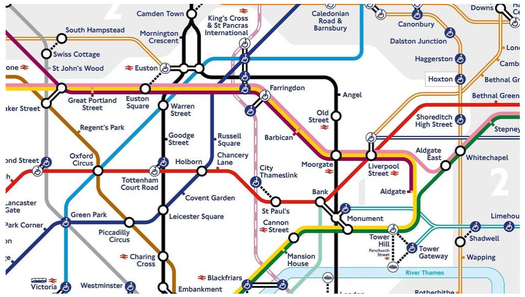
THE UNIVERSITY OF VERMONT
COLLEGE OF ENGINEERING &
MATHEMATICAL SCIENCES

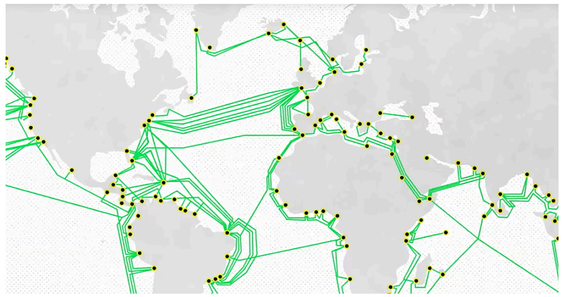
Network Flows

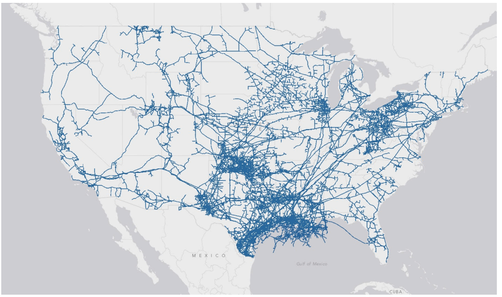
Max flow / min cut: Ford-Fulkerson, part two

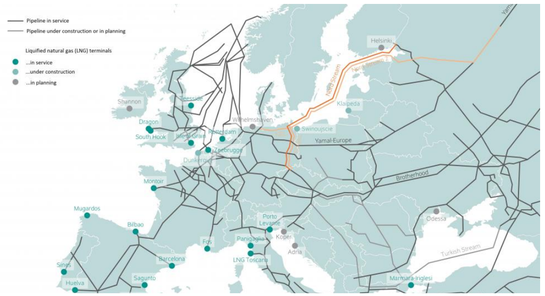
CS 124 / Department of Computer Science

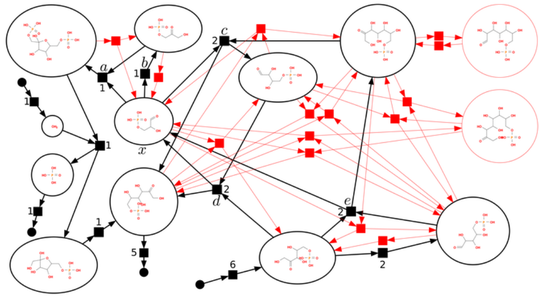
Network flow





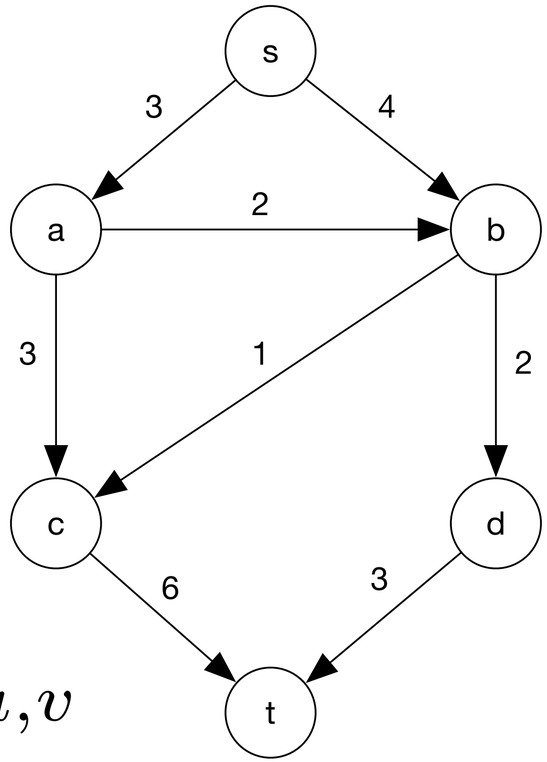






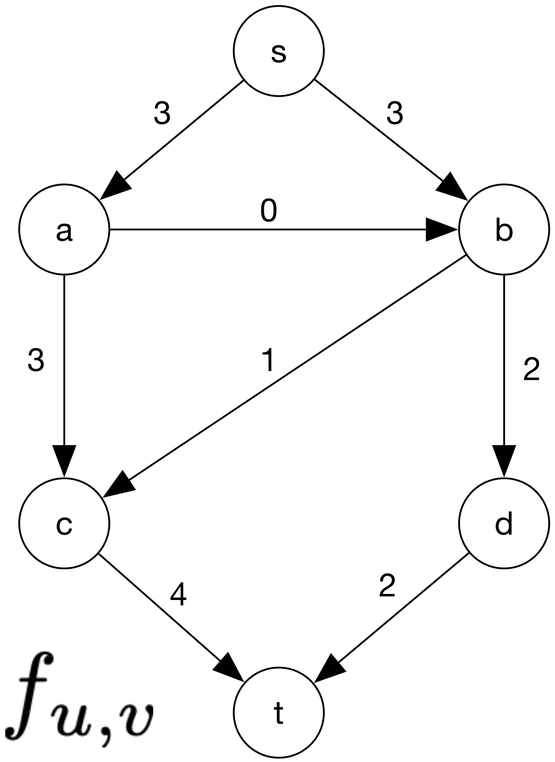
Network flow

CAPACITY



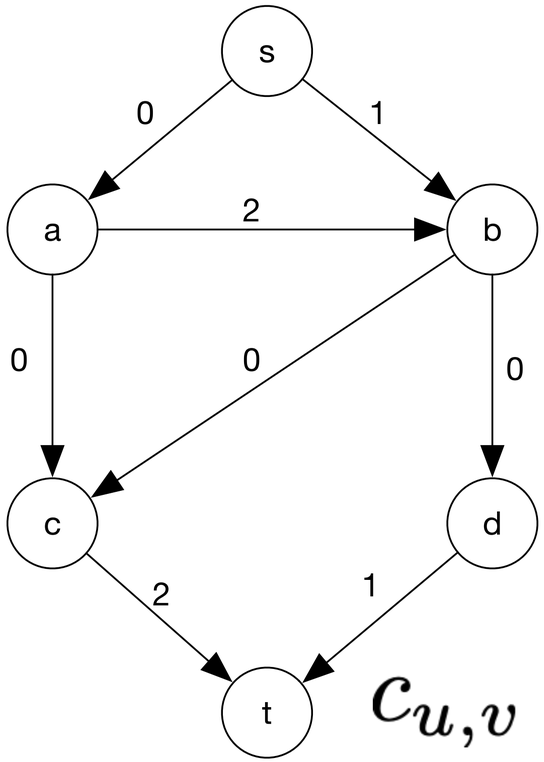
$c_{u,v}$

FLOW



$f_{u,v}$

RESIDUAL



$c_{u,v} - f_{u,v}$

Network flow

$f_{u,v}$ is the flow from u to v

$$f_{u,v} \leq c_{u,v}$$

$$\sum_{u:(u,v)\in E} f_{u,v} = \sum_{w:(v,w)\in E} f_{v,w}$$

$$\sum_{u:(s,u)\in E} = \sum_{w:(w,t)\in E}$$

Network flow

$f_{u,v}$ is the flow from u to v

$$f_{u,v} \leq c_{u,v}$$

$$\sum_{u:(u,v)\in E} f_{u,v} = \sum_{w:(v,w)\in E} f_{v,w}$$

$$\sum_{u:(s,u)\in E} = \sum_{w:(w,t)\in E}$$

Network flow

$f_{u,v}$ is the flow from u to v

$$f_{u,v} \leq c_{u,v}$$

$$\sum_{u:(u,v)\in E} f_{u,v} = \sum_{w:(v,w)\in E} f_{v,w}$$

$$\sum_{u:(s,u)\in E} = \sum_{w:(w,t)\in E}$$

Network flow

$f_{u,v}$ is the flow from u to v

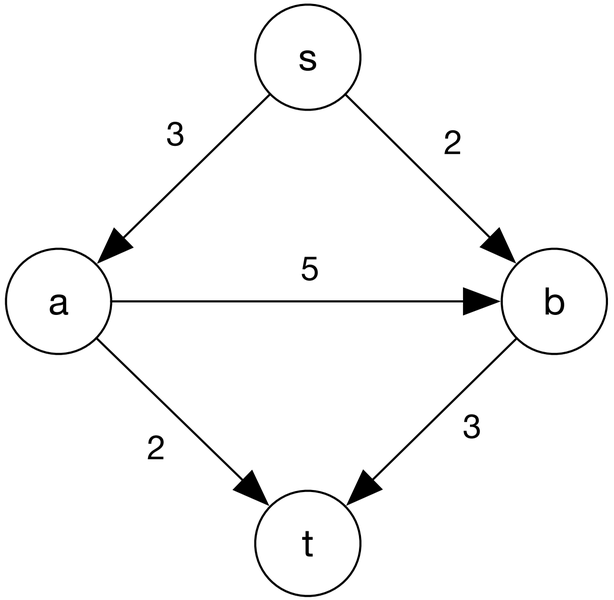
$$f_{u,v} \leq c_{u,v}$$

$$\sum_{u:(u,v)\in E} f_{u,v} = \sum_{w:(v,w)\in E} f_{v,w}$$

$$\sum_{u:(s,u)\in E} = \sum_{w:(w,t)\in E}$$

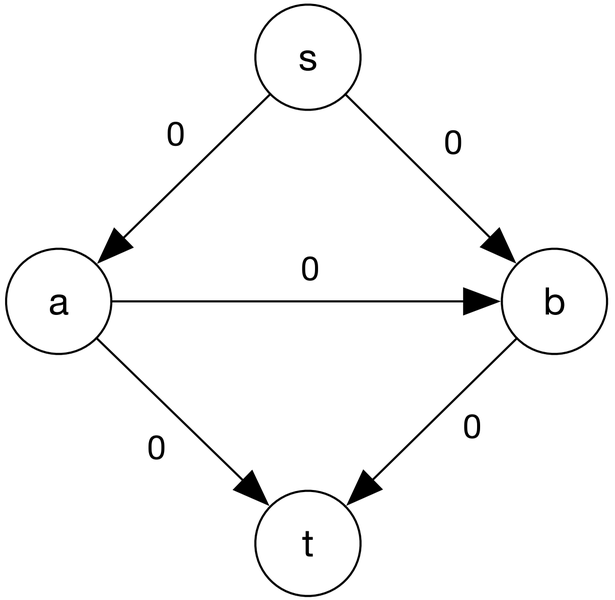
Network flow

CAPACITY



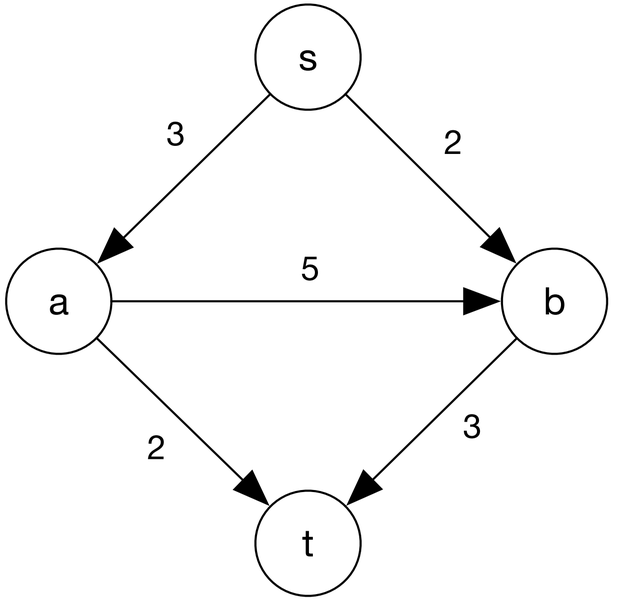
$c_{u,v}$

FLOW



$f_{u,v}$

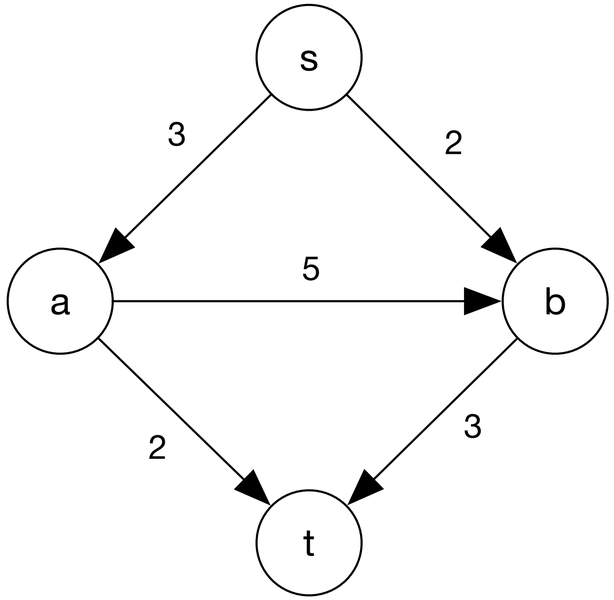
RESIDUAL



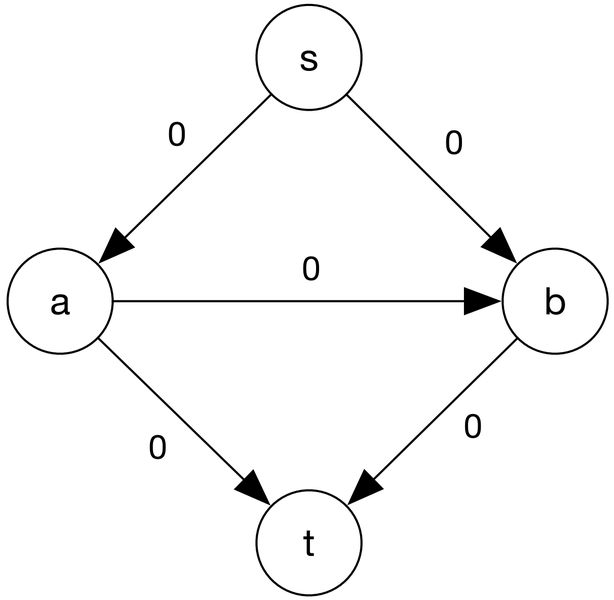
$c_{u,v} - f_{u,v}$

Network flow

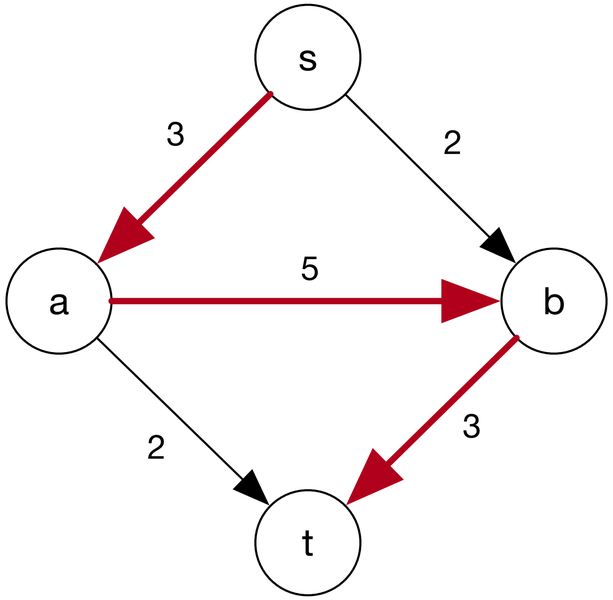
CAPACITY



FLOW

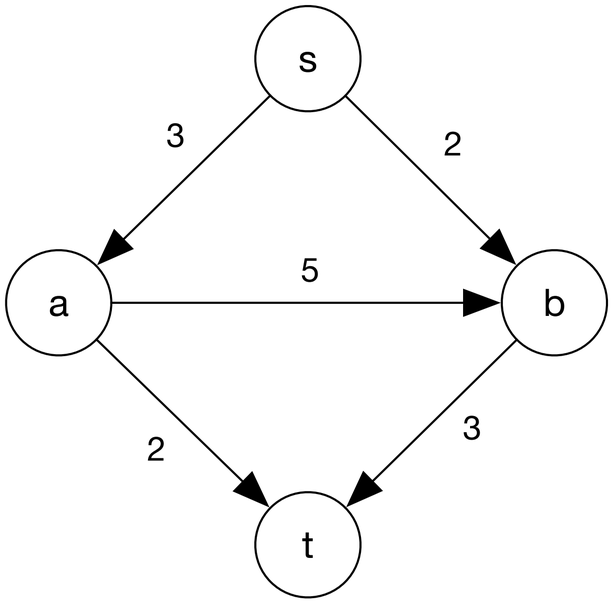


RESIDUAL

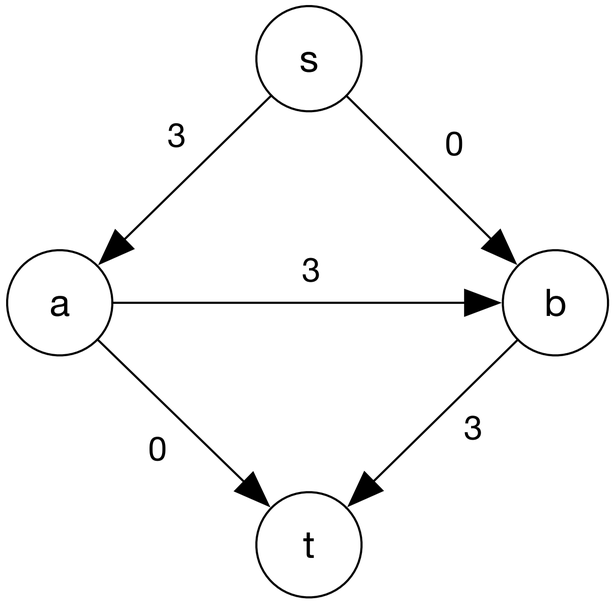


Network flow

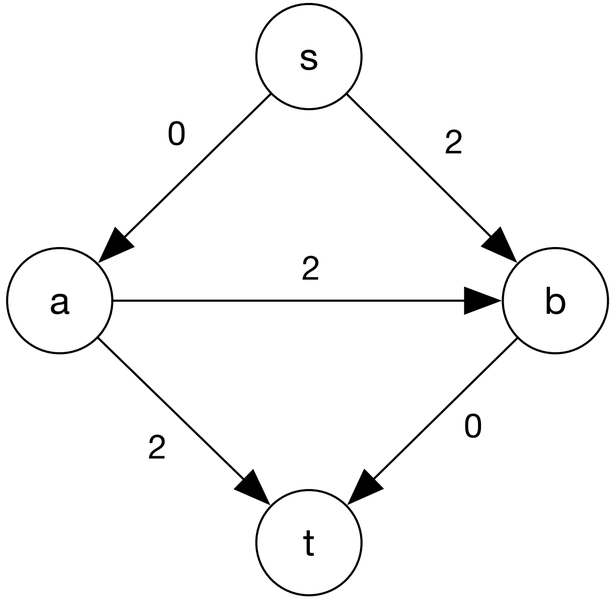
CAPACITY



FLOW



RESIDUAL

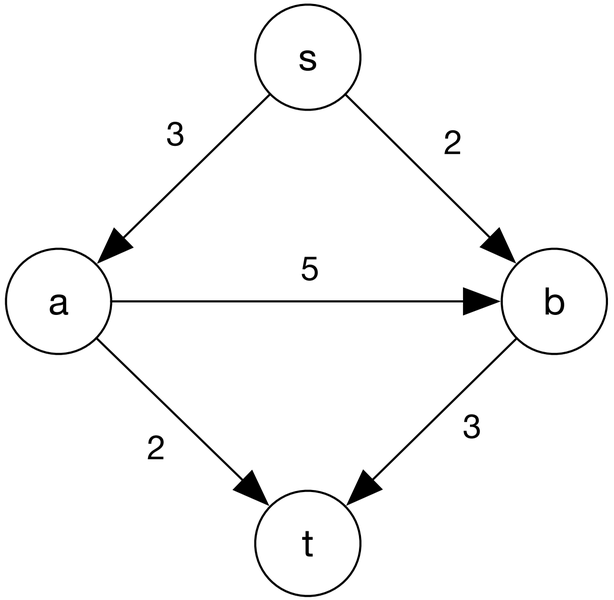


We have a problem!

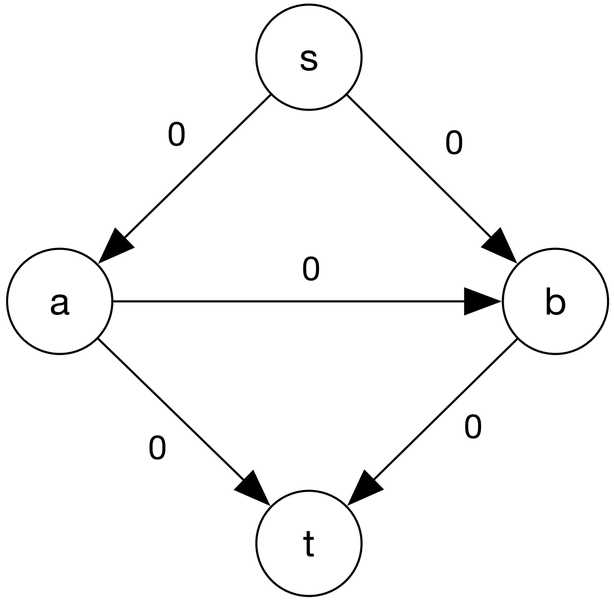
Network flow

Start over!

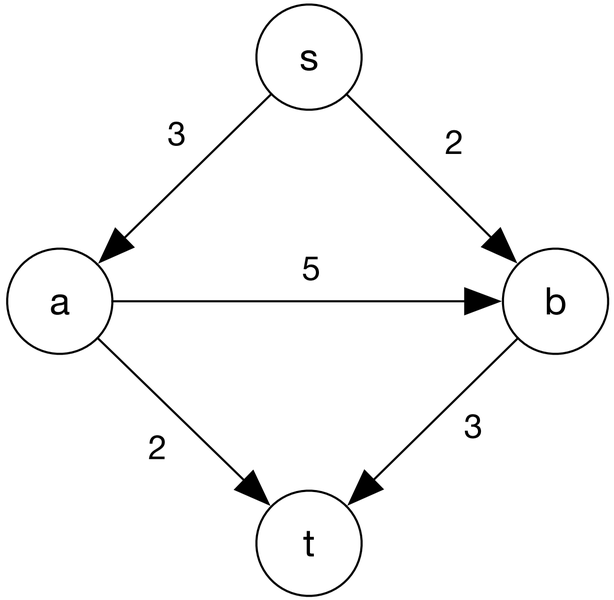
CAPACITY



FLOW

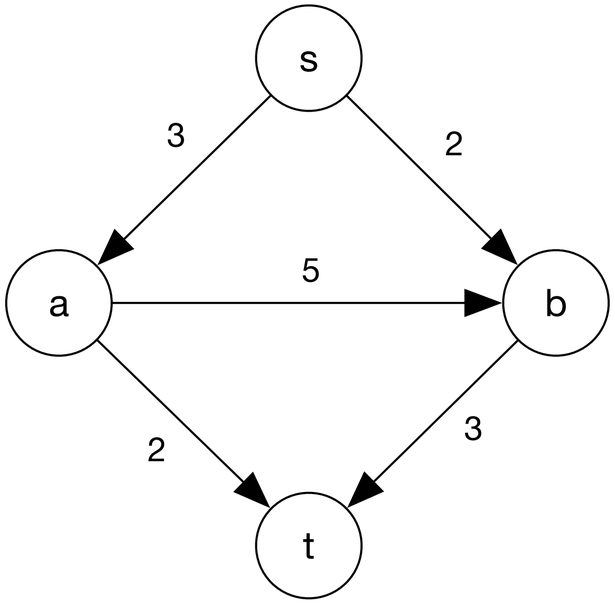


RESIDUAL

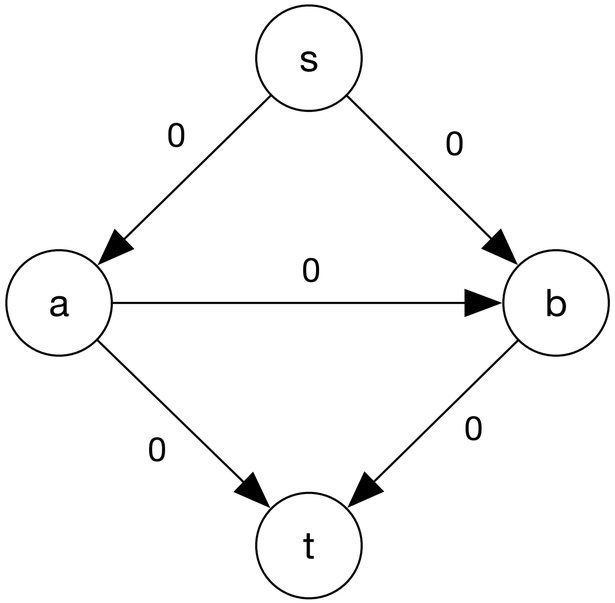


Network flow

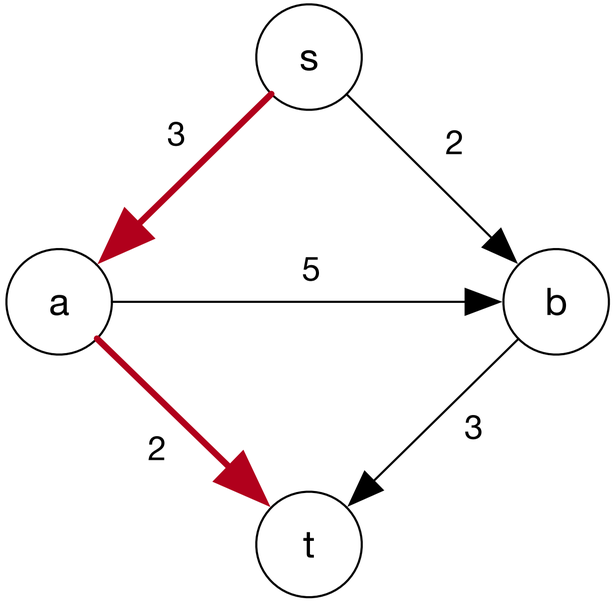
CAPACITY



FLOW

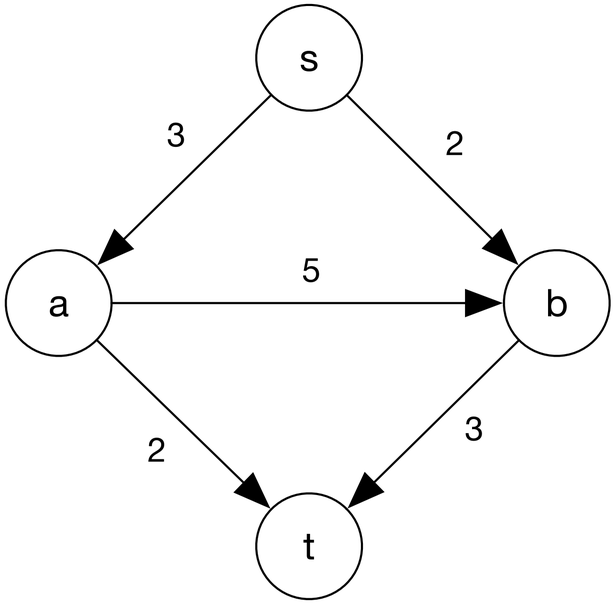


RESIDUAL

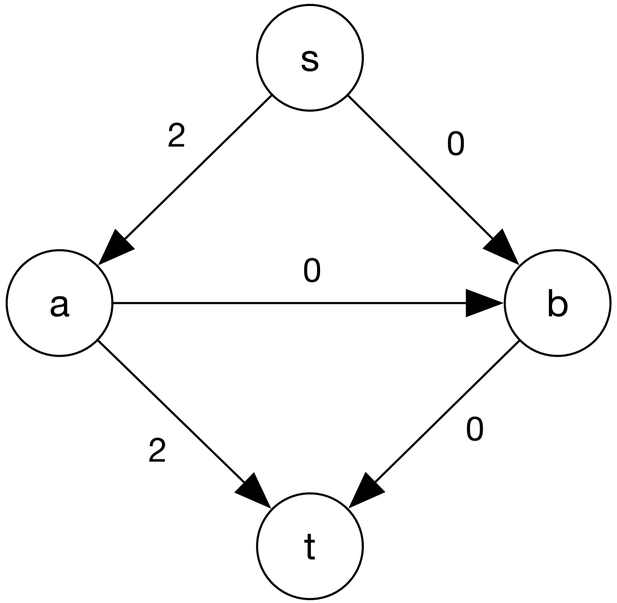


Network flow

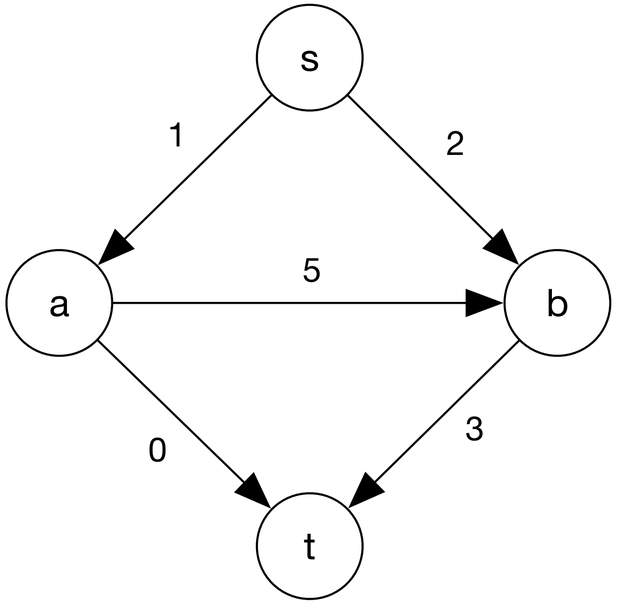
CAPACITY



FLOW

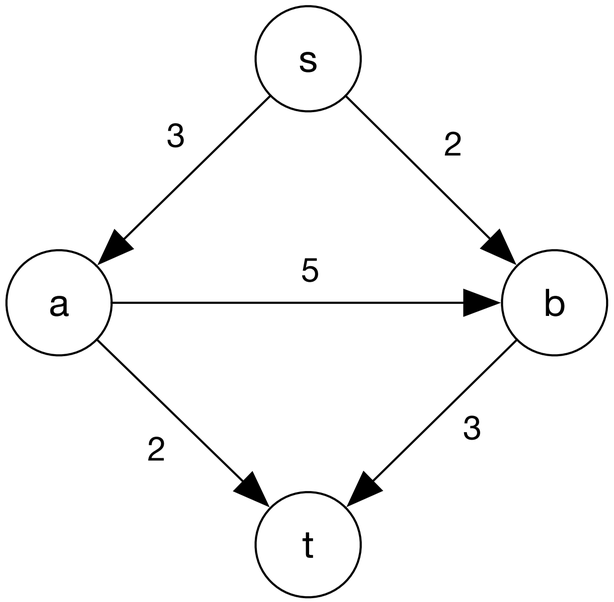


RESIDUAL

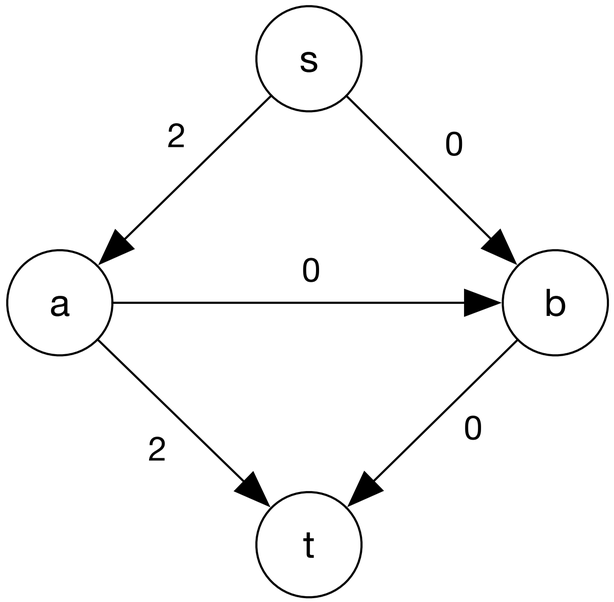


Network flow

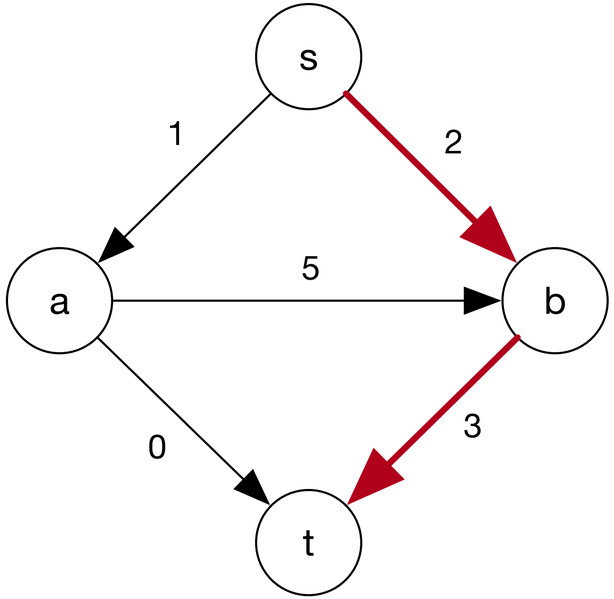
CAPACITY



FLOW

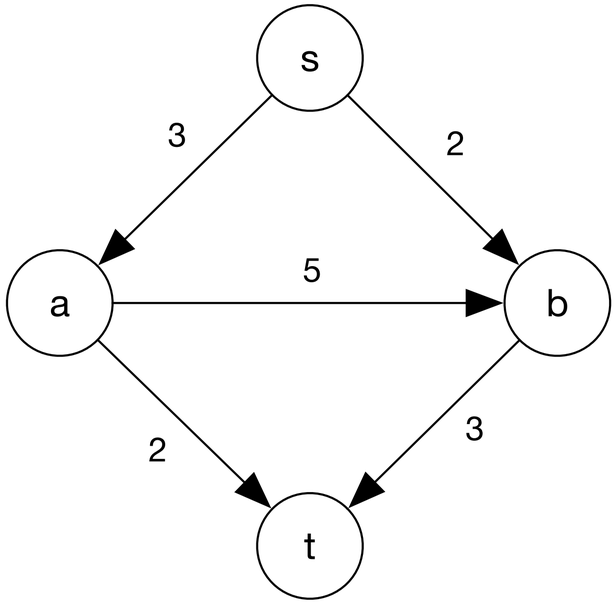


RESIDUAL

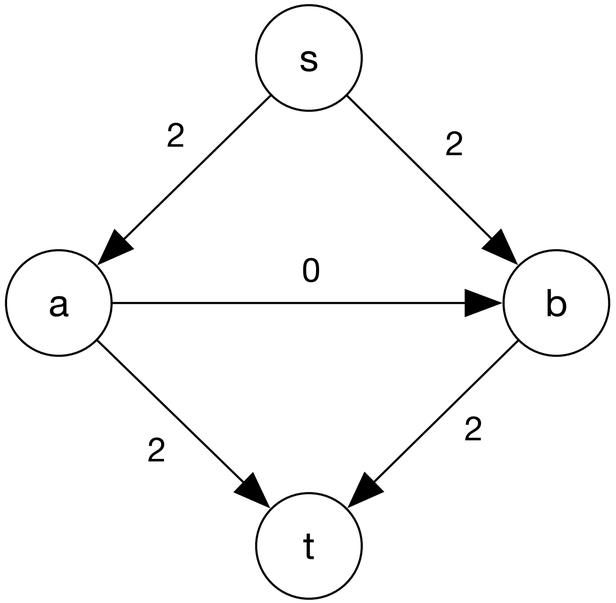


Network flow

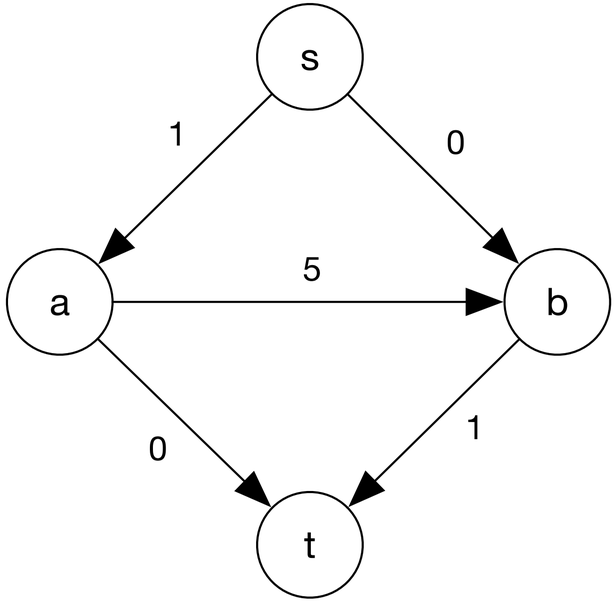
CAPACITY



FLOW

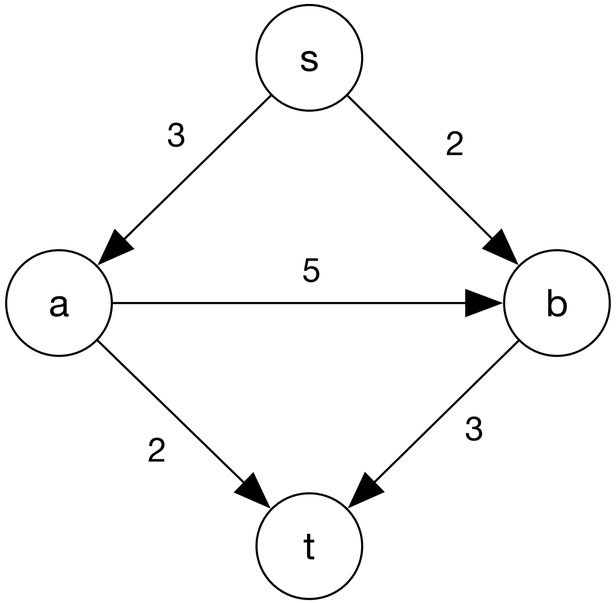


RESIDUAL

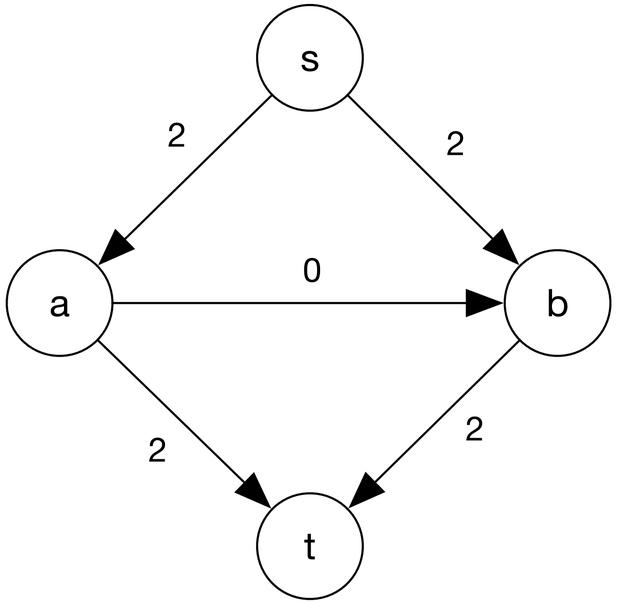


Network flow

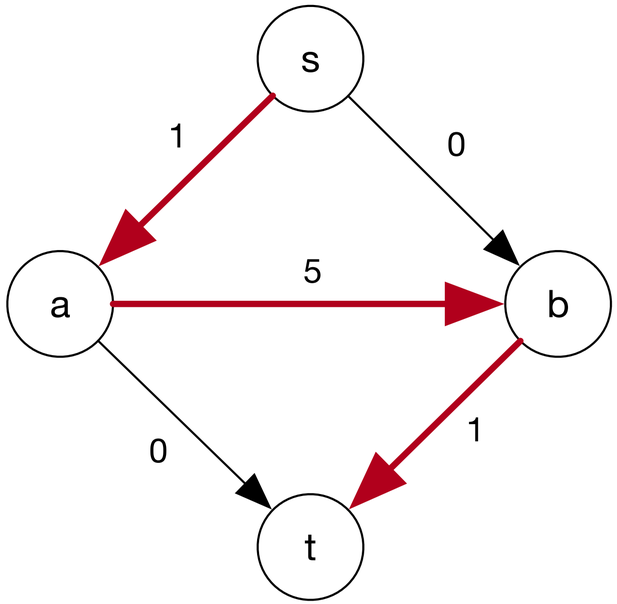
CAPACITY



FLOW

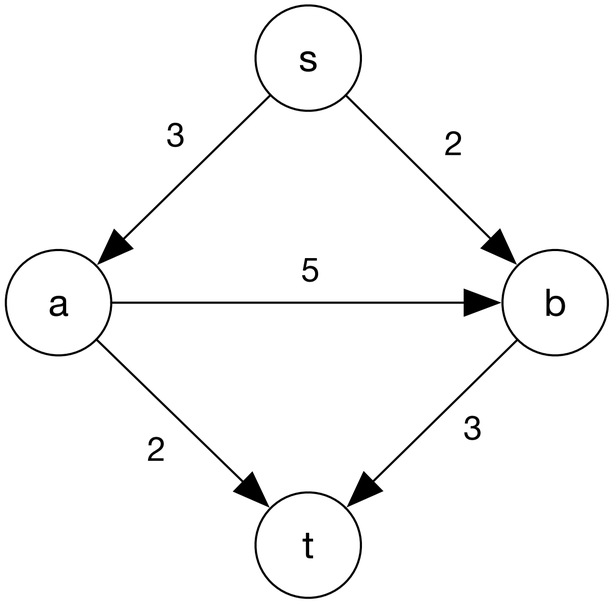


RESIDUAL

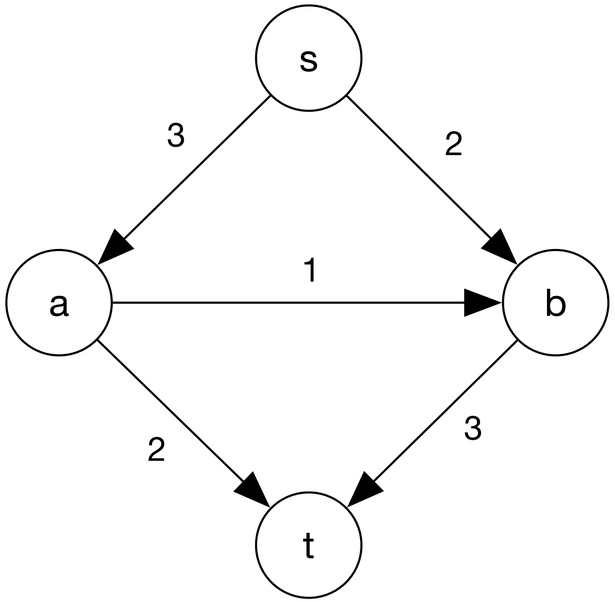


Network flow

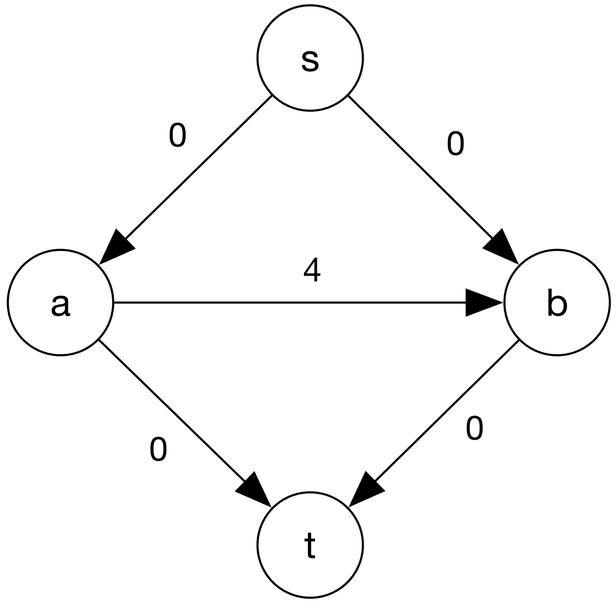
CAPACITY



FLOW



RESIDUAL



Now we have found the maximum flow

Network flow

Problem: By a "bad" sequence of discoveries of augmenting paths, we might discover a "blocking flow" which prevents us from discovering other augmenting paths.

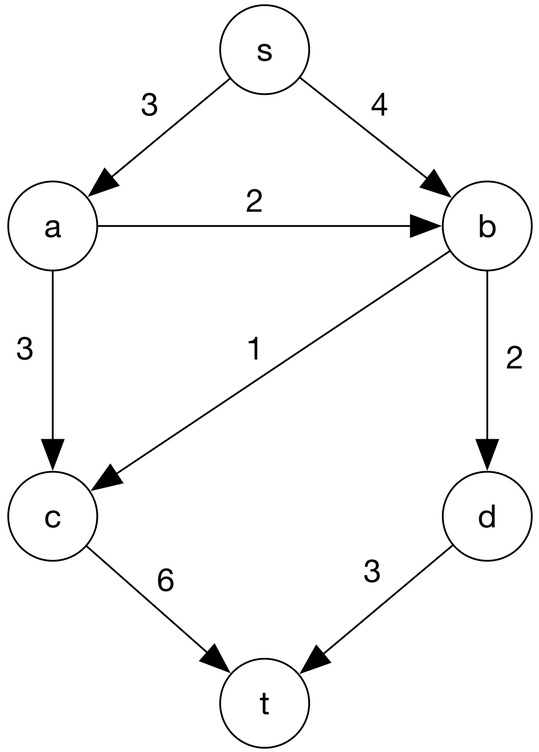
Network flow

Problem: By a "bad" sequence of discoveries of augmenting paths, we might discover a "blocking flow" which prevents us from discovering other augmenting paths.

Solution: Encode a way of revising flows in the residual graph: add "backward" edges representing flow, in addition to "forward" edges representing unused capacity.

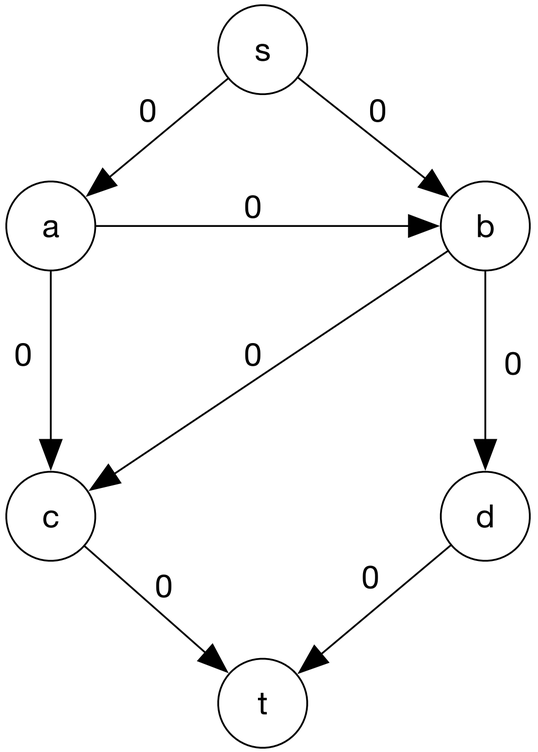
Network flow

CAPACITY



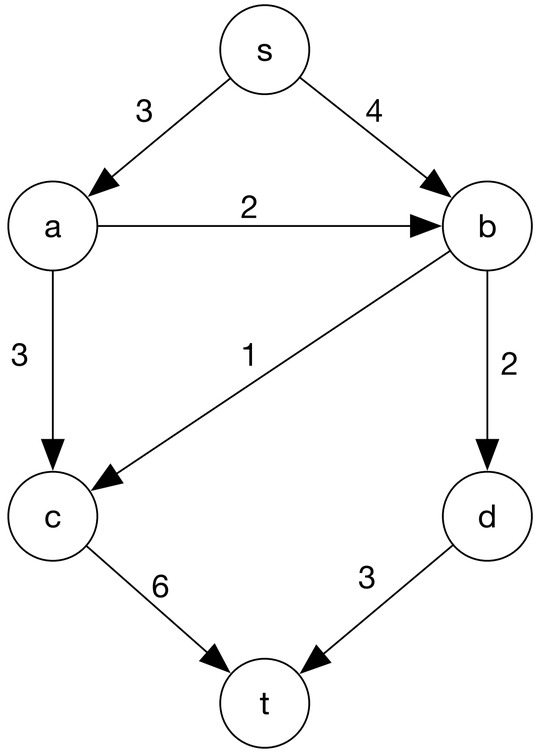
$c_{u,v}$

FLOW



$f_{u,v}$

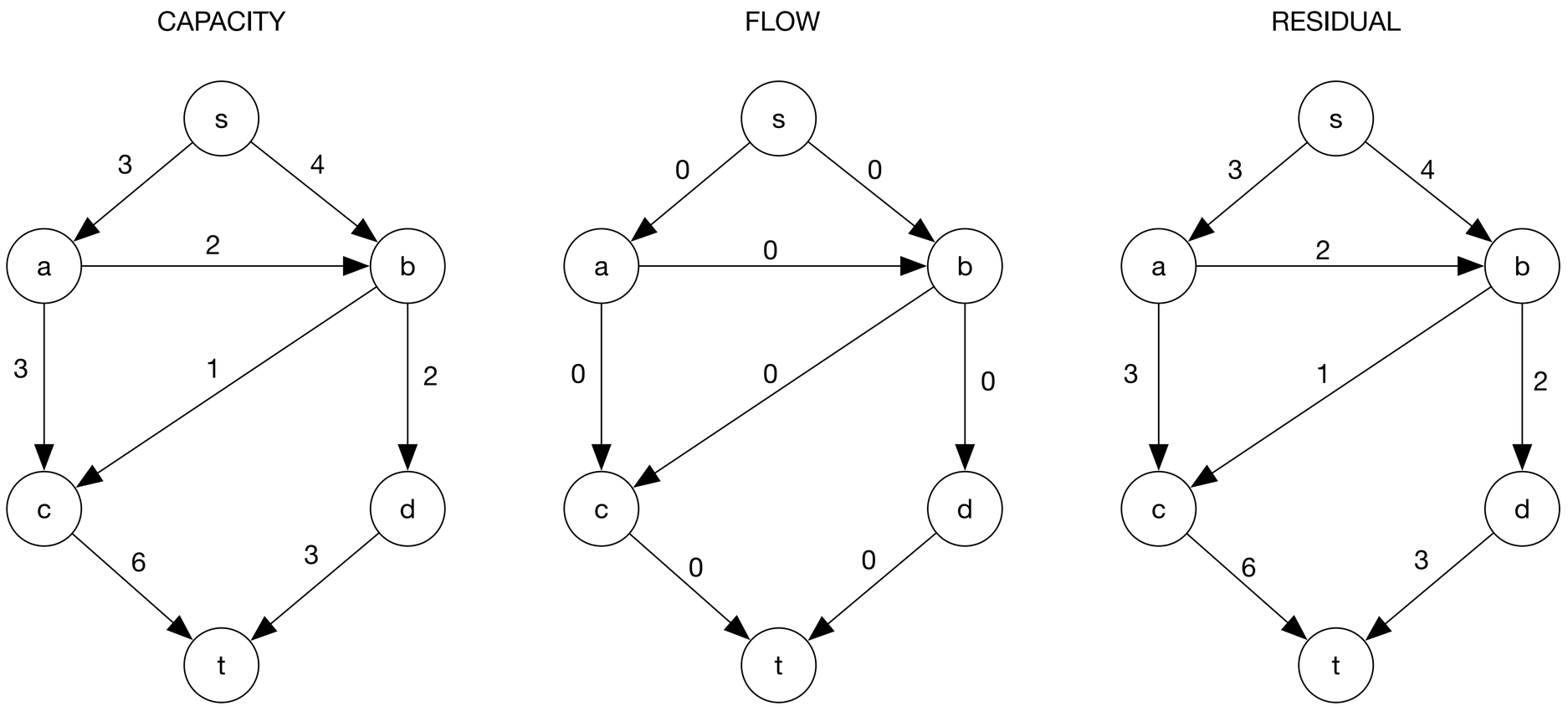
RESIDUAL



$c_{u,v} - f_{u,v}$

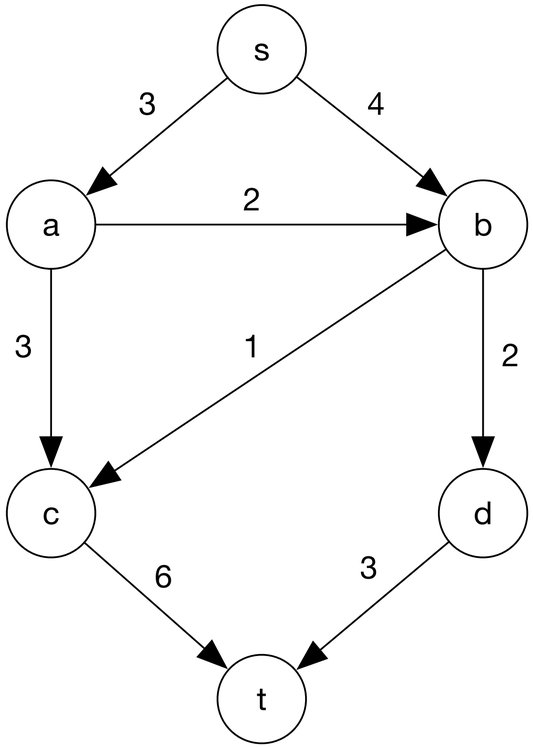
Network flow

Network flow

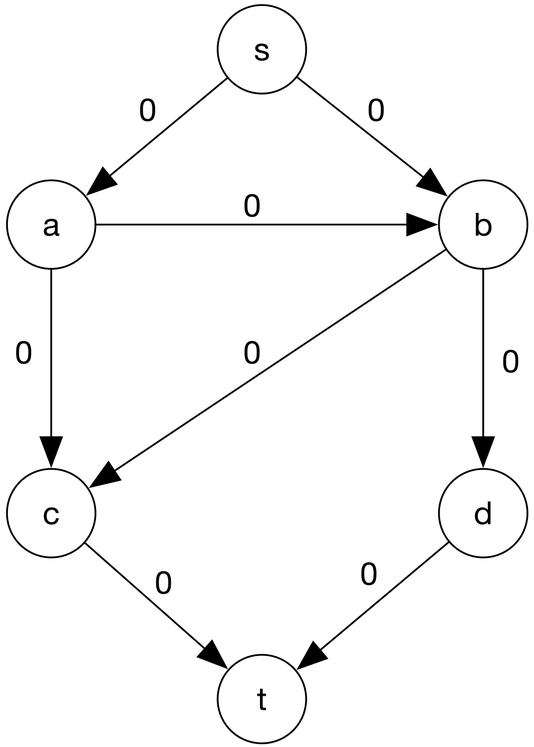


Network flow

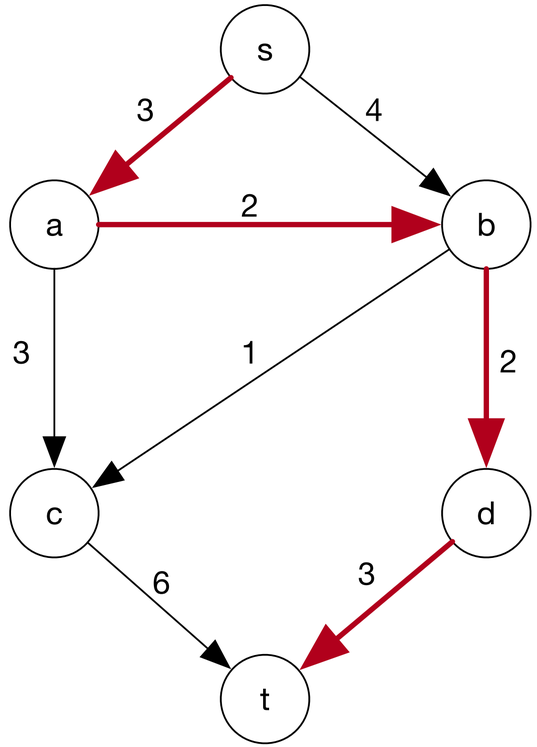
CAPACITY



FLOW

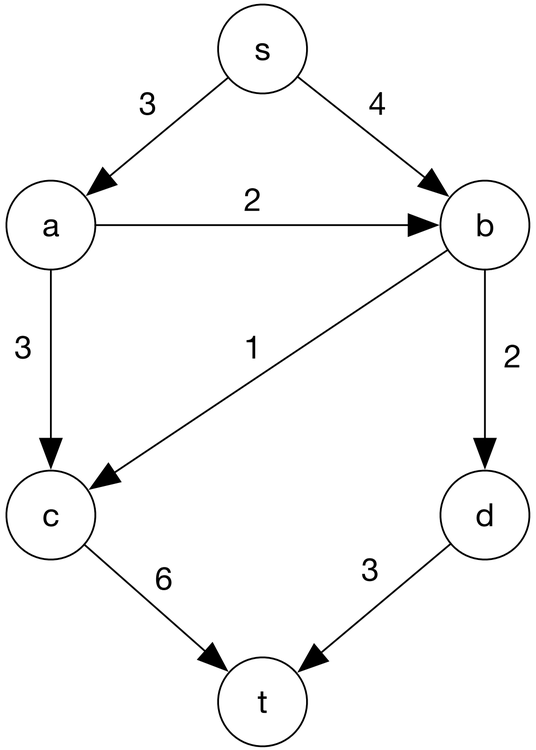


RESIDUAL

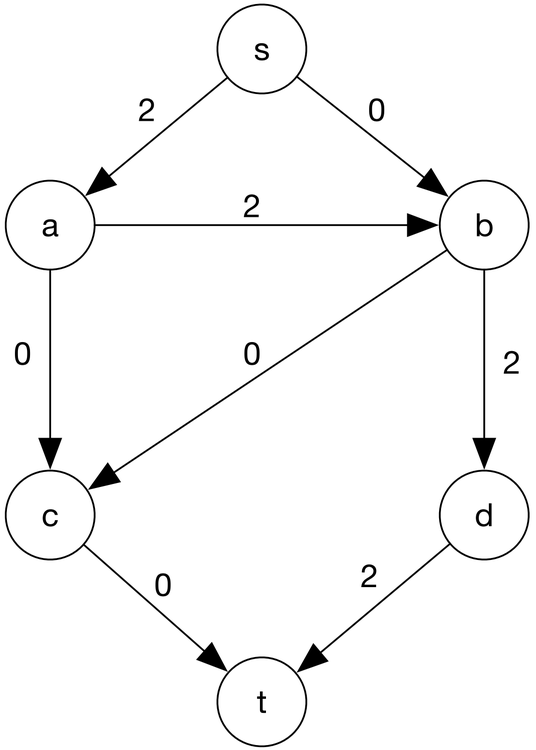


Network flow

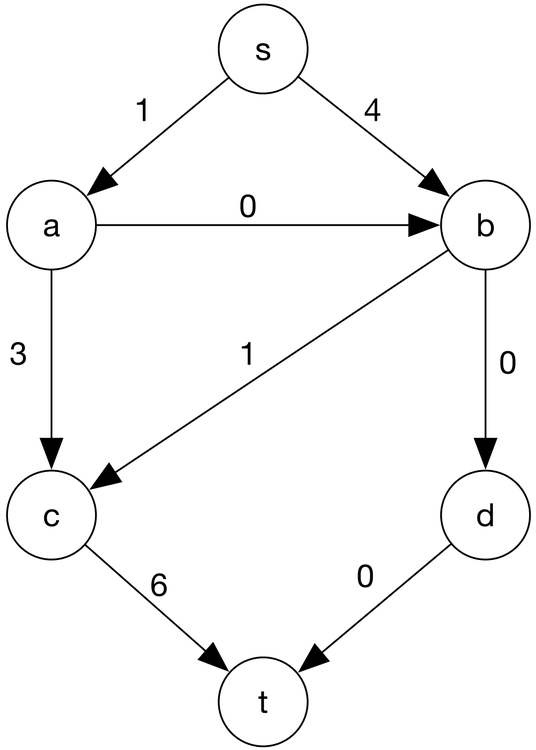
CAPACITY



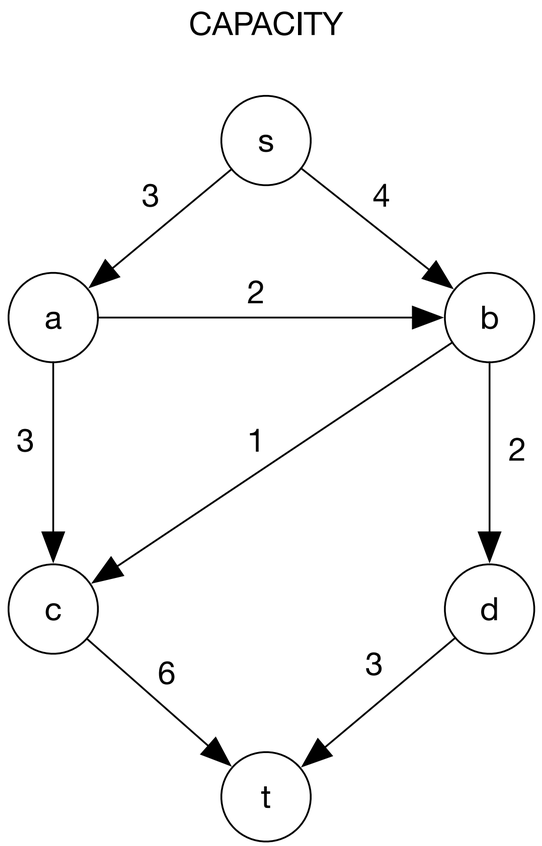
FLOW

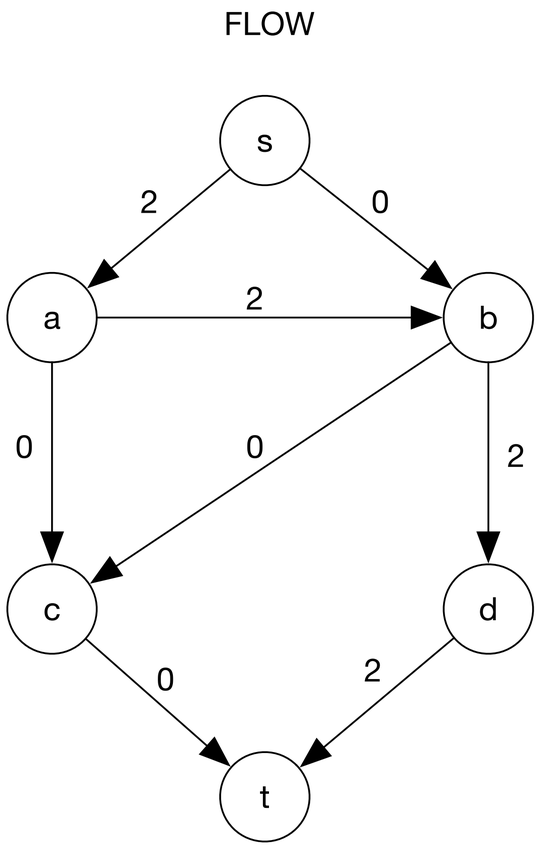


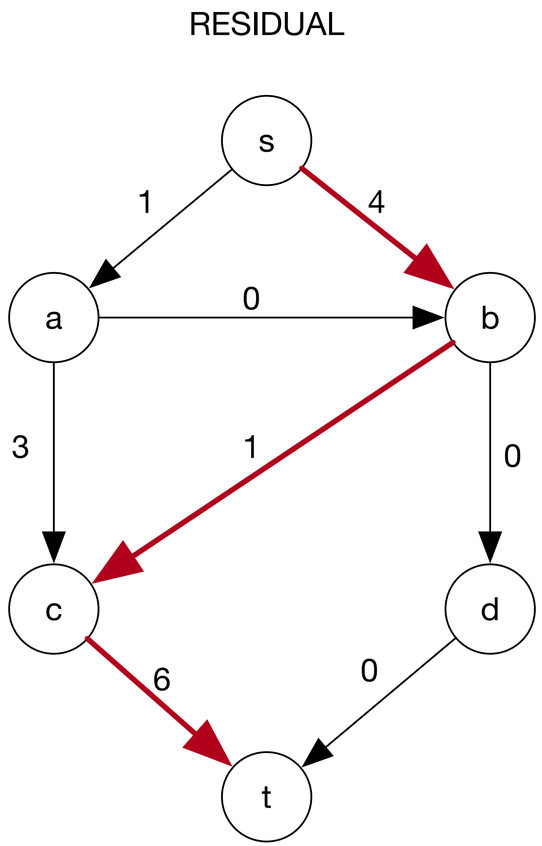
RESIDUAL



Network flow

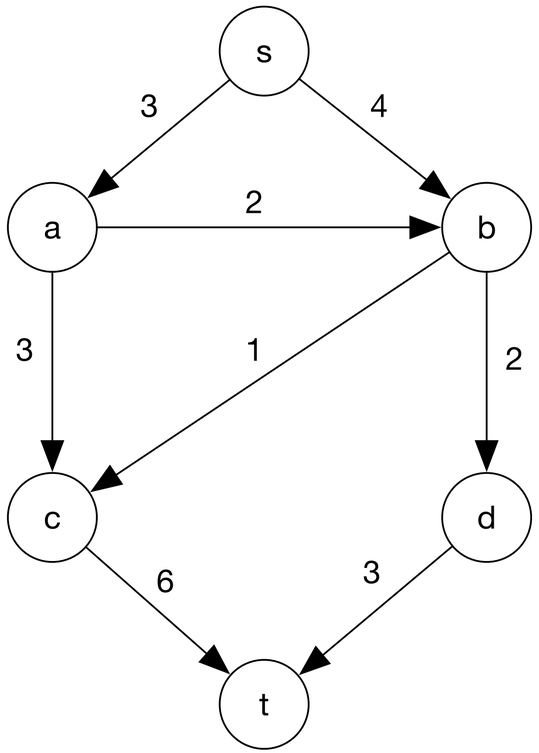




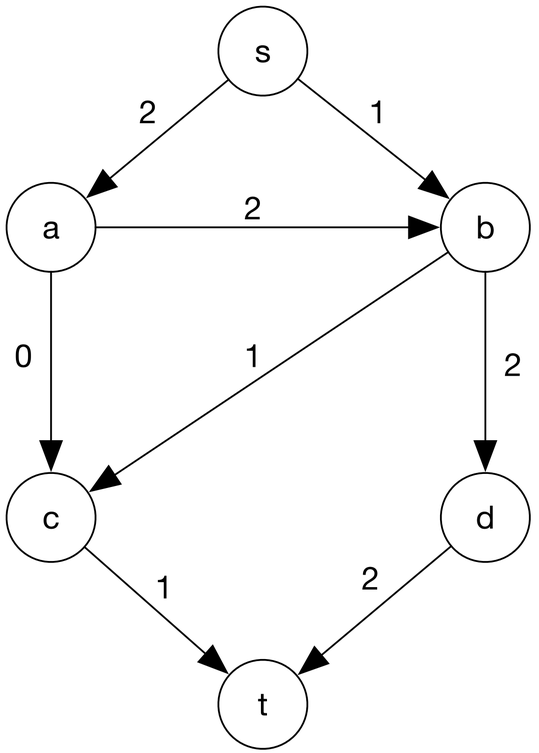


Network flow

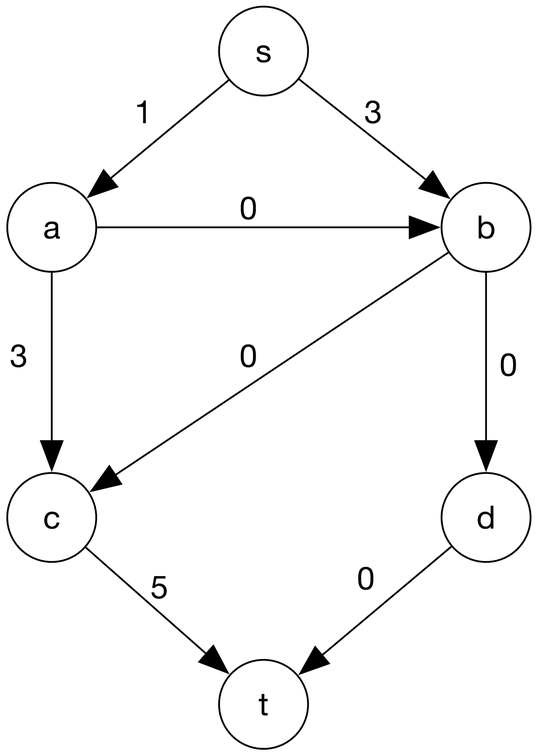
CAPACITY



FLOW

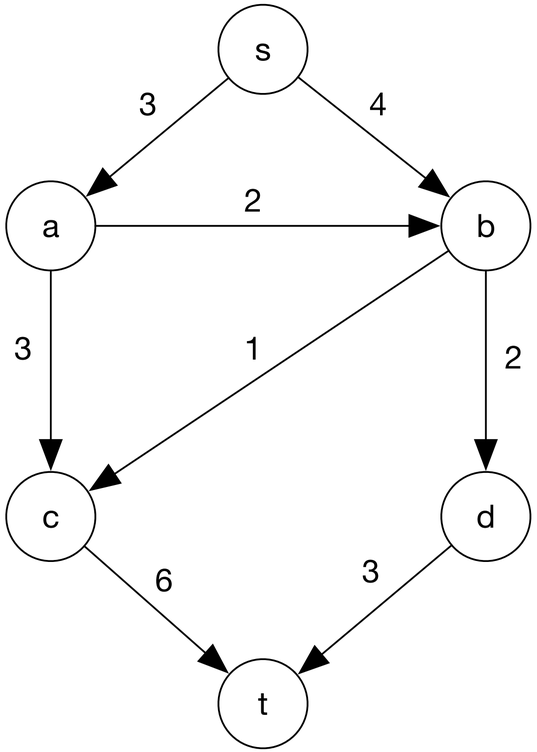


RESIDUAL

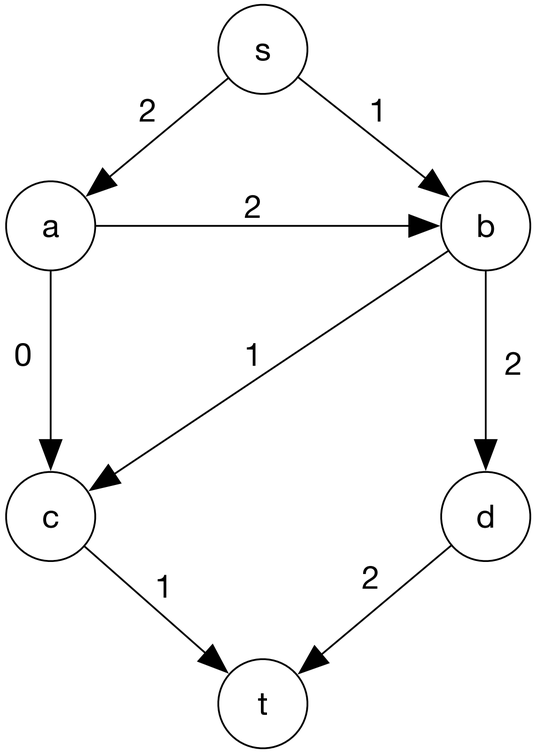


Network flow

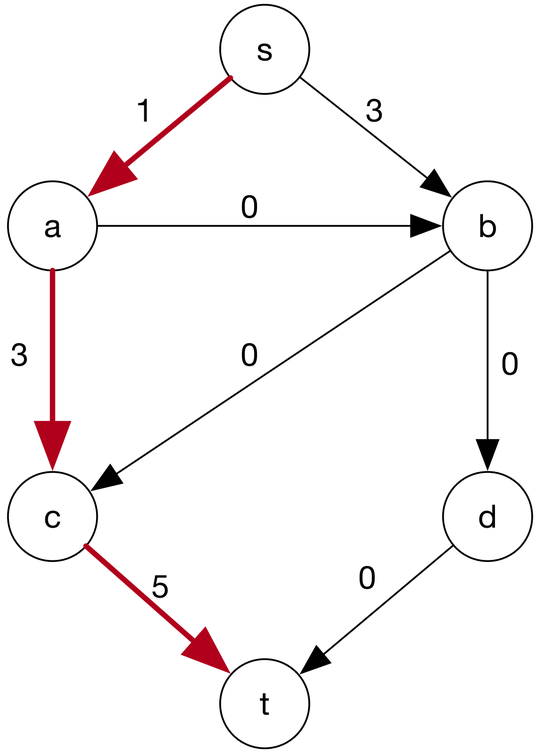
CAPACITY



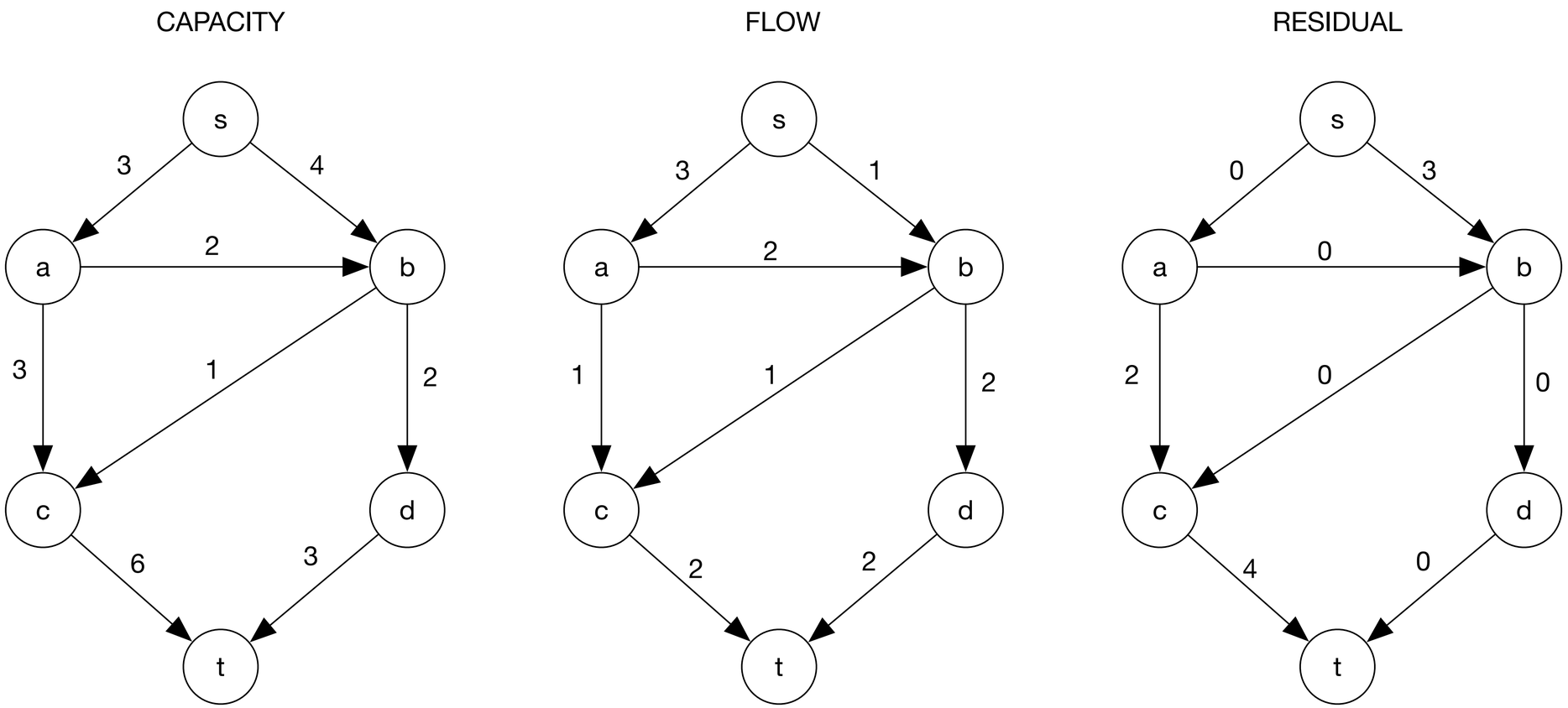
FLOW



RESIDUAL



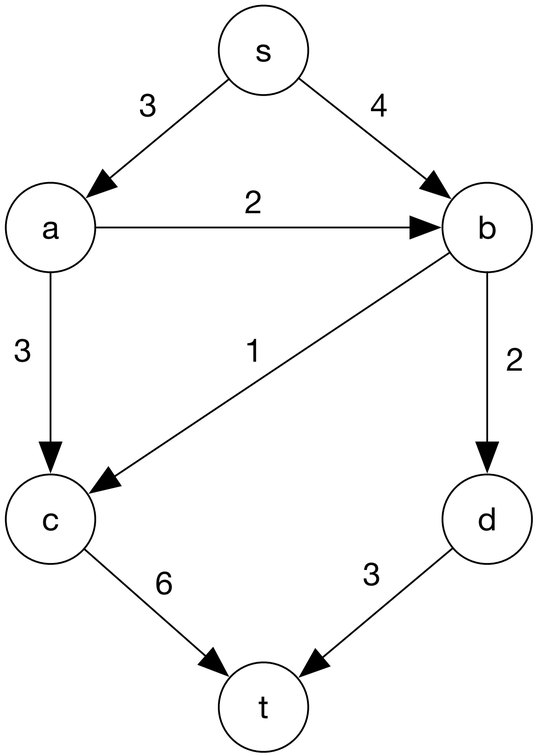
Network flow



Network flow

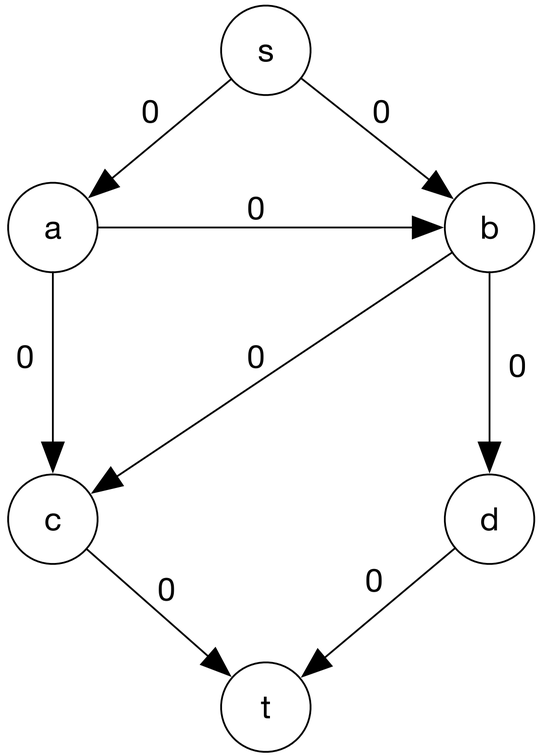
Network flow

CAPACITY



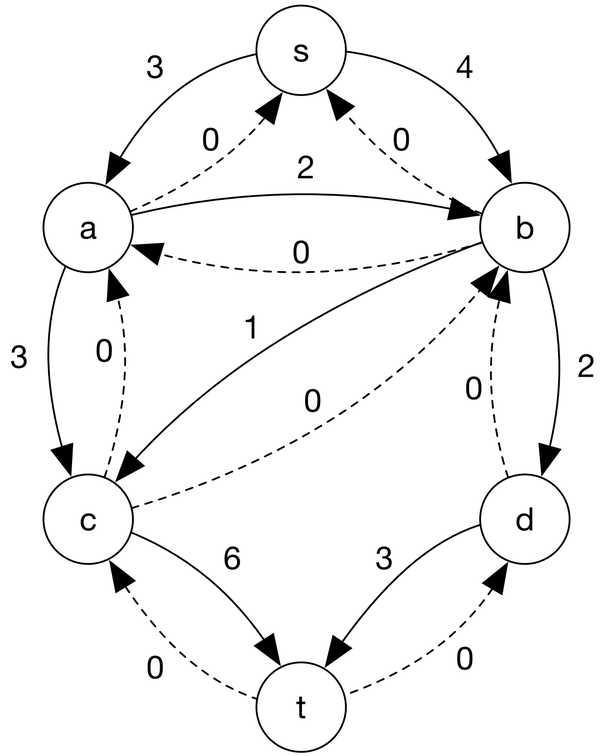
$c_{u,v}$

FLOW



$f_{u,v}$

RESIDUAL

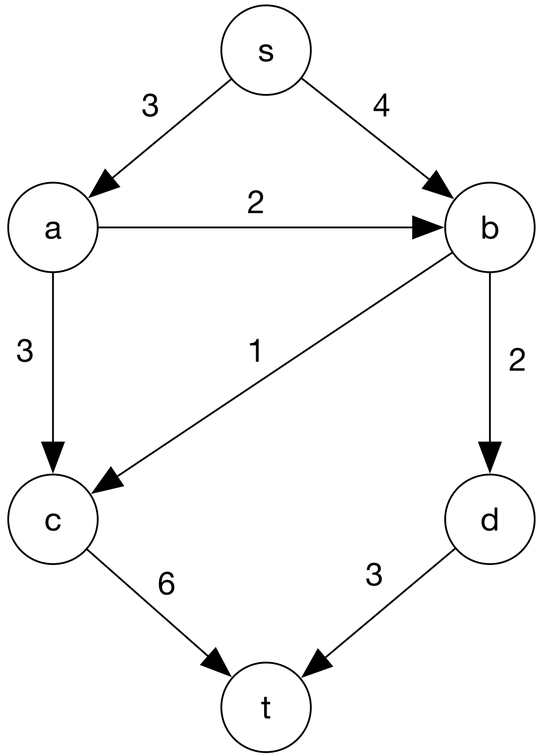


$c_{u,v} - f_{u,v}$

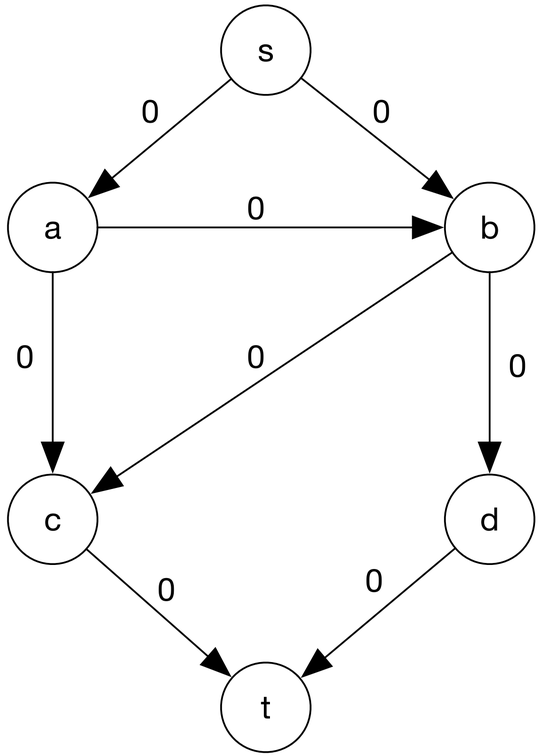
$f_{u,v}$

Network flow

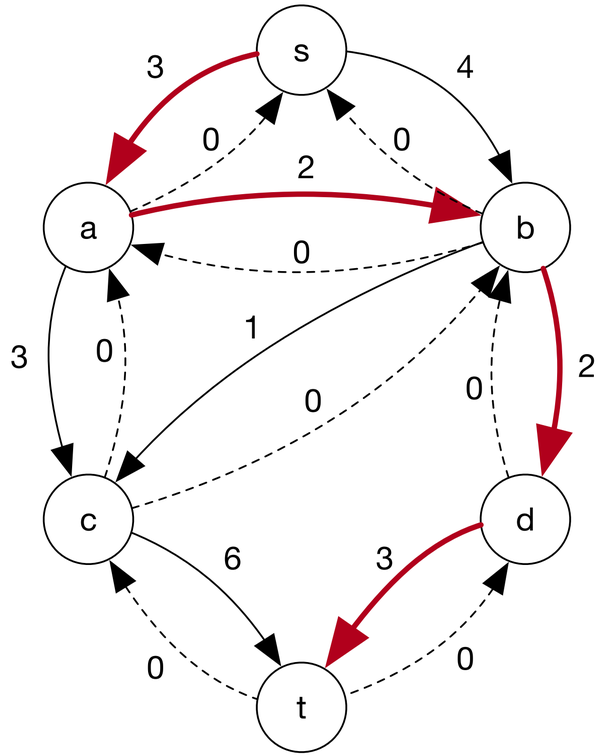
CAPACITY



FLOW

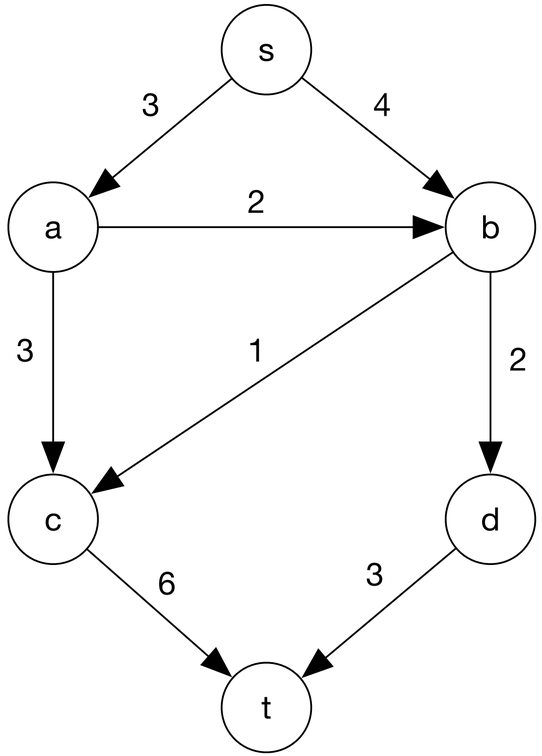


RESIDUAL

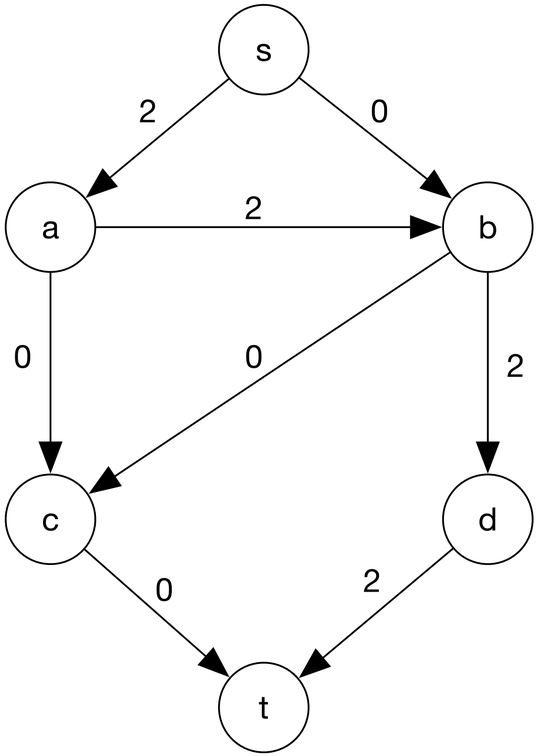


Network flow

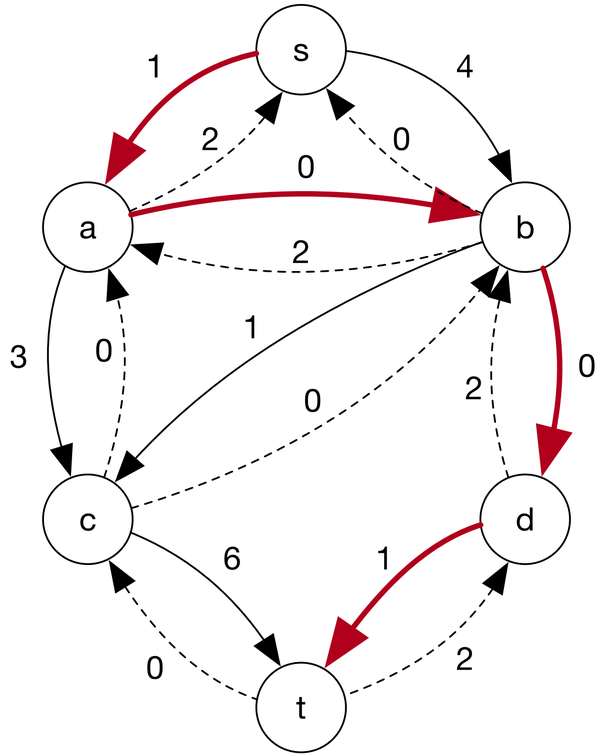
CAPACITY



FLOW

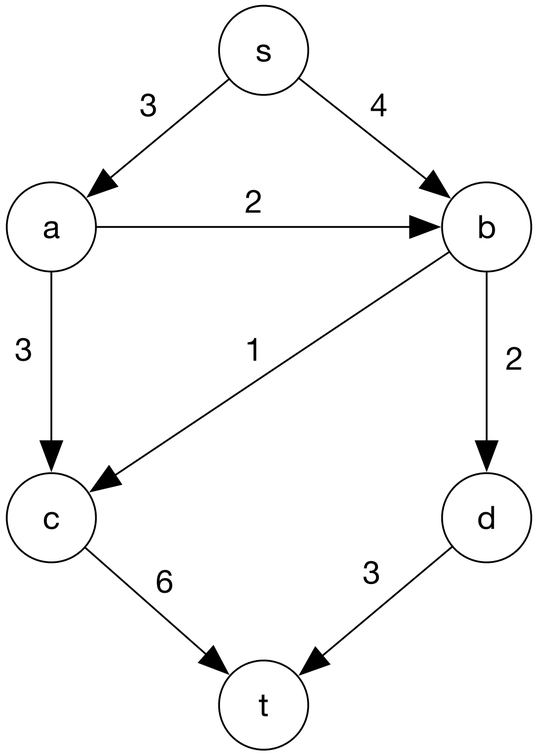


RESIDUAL

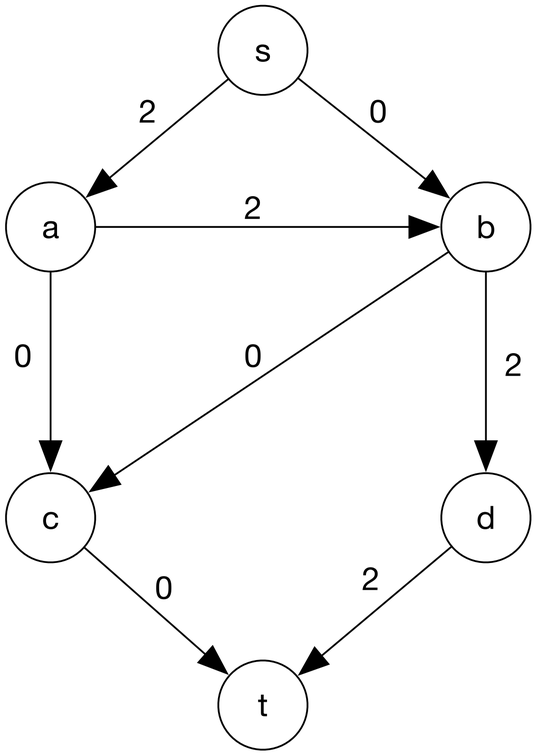


Network flow

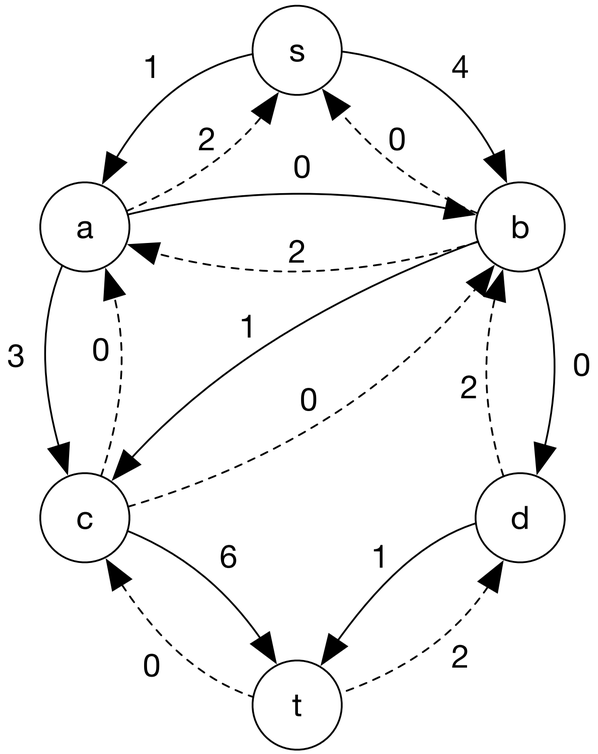
CAPACITY



FLOW

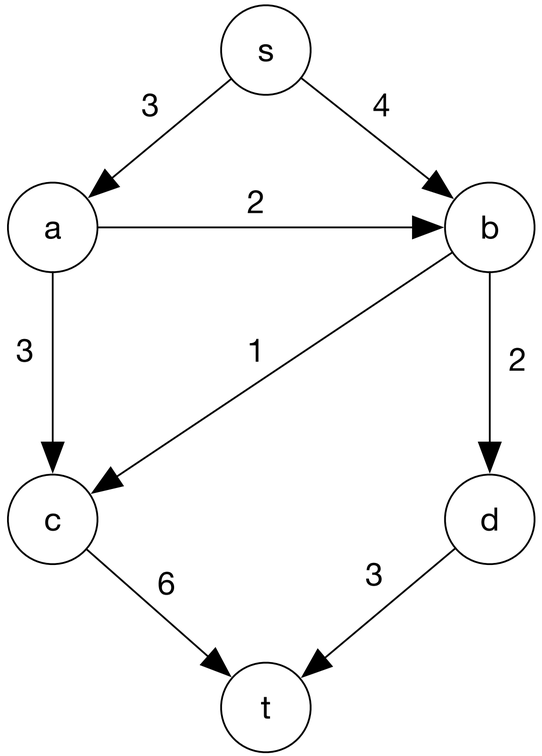


RESIDUAL

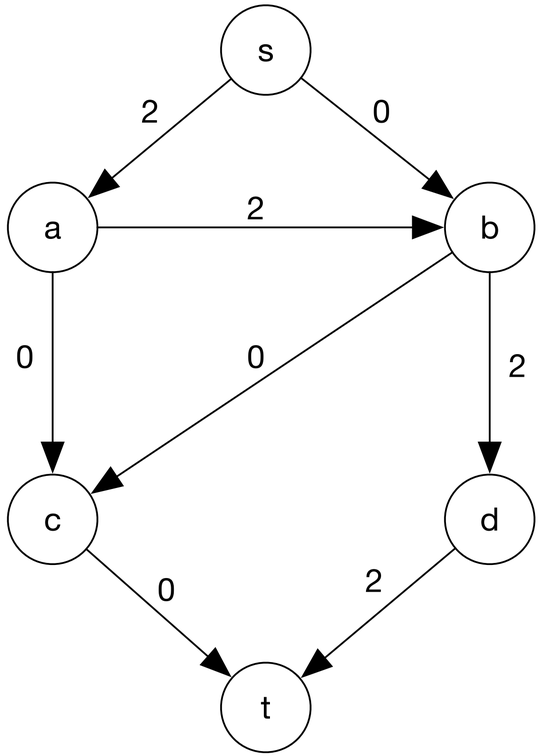


Network flow

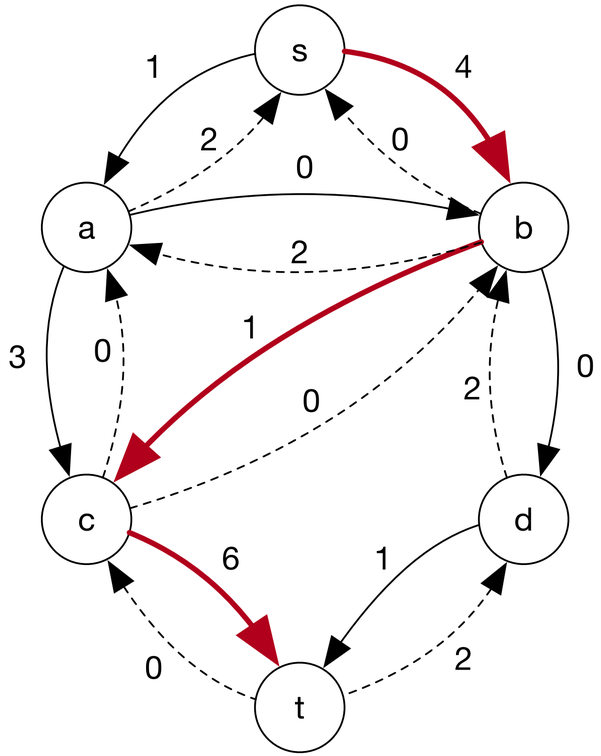
CAPACITY



FLOW

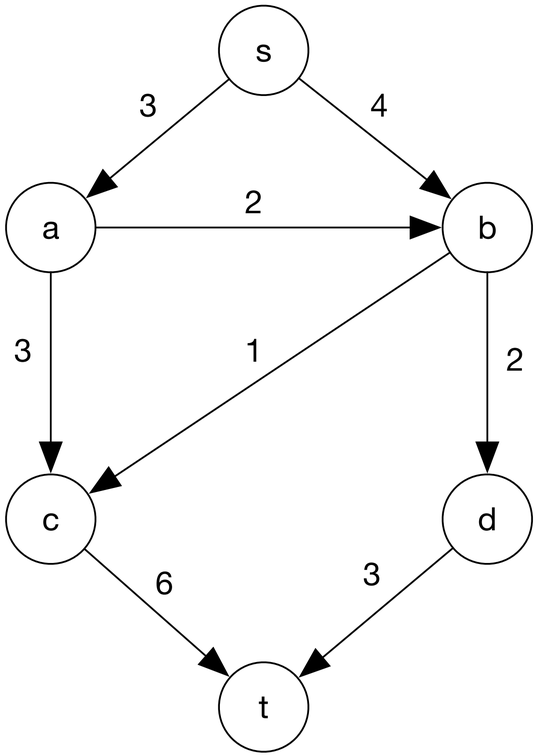


RESIDUAL

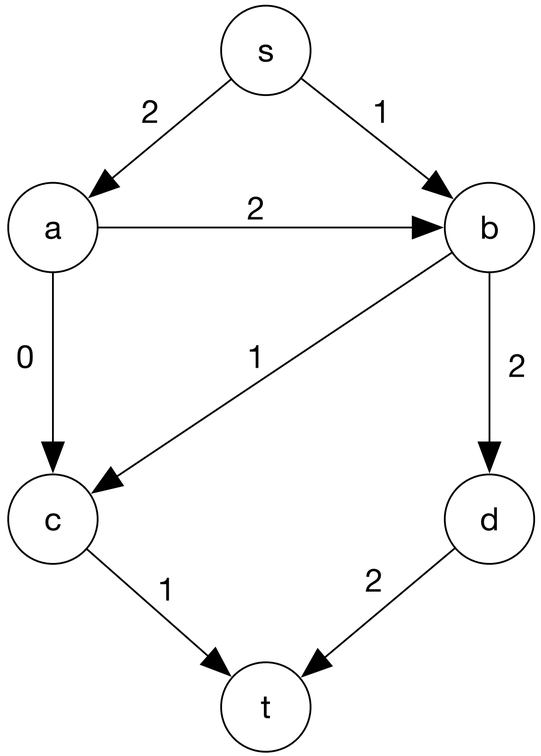


Network flow

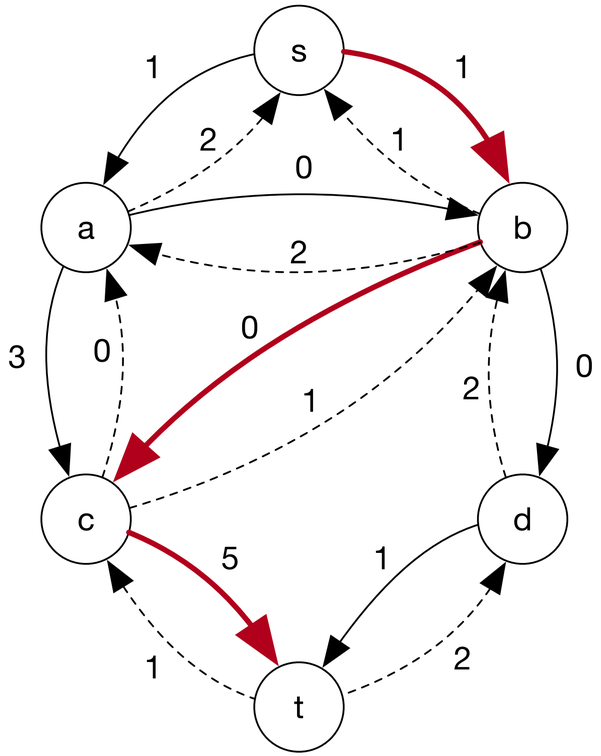
CAPACITY



FLOW

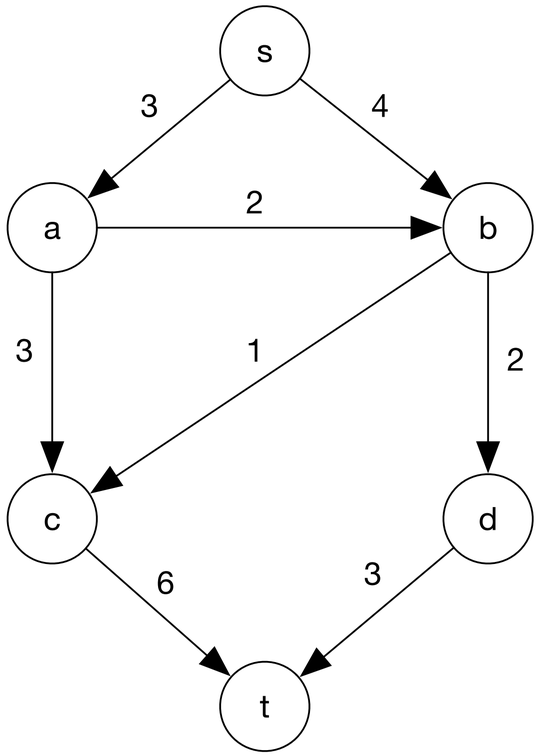


RESIDUAL

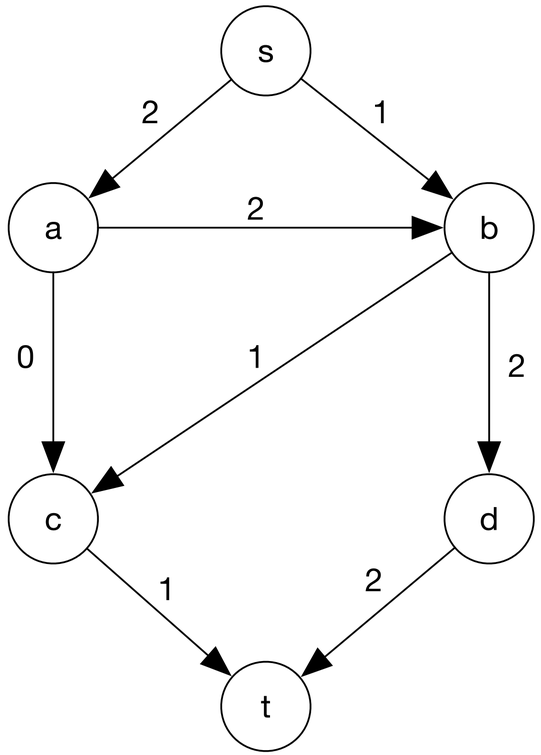


Network flow

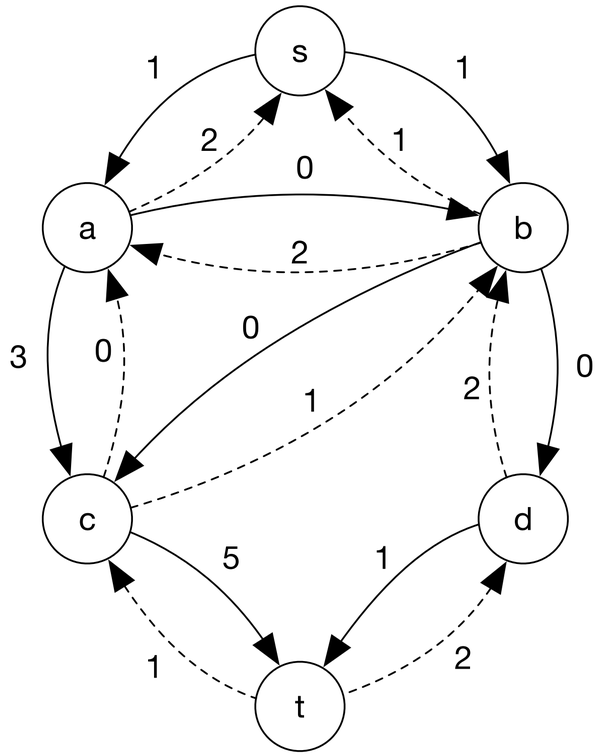
CAPACITY



FLOW

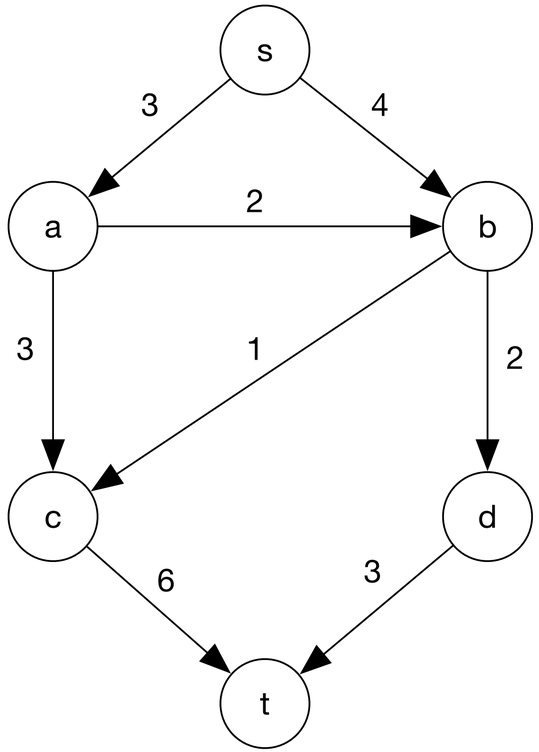


RESIDUAL

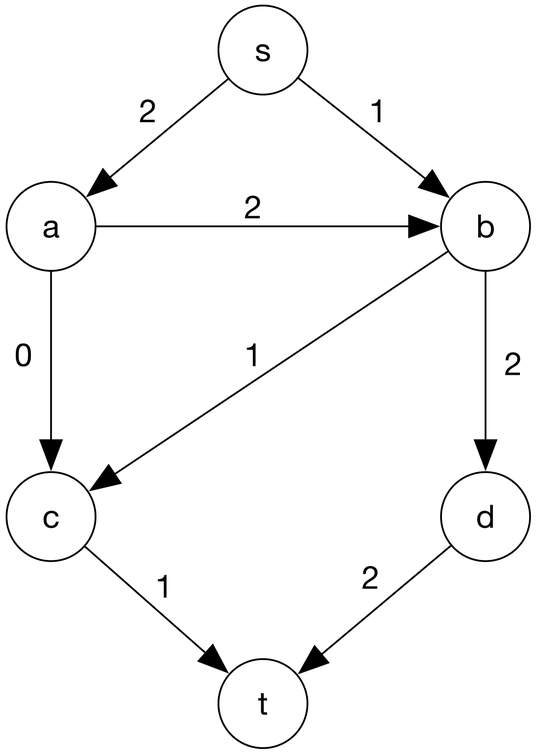


Network flow

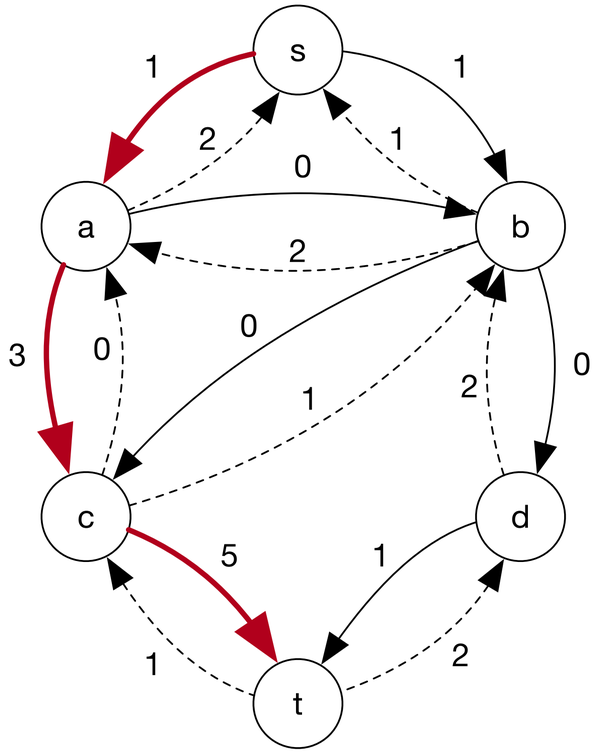
CAPACITY



FLOW

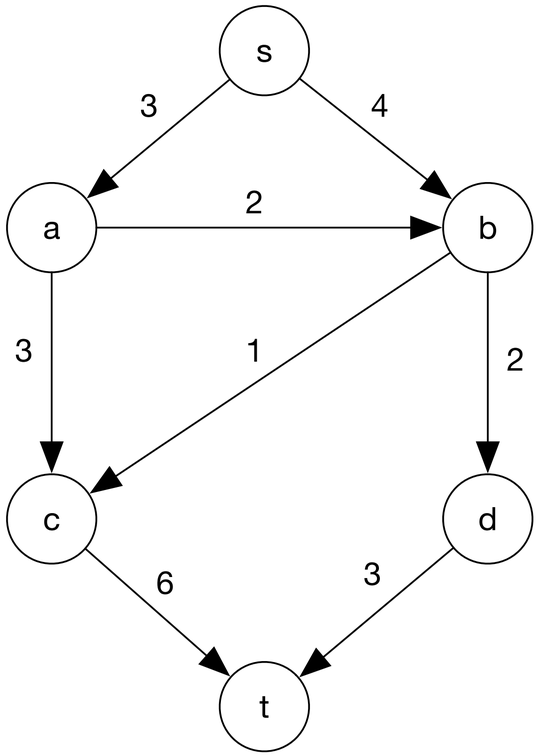


RESIDUAL

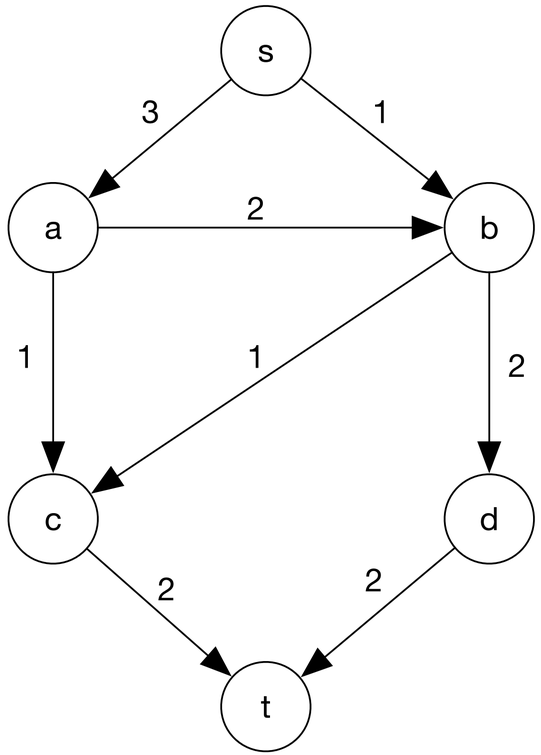


Network flow

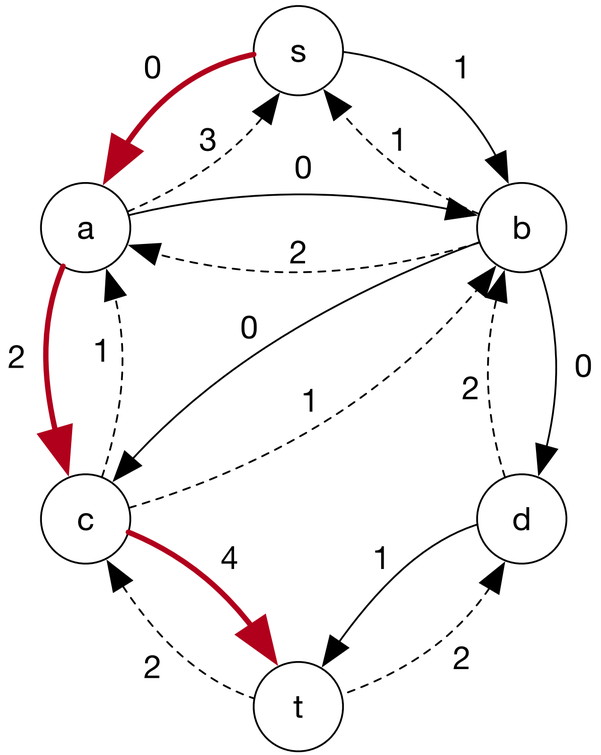
CAPACITY



FLOW

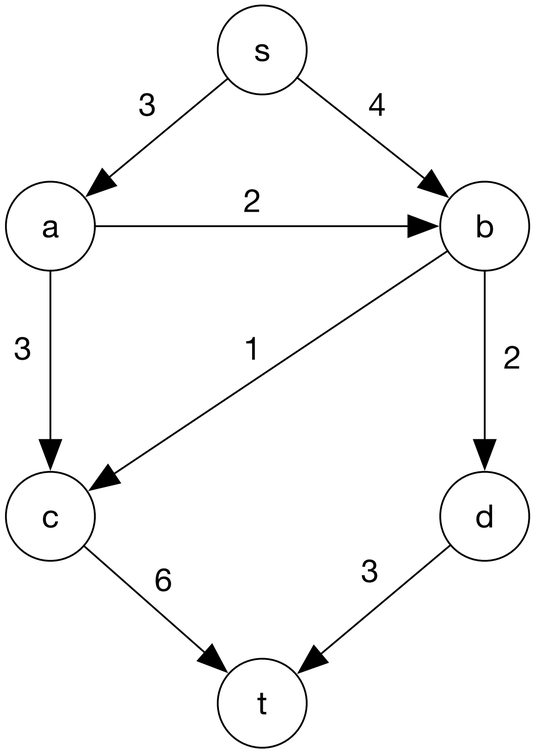


RESIDUAL

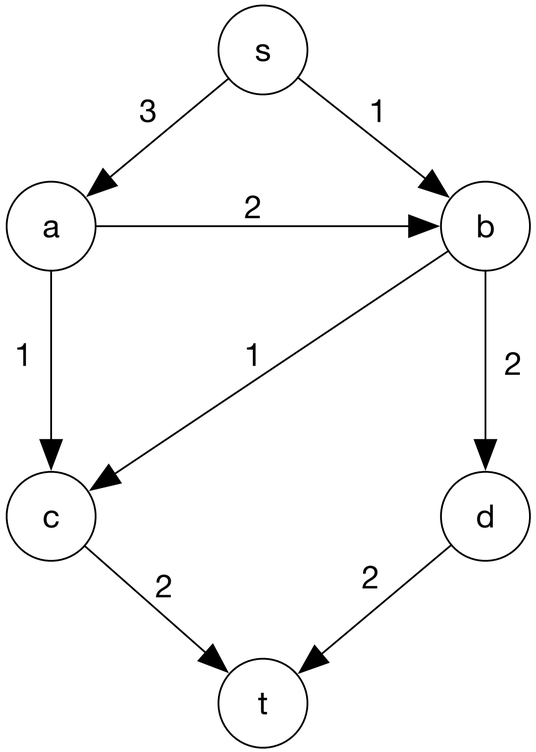


Network flow

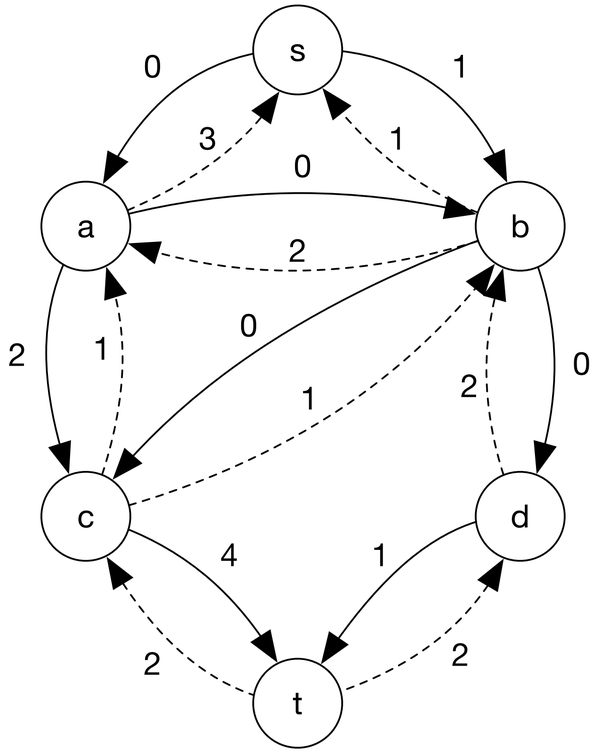
CAPACITY



FLOW

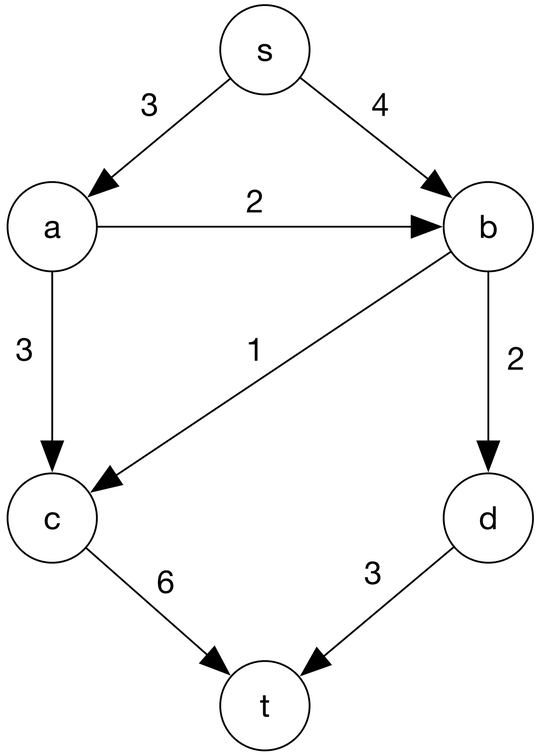


RESIDUAL

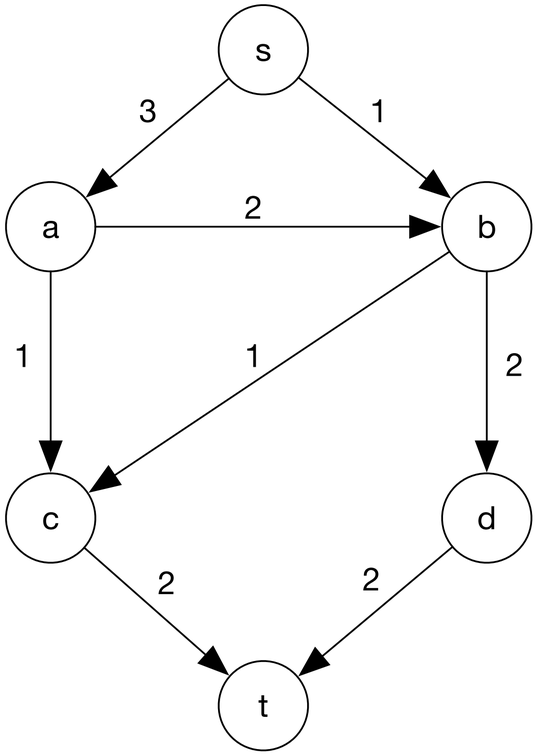


Network flow

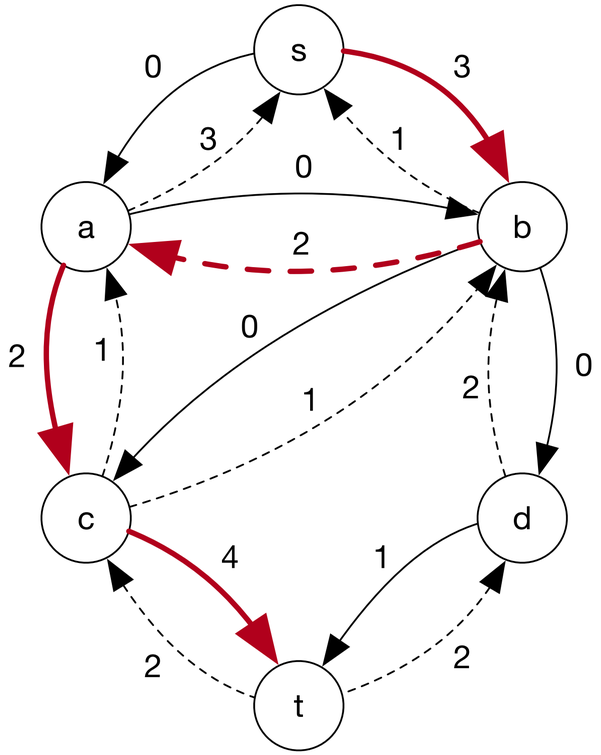
CAPACITY



FLOW

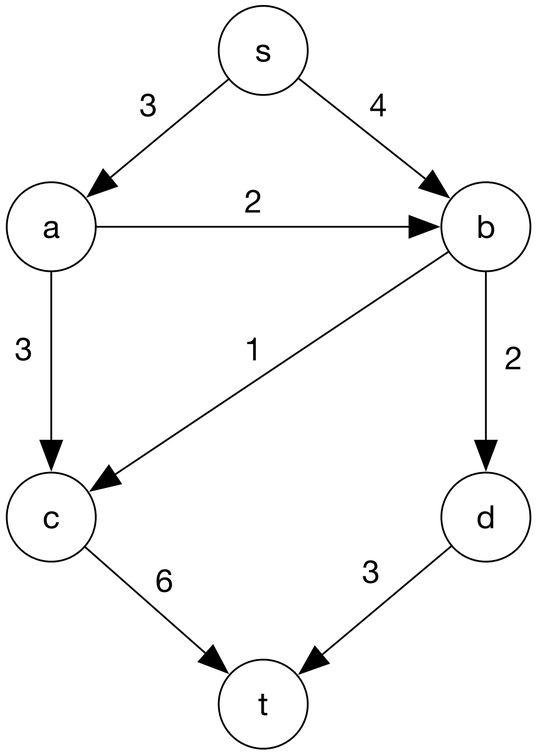


RESIDUAL

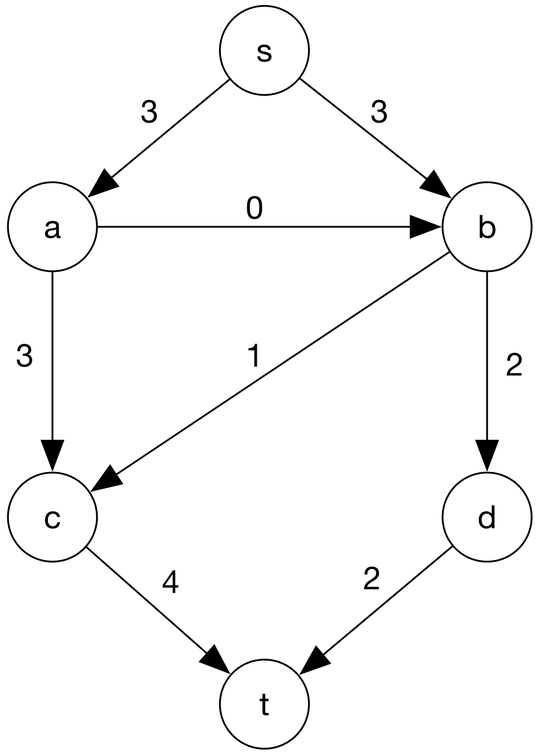


Network flow

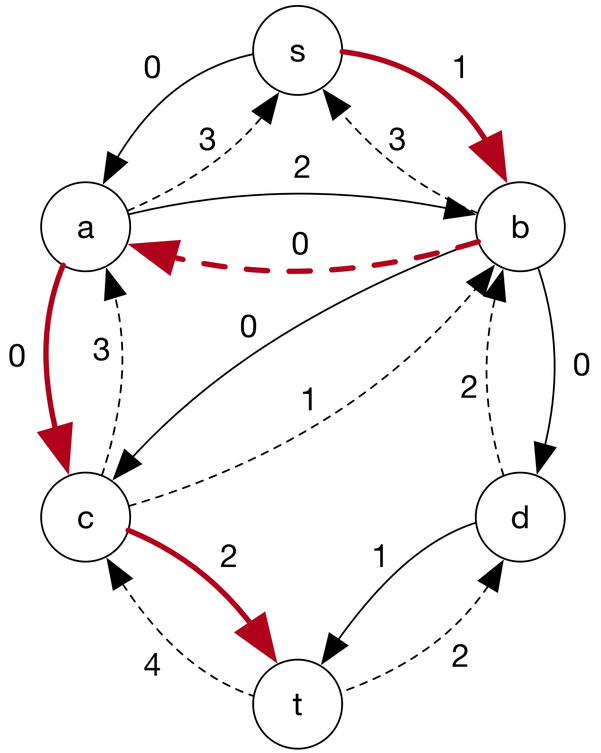
CAPACITY



FLOW

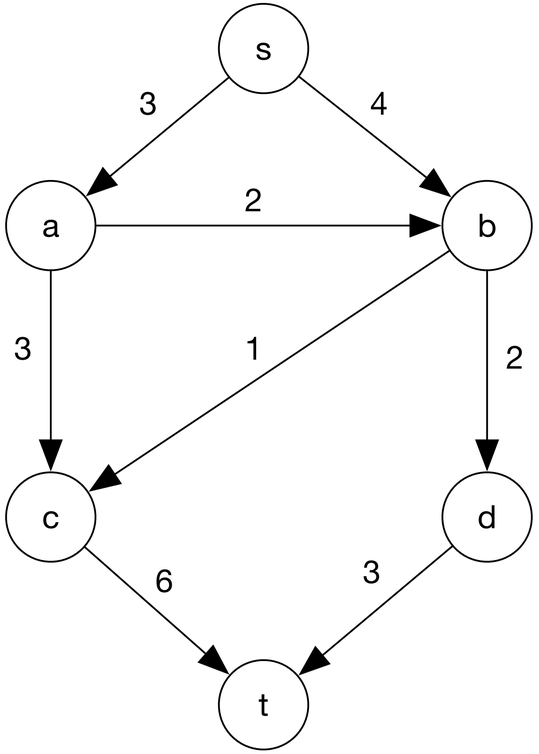


RESIDUAL

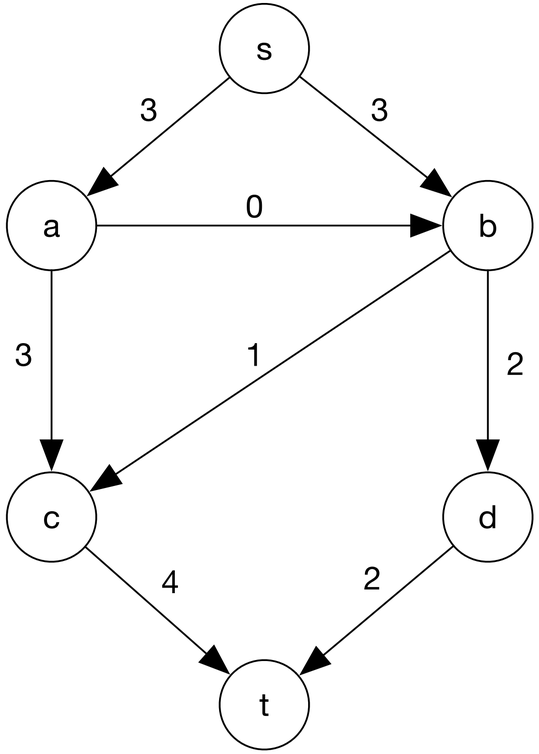


Network flow

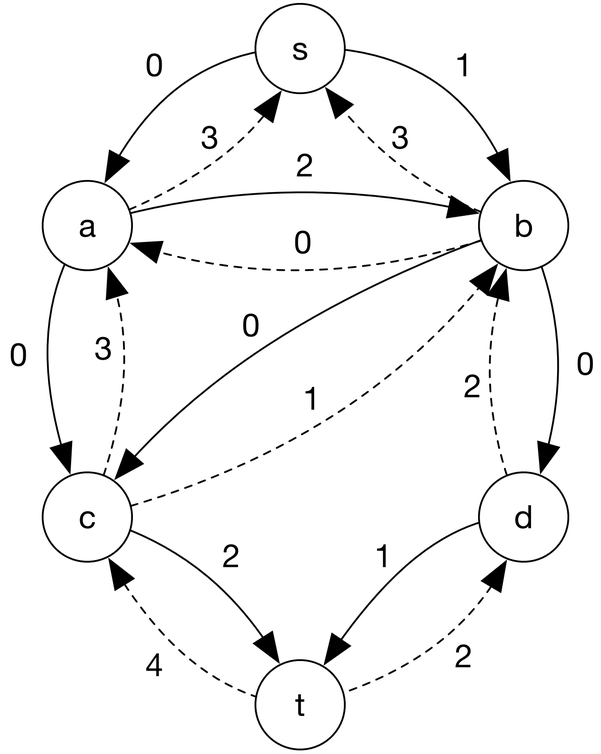
CAPACITY



FLOW



RESIDUAL



Network flow

Ford-Fulkerson

- Initialize flow network to all zeros
- Initialize residual network with weights on forward edges to correspond capacities, and weights on backward edges to zero
- While augmenting paths exist in the residual graph
  - Update flow network by adding augmenting flow to corresponding edges
  - Update forward edges in residual network by deducting augmenting flow from corresponding edges
  - Update backward edges in residual network by adding augmenting flow to corresponding edges

Network flow

Is this a greedy algorithm? Yes!

What's the complexity? $\mathcal{O}(|E|f)$

How do we find augmenting paths? It depends.

Network flow

Is this a greedy algorithm? Yes!

What's the complexity? $\mathcal{O}(|E|f)$

How do we find augmenting paths? It depends.

- If we use breadth-first search, this is called Edmonds-Karp algorithm
- Worst case complexity of Edmonds-Karp $\mathcal{O}(|V| \times |E|^2)$

Network flow

What we haven't discussed:

- Reals: rational- and irrational-valued capacities
- Proof of correctness
- Special applications of network flow (*i.e.*, bipartite matching)