CptS 515. 9/16/2020.

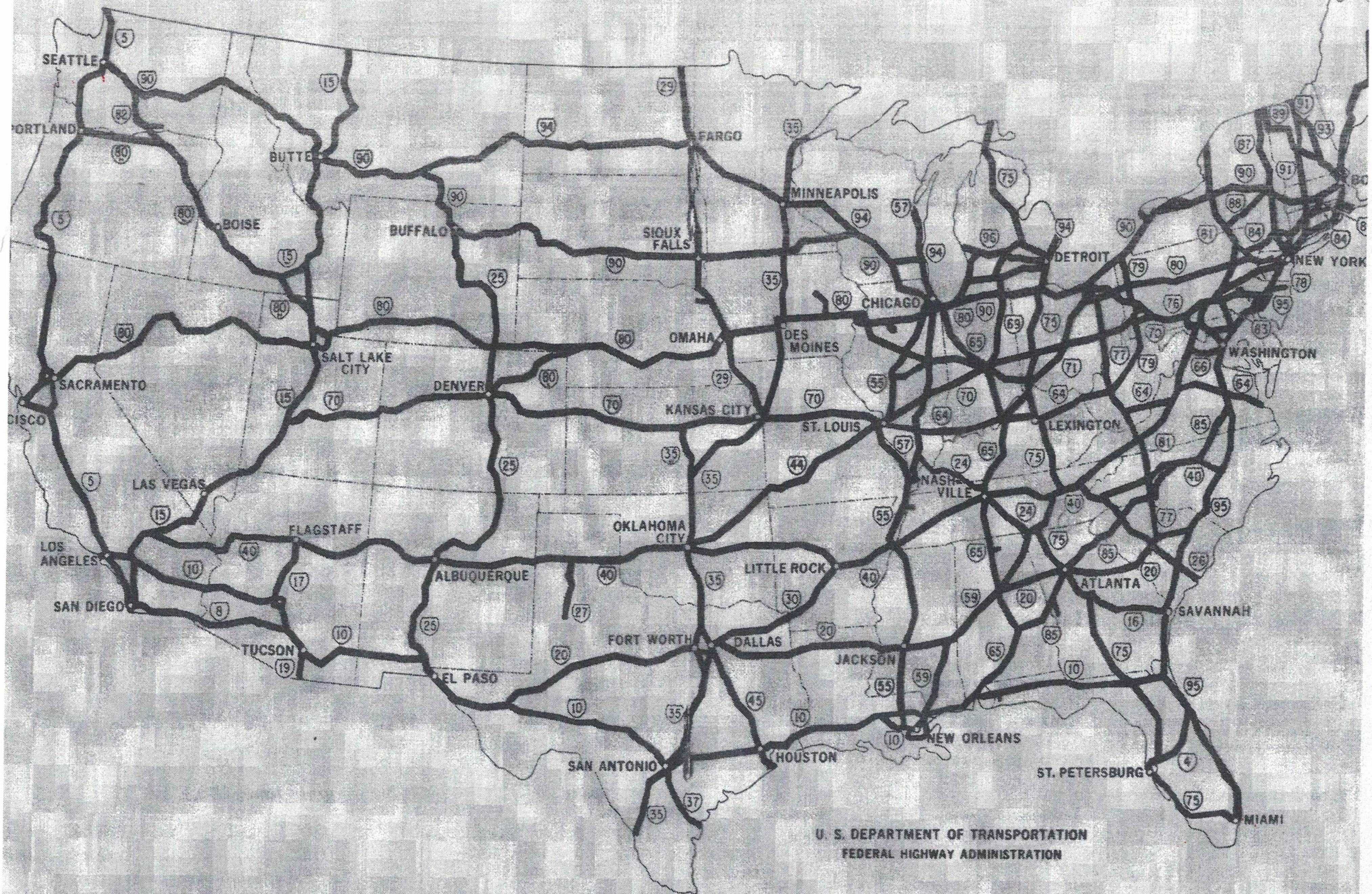
Today. Ford - Fulkerson Alg (max flow alg).
min-cut

FF-alg. (1956)

Lesson learned: (1). When naive greedy strategy doesn't work, what to do?
(local max doesn't automatically give you global max.)

(2). Even in 1950s, alg design was already sophisticated!

THE NATIONAL SYSTEM OF INTERSTATE AND DEFENSE HIGHWAYS

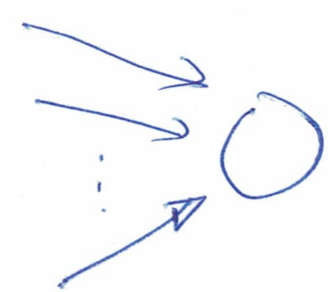


Max-flow & Minimal Cut

Consider a directed graph G where we have

(1). two designated nodes: s, t

s: source t: sink



$\neg \exists u : \langle u, s \rangle \in E$ $\neg \exists u . \langle t, u \rangle \in E$

(E is the set of edges in G)

(2). Each edge $e \in E$ is associated with a nonnegative integer, call capacity $C(e)$.

(3). Flow is a function on edges

$$f : E \longrightarrow R^+$$

(nonnegative reals)

Such that $f(e) \leq C(e)$.

There are conditions on flow:

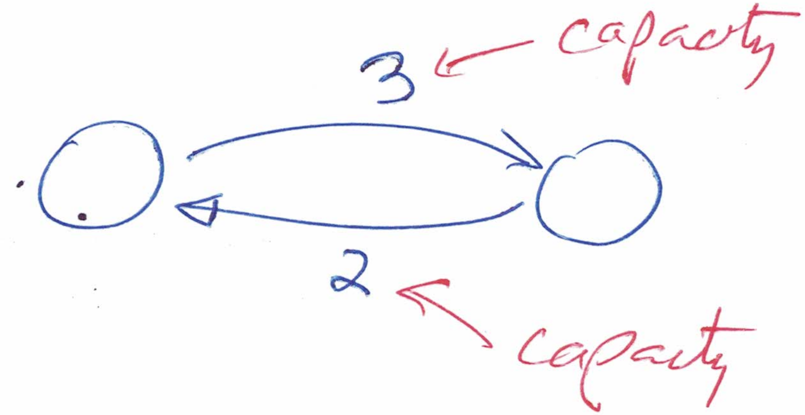
(1). flow-in == flow out:

For each node $v \neq s, t$:

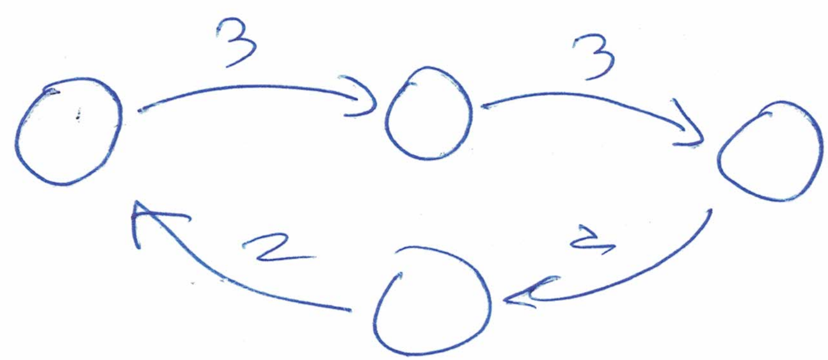
$$\sum_{u : \langle u, v \rangle \in E} f(\langle u, v \rangle) = \sum_{u : \langle v, u \rangle \in E} f(\langle v, u \rangle)$$

(flow-in of node v) (flow-out of node v)

(2). We do not assume we have edges like this:



(you can translate the above into



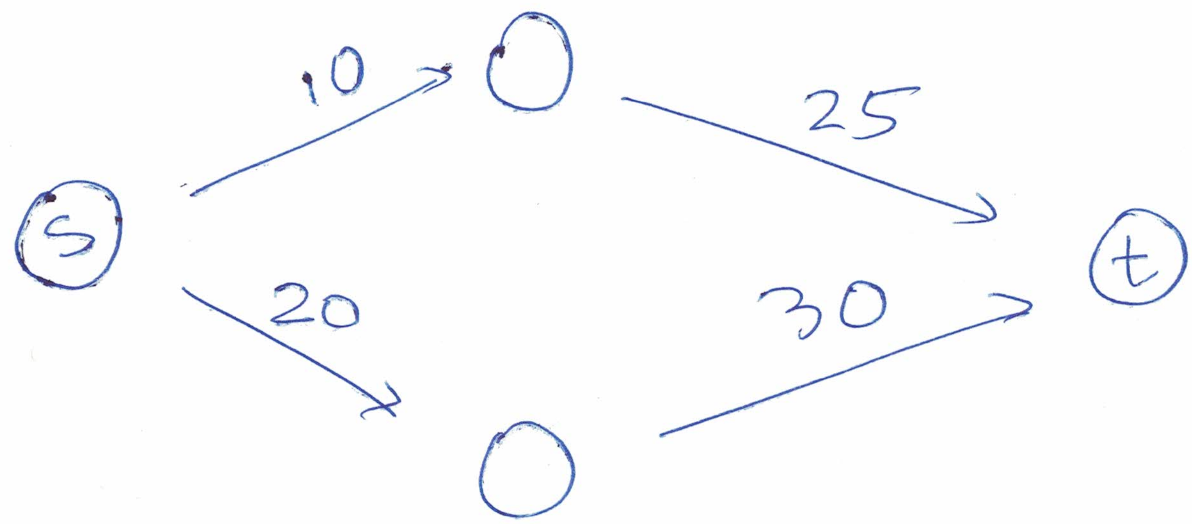
)

A flow (which is a value) of G is the total flow out of s (= the total flow into the sink t)

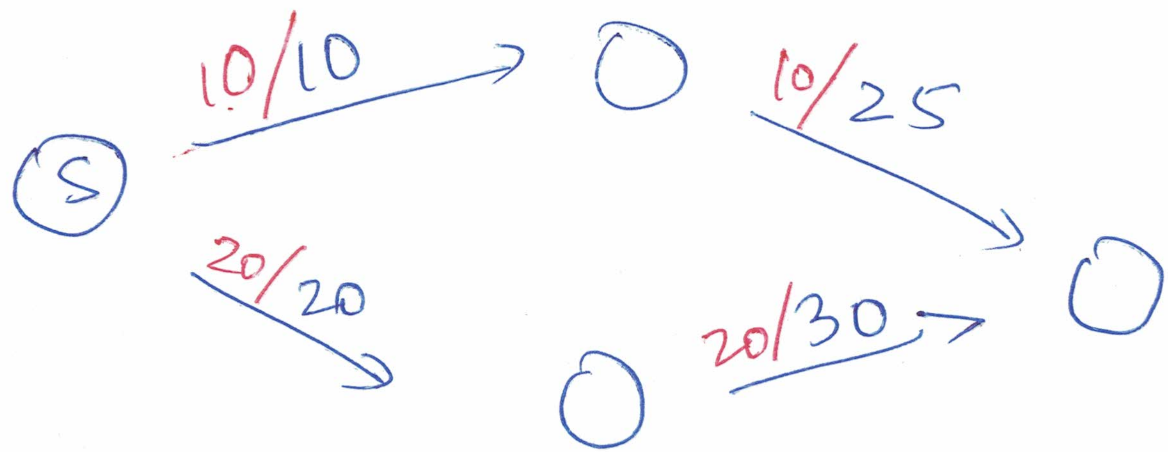
$$f(G) = \sum_{u : \langle s,u \rangle \in E} f(\langle s,u \rangle).$$

Max-flow problem is to find a flow f s.t. $f(G)$ is maximal.

Exaple:

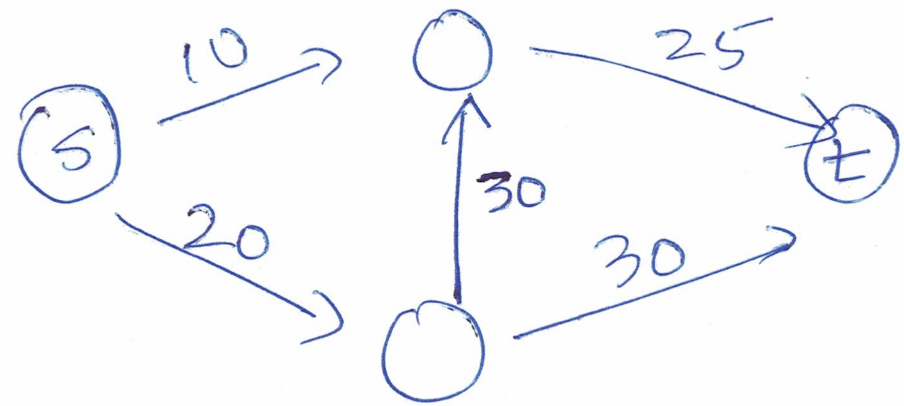


max-flow:



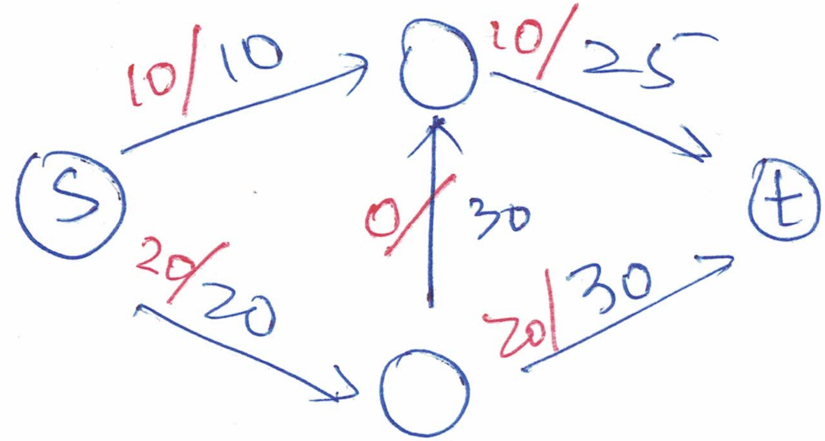
$f(G) = 10 + 20 = 30.$

more cplex exmple:



Two scenarios:

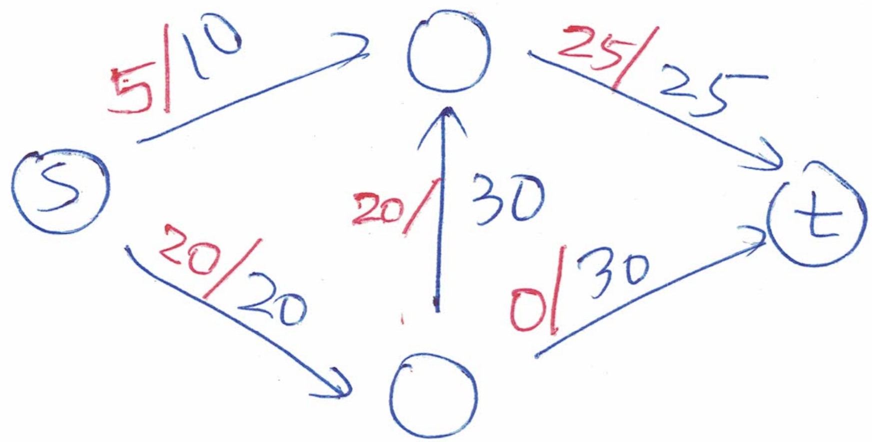
(1).



is local max!

$f(G) = 10 + 20 = 30.$

(2). different way to draw the flow

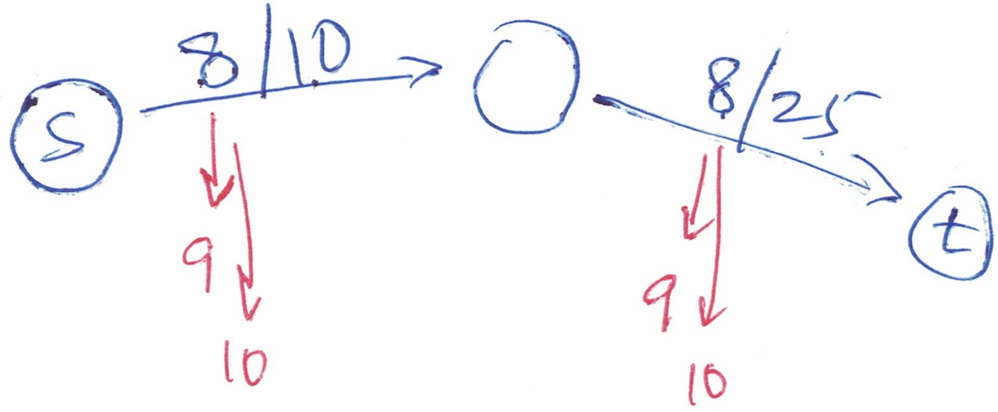


is also local max!

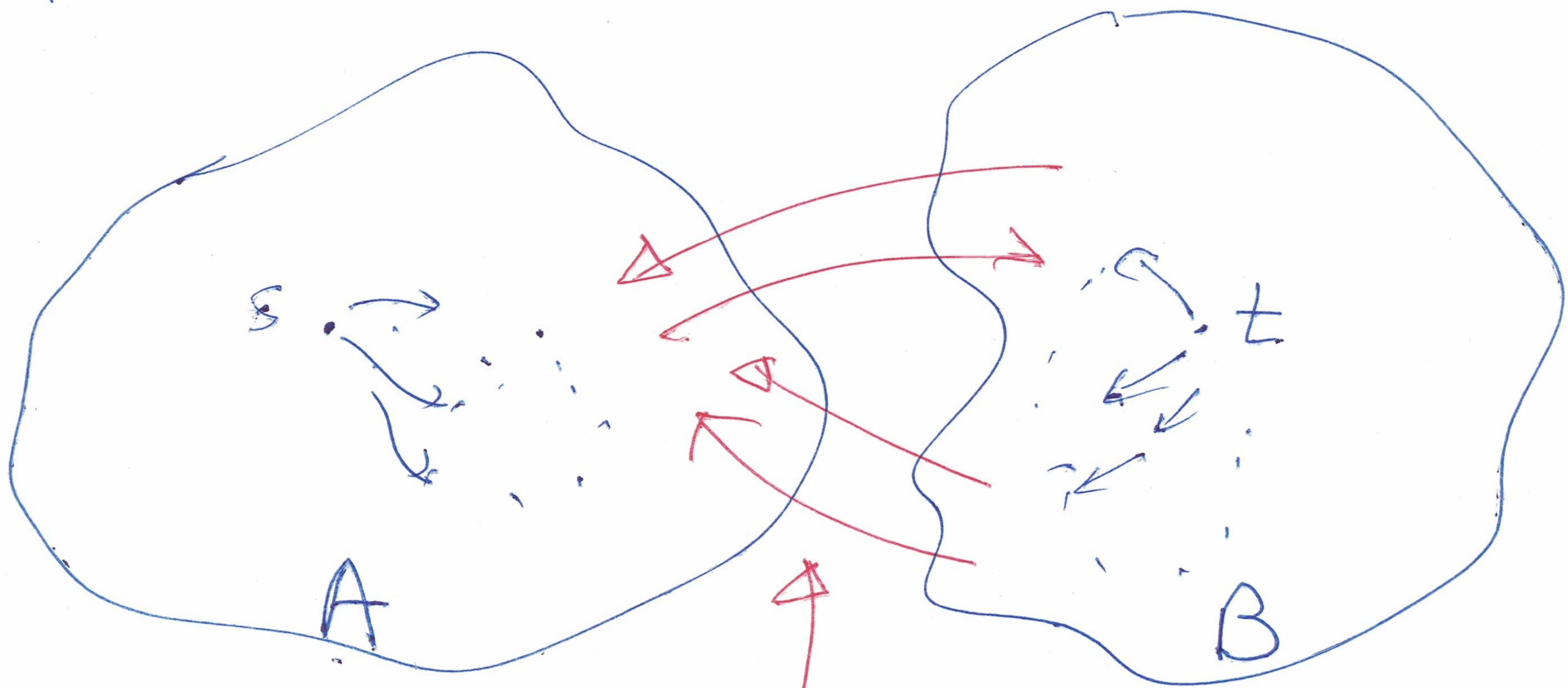
Conclusion: Since you have multiple local max scenarios, naive greedy approach doesn't work!

Study the problem <u>before</u> you even try to solve it.

Ob1. if on a path (from source to sink), for each edge, we still have room to send more flow, then the path is not yet max. out.



Obz. A cut is a partition of nodes in G such that s and t belong to two different partitions. Draw:



Called $\langle A, B \rangle$ (s-t cut).

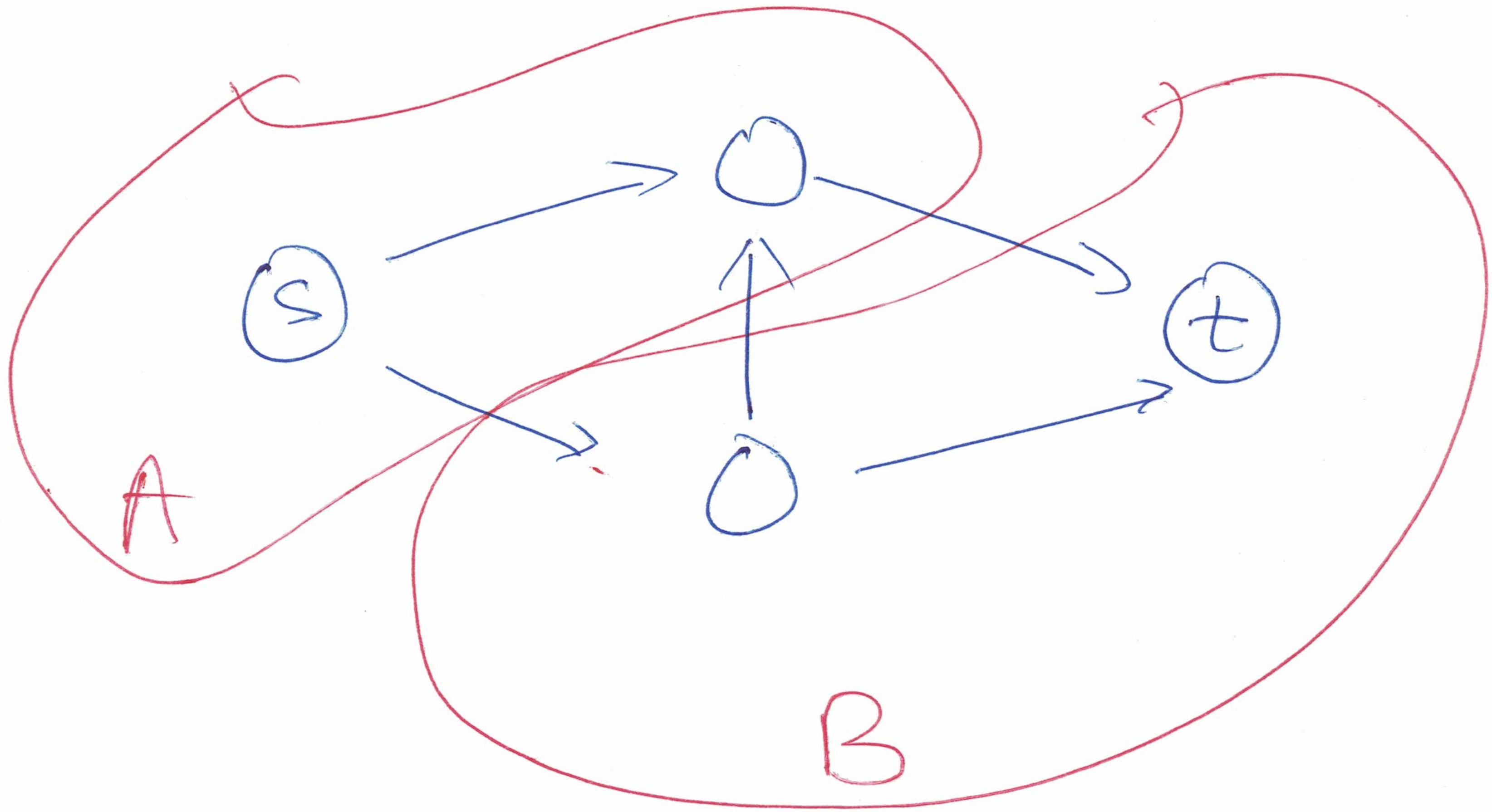
edges crossing the boundary.

Ob3. Notation,

$E_{A \to B}$: the set of all edges from a node in A to a node in B.

$E_{B \to A}$: the set of all edges from a node in B to a node in A.

Exaple on s-t cut:



For each flow f,

the total flow ~~from~~ from A to B

minus

the total flow from B to A

= the total flow value of G.

(the total flow-out of the source s)

= $f(G)$).

Formally, for each flow and each cut $\langle A, B \rangle$:

① $$\sum_{e \in E_{A \to B}} f(e) - \sum_{e \in E_{B \to A}} f(e) = f(G).$$

---

② For each flow f, the total flow value of $G \leq$ the total capacity of edges from A to B;

$$f(G) \leq \sum_{e \in E_{A \to B}} C(e).$$

// This can
// be easily proven
// by using ①.
//
//

3. Notice that in 2, the formula is true for <u>all</u> choices of $\langle A, B \rangle$ (indep. of how you do the s-t cut).

Then, for <u>all</u> flow f,

$$f(G) \le \min_{A,B} \sum_{e \in E_{A \to B}} C(e) .$$

If I have a flow f^* that reaches the upper bound, then the f^* is the max-flow that I am looking for.

Key observation by F-F:

There is a flow f^*, called max-flow that reaches the upper bound;

$$f^*(G) = \min_{A,B} \sum_{e \in E_{A \to B}} C(e).$$

Then, finding max-flow is equivalent to finding min-cut!

Key trick (to avoid naive greedy approach).