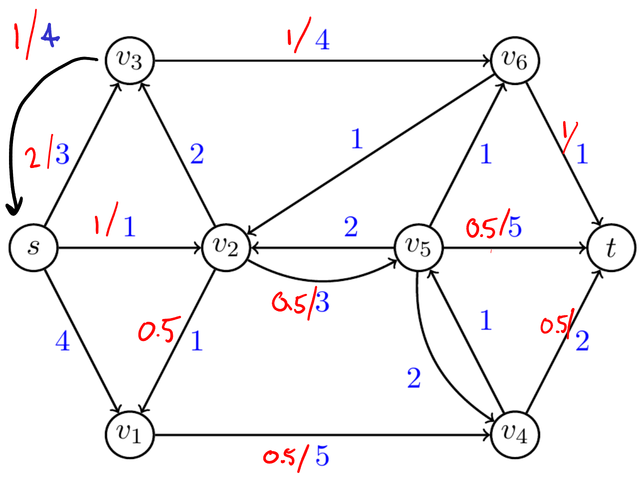
4 Flows and Cuts

4.1 Basic notions and relations

$G = (V, A)$

$u : A \to \mathbb{Z}_{\geq 0}$

$s, t \in V$, $s \neq t$



Definition 4.1: s-t flow / flow

Let $s, t \in V$, $s \neq t$. An s-t flow in G is a function $f : A \to \mathbb{R}_{\geq 0}$ satisfying the following conditions.

(i) *Capacity constraints*: $f(a) \leq u(a)\ \forall a \in A$.

(ii) *Balance constraints*: for $v \in V$,

$$\sum_{a \in \delta^+(v)} f(a) \qquad f(\delta^+(v)) - f(\delta^-(v)) \begin{cases} = 0 & \text{if } v \in V \setminus \{s, t\} \ , \\ \geq 0 & \text{if } v = s \ , \\ \leq 0 & \text{if } v = t \ . \end{cases}$$

The *value* of a flow f is $\nu(f) := f(\delta^+(s)) - f(\delta^-(s))$.

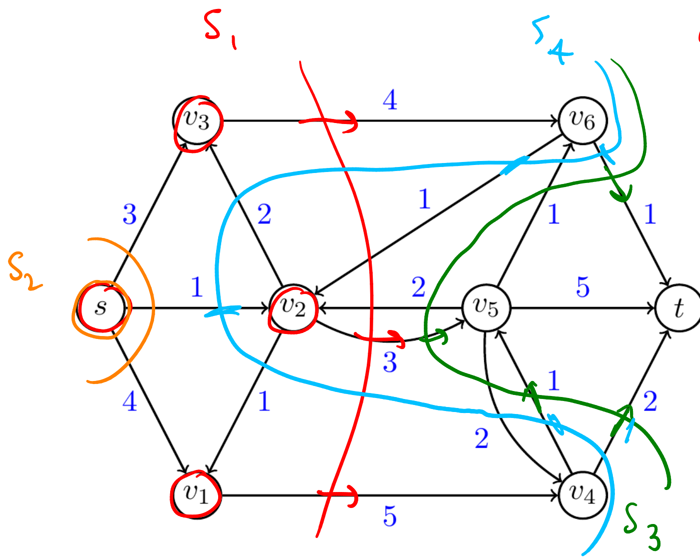
Maximum flow problem, or maximum s-t flow problem

Input: A directed graph $G = (V, A)$, arc capacities $u : A \to \mathbb{Z}_{\geq 0}$, and $s, t \in V$, $s \neq t$.

Task: Find a maximum s-t flow in G, i.e., an s-t flow f that maximizes $\nu(f)$.

Definition 4.2: s-t cut

An s-t cut is a set $C \subseteq V$ such that $s \in C$ and $t \notin C$. Furthermore, in the context of a maximum flow problem with capacities $u\colon A \to \mathbb{Z}_{\geq 0}$, the *value* of an s-t cut C is defined as $u(\delta^+(C))$. An s-t cut C is called *minimum* if it has minimum value among all s-t cuts.



$u(\delta^+(S_1)) = 4 + 3 + 5 = 12$

$u(\delta^+(S_2)) = 3 + 1 + 4 = 8$

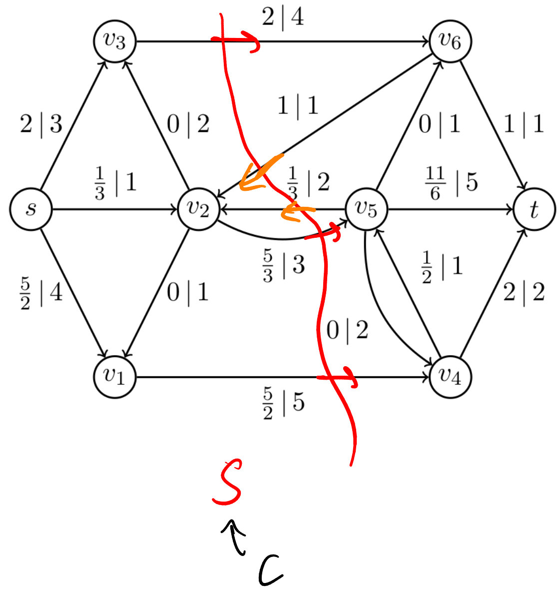
$u(\delta^+(S_3)) = 1 + 3 + 1 + 2 = 7$

$u(\delta^+(S_4)) = 6$

Lemma 4.3: Value of a flow expressed via an s-t cut

Let f be an s-t flow and $C \subseteq V$ an s-t cut. Then

$$\nu(f) = f(\delta^+(C)) - f(\delta^-(C)) \ .$$



$$\nu(f) = \underbrace{f(\delta^+(s))}_{\underbrace{2+\frac{1}{3}+\frac{5}{2}}_{4+\frac{5}{6}}} - \underbrace{f(\delta^-(s))}_{=0} = 4+\frac{5}{6}$$

$$\nu(f) = \underbrace{f(\delta^+(S))}_{\underbrace{2+\frac{5}{3}+\frac{5}{2}}_{6+\frac{1}{6}}} - \underbrace{f(\delta^-(S))}_{1+\frac{1}{3}} = 4+\frac{5}{6}$$

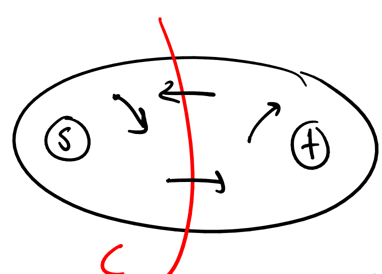
Proof of Lemma 4.3

$$\nu(f) = f(\delta^+(s)) - f(\delta^-(s)) + \sum_{v \in C\setminus\{s\}} \Big(\underbrace{f(\delta^+(v)) - f(\delta^-(v))}_{=0}\Big)$$

balance constraints

$$= \sum_{v \in C} \Big(f(\delta^+(v)) - f(\delta^-(v))\Big)$$

$$= f(\delta^+(C)) - f(\delta^-(C))$$



#

Theorem 4.5: Weak max-flow min-cut theorem

Let f be an s-t flow and let $C \subseteq V$ be an s-t cut. Then

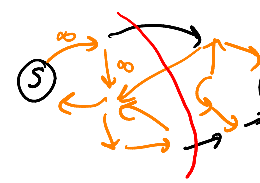
$$\nu(f) \leq u(\delta^+(C)) \ .$$

In other words, the value of a maximum s-t flow is upper bounded by the value of a minimum s-t cut.

<u>Proof</u>

$$\nu(f) \overset{\text{Lemma 4.3}}{=} \underbrace{f(\delta^+(C))}_{\leq u(\delta^+(C))} - \underbrace{f(\delta^-(C))}_{\geq 0} \leq u(\delta^+(C))$$

#

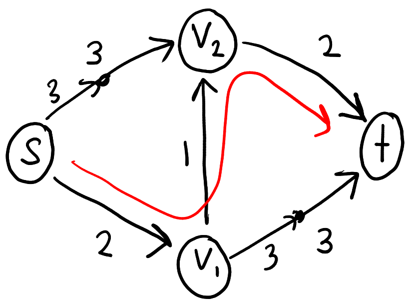


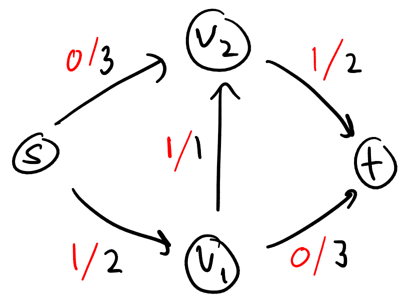
<u>Remark</u>: use of infinite (∞) capacities

Even though the capacities $u: A \to \mathbb{Z}_{\geq 0}$ in a flow problem are are assumed to be non-negative integers, it is common to also allow the use of infinite capacities, i.e., $u(a) = \infty$.
This can be reduced to case of finite capacities because

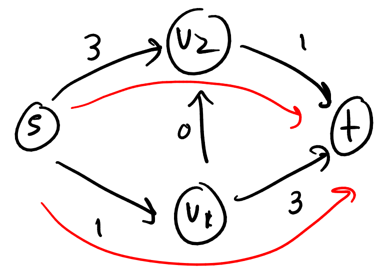
- Either there is an s-t path consisting only of arcs with infinite capacity, in which case the max flow value is ∞. (Can be checked via BFS.)
- Or the infinite capacities can be replaced by large finite capacities (e.g. the sum of all finite capacities).

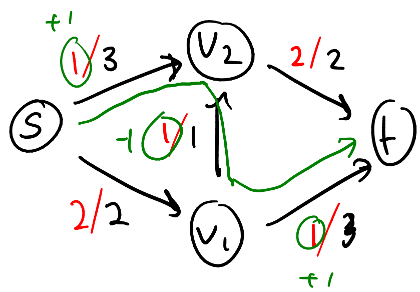
4.2 Algorithm of Ford-Fulkerson and strong max-flow min-cut theorem

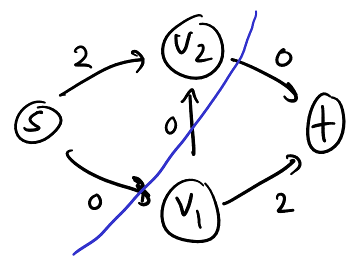




"leftover capacities"



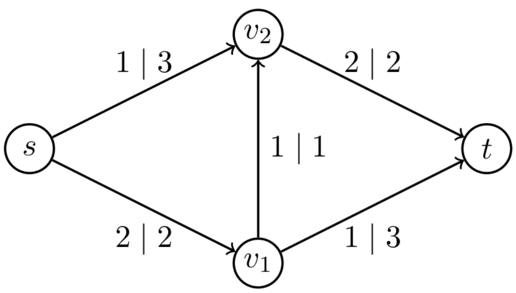


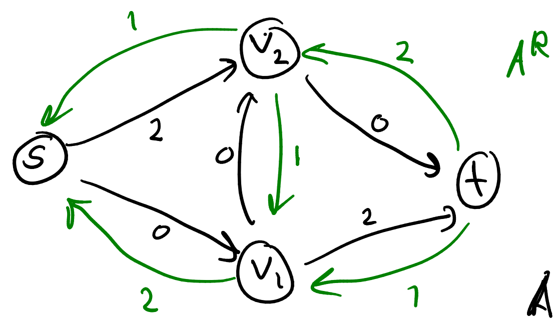


Definition 4.7: f-residual graph & f-residual capacities

Let f be an s-t flow in G. The *f-residual graph* $G_f = (V, B)$ with *f-residual capacities* $u_f : B \to \mathbb{Z}_{\geq 0}$ is defined as follows. The set of arcs $B := A \cup A^R$ contains all original arcs A together with reverse arcs A^R , where for $a \in A$, the set A^R contains an arc a^R that is *antiparallel* to a, i.e., the head of a^R is the tail of a and vice versa. Furthermore,

$$u_f(b) := \begin{cases} u(b) - f(b) & \text{if } b \in A \ , \\ f(a) & \text{if } b = a^R \in A^R \ . \end{cases}$$





$G = (V, A)$

Definition 4.8: f-augmenting path/augmenting path

Let f be an s-t flow in G. An *f-augmenting* path $P \subseteq B$ is an s-t path in $G_f = (V, B)$ with $u_f(b) > 0 \ \forall b \in P$.

Definition 4.9: Augmentation

The *augmentation* of an s-t flow f in G along an f-augmenting path $P \subseteq B$, where $G_f = (V, B)$ is the f-residual graph, is the flow f' in G defined as

$$f'(a) = \begin{cases} f(a) + \gamma & \text{if } a \in P \ , \\ f(a) - \gamma & \text{if } a^R \in P \ , \\ f(a) & \text{if } a, a^R \notin P \ , \end{cases}$$

where $\gamma := \min\{u_f(b) : b \in P\} > 0$. We call the value γ the *augmentation volume* of the augmenting path P.

Lemma 4.10: Running time for finding f-augmenting paths

Let f be an s-t flow in G and denote the number of arcs and vertices in G by m and n, respectively. If there is an f-augmenting path, then such a path can be found in $O(m+n)$ time via breadth-first search.

Proof

See script.

Algorithm 3: Ford and Fulkerson's algorithm to find a maximum s-t flow

Input : Directed graph $G = (V, A)$ with arc capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and $s, t \in V$, $s \neq t$.
Output: A maximum s-t flow f.

1. **Initialization:**
   $f(a) = 0 \; \forall a \in A$.
2. **while** ($\exists$ f-augmenting path P in G_f) **do:**
   Augment f along P and set f to be the augmented flow.
3. **return** f.

Example run of Ford and Fulkersons algorithm