The augmenting path algorithm

<u>Def</u> : given a flow x, an <u>augmenting path</u> is a directed path from s to t in $G(x)$; its <u>residual capacity</u> is the minimum residual capacity of its arcs.

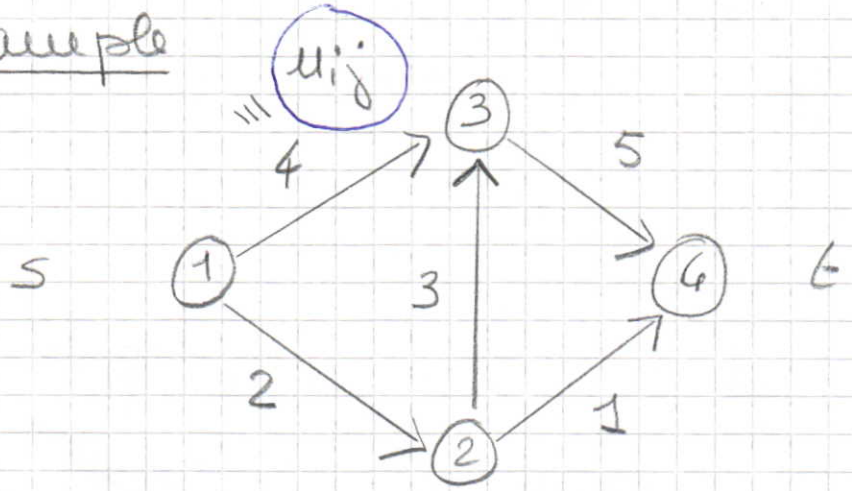
<u>The algorithm</u> : at each iteration finds an augmenting path and augments flow on this (= residual capacity) until there is no augmenting path

<u>Theorem</u> : a flow x^* is a maximum flow if and only if $G(x^*)$ contains no augmenting path

<u>Proof</u> : from the max-flow min-cut theorem (course RO)

Theorem (integrality): if the arc capacities are integer, then there exists an integer maximum flow.

example



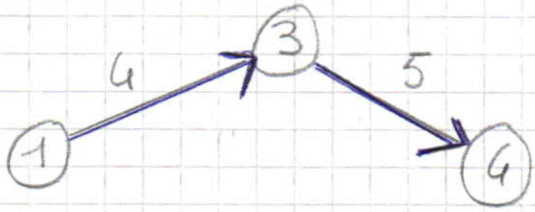
Initial flow

$x = 0$

$v = 0$

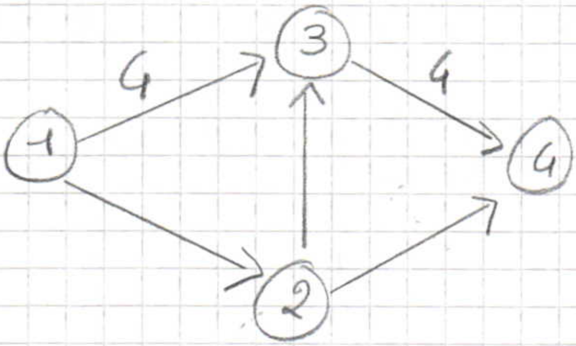
1) $G(x) = G$

aug. path



$\delta = \min\{4, 5\} = 4$ residual capacity

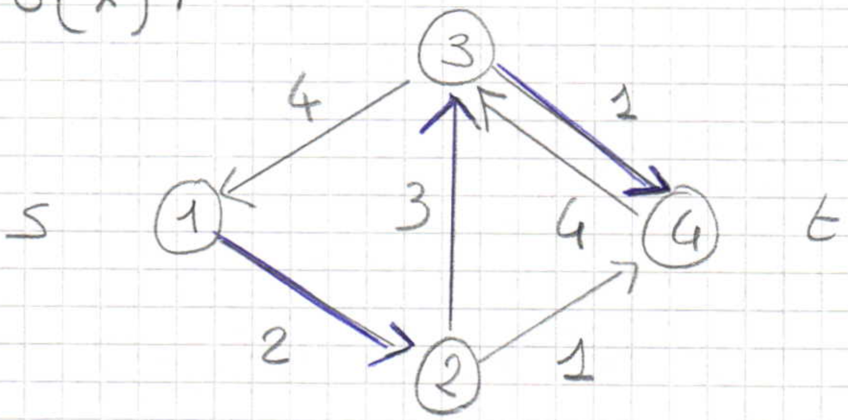
flow x



$v = 4$

NB: an integer flow at each iteration!

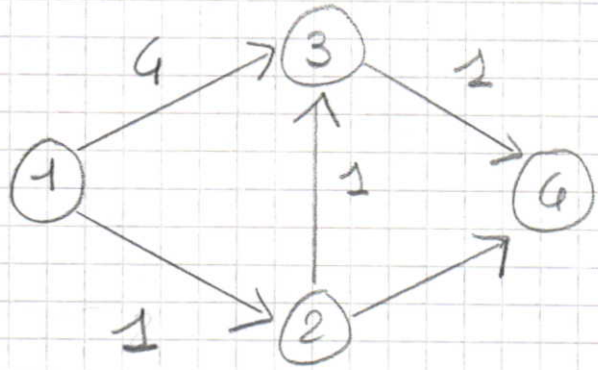
2) $G(x)$:



aug. path : $(1, 2, 3, 4)$

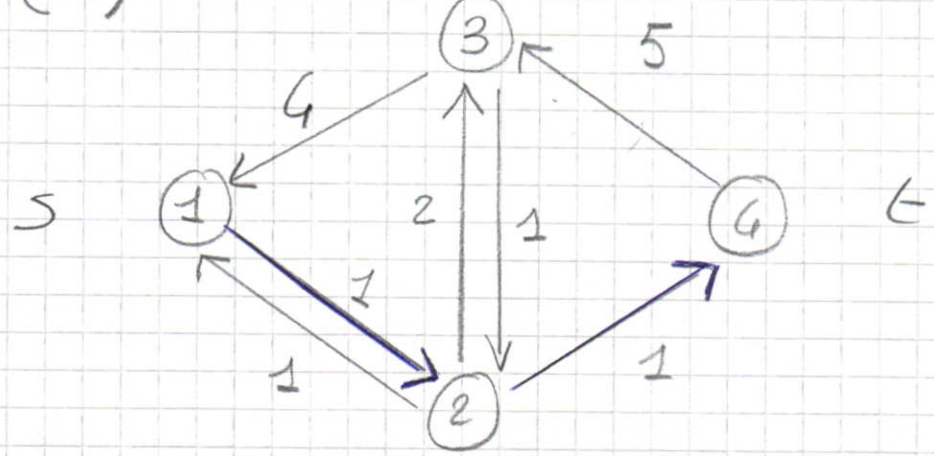
$$\delta = \min\{2, 3, 1\} = 1$$

flow x :



$v = 4 + \delta = 5$

2) $G(x)$:

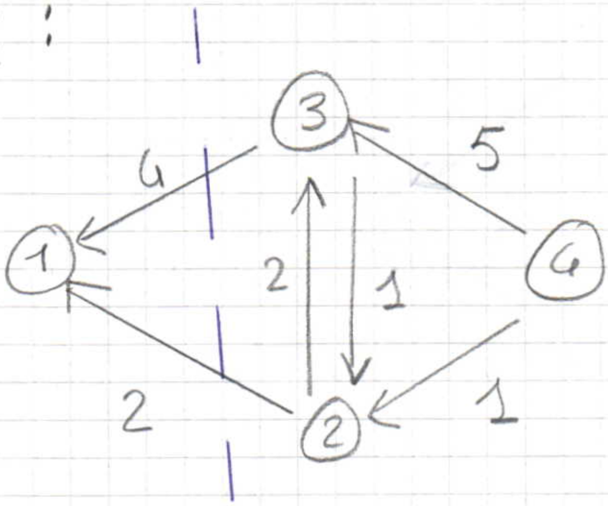


aug. path : $(1, 2, 4)$

$$\delta = \min\{1, 1\} = 1$$

<u>flow x :</u>

G(x): s



t

$v = 5 + \delta = 6$

<u>no augmenting path</u> :

STOP, x is an (integer) maximum flow

A minimum cut : $S = \{1\}$ $\bar{S} = \{2, 3, 4\}$

$$u[S, \bar{S}] = u_{12} + u_{13} = 6 = v$$

<u>Time complexity</u> $(u_{ij} \in \mathbb{Z}^+)$: $O(n \cdot m \cdot U)$,

where U maximum arc capacity

<u>Proof</u> :

- capacity of any s-t cut : $O(n \cdot U)$
- time to discover an augmenting path : $O(m)$

□