CS 218 Design and Analysis of Algorithms

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Module 2: Flow networks, Max-flow, Min-cut and applications

Weak duality

Lemma (Max Flow is at most as much as the Min Cut)

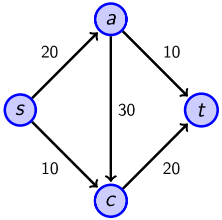
Let f be any flow in the flow network G. Let (S,T) be any (s,t)-Cut in G. Then $|f| \leq cap(S,T)$. Moreover, $|f| = cap(S,T)$ if and only if f saturates every edge from S to T and avoids every edge from T to S.

Proof.

$$
\begin{aligned}
|f| &= f^{\rightarrow}(s) && \\
&= f^{\rightarrow}(S) - f^{\leftarrow}(S) && \text{due to the conservation constraint} \\
&\leq \sum_{u \in S} \sum_{v \in T} f(u,v) && \text{recall, if } (u,v) \notin E \text{ then } f(u,v) = 0 \\
&\leq \sum_{u \in S} \sum_{v \in T} c(u,v) && \text{due to the capacity constraint} \\
&= \mathrm{cap}(S,T)
\end{aligned}
$$

Solving the Max Flow Problem

How do we plan to solve this problem efficiently?



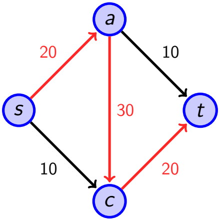
Suppose we start with $f(e) = 0$ for each $e \in E$.

This satisfies the capacity constraint and the conservation constraint.

Let us try to push some flow along an s to t path, allowed by the capacity constraints.

Solving the Max Flow Problem

How do we plan to solve this problem efficiently?



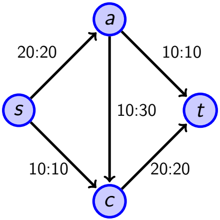
Here is an s to t path.

We can push a flow of 20 along this.

Is this the maximum flow possible? No!

Solving the Max Flow Problem

How do we plan to solve this problem efficiently?



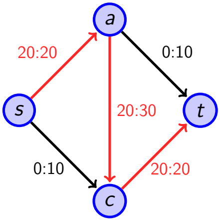
By not sending the full flow of 20 through (a, c), we are able to use the (a, t) edge to send extra flow.

This way we have achieved a flow of value 30 through the network.

How did we arrive at this answer?

Solving the Max Flow Problem

Suppose we have the first flow that we came up with.



Suppose now we push 10 units of flow from s to c.

This results in a flow of 30 into c.

But due to the capacity constraints, we cannot push a flow of 30 out of c to t.

So we undo a flow of 10 along (a, c).

But this results into too little flow out of a.

So we push 10 from a to t.

Now we have a valid flow and the value has increased to 30.

Residual Graph

In order to formalise the idea of undoing a flow, we define residual graphs.

Definition (Residual Graphs)

Given a graph G with capacity function $c : E \to \mathbb{N}$ and a flow f, a residual graph of G with respect to the flow f, denoted as G_f, is

The vertices of G_f are the same as the vertices of G.

If e is an edge in G such that $f(e) < c(e)$, then G_f has the edge e with capacity $c(e) - f(e)$ on it. This is called the forward edge.

If $e = (u, v)$ is such that $f(e) > 0$, then we add an edge (v, u) in G_f with capacity $f(e)$ on it. This is called a backward edge.

Residual Graph

Definition (Residual Graphs)

Given a graph G with capacity function $c : E \to \mathbb{N}$ and a flow f, a residual graph of G with respect to the flow f, denoted as G_f, is

- The vertices of G_f are the same as the vertices of G.
- If e is an edge in G such that $f(e) < c(e)$, then G_f has the edge e with capacity $c(e) - f(e)$ on it. This is called the forward edge.
- If $e = (u, v)$ is such that $f(e) > 0$, then we add an edge (v, u) in G_f with capacity $f(e)$ on it. This is called a backward edge.

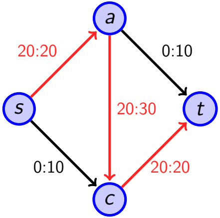
Residual Graph

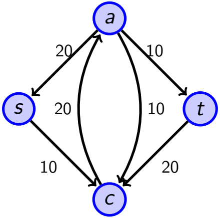
Definition (Residual Graphs)

Given a graph G with capacity function $c : E \to \mathbb{N}$ and a flow f, a residual graph of G with respect to the flow f, denoted as G_f, is

- The vertices of G_f are the same as the vertices of G.
- If e is an edge in G such that $f(e) < c(e)$, then G_f has the edge e with capacity $c(e) - f(e)$ on it. This is called the forward edge.
- If $e = (u, v)$ is such that $f(e) > 0$, then we add an edge (v, u) in G_f with capacity $f(e)$ on it. This is called a backward edge.

Example.





Residual graph and residual capacity

For each edge of G, there are at most two edges in G_f.

If $0 < f(e) < c(e)$, then two edges get added to G_f in place of e. One forward, one backward.

Size of G_f is at most twice the size of G.

The capacities on the edges of G_f is called residual capacity.

Augmenting paths: pushing more flow from s to t

How should we systematically increase the flow from s to t?

Let π be any s to t path in G_f.

Let $\theta(\pi, f)$ denote the smallest residual capacity along π in G_f.

Now consider the following subroutine.

$\texttt{Aug}(\pi, f)$

> $b \leftarrow \theta(\pi, f)$
> **for** every edge $e = (u, v) \in \Pi$ **do**
>   **if** e is a forward edge **then**
>   increase $f(e)$ in G by b
>   **else**
>   decrease $f(v, u)$ in G by b
>   **end if**
> **end for**