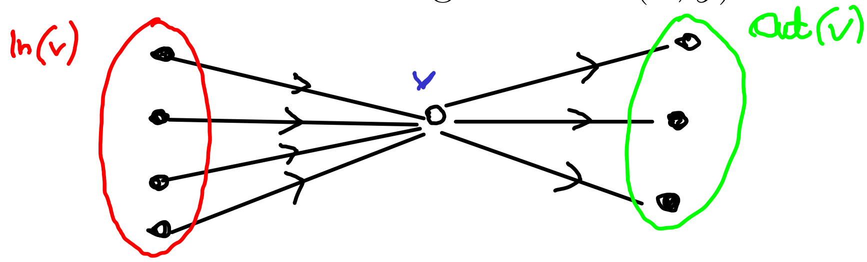
Flows in networks

A network is a directed graph $G = (V, A)$ together with some numerical data.

A flow is a function $f : A \longrightarrow \mathbb{R}$; for $(x, y) \in A$ the value $f(x, y)$ determines the flow through the arc (x, y).



Notation.

$$\mathrm{In}(v) = \{x \in V : x, v) \in A\},$$
$$\mathrm{Out}(v) = \{y \in V : (v, y) \in A\}$$

Flow Conservation Principle at the vertex $v \in V$

$$\sum_{x \in \mathrm{In}(v)} f(x, v) = \sum_{y \in \mathrm{Out}(v)} f(v, y)$$

Max Flow Problem

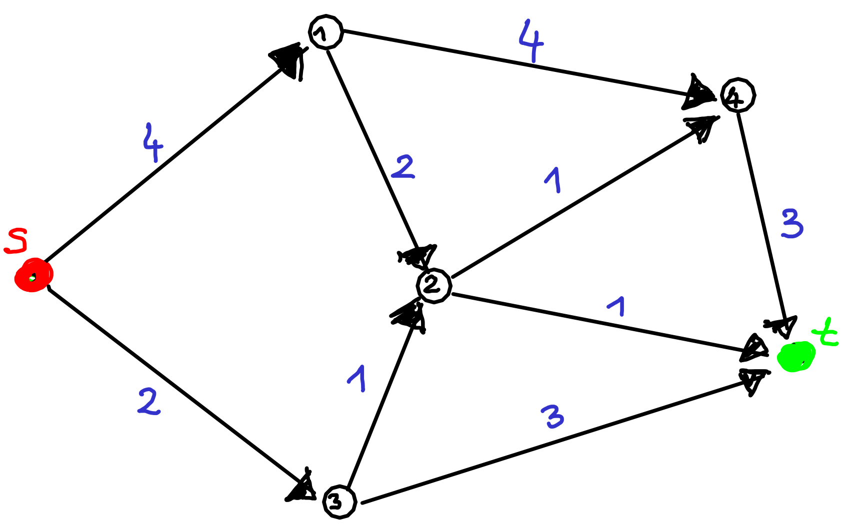
Given $G = (V, A)$ with a function $u : A \to \mathbb{R}_+ \cup \{\infty\}$; here u_{ij} is the capacity of the arc $(x, y) \in A$. We fix two different vertices $s, t \in V$ and assume that $\mathrm{In}(s) = \mathrm{Out}(t) = \emptyset$.

Max flow problem. We consider flows f such that

$$\left.\begin{array}{l}\text{(i) } 0 \leqslant f(x,y) \leqslant u_{(x,y)} \text{ for every } (x,y) \in A;\\ \text{(ii) FCP holds in every } x \in V \setminus \{s,t\}.\end{array}\right\} \text{feasible}$$

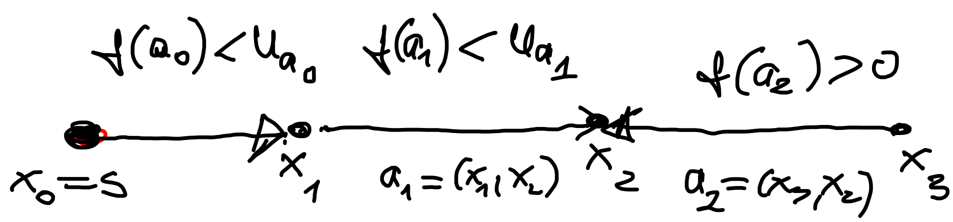
Find a maximal flow f through such a network, i.e. f maximizing the value

$$vol(f) = \sum_{y \in \mathrm{Out}(s)} f(s, y) = \sum_{x \in \mathrm{In}(t)} f(x, t).$$



Ford-Fulkerson Algorithm — an outline

(1) Start from some feasible flow f (e.g. $f = 0$).
(2) Check if f can be augmented (=made larger); if not then STOP.
(3) Augment f to f' and repeat.

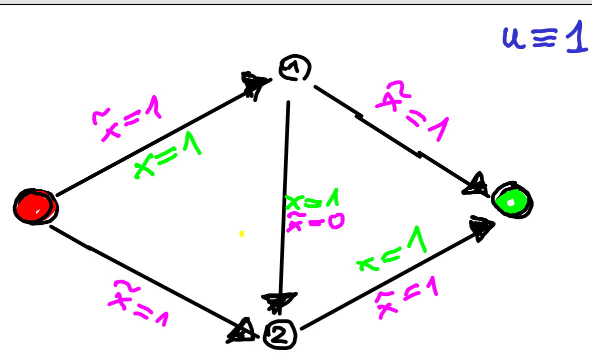


Augmenting path P (for some feasible flow f given) is a sequence

$$s = x_0, a_0, x_1, a_1, x_2, a_2, \ldots, a_{n-1}, x_n = t,$$

where for every i
(1) $a_i = (x_i, x_{i+1}) \in A$ and $f(a_i) < u_{a_i}$ (a forward arc); or
(2) $a_i = (x_{i+1}, x_i) \in A$ and $f(a_i) > 0$ (a backward arc).
If P is such a path then, writing F and B for the forward and backward arc on P we define

$$\delta(P) = \min\left(\min_{a_i \in F}(u_{a_i} - f(a_i)), \min_{a_i \in B} f(a_i)\right) > 0$$



Augmenting the flow

Theorem. Let f be a feasible flow and let P be an augmenting path (with forward arcs F and backward arc B). Then the flow $\widehat{f}$ given by

$$\widehat{f}(x,y) = \begin{cases} f(x,y) + \delta(P) & \text{when } (x,y) \in F \\ f(x,y) - \delta(P) & \text{when } (x,y) \in B \\ f(x,y) & \text{otherwise} \end{cases}$$

is also feasible and $vol(\widehat{P}) = vol(P) + \delta(P)$.

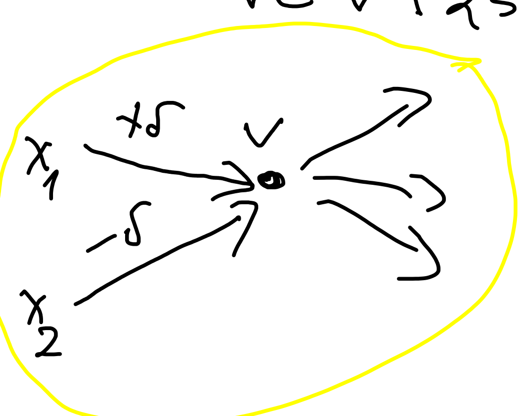
from the definition of $\delta(P)$:

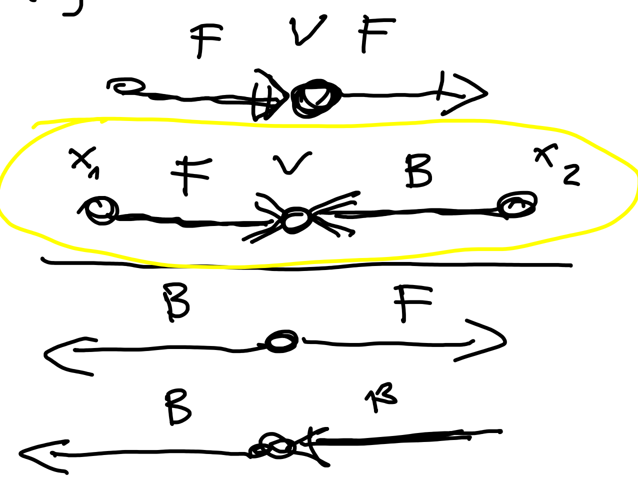
$$\hat{f}(x,y) \geq 0$$

$$\hat{f}(x,y) \leq u_{(x,y)}$$

Note that $\hat{f}$ ~~is~~ ~~...~~ satisfies FCP at each $\cancel{v} \in V \setminus \{s,t\}$

$v \in V \setminus \{s,t\}$

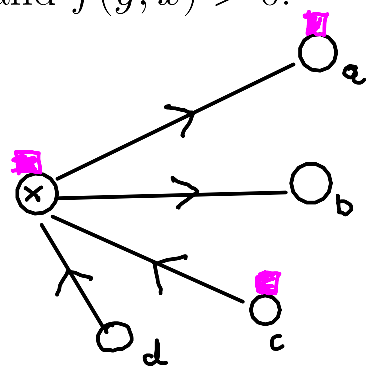




Serching for augmenting paths

We fix some feasible flow f. We use the following jargon:

- **scanning a vertex** $x \in V$: labelling $y \in V$ such that $(x, y) \in A$ and $f(x, y) < u_{(x,y)}$ and $y \in V$ such that $(y, x) \in A$ and $f(y, x) > 0$.



$f(x,a) < u_{(x,a)}$

$f(x,b) = u_{(x,b)}$

$f(c,x) > 0$

$f(d,x) = 0$

Labeling algorithm.

(1) Put $I = \{s\}$; $L = \{s\}$.

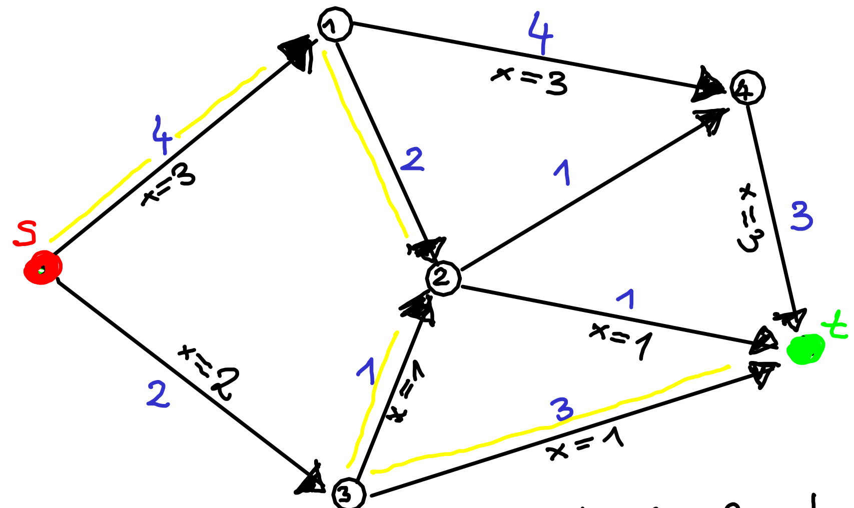
(2) If $t \in L$ then STOP (announce finding an augmenting path).

(3) If $I = \emptyset$ then STOP (no such paths).

(4) Choose any $x \in I$ and scan x; remove x from I and incorporate to I and to L all the vertices with new labels.

(5) GoTo (2).

.



$I = E = \{s\}$	Scan s $I = \{1\}$ $E = \{s, 1\}$	Scan 1 $I = \{2, 4\}$ $E = \{s, 1, 2, 4\}$	Scan 2 $I = \{3, 4\}$ $E = \{s, 1, 2, 4\}$	Scan 3 $t \in E$!

Why the flow is maximal (if there are no AP)?

Definition. A cut is any $S \subseteq V$ such that $s \in S$, $t \notin S$. The capacity of the cut S is defined as

$$c(S) = \sum_{x \in S. y \notin S, (x,y) \in A} u_{(x.y)}.$$

Theorem. We have $vol(f) \leqslant c(S)$ for every feasible flow f and every cut S.
If f and S satisfy $vol(f) = c(S)$ then the flow f is maximal.

Theorem. If the labelling algorithm finds no augmenting paths then the given flow is maximal.