has value 1.5, which we can see both by computing the net out-flow from Uppsala and the net flow into Södertälje.

Remark 3. It follows from the conservation constraint that $|f|$ is also equal to the net in-flow at the sink.[3] Therefore, we can always assume that $|f| \geq 0$, since if it were not, we could just swap the roles of s and t.

[3] *Exercise* 1. Prove this.

Having defined what we mean by a flow, let us next formalize the intuitive notion of a bottleneck in the graph.

Definition 4. Let $G = (V, E, w, s, t)$ be a flow network with source s and sink t. An *s-t-cut* is a partitioning of V into two sets S, T such that $s \in S$ and $t \in T$. The *capacity* of the cut is
$$c(S, T) = \sum_{(v, v') \in S \times T} c(v, v') = \sum_{e \in E(S,T)} w(e).$$

Considering our intuition about the bottlenecks, it should be clear that any flow from s to t has to at some point pass from S into T, and so use some of the capacity of the cut. So the total flow cannot be greater than the capacity of the cut, that is, $|f| \leq c(S,T)$ for any flow f and any s-t-cut S,T.

In fact, it turns out that equality is only achieved in this inequality in the most extreme case.

Lemma 5. *Let G be a flow network with a flow f and an s-t-cut of V into S and T. If $|f| = c(S,T)$, then $|f|$ is maximal among all flows, and $c(S,T)$ is minimal among all s-t-cuts.*

Proof. As we saw, for any other flow f', we must have

$$|f'| \leq c(S,T) = |f|,$$

and so f is maximal. Similarly, for any other s-t-cut S', T' we have

$$c(S,T) = |f| \leq c(S',T')$$

and so S,T is minimal. □

A central construction in the theory of flow networks is the *residual network*, which as the name suggests encodes how much capacity is left over by a flow.

Definition 6. Let G be a flow network with capacity function c, and let f be a flow on G. The *residual capacity* c_f is a function from $V \times V$ into $[0,\infty)$ defined by[4]

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \text{ is an edge in } G \\ f(v,u) & \text{if } (v,u) \text{ is an edge in } G \\ 0 & \text{otherwise.} \end{cases}$$

[4] Notice how we use the assumption that there are no back-and-forth edges in G here – otherwise the definition would not make sense.

The *residual network* G_f is the weighted directed graph which has an edge $u \to v$ whenever $c_f(u,v) > 0$, and this edge has weight $c_f(u,v)$ if so.

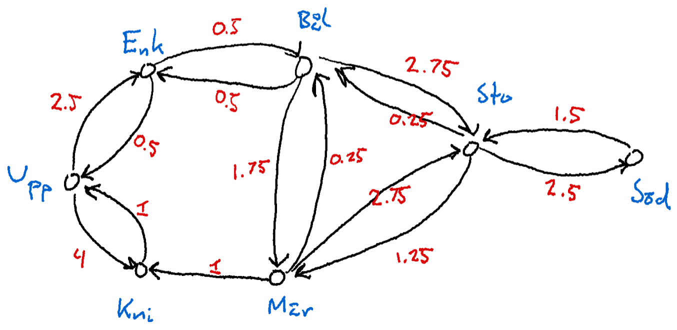

Figure 4: The residual network of the flow in Figure 3.

We give an example of this in Figure 4, using the same flow network we have seen before.

Exercises