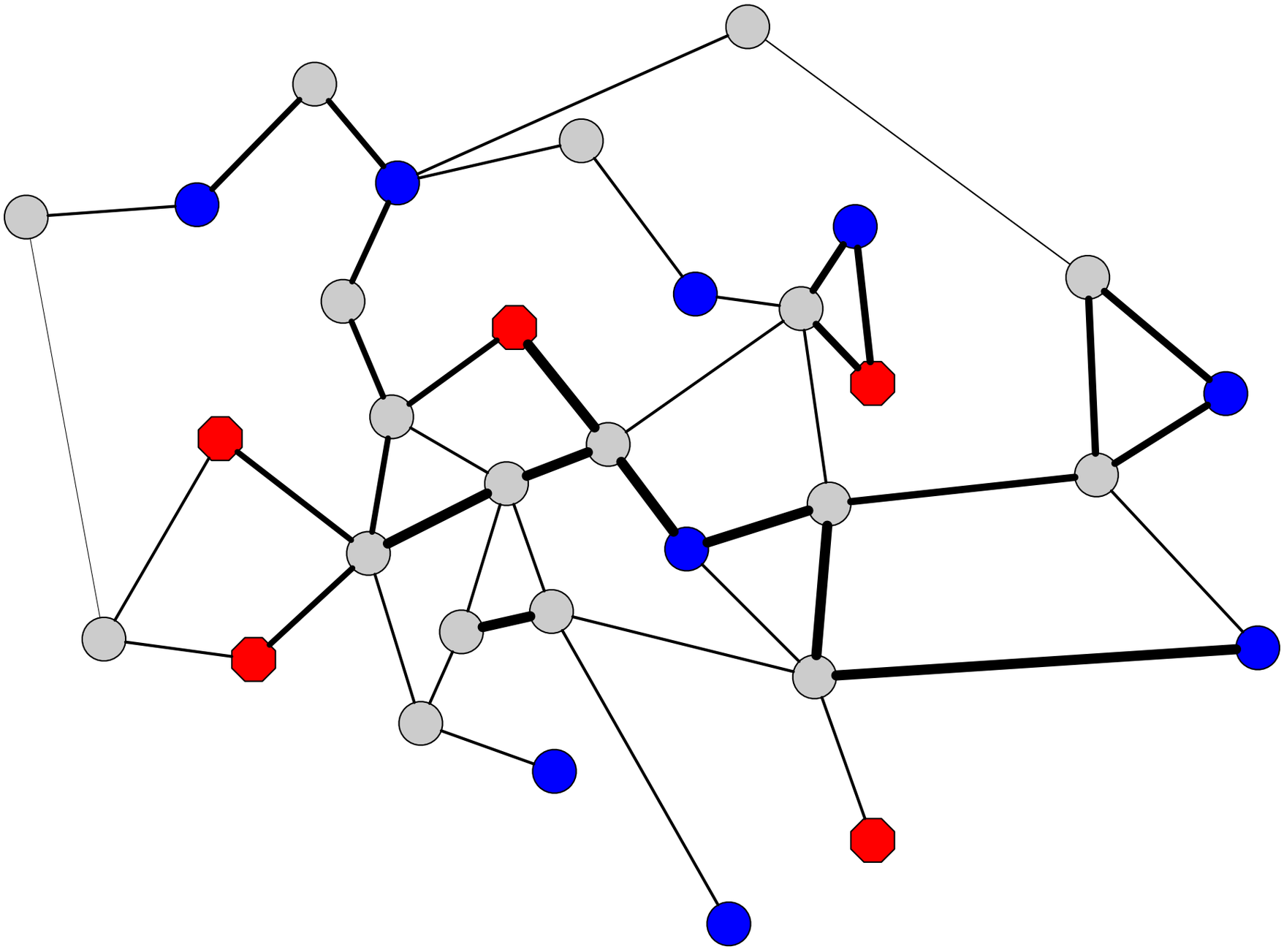
Maximum Flow Problems

Thomas Schwarz, SJ

Flow Networks

- Assume we have several power generators and several large consumers of power
- Transmission grid has limited capacity
- Power generators have maximum power generation
- Consumers have maximum power consumption
- Transmission lines have limited capacity
- What is the maximum amount of power that can be moved from generators to consumers?

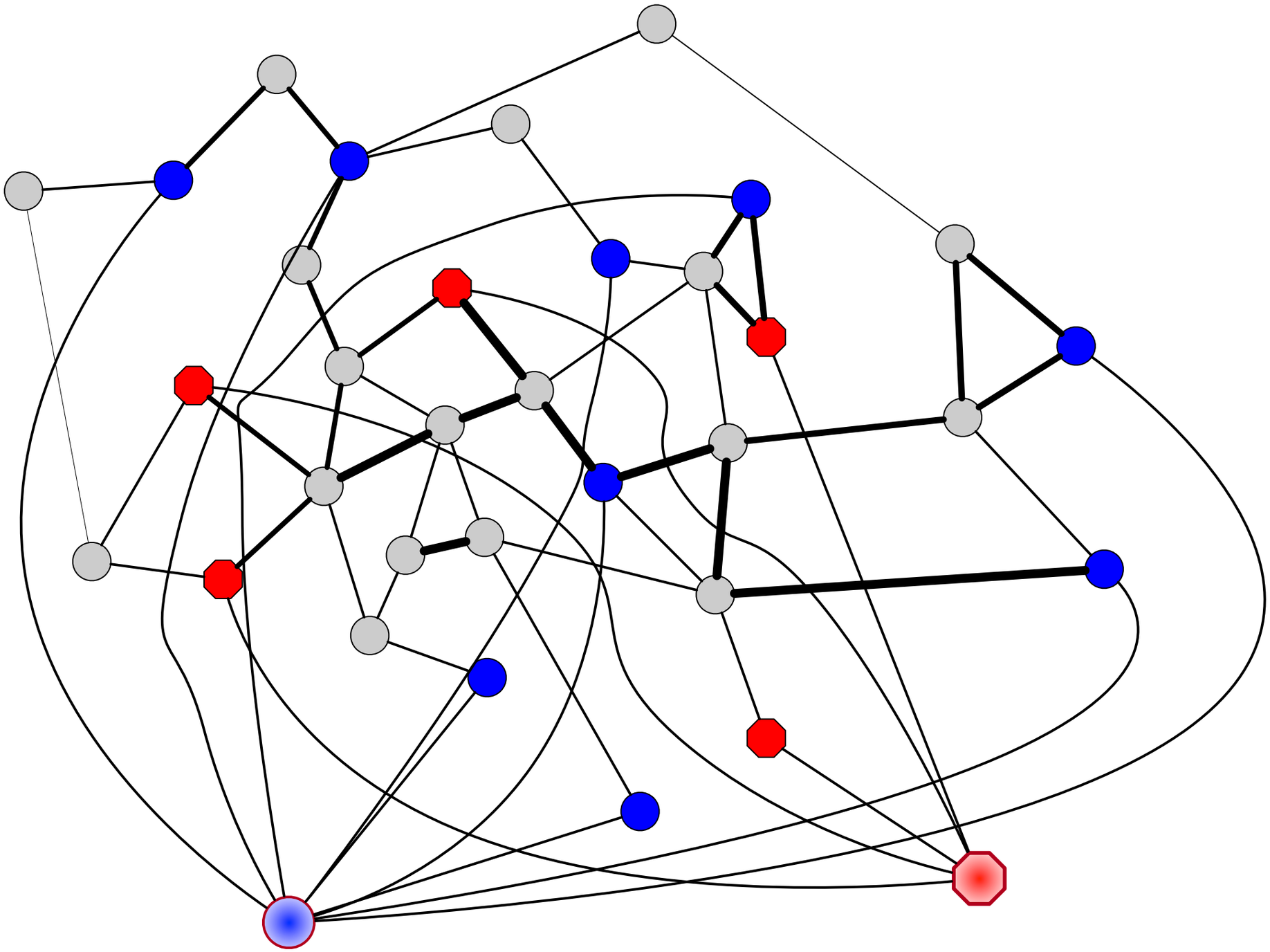
Flow Networks



Flow Networks

- Create a super-source node and a super-consumer
- Connect all generators to the super source with a transmission capacity equal to the power source
- Connect all consumers to the super-consumer with a transmission capacity equal to the maximum demand

Flow Networks



Flow Networks

- Question becomes:
  - What is the maximum flow from producer to consumer?

Flow Networks

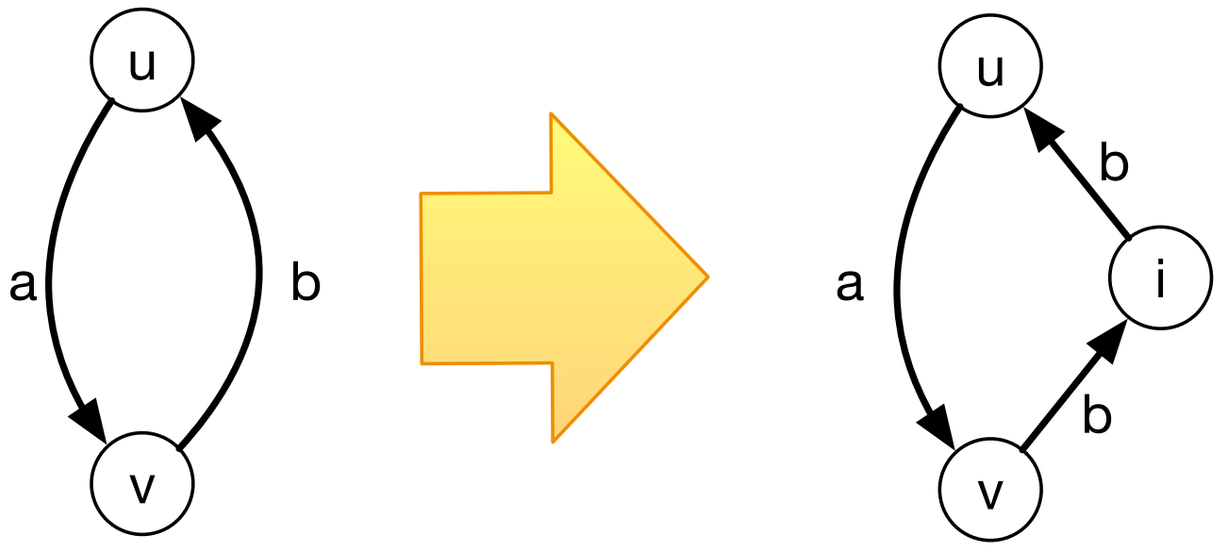
- Flow networks:
  - Directed graph (V, E)
  - Each edge (u, v) has a capacity of $c(u, v)$
  - Two special nodes: s (source), t (sink)
  - Can remove nodes that do not lie on a path from s to t
  - A flow is a function $f : V \times V \to \mathbb{R}, \quad u, v \mapsto f(u, v)$
    - $\forall u, v, \in V : \ 0 \leq f(u, v) \leq c(u, v)$
    - $\forall u \in V \backslash \{s, t\} : \sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v)$ (influx = outflux)

Flow Networks

- The maximum flow problem:
  - Maximize $|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$
    - Typically, a flow network will have no edges into the source and the second addend is zero

Flow Networks

- In Mathematics, a directed graph cannot have simultaneously an edge (u, v) and an edge (v, u)
  - These are called antiparallel edges
- We can get around this by introducing artificial vertices

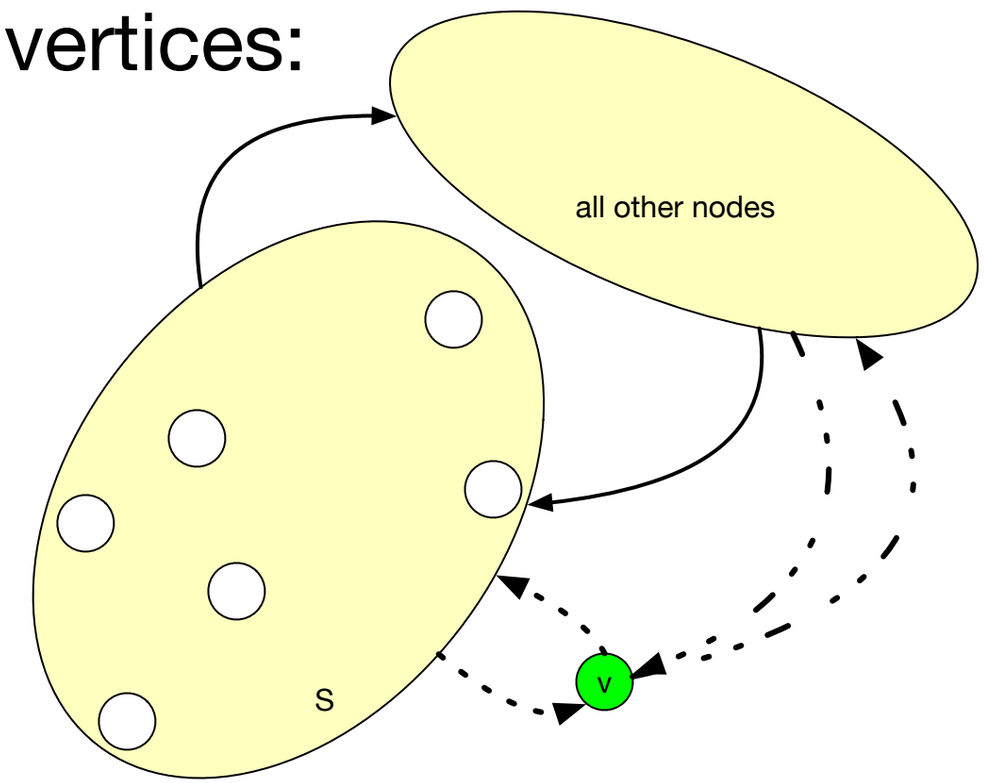


Ford Fulkerson

- Family of algorithms based on
  - Residual networks
  - Augmenting paths
  - Cuts (as defined before)

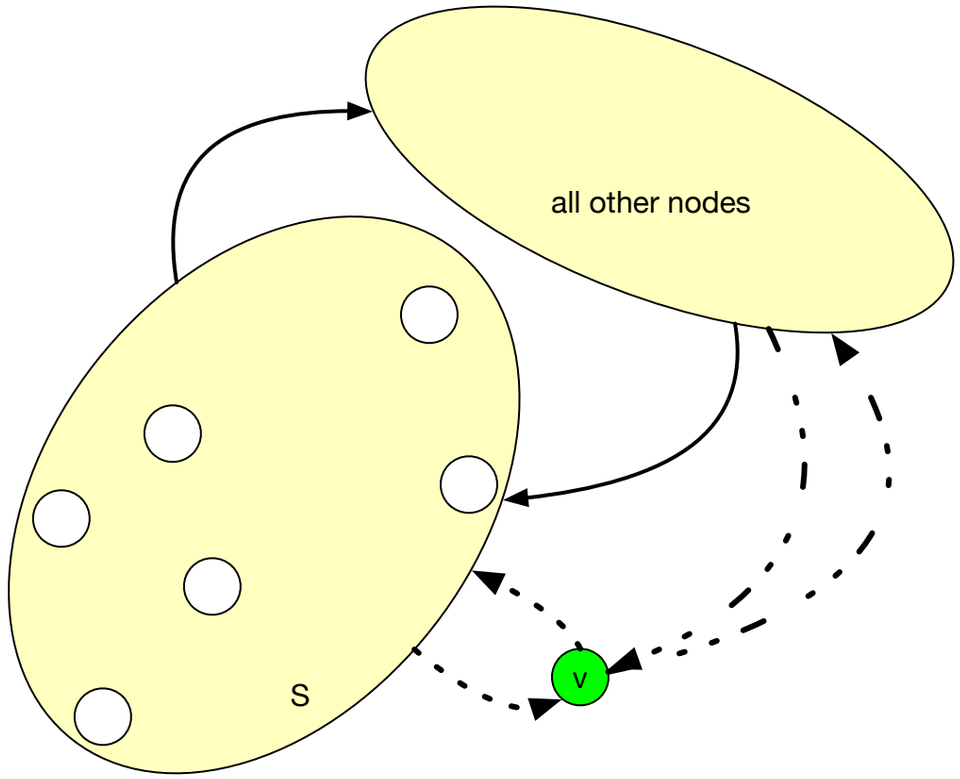
Local Equilibrium

- Inflow is equal to Outflow for any set of vertices (not including source and sink
  - Proof by induction:
    - For one vertex: This is the equilibrium condition
    - n vertices to $n + 1$ vertices:



Local Equilibrium

- Let v be a new vertex and form $S' = S \cup \{v\}$. Let
- Inflow and Outflow between S and $\complement(S)$ are equal
- Inflow into S' changes by
  - add flow from $\complement(S) \backslash \{v\}$ to $\{v\}$
  - subtract flow from $\{v\}$ to S
- Outflow from S' changes by
  - subtract flow from S to $\{v\}$
  - add flow from $\complement(S) \backslash \{v\}$ to $\{v\}$
- Because of equilibrium in v, the difference is zero



Local Equilibrium

- Corollary: Outflow from source = Inflow to sink
- Introduce a dummy link between sink and source:
  - Flow has now equilibrium for all set of vertices

Local Equilibrium

- Cuts: Partition of the vertex set with source in one and sink in the other partition set
- For a cut, we can calculate the net-flow:
    - Sum of the individual flows crossing the cut away from source towards the sink
- Corollary:
    - The maximum flow is equal to the minimum flow crossing a cut

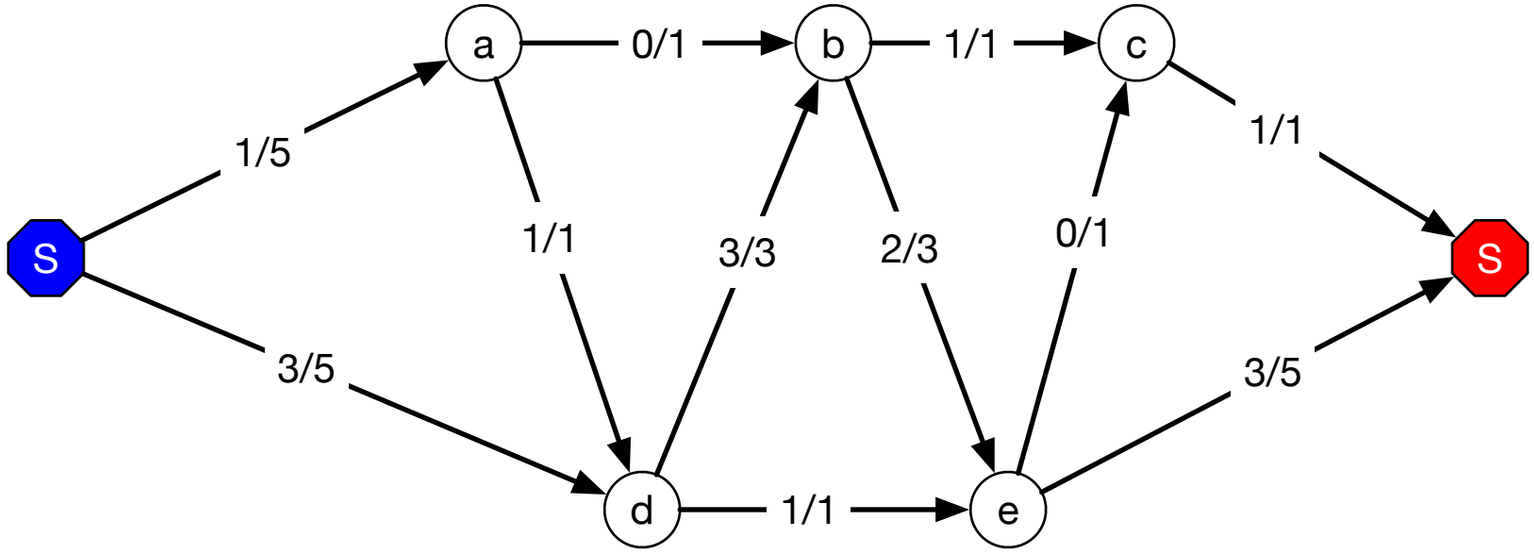
Local Equilibrium

- Given a cut S, T. Define $c(S, T)$ to be the sum of the capacities of edges from S to T

- Max-flow min-cut theorem:
  - The following is equivalent
    - f is a maximum flow
    - The residual network contains no augmenting paths
    - $|f| = c(S, T)$ for some cut (S, T)

Ford Fulkerson

- Residual Networks Motivating Example:
  - Check that the flow is balanced at all nodes



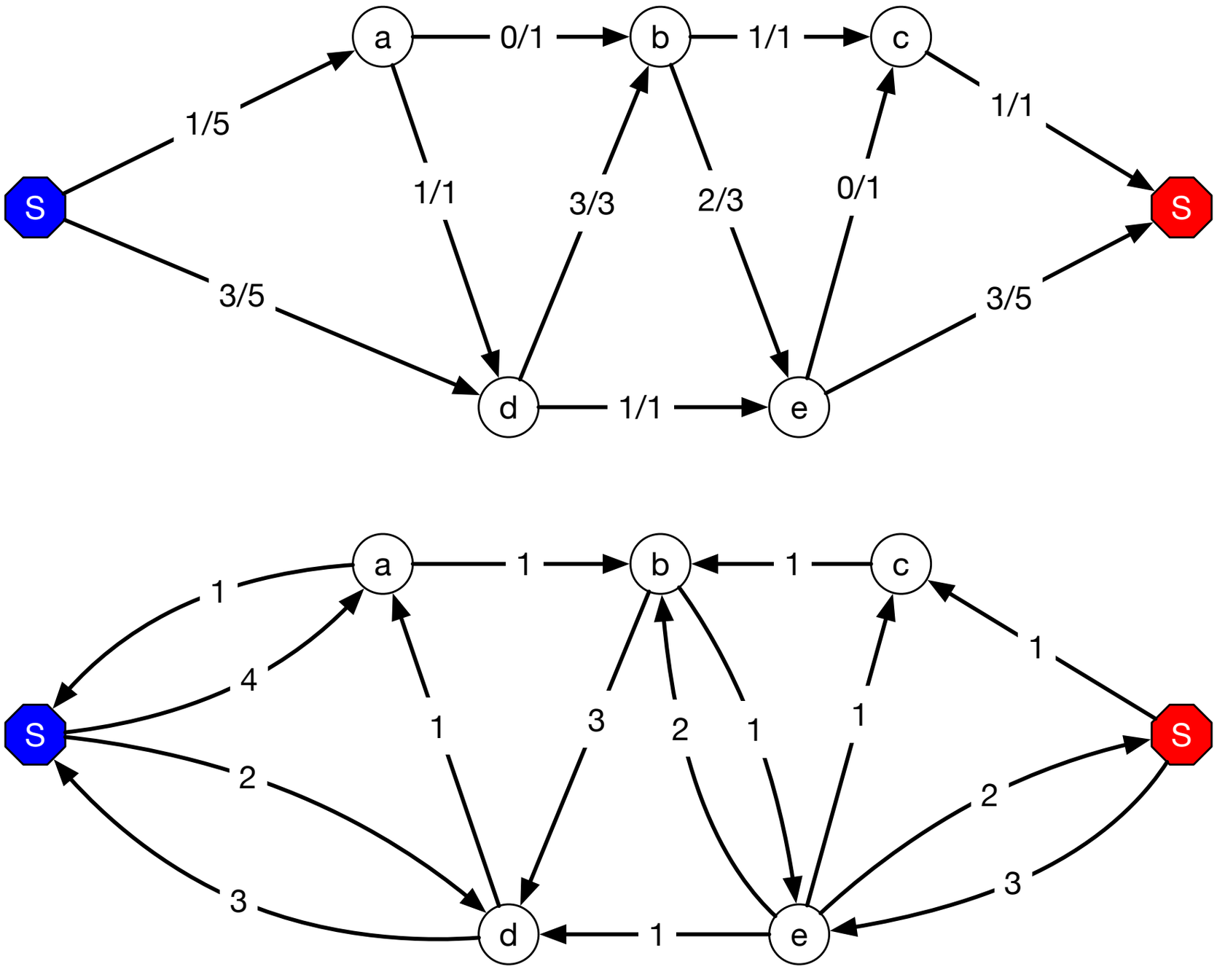
Ford Fulkerson

- The residual network captures how this flow can be changed
  - If there is a flow $f(u, v) < c(u, v)$, we can:
    - Augment the flow by up to $c(u, v) - f(u, v)$
    - Decrement the flow by up to $f(u, v)$
- Therefore:
  - Create an edge $c'(u, v) = c(u, v) - f(u, v)$
  - Create an edge $c'(v, u) = f(u, v)$

Ford Fulkerson

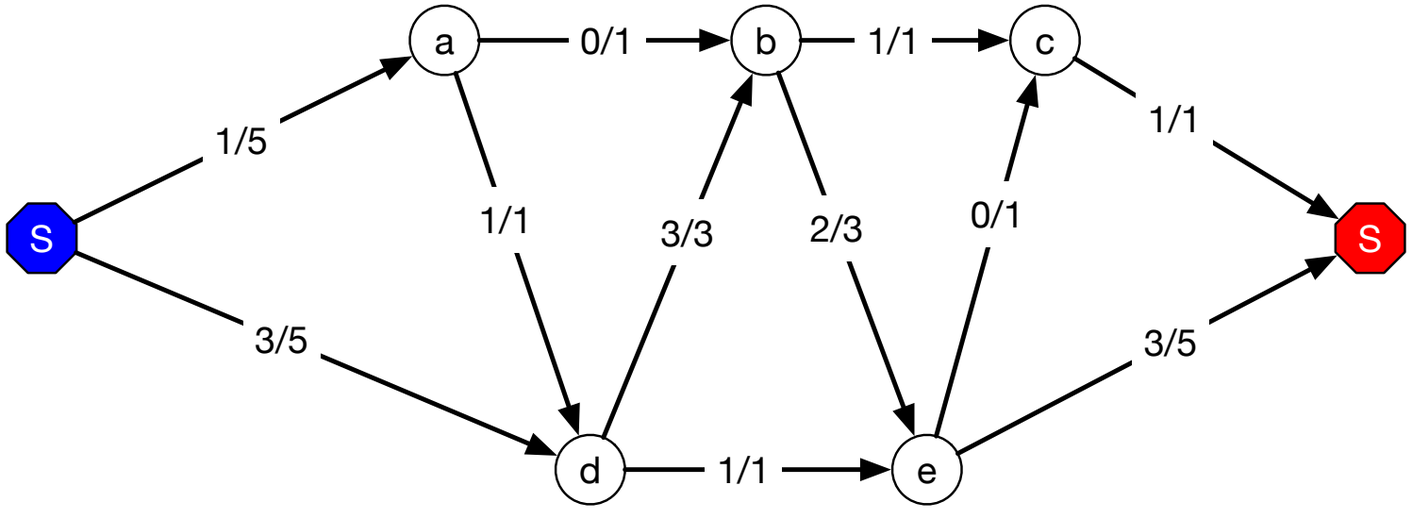
- Residual network
  - If a flow is not at capacity in an edge, we have a "residual" capacity
  - If a flow through an edge is positive, we can reduce this flow

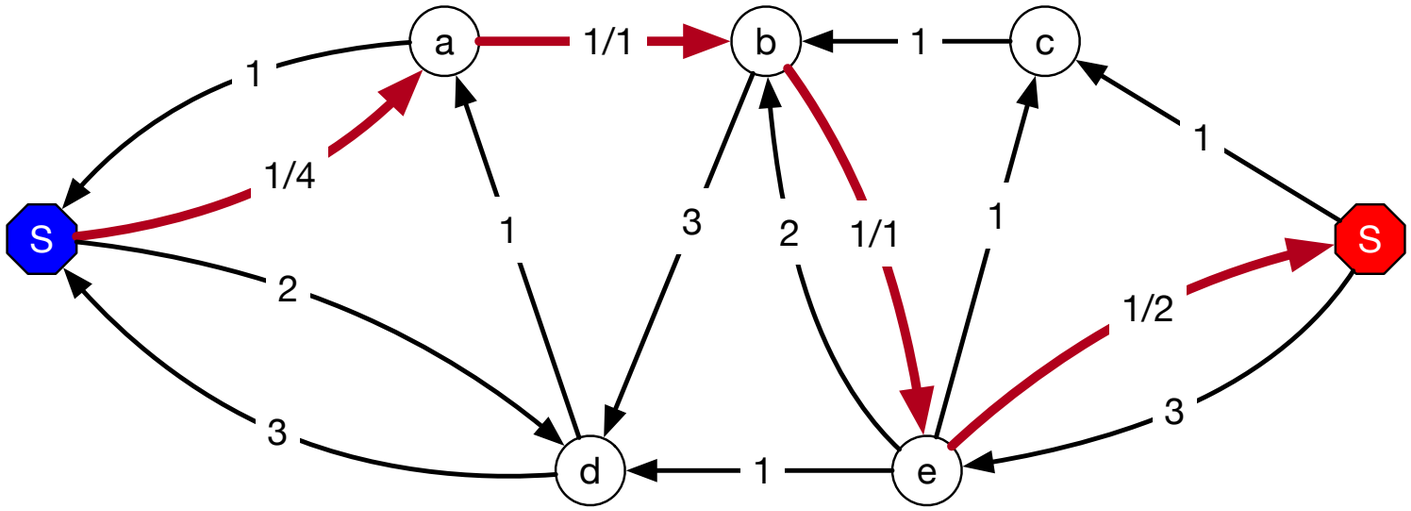
Ford Fulkerson

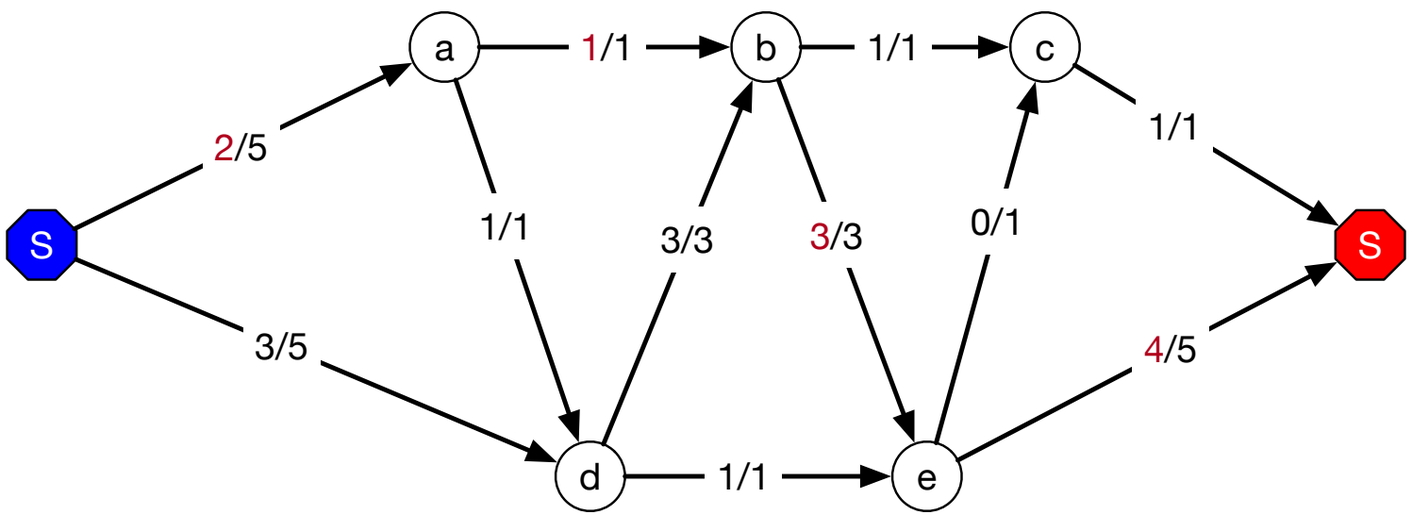


Ford Fulkerson

- Because we have anti-parallel edges, the residual network is not a graph in the sense of Mathematics
- But we can still treat this as a flow network
- Assume we can find a flow from source to sink in the residual graph
  - This is called an augmenting path
- We can then add the augmenting path flow to the previous flow
- Which carries more from sink to source


a
b
c
S
S
d
e
0/1
1/1
1/5
1/1
3/3
2/3
0/1
1/1
3/5
3/5
1/1


a
b
c
S
S
d
e
1/1
1
1
1/4
1
3
2
1/1
1
1
2
1/2
3
3
1


a
b
c
S
S
d
e
1/1
1/1
2/5
1/1
3/3
3/3
0/1
1/1
3/5
4/5
1/1

Ford Fulkerson

- More formally: let f' be a flow in the residual network
- Then define a new flow for edges (u, v)
  - $(f \uparrow f')(u, v) = f(u, v) + f'(u, v) - f'(v, u)$

Ford Fulkerson

- When we "push" a flow along the reverse of an edge, then we have a "cancellation"

Ford Fulkerson

- Lemma: $f \uparrow f'$ is a flow
  - Capacity constraint
    - Follows from definition of a residual network
    - Inflow = Outflow follows from both being flows
- Lemma: $f \uparrow f'$ has a larger flow from source to sink

Ford Fulkerson

- Ford Fulkerson algorithms:

  1. Create a zero flow

  2. While possible: Find a path from source to sink in the residual network.

  3. Calculate the minimum flow on the path

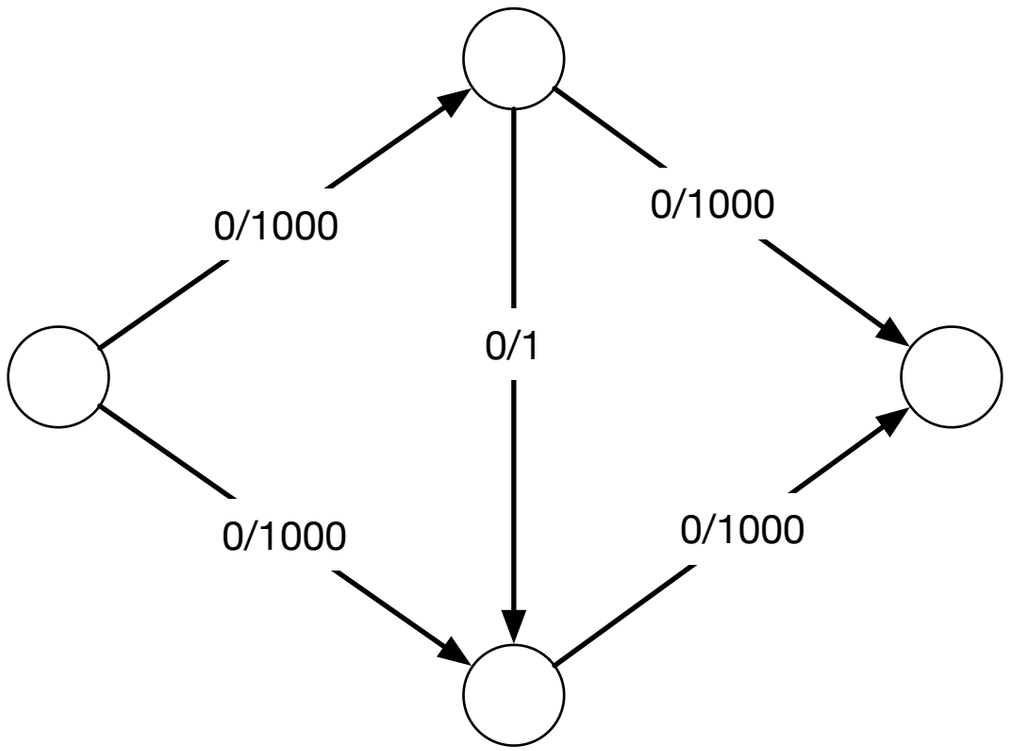
  4. Adjust the flow along the path

Ford Fulkerson

- Run-time for Ford Fulkerson algorithms depend on the way to find the path in the residual network
  - Example: There is no guarantee that we approach the maximum flow while we continue to improve
- Assume that flow capacities are integers
- For path detection: use Breadth First Search in the residual network
  - Costs time $\Theta(|E| + |V|)$ to detect a path
  - Improves flow from source to sink by 1
  - Total time is $|f| \cdot \Theta(|E| + |V|)$

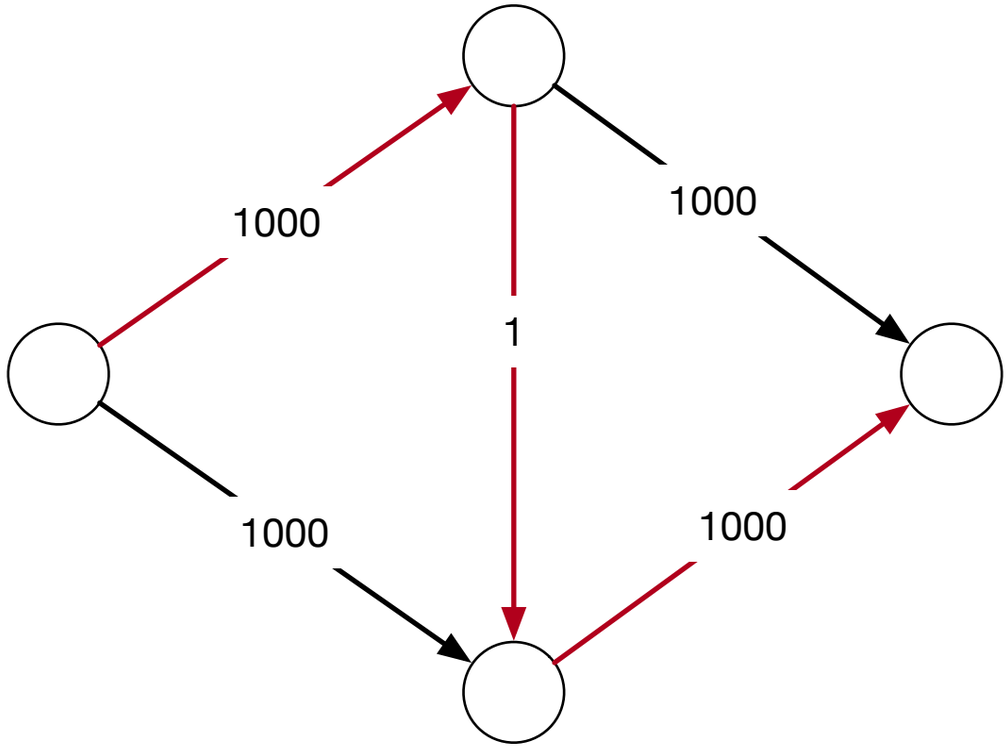
Ford Fulkerson

- Classic Example how Ford Fulkerson can be slow



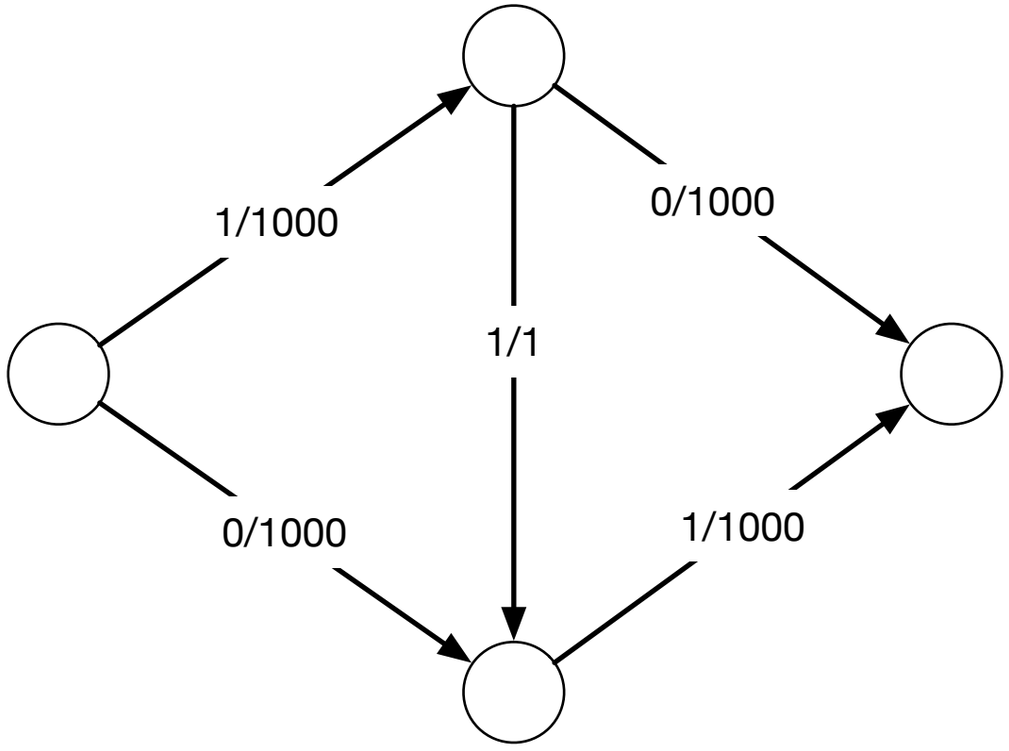
Ford Fulkerson

- Example
  - Path in the residual network



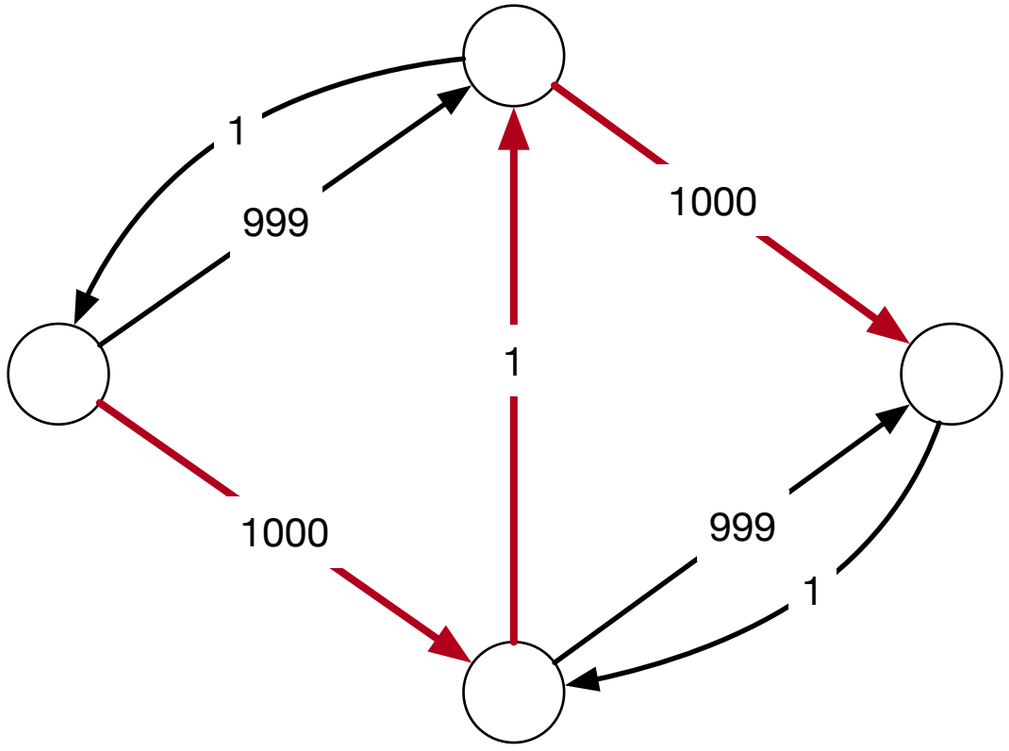
Ford Fulkerson

- Example
  - becomes flow



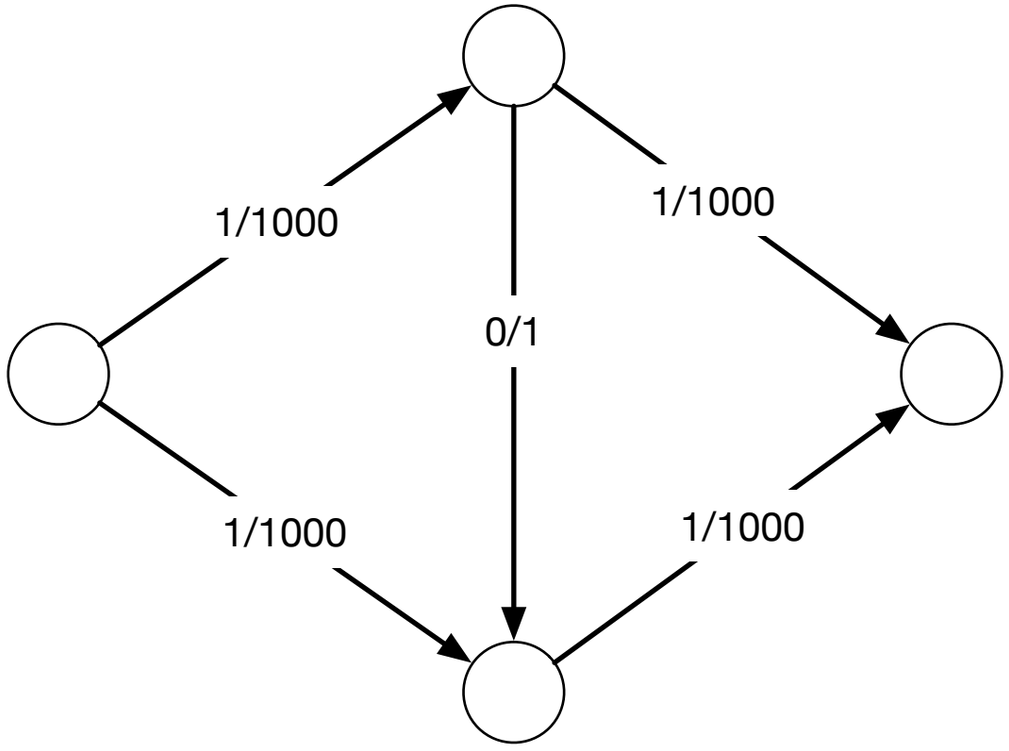
Ford Fulkerson

- Example
  - Residual network still has a path



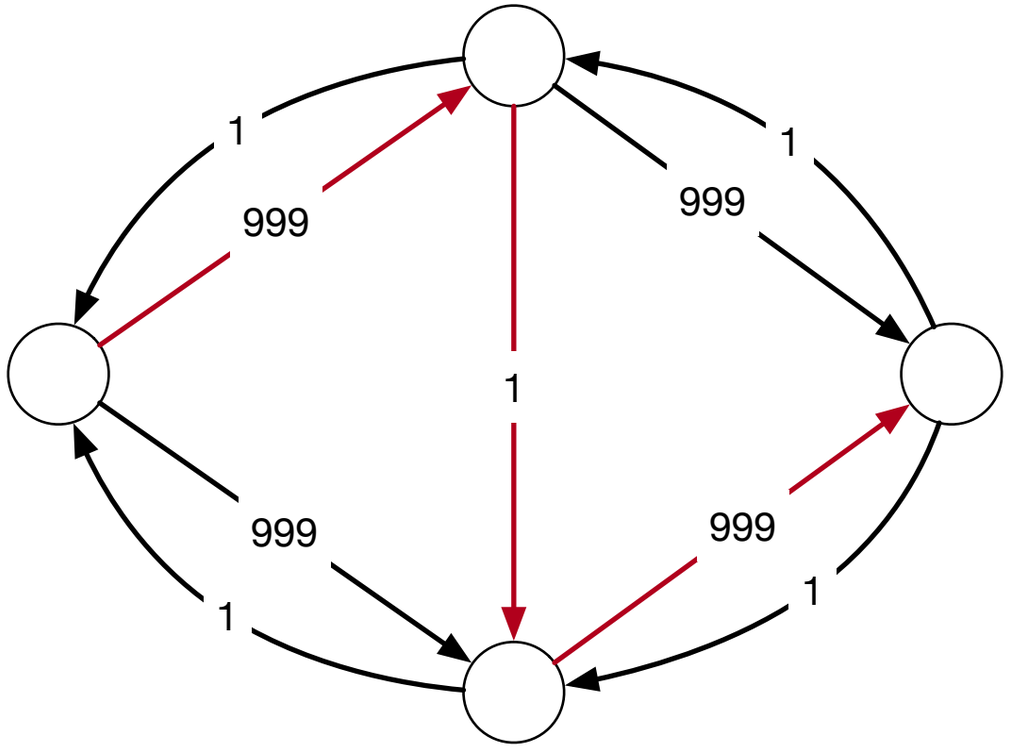
Ford Fulkerson

- Example
  - Gives an increased flow:



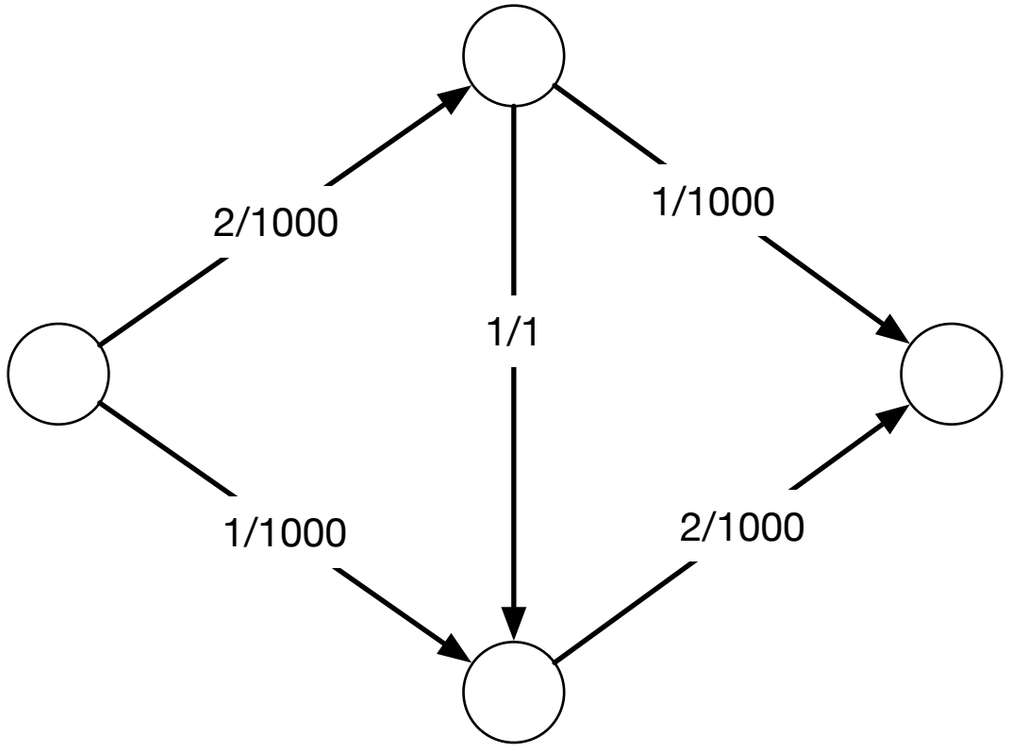
Ford Fulkerson

- Example:
  - Residual network has a path



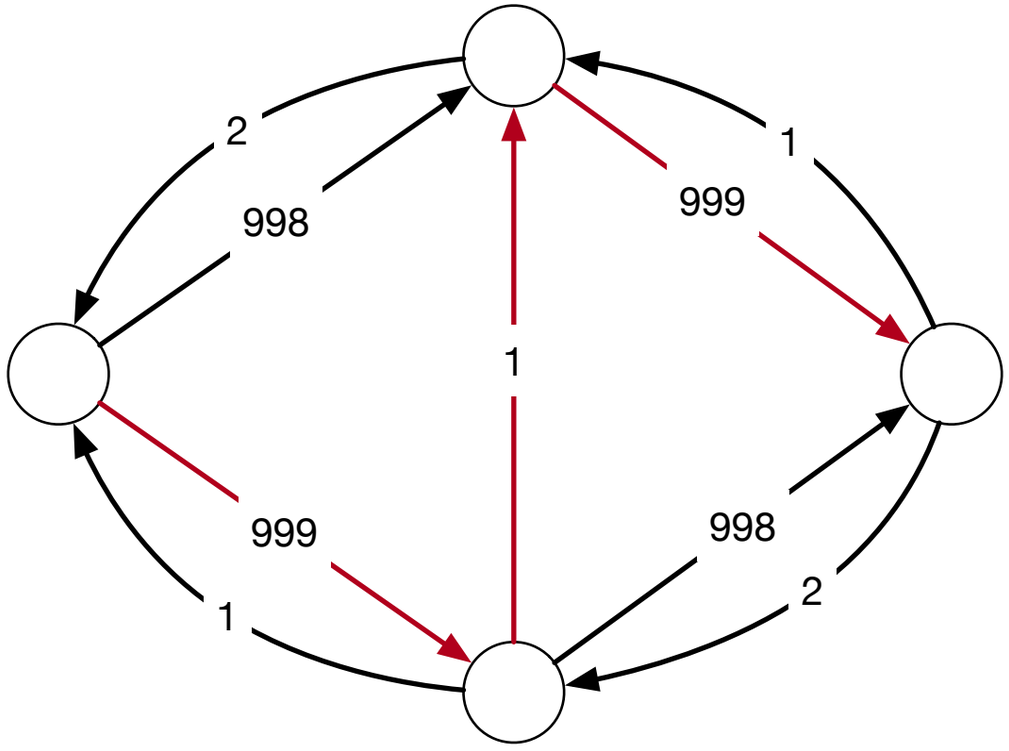
Ford Fulkerson

- Example
  - Increased flow



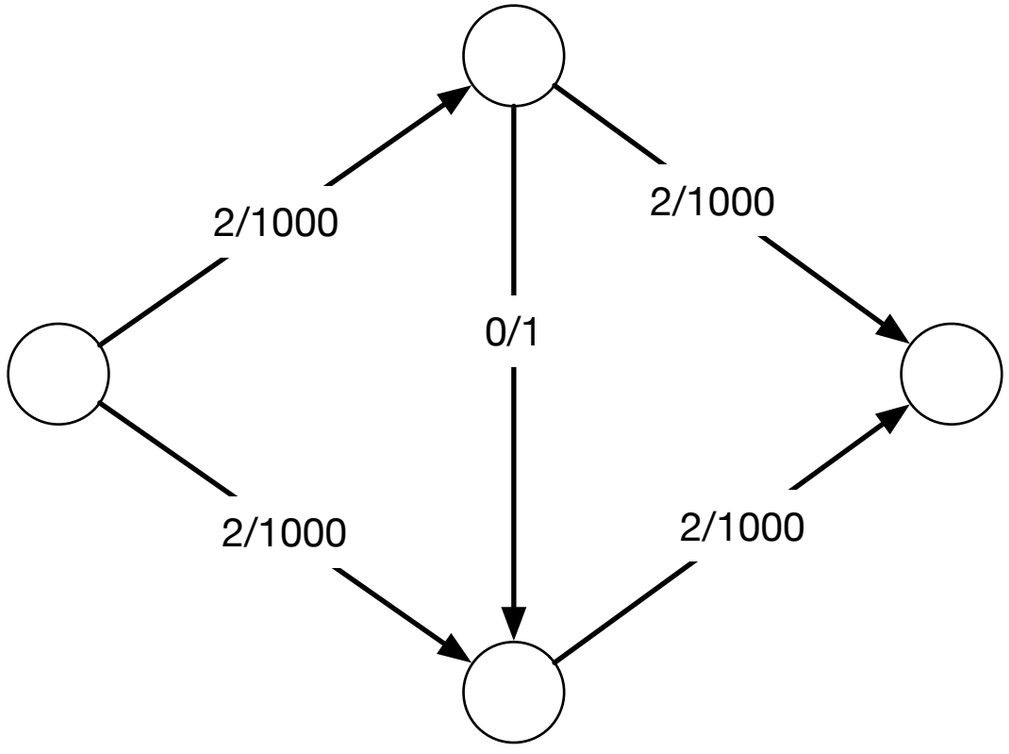
Ford Fulkerson

- Example
  - Residual Network



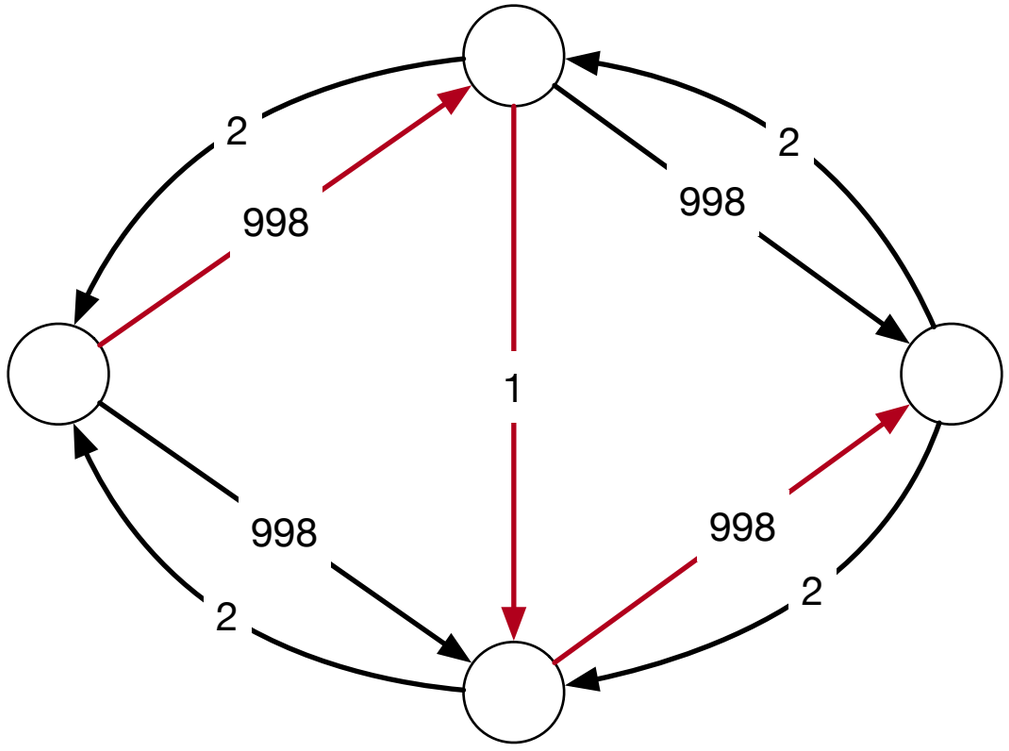
Ford Fulkerson

- Example



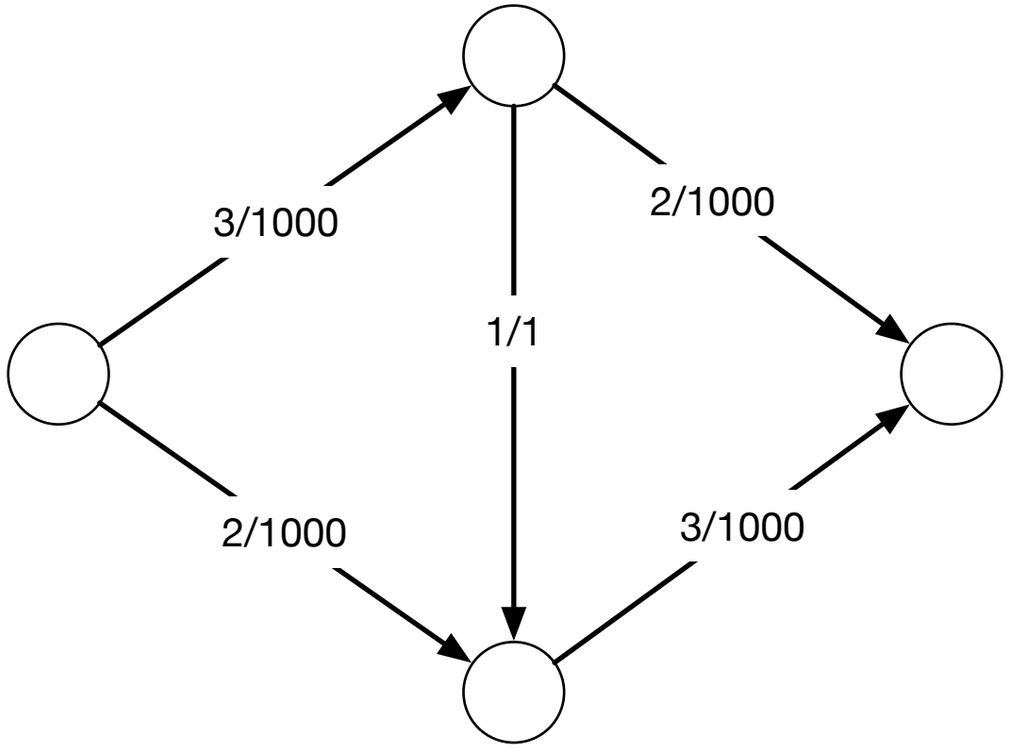
Ford Fulkerson

- Example



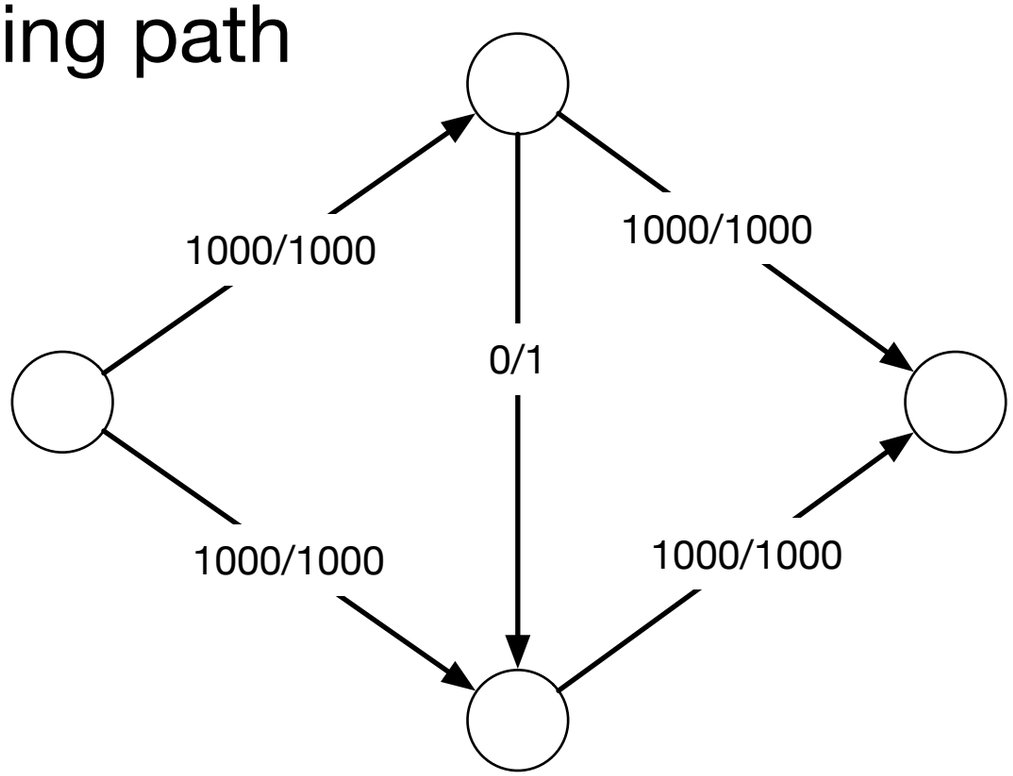
Ford Fulkerson

- Example

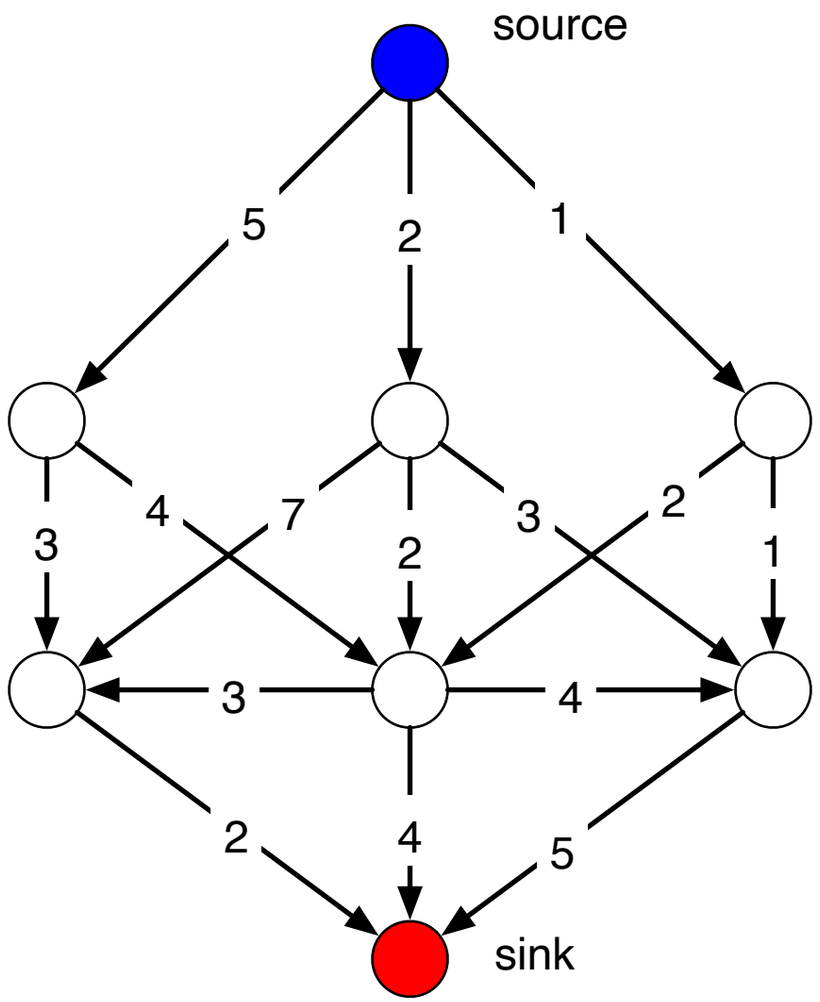


Ford Fulkerson

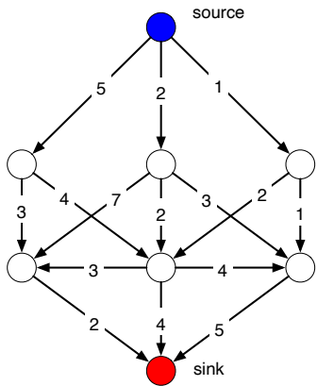
- Example:
  - It takes 2000 steps to go to the optimal flow
    - Because we were unlucky in selecting the augmenting path



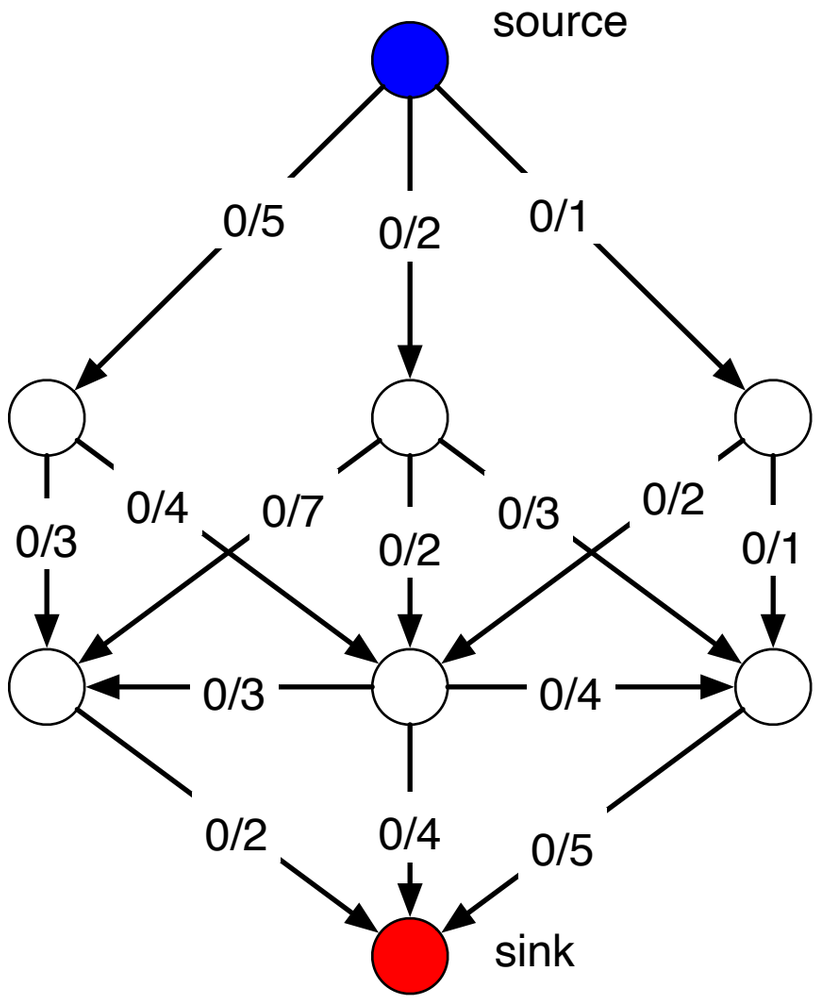
Ford Fulkerson Example



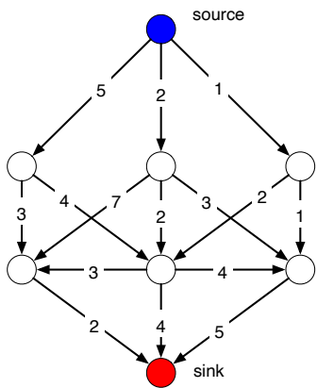
Ford Fulkerson Example



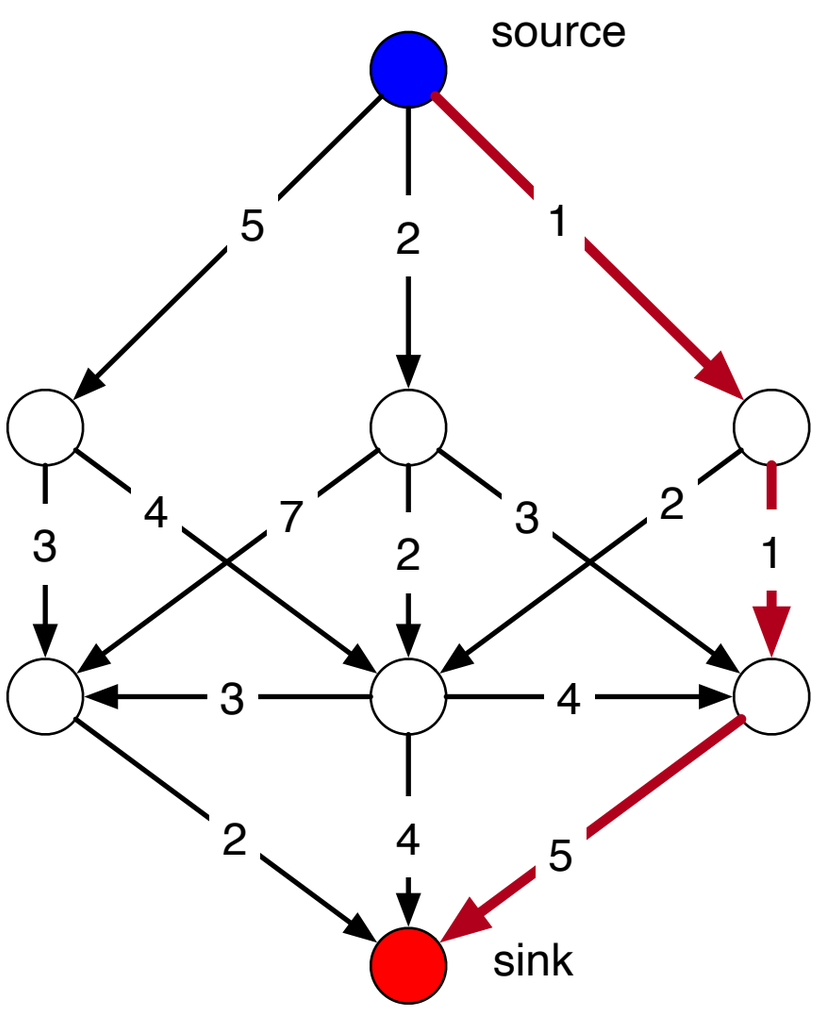
Step 1: Initialize flow to 0



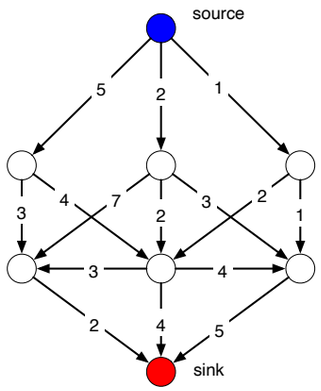
Ford Fulkerson Example

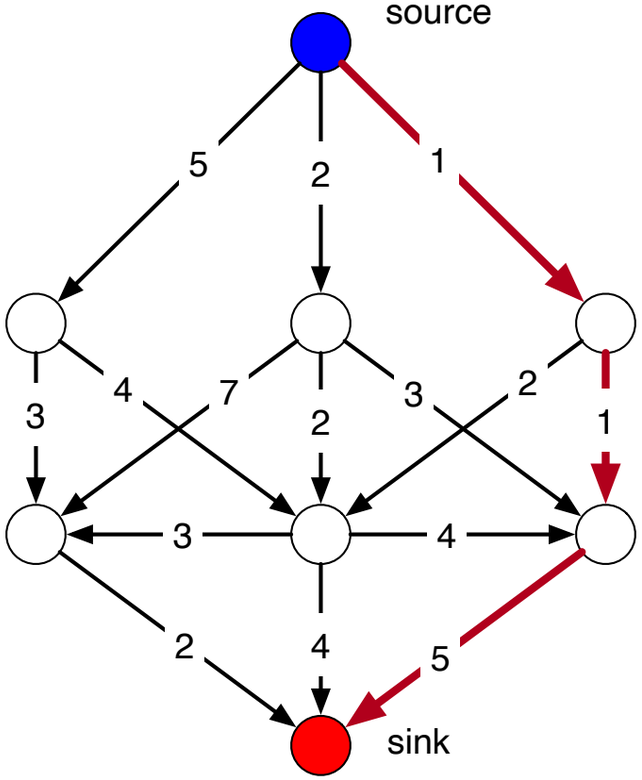


Calculate the residual graph
and find a path
from source to sink

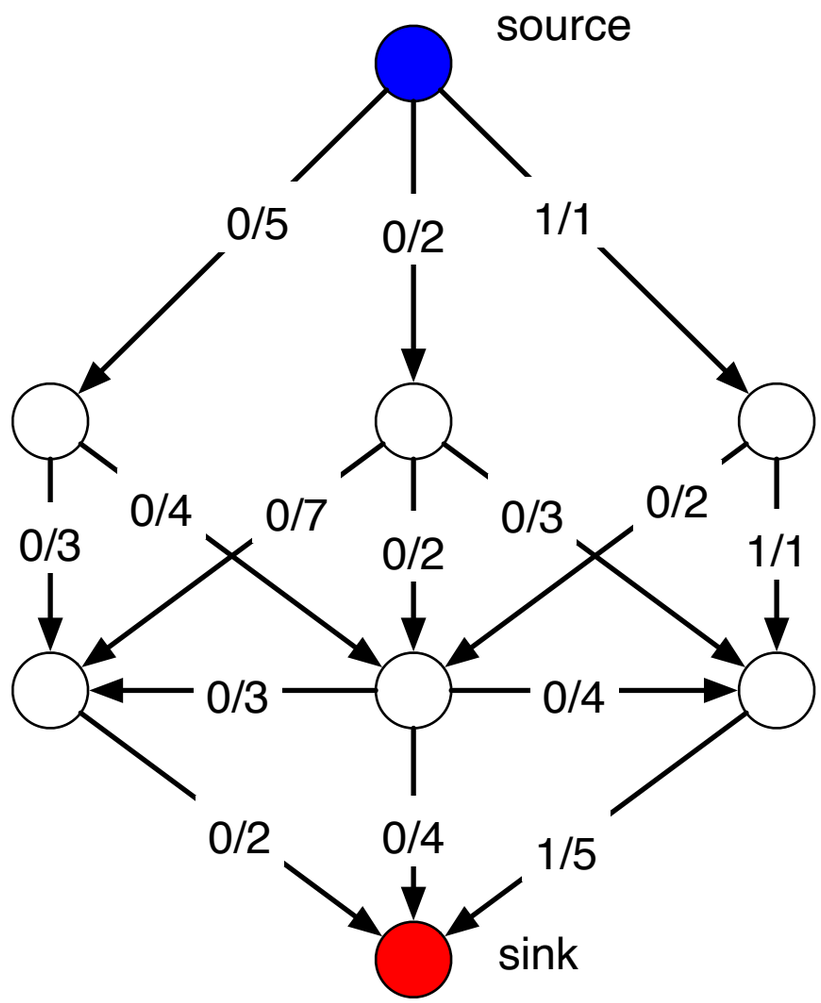


Ford Fulkerson Example

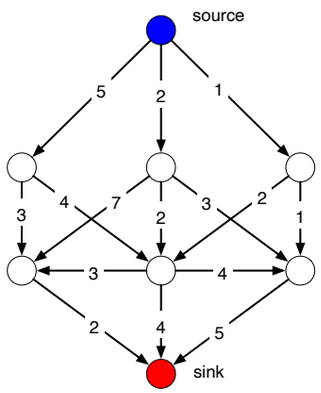




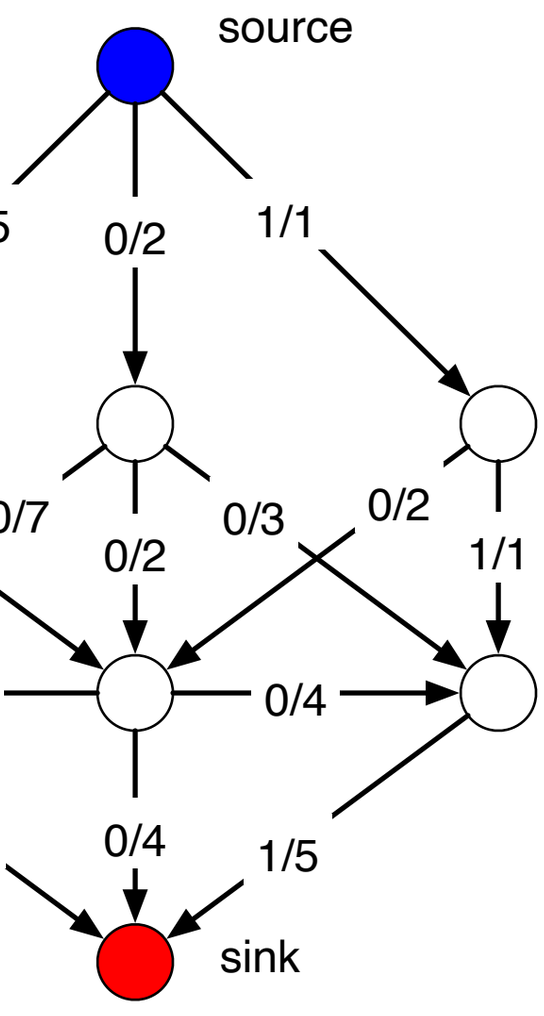
Add the residual flow to the network

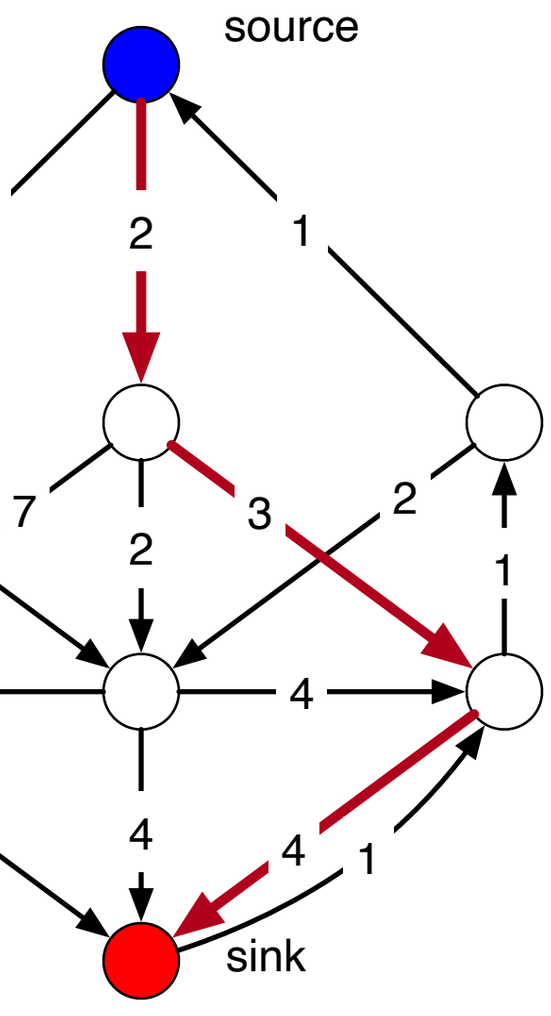


Ford Fulkerson Example

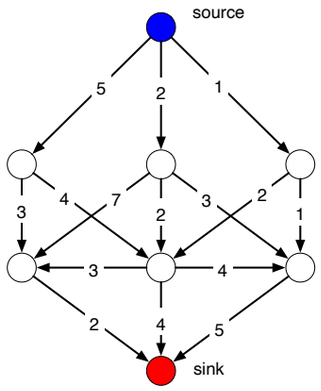


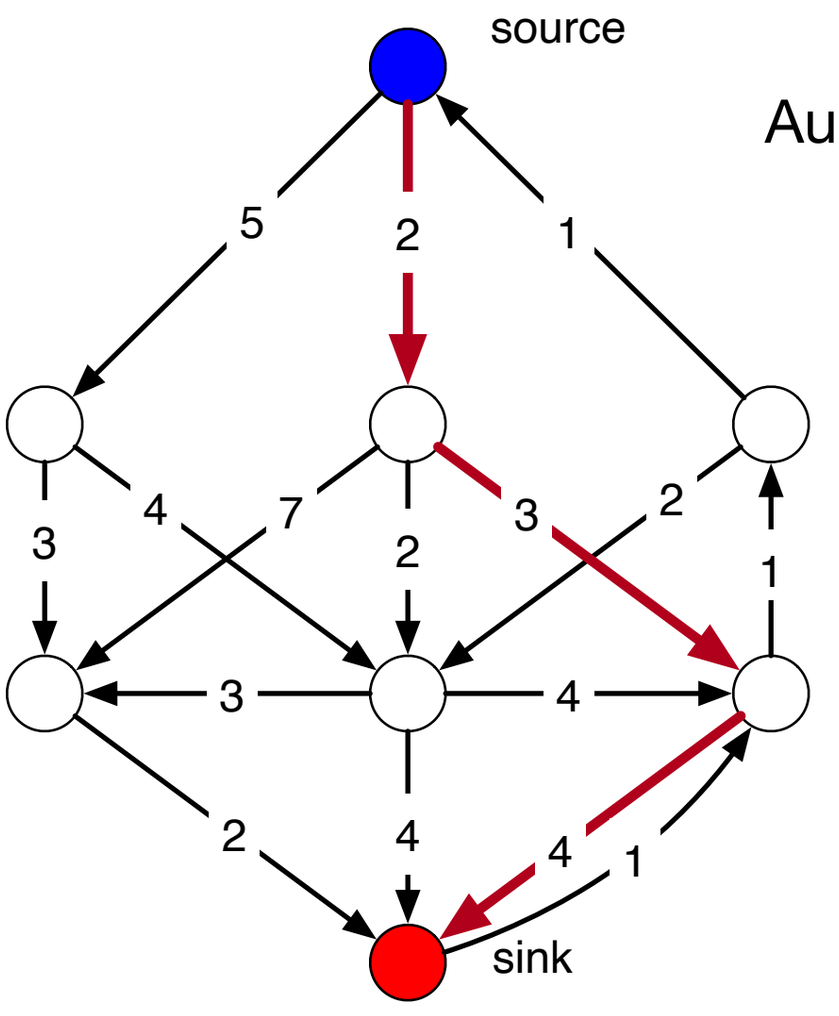
Calculate the residual graph
and find a path from
source to sink



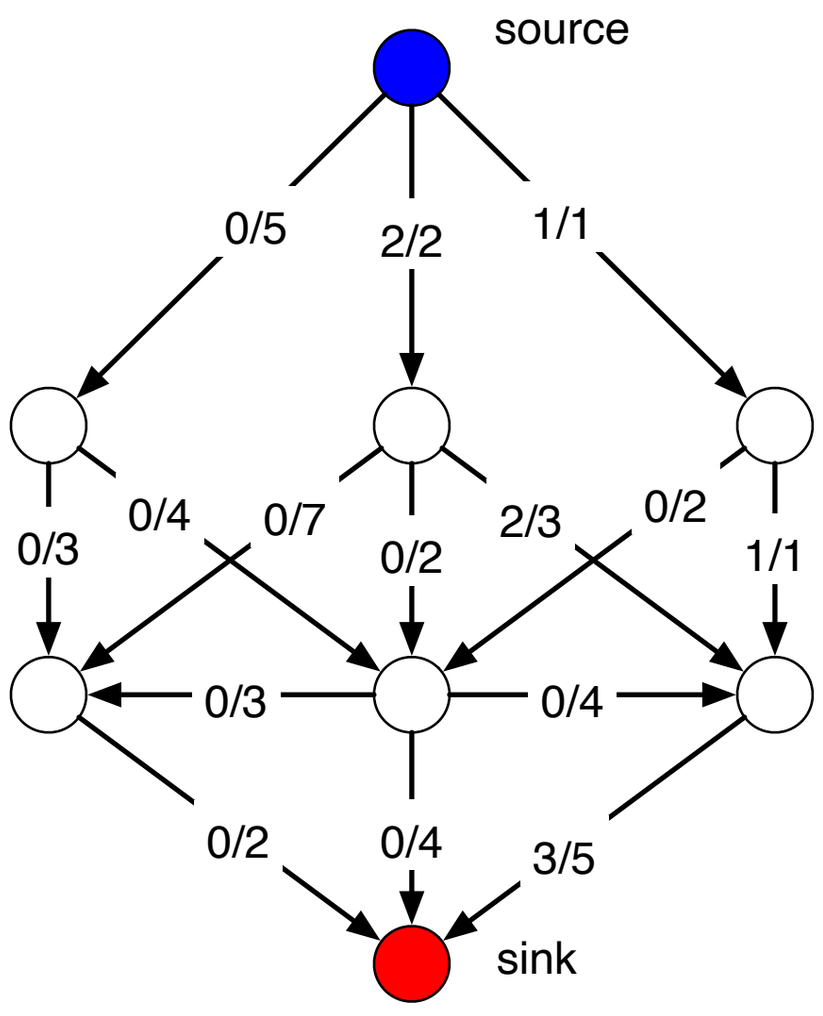


Ford Fulkerson Example

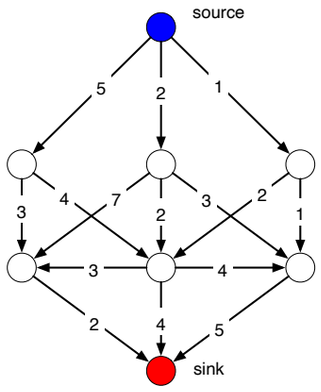


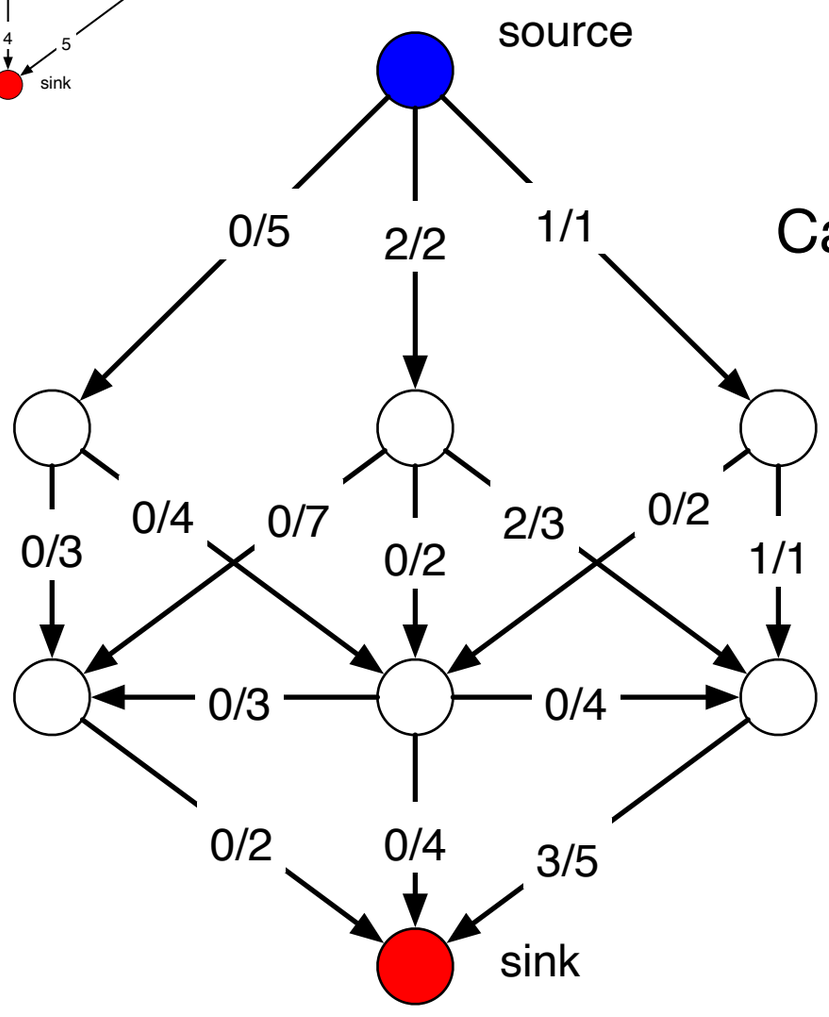


Augment the flow

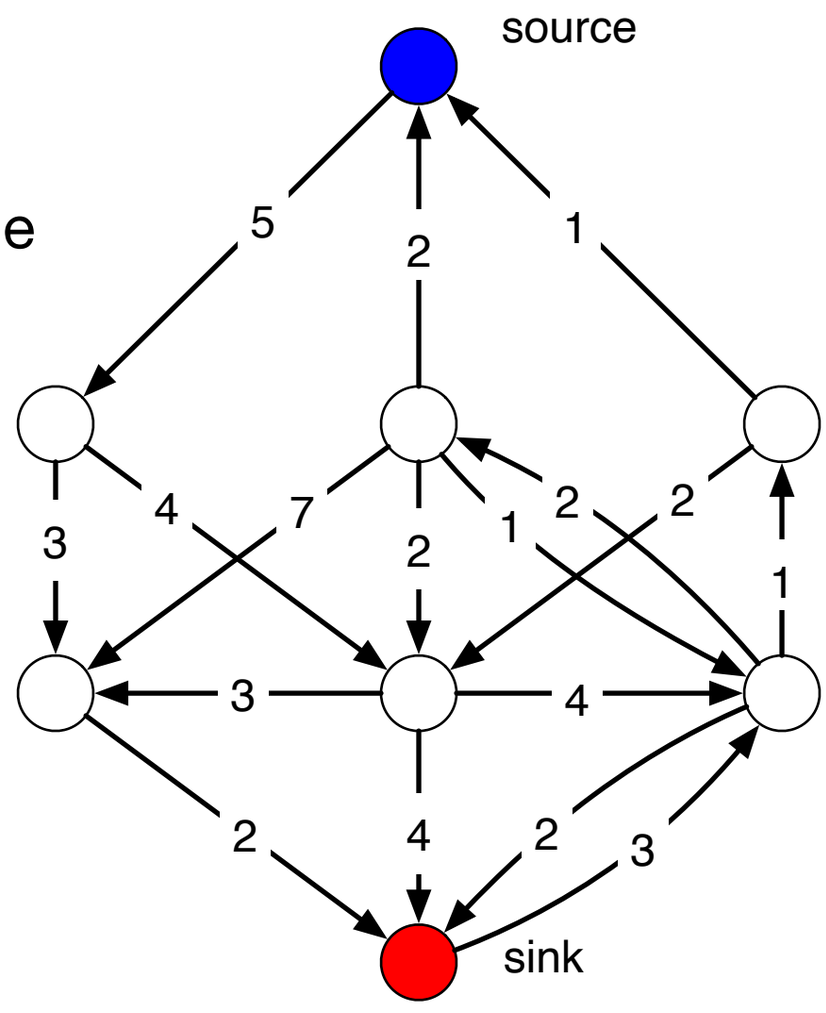


Ford Fulkerson Example

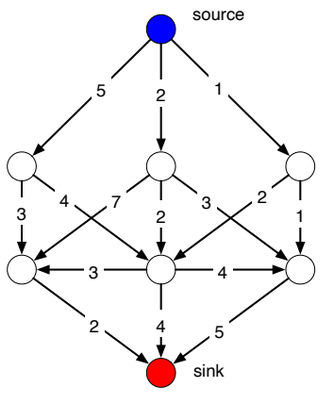




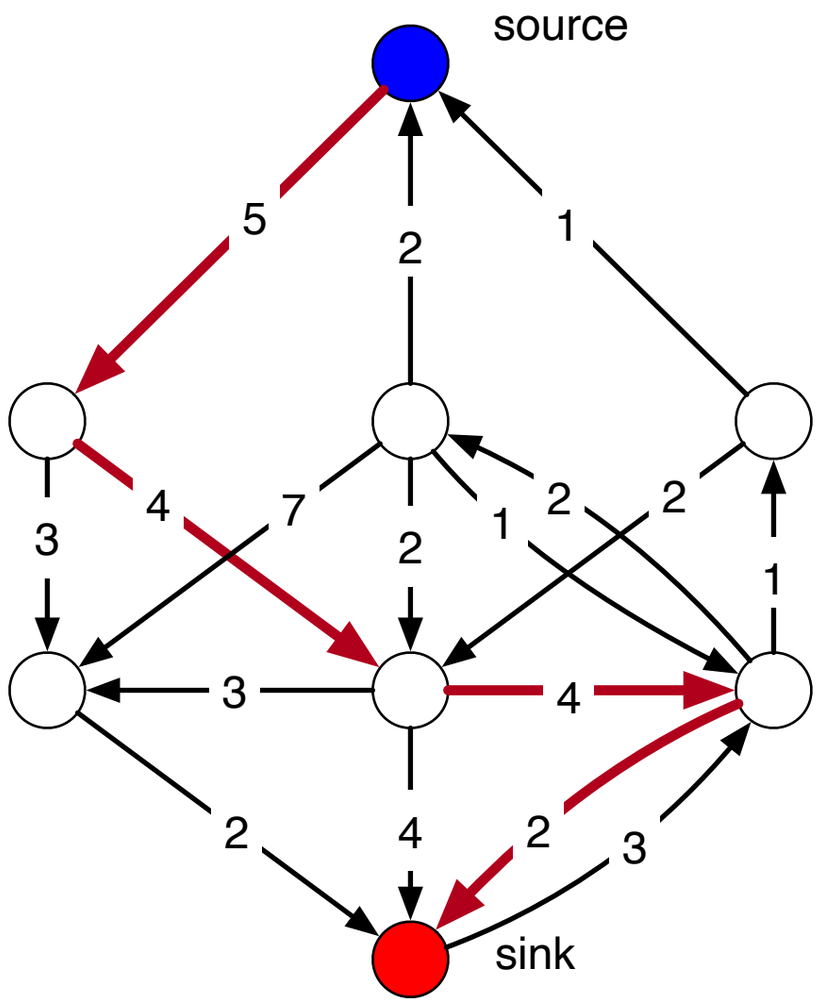
Calculate the residual



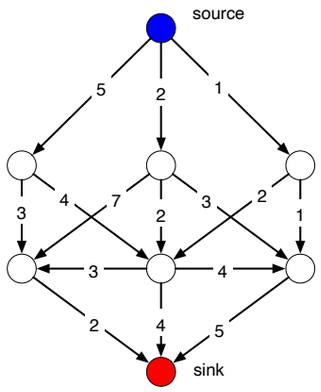
Ford Fulkerson Example

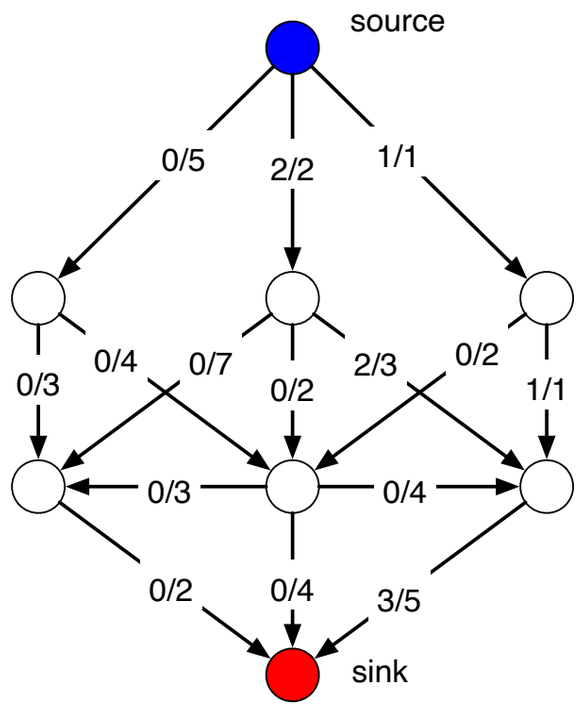


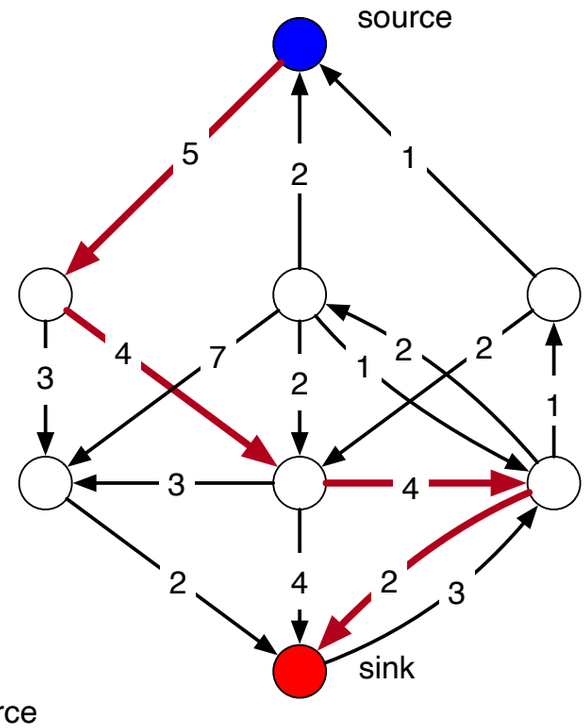
Find a path from source to sink



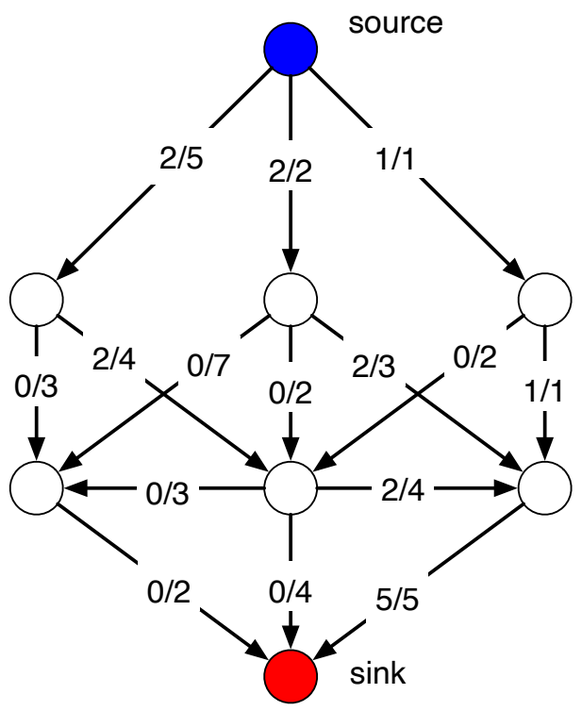
Ford Fulkerson Example



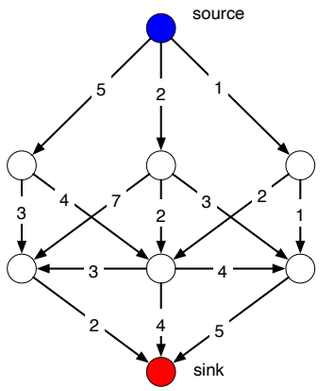


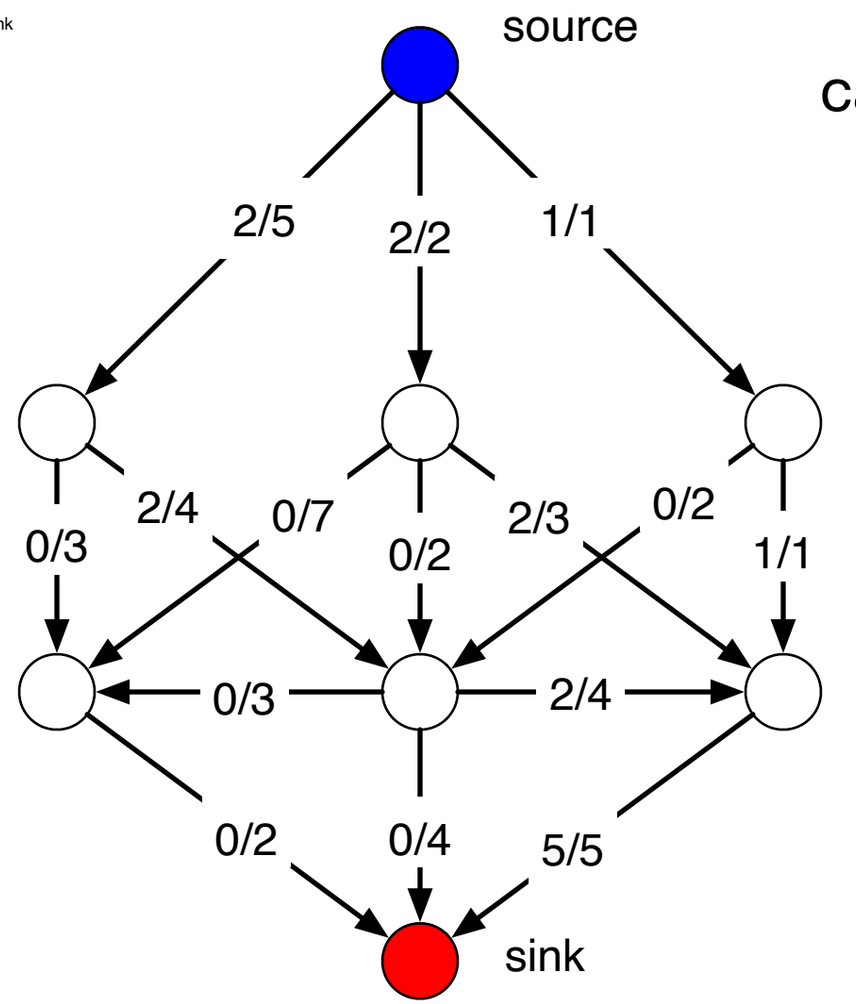


Augment the flow

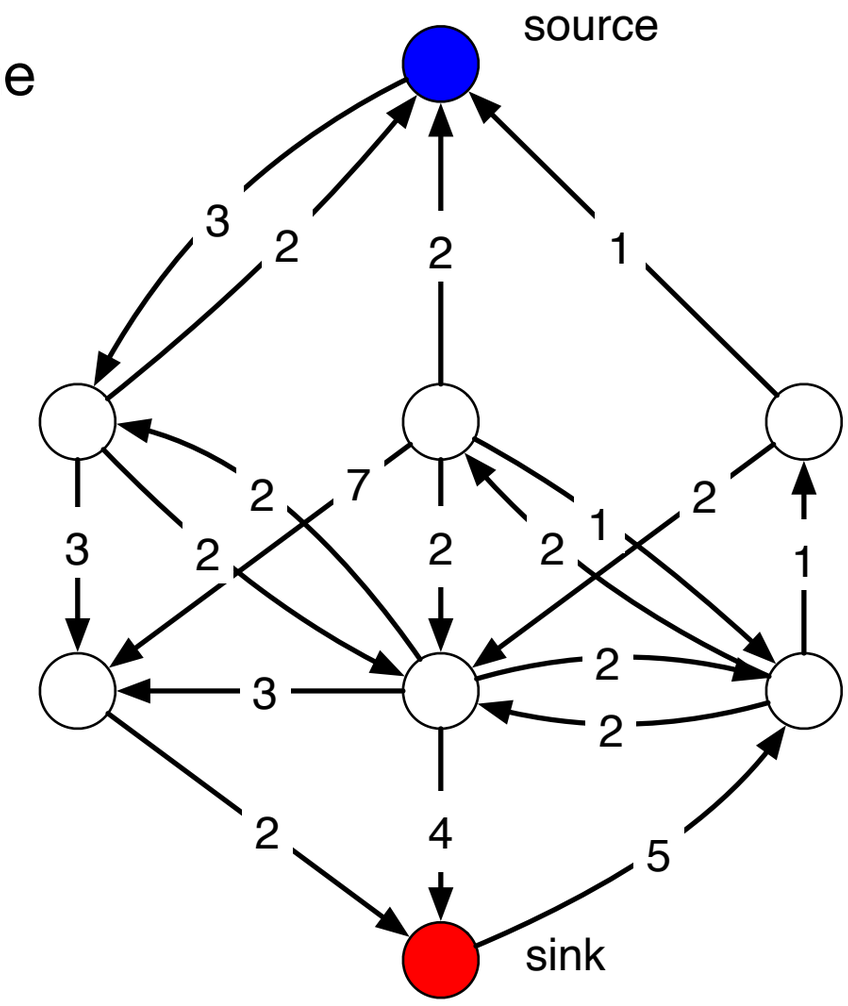


Ford Fulkerson Example

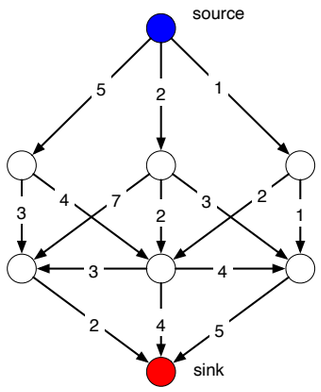




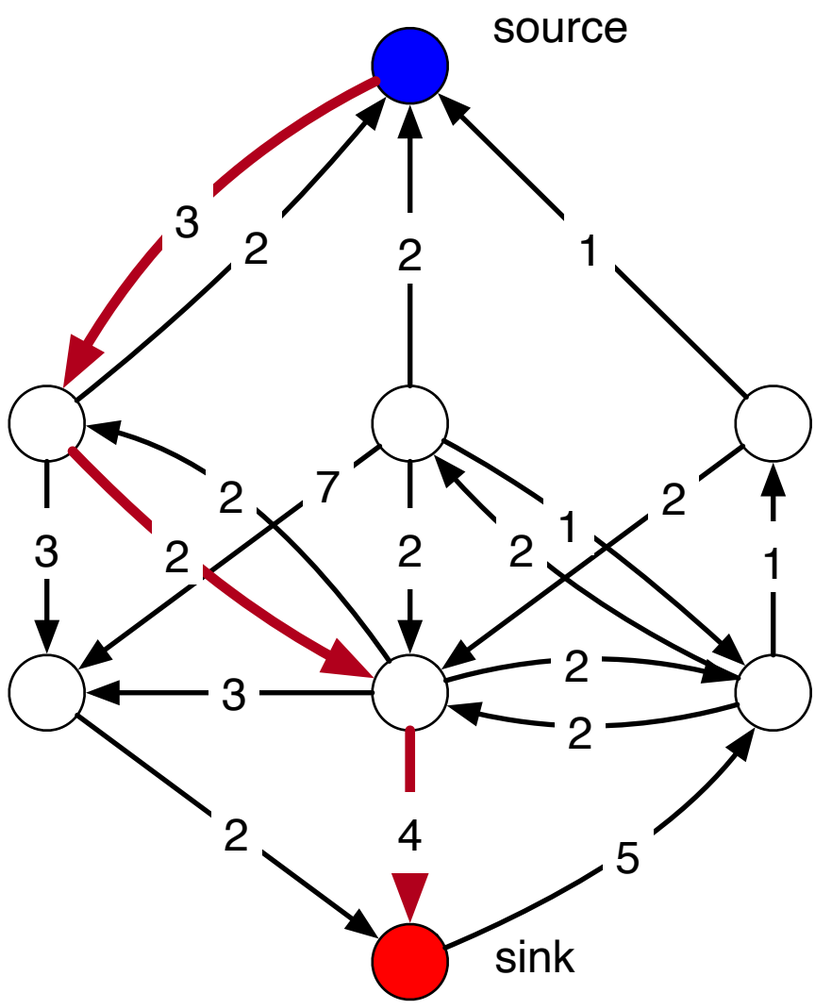
calculate the residual



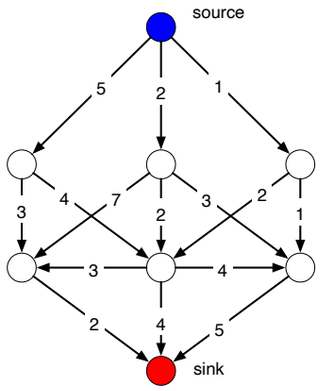
Ford Fulkerson Example

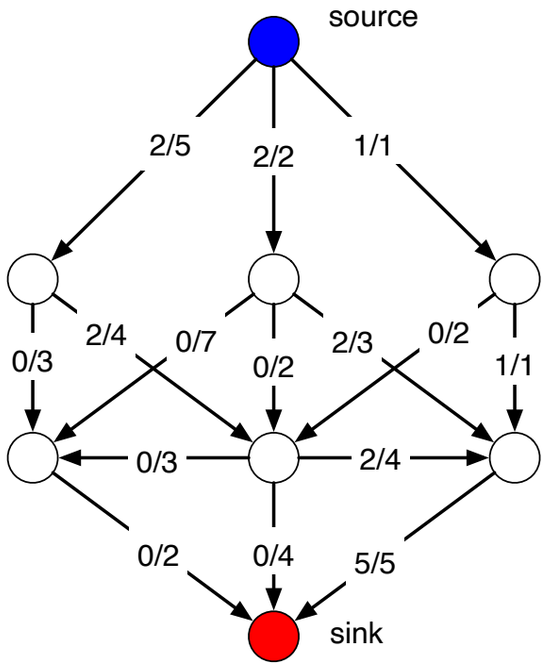


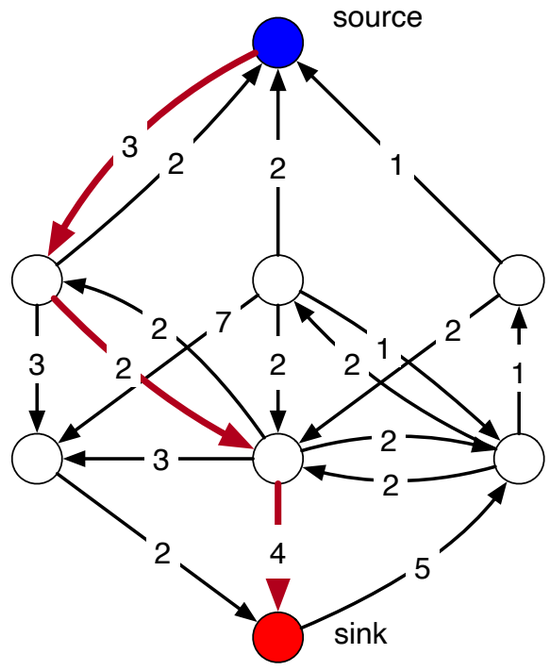
Find a path in the residual
from source to sink



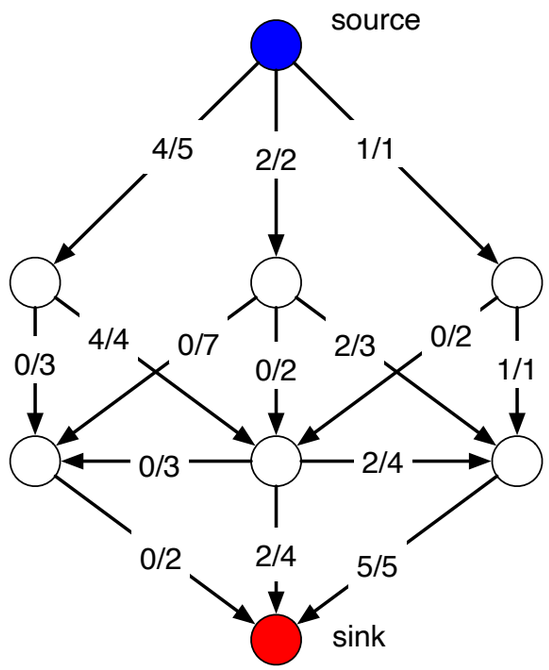
Ford Fulkerson Example



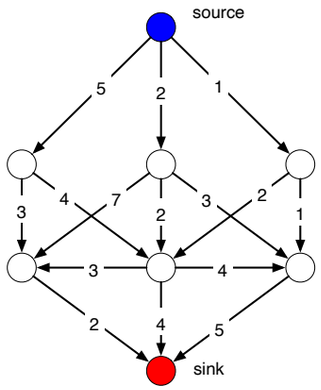


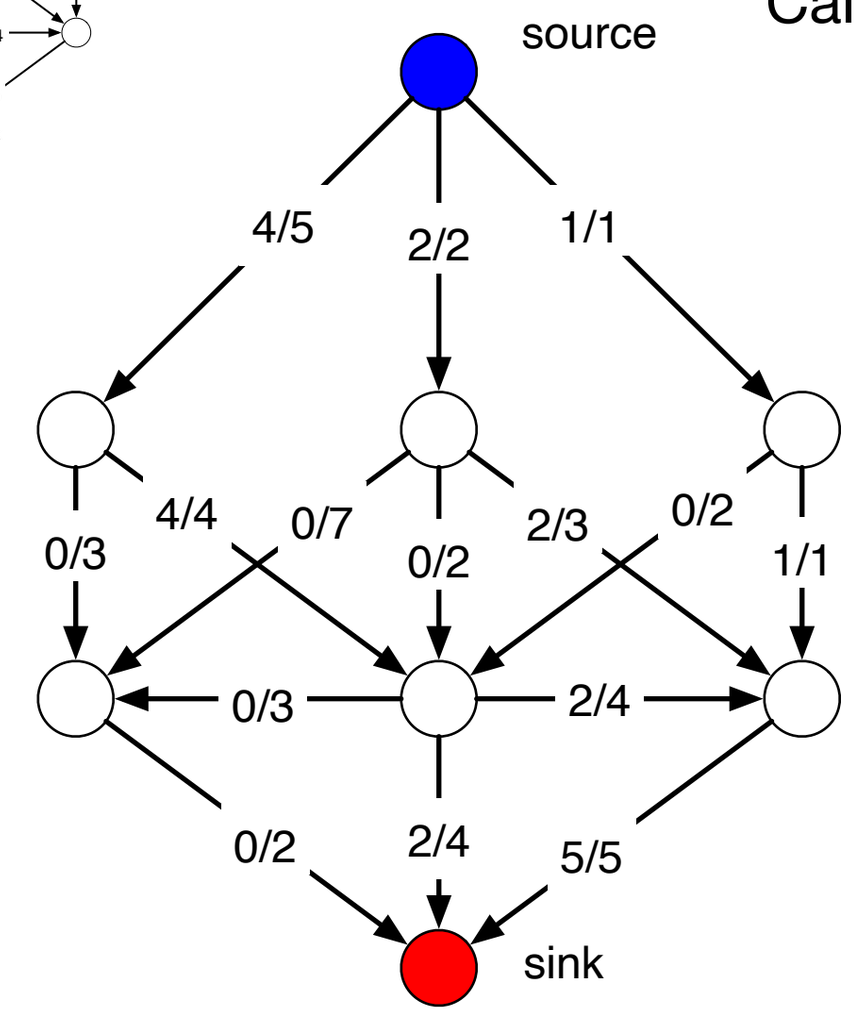


Augment the current flow along the path

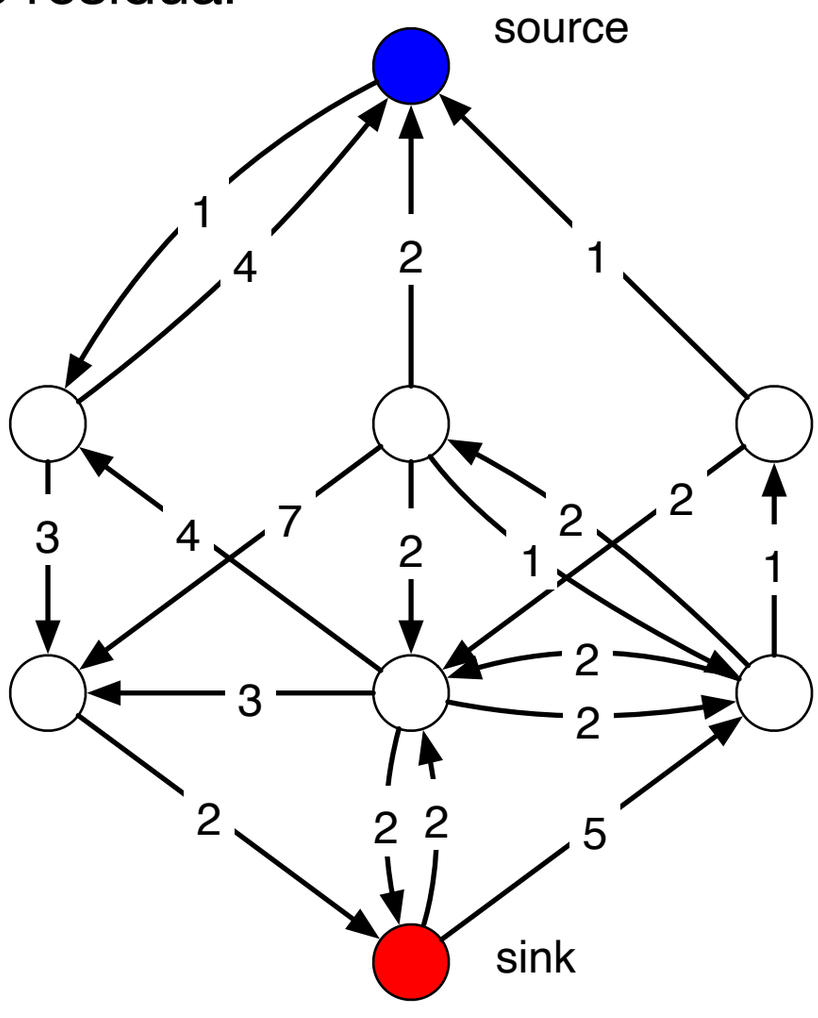


Ford Fulkerson Example

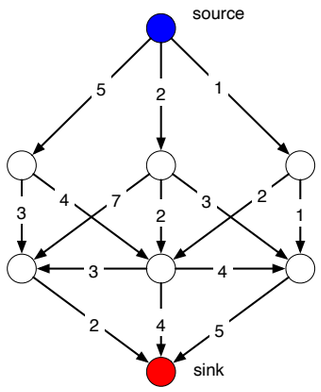




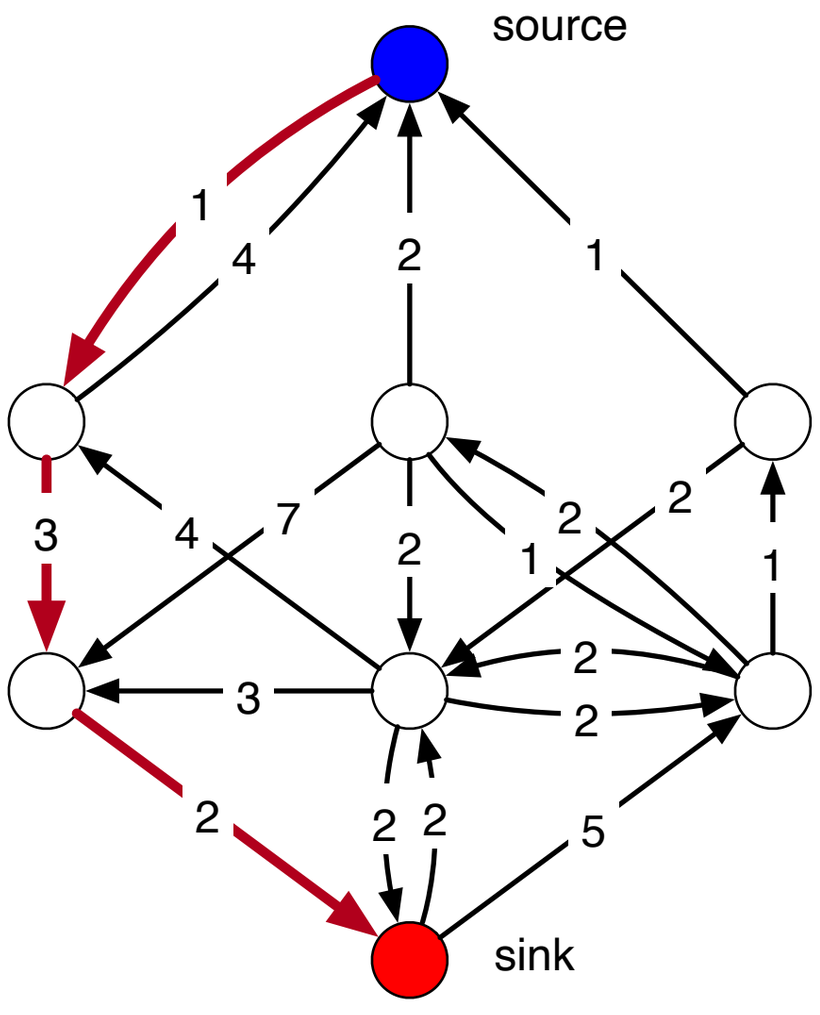
Calculate the residual



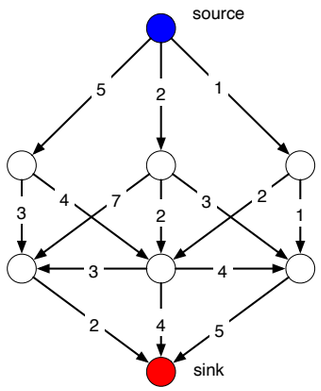
Ford Fulkerson Example

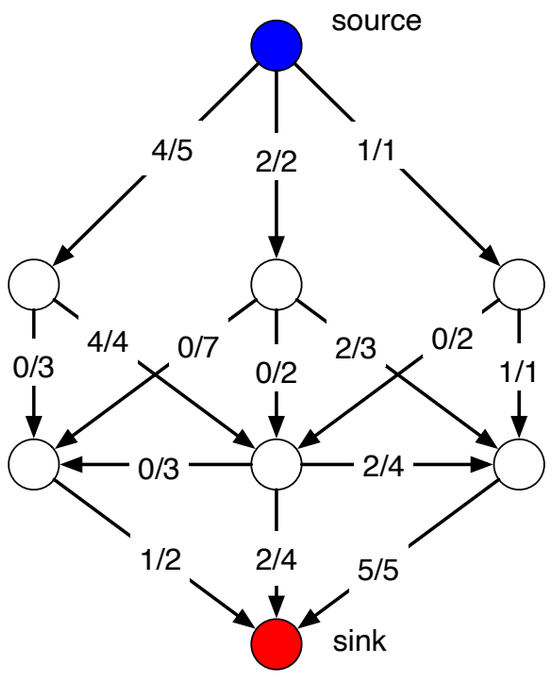


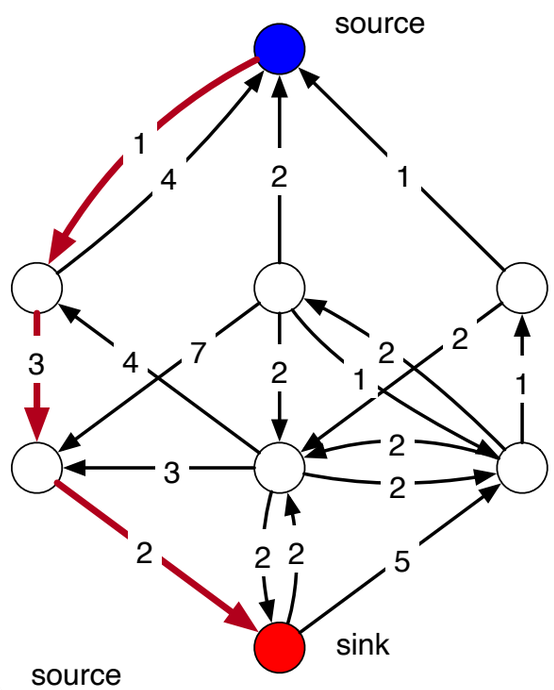
Find a path from source to sink



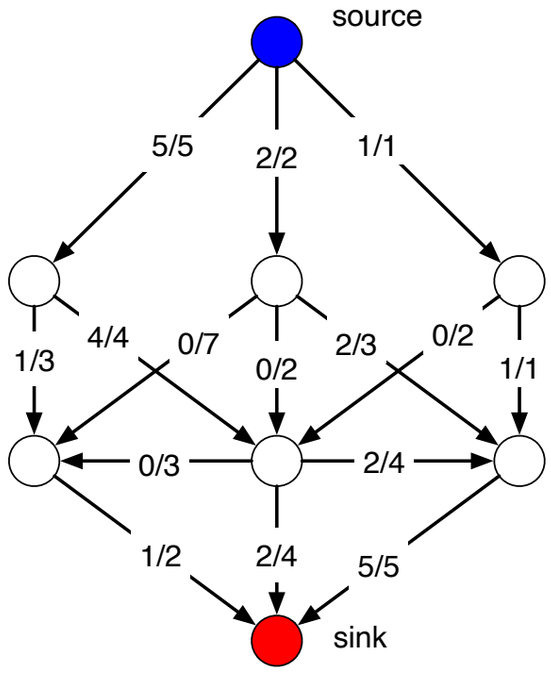
Ford Fulkerson Example



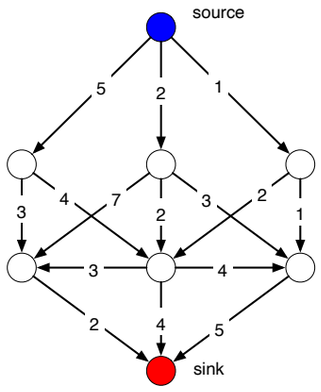




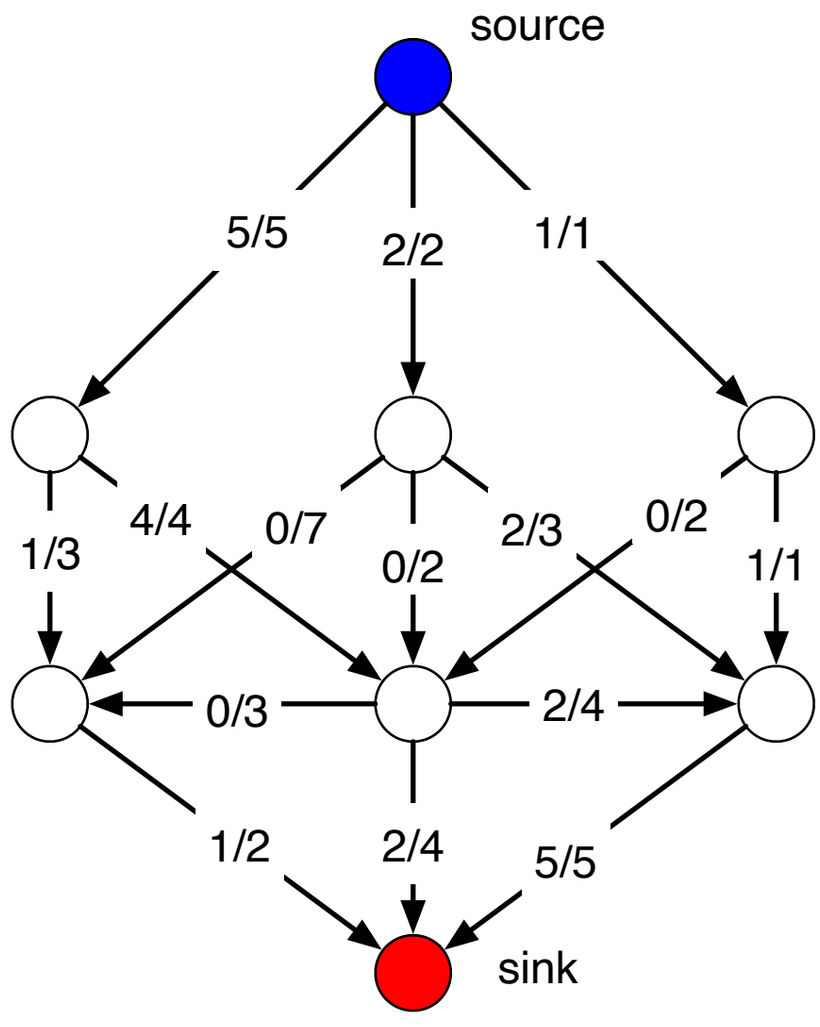
Augment the current flow

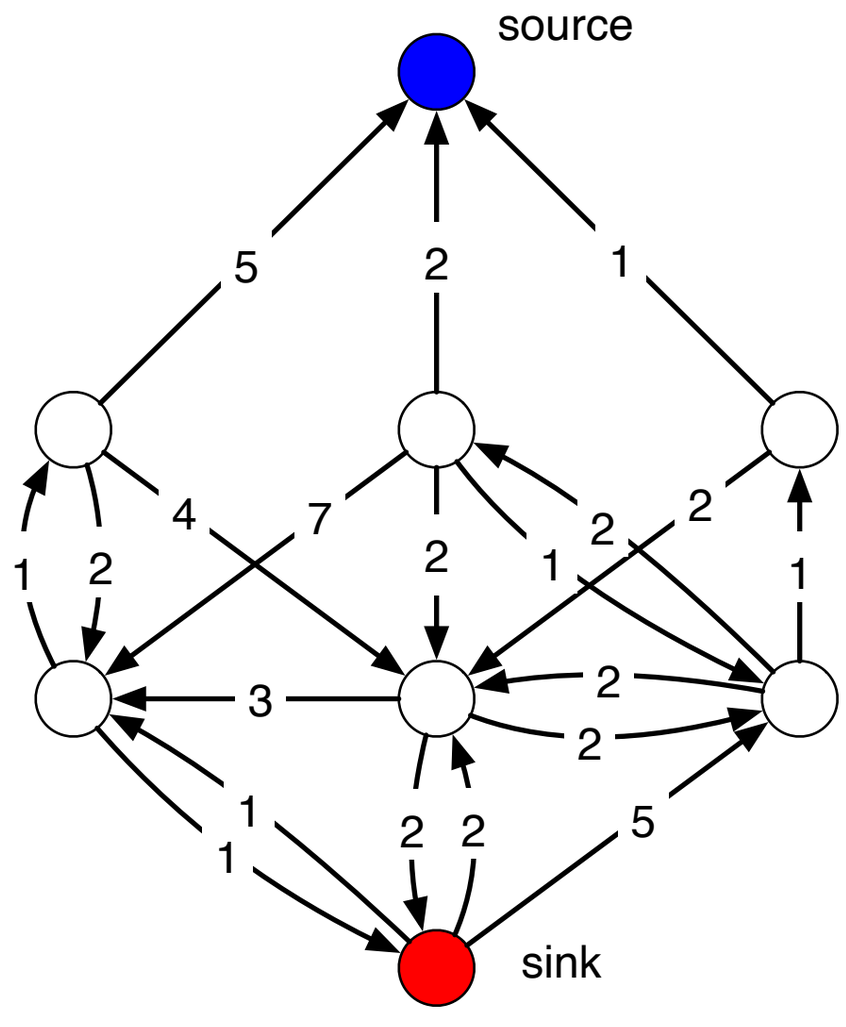


Ford Fulkerson Example

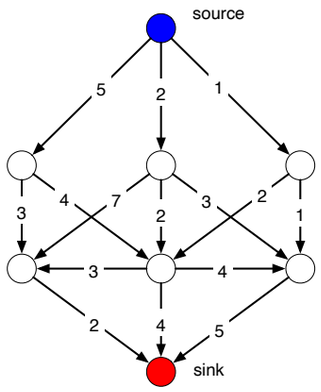


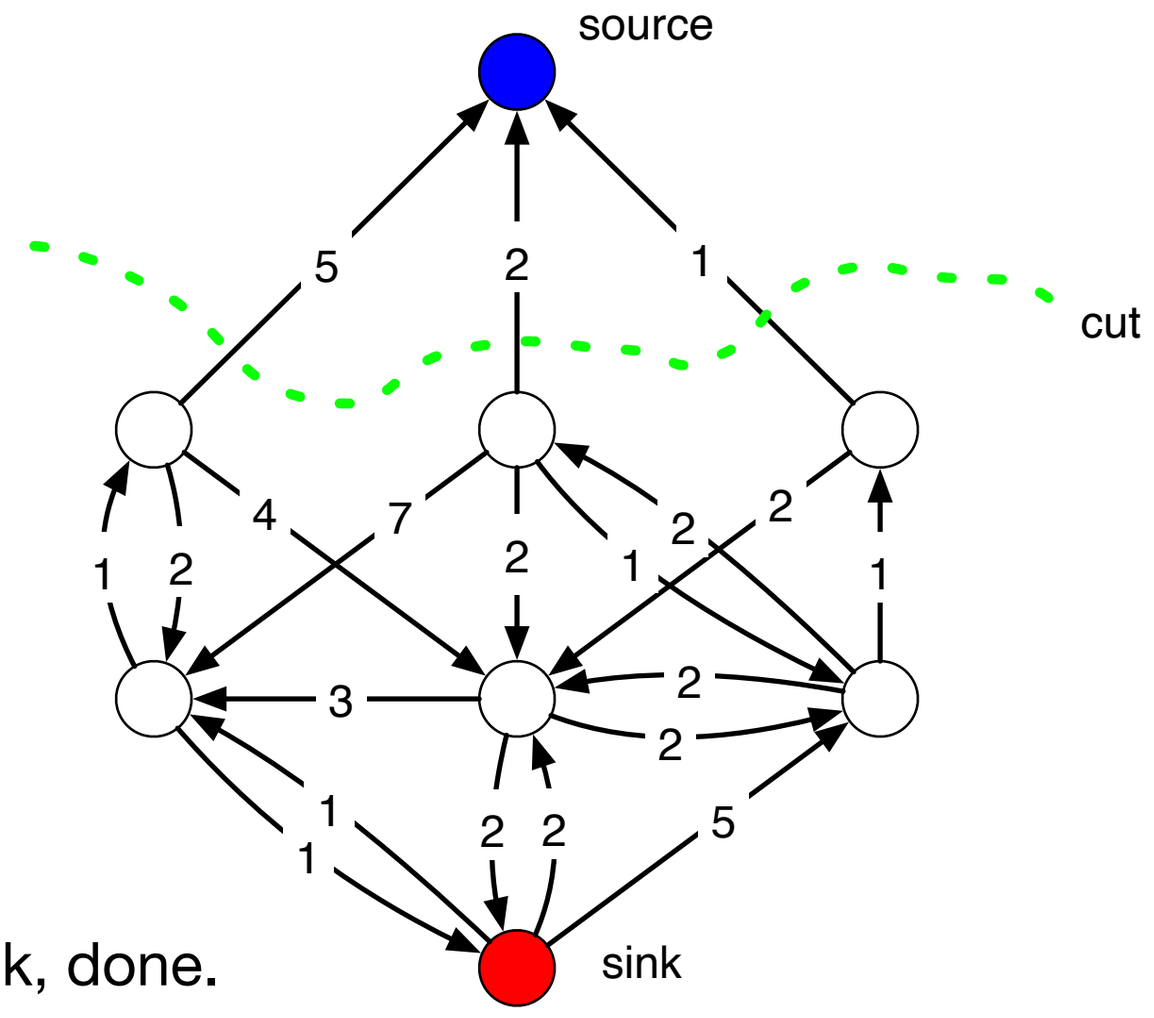
Calculate the residual





Ford Fulkerson Example





No path from source to sink, done.

Edmonds Karp Algorithm

- Use Ford Fulkerson
  - Determine an augmenting path use BFS
    - Favoring shortest number of edges in a path

- Can show: total number of flow augmentations is $O(|V| \cdot |E|)$