CIS 575. Introduction to Algorithm Analysis
Material for April 24, 2024

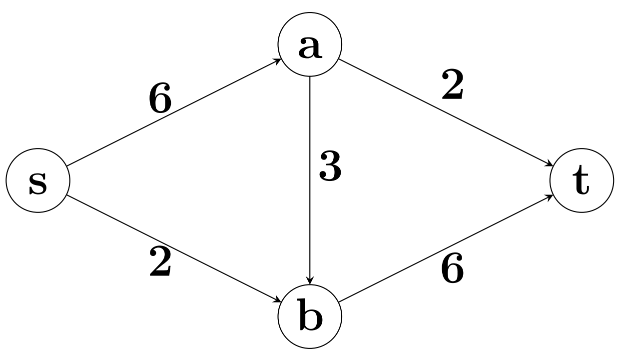
Minimal Cuts in Networks

Torben Amtoft

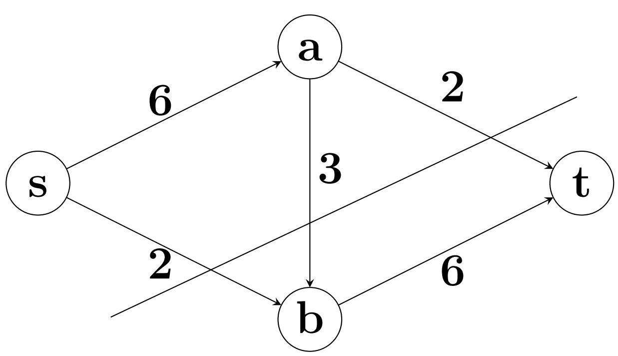
The topic of this note is covered in *Cormen*'s Section 24.2.

1 Cutting Networks

Consider the network



whose maximum flow is only 7, despite the edges from the source having total capacity 8, as have the edges going into the sink. To understand why it is not possible to achieve a flow with value 8, observe that there is a "bottleneck" that limits to 7 the flow from the upper left nodes (s and a) to the lower right nodes (b and t):



Such a bottleneck is a special case of a **cut**, where a cut in a flow network (V, E) is a subset of V that contains the source s but does not contain the sink t. We often use U to denote a cut, and let W be the remaining nodes in V.

The **capacity** of such a cut, $C(U)$, is the sum of the capacities on edges going from U to W:

$$C(U) = \sum_{u \in U, w \in W} C(u, w)$$

For the given network, there are 4 cuts, with capacities given by

$\{s\}$	$6 + 2 =$	8
$\{s, a\}$	$2 + 3 + 2 =$	7
$\{s, b\}$	$6 + 6 =$	12
$\{s, a, b\}$	$2 + 6 =$	8

We see that the cut with the smallest capacity is $\{s, a\}$, confirming that this is indeed the bottleneck with a capacity that equals the maximum flow.

It turns out, as we shall argue for in the next paragraphs, that the value of a flow can never exceed the capacity of a cut, but that the maximum flow will equal the capacity of least one cut (a bottleneck).

Max Flow Not Bigger Than Min Cut With U an arbitrary cut (thus $s \in U$) and W the remaining nodes (thus $t \in W$), we for an arbitrary flow F have $\mathbf{V(F) \leq C(U)}$ since

$$\begin{array}{rcl}
 & & V(F) \\
\text{(definition of flow value)} & = & \sum_{v \in V} F(s, v) - \sum_{v \in V} F(v, s) \\
\text{(flow conservation)} & = & \sum_{v \in V} (F(s, v) - F(v, s)) + \sum_{u \in U \setminus \{s\}, v \in V} (F(u, v) - F(v, u)) \\
(U = \{s\} \cup (U \setminus \{s\})) & = & \sum_{u \in U, v \in V} F(u, v) - \sum_{u \in U, v \in V} F(v, u) \\
(U, W \text{ partition } V) & = & \sum_{u \in U, v \in W} F(u, v) + \sum_{u \in U, v \in U} F(u, v) - \sum_{u \in U, v \in U} F(v, u) - \sum_{u \in U, v \in W} F(v, u) \\
\text{(cancel out)} & = & \sum_{u \in U, v \in W} F(u, v) - \sum_{u \in U, v \in W} F(v, u) \\
\text{(flows non-negative)} & \leq & \sum_{u \in U, v \in W} F(u, v) \\
\text{(flow bounded by capacity)} & \leq & \sum_{u \in U, v \in W} C(u, v) \\
\text{(definition of cut capacity)} & = & C(U)
\end{array}$$

Max Flow Equals Min Cut With M the maximum value of a flow in (V, E), M is also the minimal capacity of a cut.

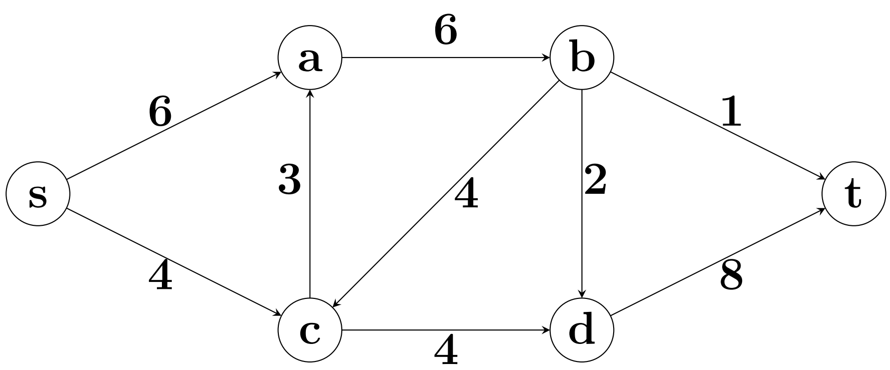
We shall prove that claim for the special case when $M = 0$. Then there exists no augmenting path. Hence

$$U = \{v \in V \mid \text{there is a path from } s \text{ to } v\}$$

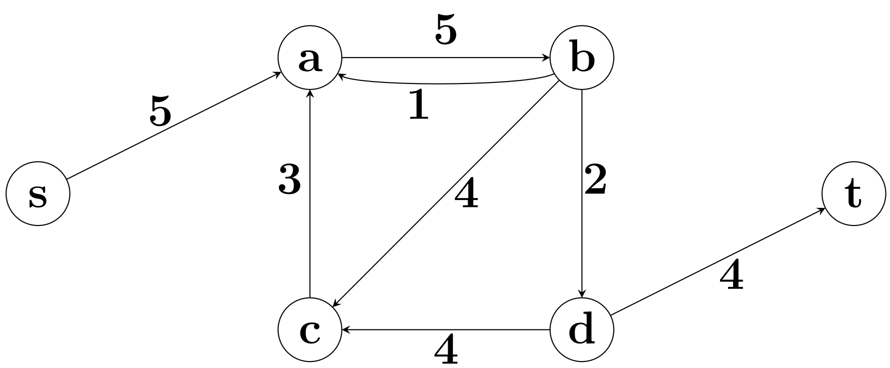
contains s but not t, and thus U is a cut. And the capacity of that cut is 0 since if (u, w) is an edge with $u \in U$ and $w \in V \setminus U$ then there is a path from s to w which *contradicts* $w \notin U$.

Finding Min Cut To find a cut with minimum capacity, one may try to examine all possible cuts, but this is not a feasible approach since the number of cuts is exponential in the number of nodes. Fortunately, the Edmonds-Karp algorithm allows us to efficiently construct not just a flow with maximum value, but as a byproduct also a cut with minimum capacity.

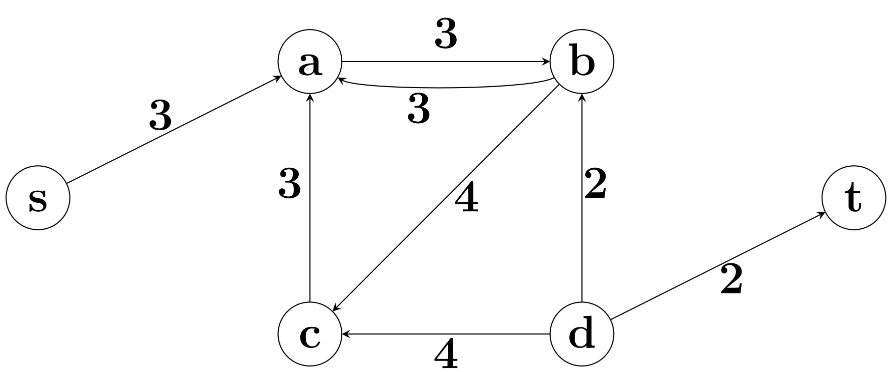
We shall illustrate this by considering the network



As augmenting paths, the Edmonds-Karp algorithm will first consider (in any order) *sabt* with minimum capacity 1, and *scdt* with minimum capacity 4. After processing these two paths, the residual network will be



which has only one augmenting path: *sabdt*, with minimum capacity 2. We get the residual network



which has no augmenting path since from s it is possible to reach a, b and c but *not* d or t. Accordingly, $\{s, a, b, c\}$ is a cut with capacity 0, and is indeed also in the original graph a cut with minimum capacity:

$$C(\{s, a, b, c\}) = C(b, t) + C(b, d) + C(c, d) = 1 + 2 + 4 = 7$$

which equals the value, $4 + 1 + 2 = 7$, of the flow we constructed.