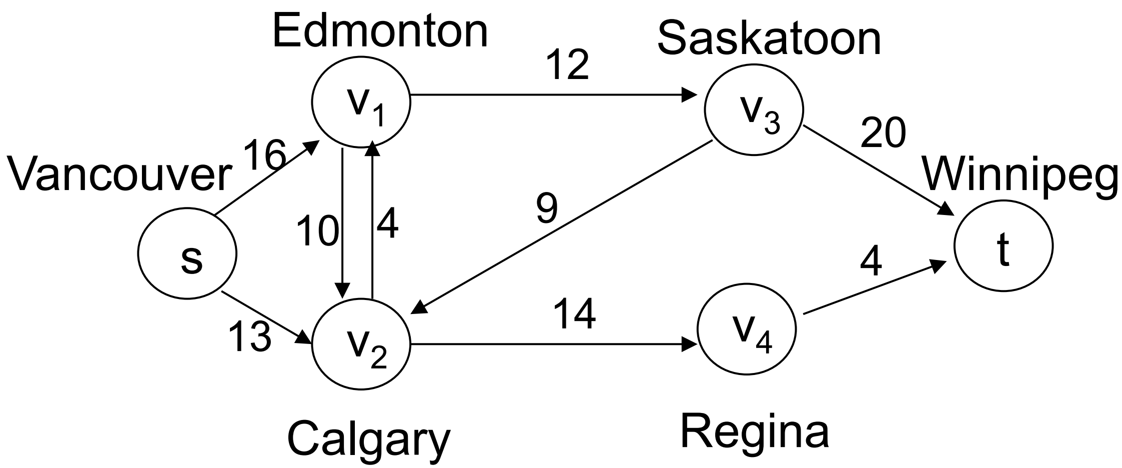
Flow problems



Let G=(V,E) be the graph of a flow problem. The company in Vancouver is called source s, the market in Winipeg target t. many goods must be brought to the target via some streets. Only $c_{u,v}$ such goods can be transported over (u,v). How can we transport as many goods as possible from Vancouver to Winnipeg, utilizing all paths from s to t?

Flow networks are used in order to model distribution problems, transport- and reload problems … Transported are water power, gas, cars …

Flow problem

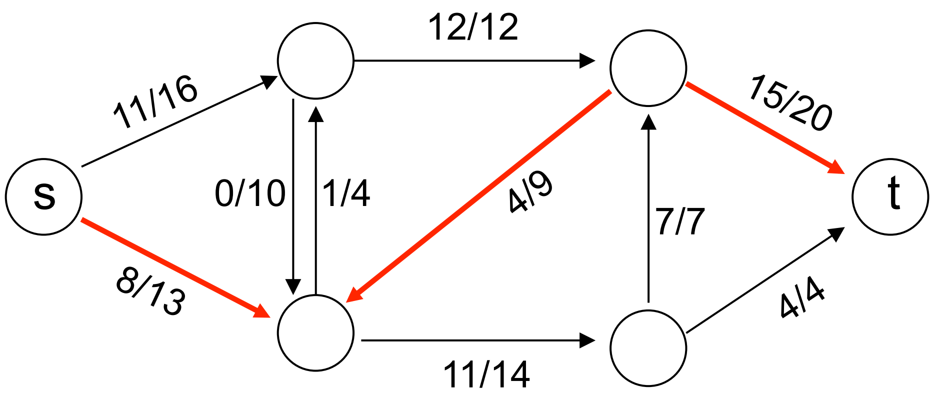
Flow network:
- $G=(V,E)$, directed graph,
- for all $(u,v) \in E$, we have a non-negative capacity restriction $c(u,v) > 0$
- if $(u,v) \notin E$, then $c(u,v)=0$
- there are two special nodes: source s and target t (also called sink)
- for each node v, a path from s to v and from v to t exists

Let $G=(V,E)$ be a flow network, let s be the source and t the sink. A **flow** in G is a function $f: V \times V \rightarrow \mathbb{R}$ with:
- capacity constraint: $f(u,v) \leq c(u,v)$ für alle $u,v \in V$
- symmetry: $f(u,v) = -f(v,u)$ für alle $u,v, \in V$
- flow conservation: $\sum_{v \in V} f(u,v) = 0$

The **value** of a flow is
  $|f|$ $= \sum_{v \in V} f(s,v)$, i.e. the total flow out of s

Def.: Given is a flow network and a feasible flow x from s to t. An „augmenting path“ (or „improving path“) is a path P from s to t, where the edge directions are ignored, with the following properties:

- For each edge (a,b), which is forward-directed in P, it is valid: f(a,b) < c(a,b). I.e. Forward edges have free capacities.
- For each edge (b,a), which is backwards-directed in P, it is valid: f(a,b) > 0.



Maximum change along P:

$$\min_{\text{edges of } P} \begin{cases} c(a,b)-f(a,b) & \text{following forward edges} \\ f(a,b) & \text{following backward edges} \end{cases}$$

Flow problem

Ford-Fulkerson Algorithm

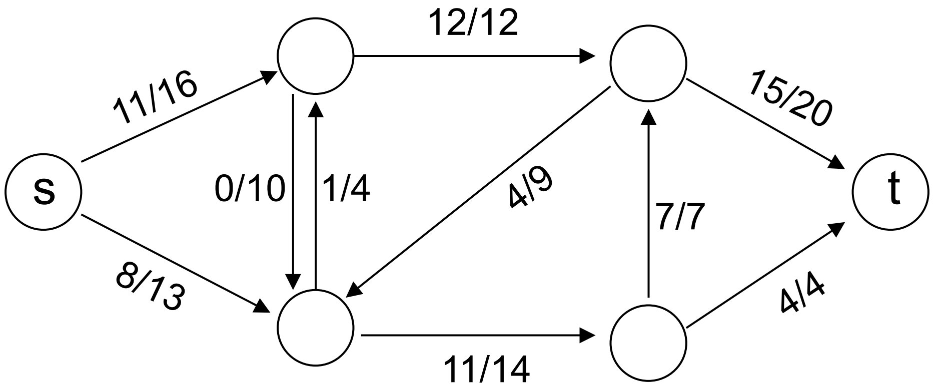
Ford-Fulkerson(G,s,t)
1. initialize flow to 0
2. while there is an augmenting path p do
3.   improve the flow along p
4. return f

We start with flow value 0 and increase the flow step by step.

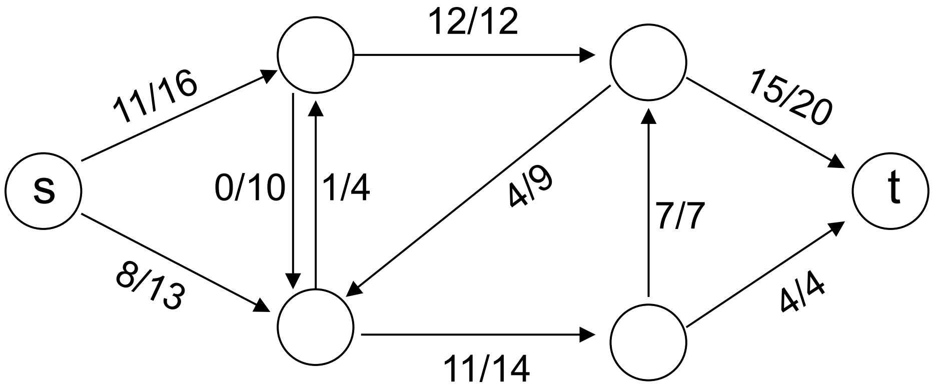
Flow problem
Residual Networks

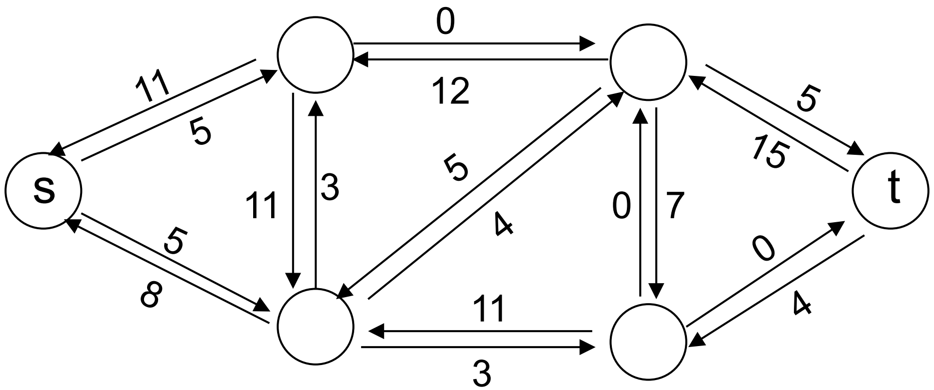
Let f be a flow in G. $c_f(u,v) = c(u,v) - f(u,v)$ is called **residual capacity**.

Let G=(V,E) be a flow network and f a flow. The **residual network** then is $G_f = (V,E_f)$ with $E_f = \{(u,v) \in V \times V \mid c_f(u,v)>0\}$. Note, in the residual graph may be more nodes than in the original graph.
(why? $f(v,u)>0 \Rightarrow f(u,v) < 0 \Rightarrow c_f(u,v) > c(u,v)$):

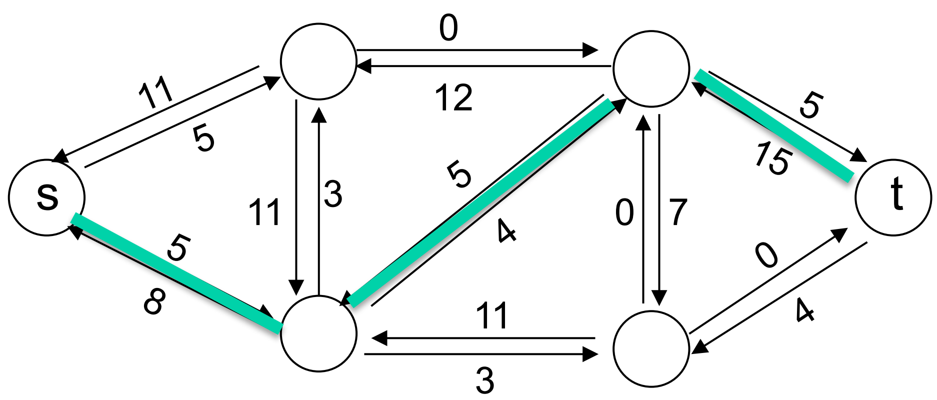


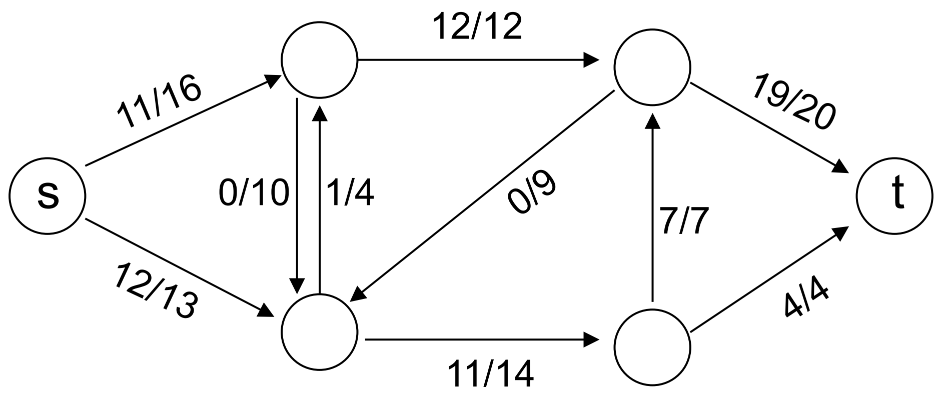
Flow problem
Residual Networks





Flow problem
Residual Networks





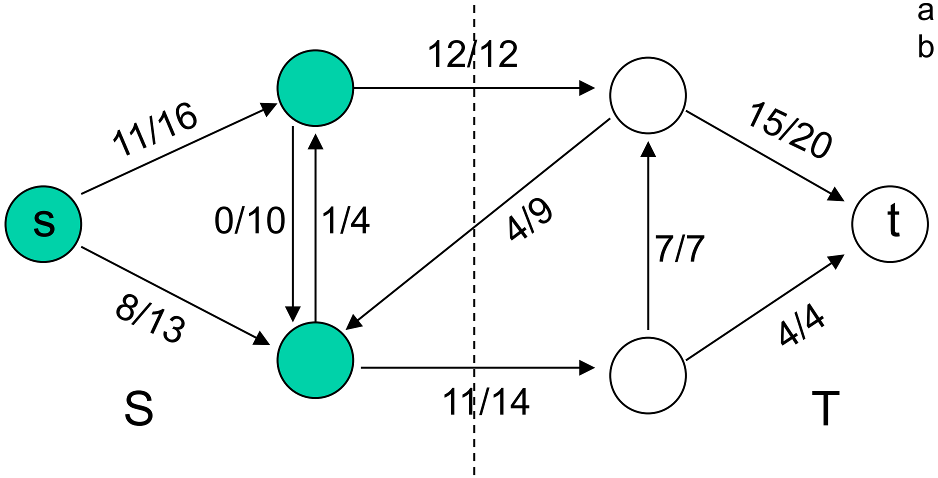
Flow problem

Cut:

A **cut** of a network is a partition of V into S and T = V \ S, such that $s \in S$ and $t \in T$.

If f is a flow, then a so called **netflow over a cut (S,T) is :** $\sum_{\substack{(a,b)\in E \\ a \in S \\ b \in T}} f(a,b)$

The **capacity** of a cut (S,T) is: $C(S,T) := \sum_{\substack{(a,b)\in E \\ a \in S \\ b \in T}} c(a,b)$



Netflow over (S,T) is 19
Capacity C(S,T) = 26

Note: If $(a,b) \notin E$, then it is $c(a,b) = 0$, but $f(a,b)$ possiblly is < 0.

Flow problems
Residual networks

Claim ResNet1: Let G = (V,E) be a flow network and let f be a flow. Let G‘ be the residual network of G and let f‘ be a flow in G‘ along an improving path. Then it is valid for the sum of the flows f + f‘: |f + f‘| = |f| + |f‘|

Proof: follows directly from the construction of G‘

Claim ResNet2: If (S,T) is a cut, the flow from S to T cannot be larger then the capacity of the cut.
Proof: Für each single edge (u,v) from S to T it is f(u,v) ≤ c(u,v). Thus it is also valid for the sum over all edges from S to T:
$\sum_{u \in S, v \in T} f(u,v) \leq \sum_{u \in S, v \in T} c(u,v)$

Max-flow min-cut Theorem

Let f be a flow in a flow network G=(V,E) with source s and sink t. Then, the following statements are equivalant to each other:

1. f is a maximum flow
2. the residual network G_f contains no augmenting path
3. there is a cut (S,T) such that $|f| = \sum_{u \in S, v \in T} c(u,v)$

1 ⇒ 2: Let us assume that f is a maximum flow, and G_f contains an augmenting (improving) path f'. However, the augmenting path is chosen such that it helps improving the flow. This would imply $|f + f'| > |f|$. Then f was not maximum.

2 ⇒ 3: Let no augmenting path exist. Then there is no path in G_f from s to t (, with capacities > 0). Define

$$S := \{v \in V \text{ with: there is a path from s to v in } G_f\}$$

Then it is (S,T=V\S) a partition and for each edge (u,v) with $u \in S$ and $v \in T$ it is $f(u,v)=c(u,v)$, because otherwise: $(u,v) \in E_f$.

Flow problems
Residual networks

Max-flow min-cut Theorem

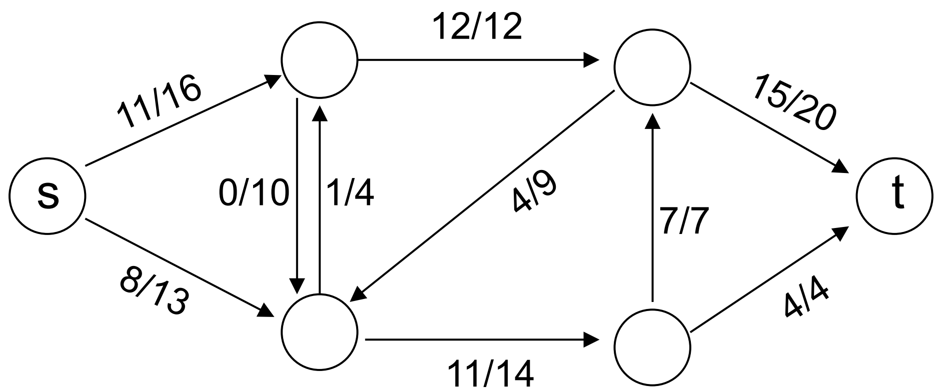
Let f be a flow in a network G=(V,E) with source s and sink t. Then the following statements are equivalent to each other:

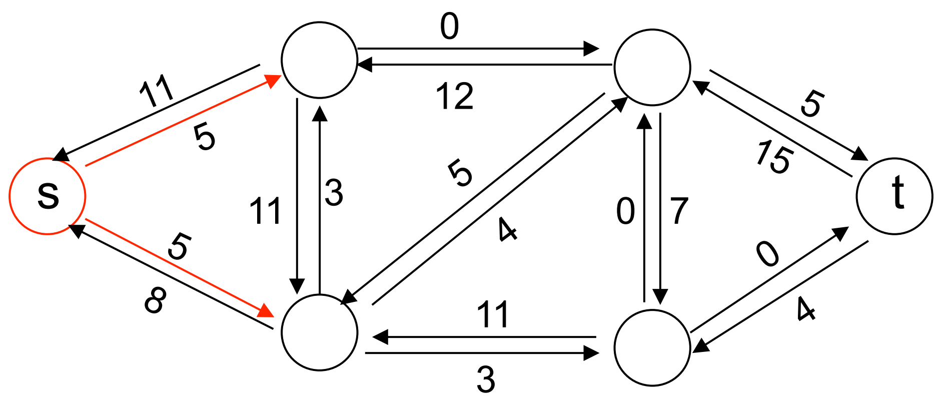
1. f is a maximum flow
2. The residual network G_f contains no augmenting path
3. there is a cut (S,T) such that $|f| = \sum_{u \in S, v \in T} c(u,v)$

3 ⇒ 1: Let $|f| = \sum_{u \in S, v \in T} c(u,v)$, for S and T as in point 2. Because of Caim ResNet2 there is no increasing flow.

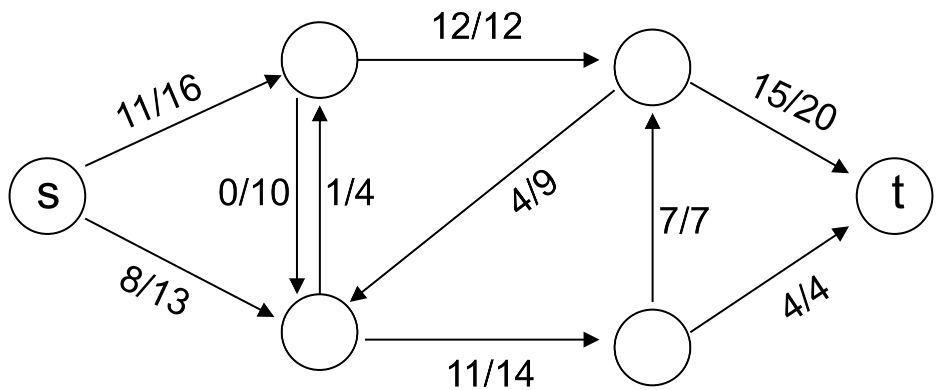
How do we find imrpving paths? With Breadth First Search.

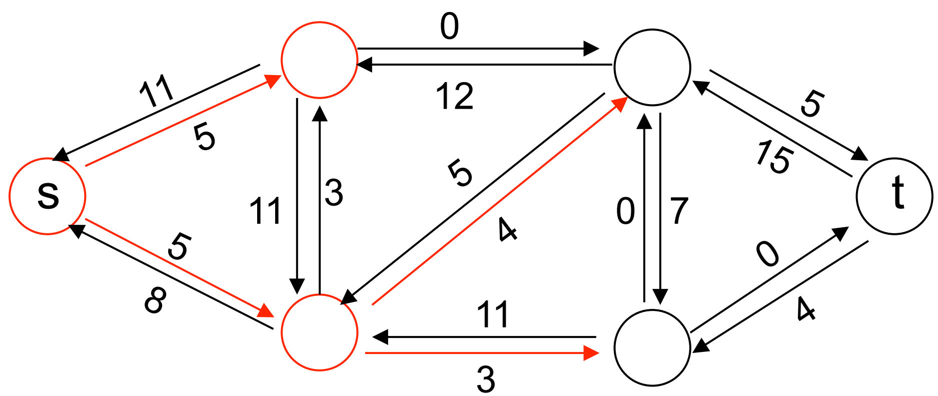
Flow problem
Residual Network



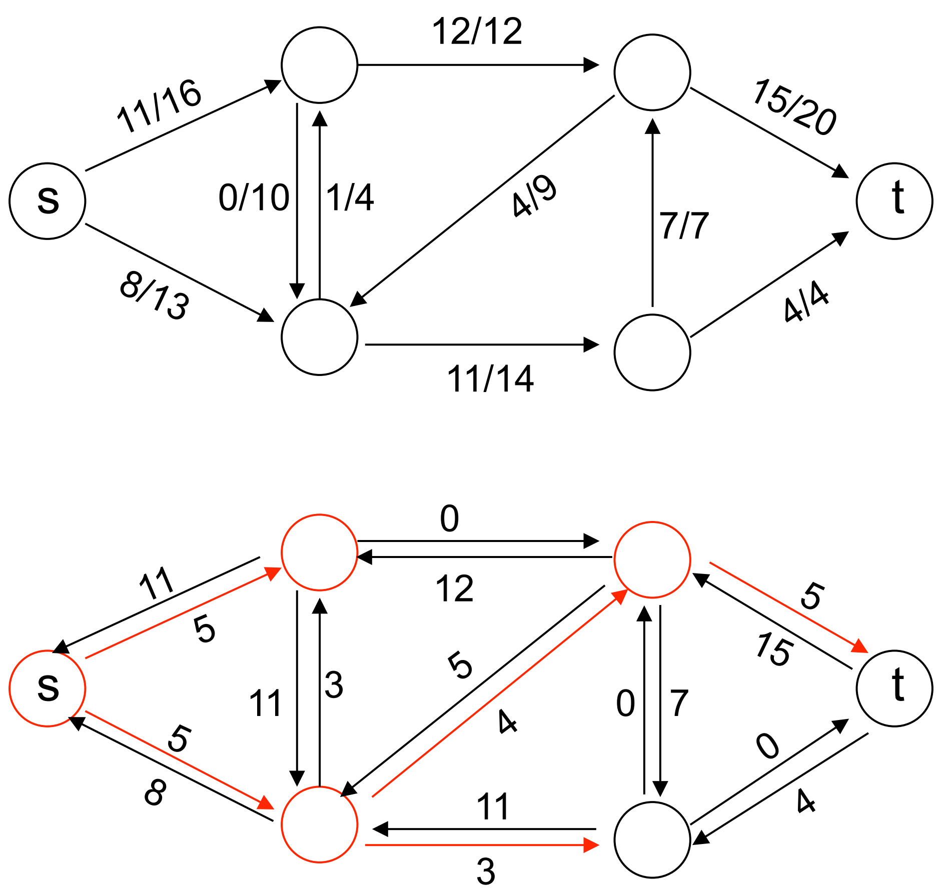


Flow problem
Residual Network





Flow problem
Residual Network



Flow problem

Ford-Fulkerson Algorithm

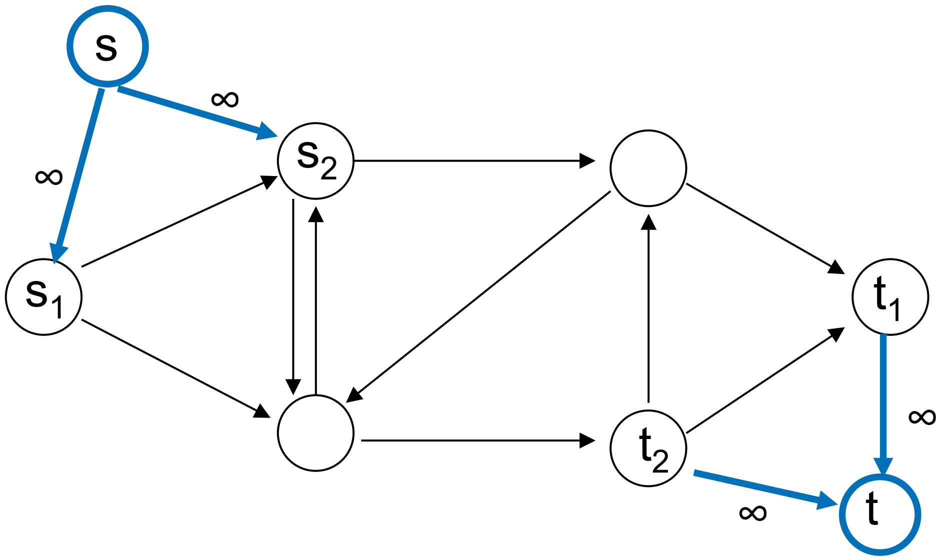
ClaimFF1: When the Ford-Fulkerson Algorithm halts, it terminates with optimal solution.
Prf.: After termination, build the sets S and T as in the Max-flow min-cut Theorem. All forward edges are then saturated, all backward edges empty. (Otherwise, the algorithm would not have halted). The (S,T)-cut has the same value as the flow delivered by the algorithm.

ClaimFF2: The Ford-Fulkerson Algorithm terminates after finitely many steps, as long as all input parameters are natural or rational numbers.

Prf.: natural numbers: clear, because the flow is increased by integer units.
Rational numbers: clear, because we can multiply all numbers with a common denominator.

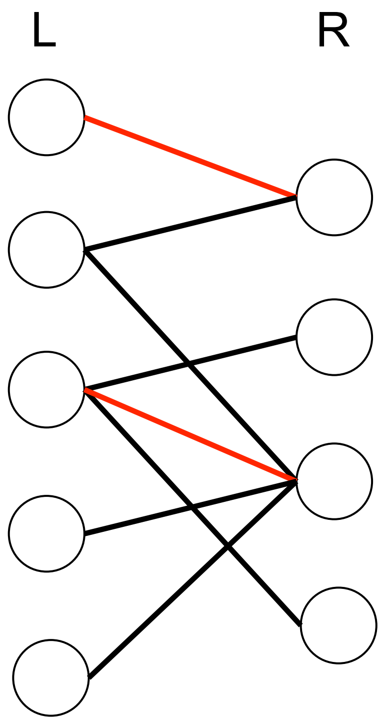
Adjunct problems of the maximum flow problem

1. Several sources and sinks

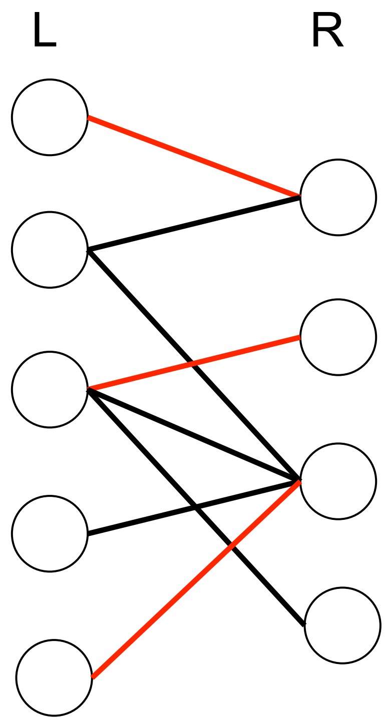


Adjunct problems of the maximum flow problem

2. Maximum Matching in bipartite graphs



Matching of size 2

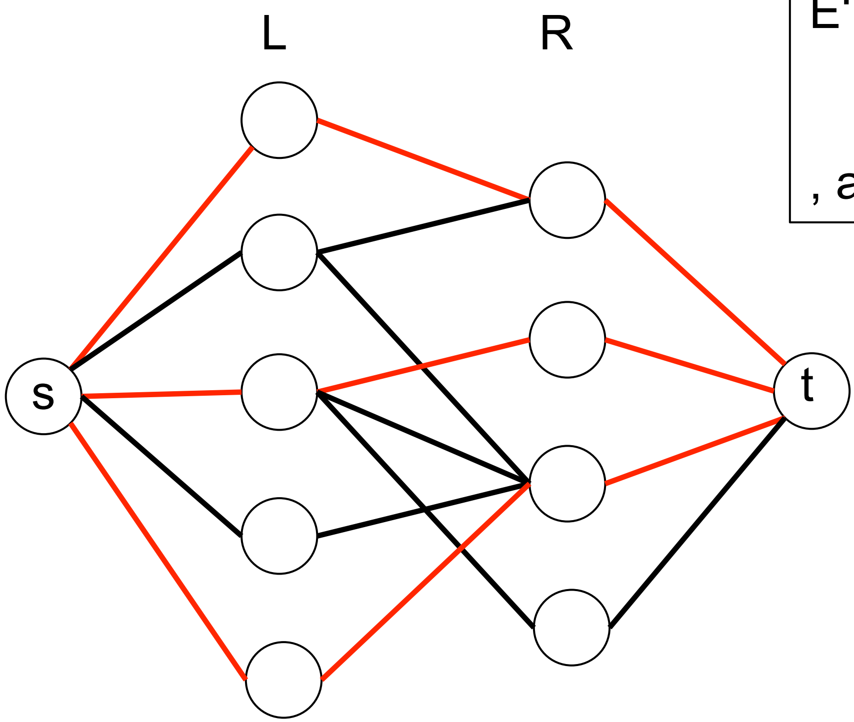


Maximum-Matching of size 3

Adjunct problems of the maximum flow problem

TECHNISCHE UNIVERSITÄT DARMSTADT

2. Maximum Matching in bipartiten Graphs



E‘ = { (s,u) : u∈L }
∪{(u,v) : u∈L,v∈R, (u,v)∈E}
∪{ (u,t) : u∈R }
, and unit weights for edges

Adjunct problems of the maximum flow problem

2. Maximum Matching in bipartiten graphs

Claim MaxBiMa: Let G=(V,E) be a bipartite graph with node partitioning $V = L \cup R$. Let G‘=(V‘,E‘) the corresponding flow network. Then:

If M is a matching in G, then there is an integer flow G‘ with $|f| = |M|$. If vice versa f is an integer flow in G‘, then there is a matching M in G with $|f|=|M|$.

Proof: Exercise