Lecture 20

Max-Flow Problem
and Augmenting Path Algorithm

October 28, 2009

Outline

- Max-Flow problem
  - Description
  - Some examples
- Algorithm for solving max-flow problem
  - Augmenting path algorithm

Max-Flow Problem: Single-Source Single-Sink

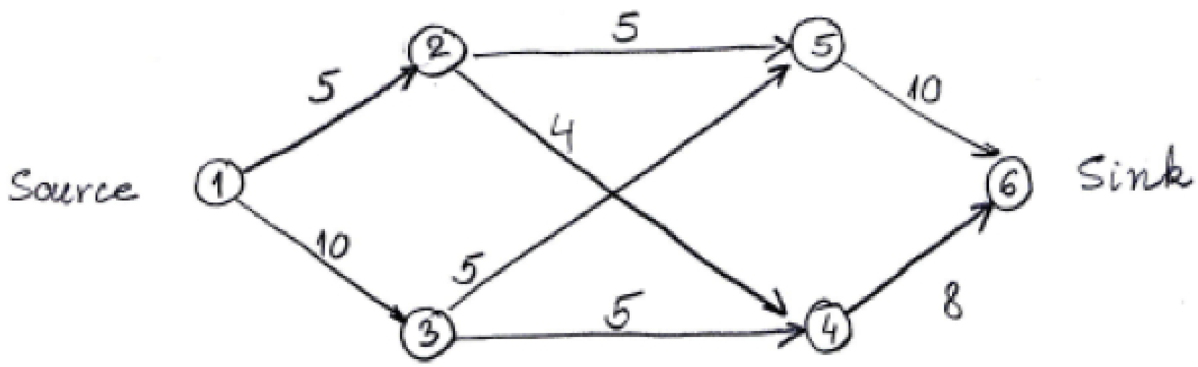
We are given a directed capacitated network (V, E, C) connecting a source (origin) node with a sink (destination) node.

The set V is the set of nodes in the network.

The set E is the set of directed links (i, j)

The set C is the set of capacities $c_{ij} \geq 0$ of the links $(i, j) \in E$.

The problem is to determine the **maximum amount of flow** that can be sent from the source node to the sink node.



This is **Max-Flow Problem** for single-source and single-sink

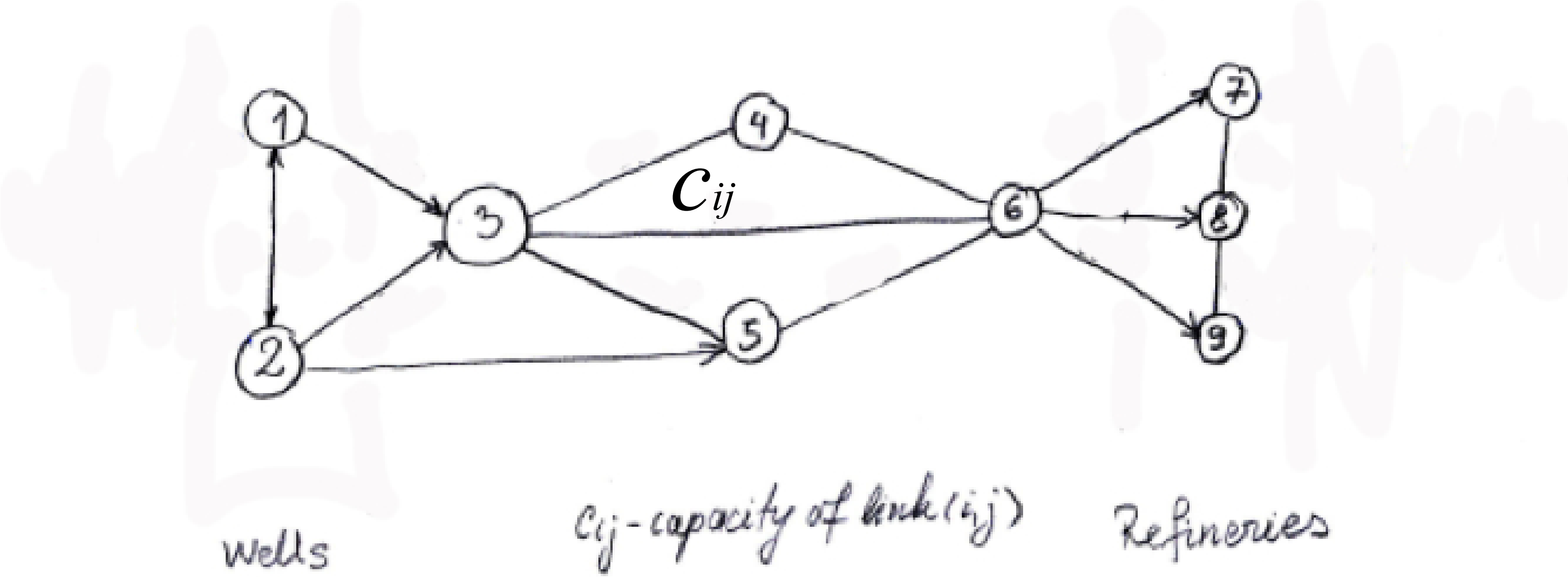
Max-Flow Problem: Multiple-Sources Multiple-Sinks

We are given a directed capacitated network (V, E, C) connecting multiple source nodes with multiple sink nodes.

The set V is the set of nodes and the set E is the set of directed links (i, j)

The set C is the set of capacities $c_{ij} \geq 0$ of the links $(i, j) \in E$

The problem is to determine the **maximum amount of flow** that can be sent from the source nodes to the sink nodes.

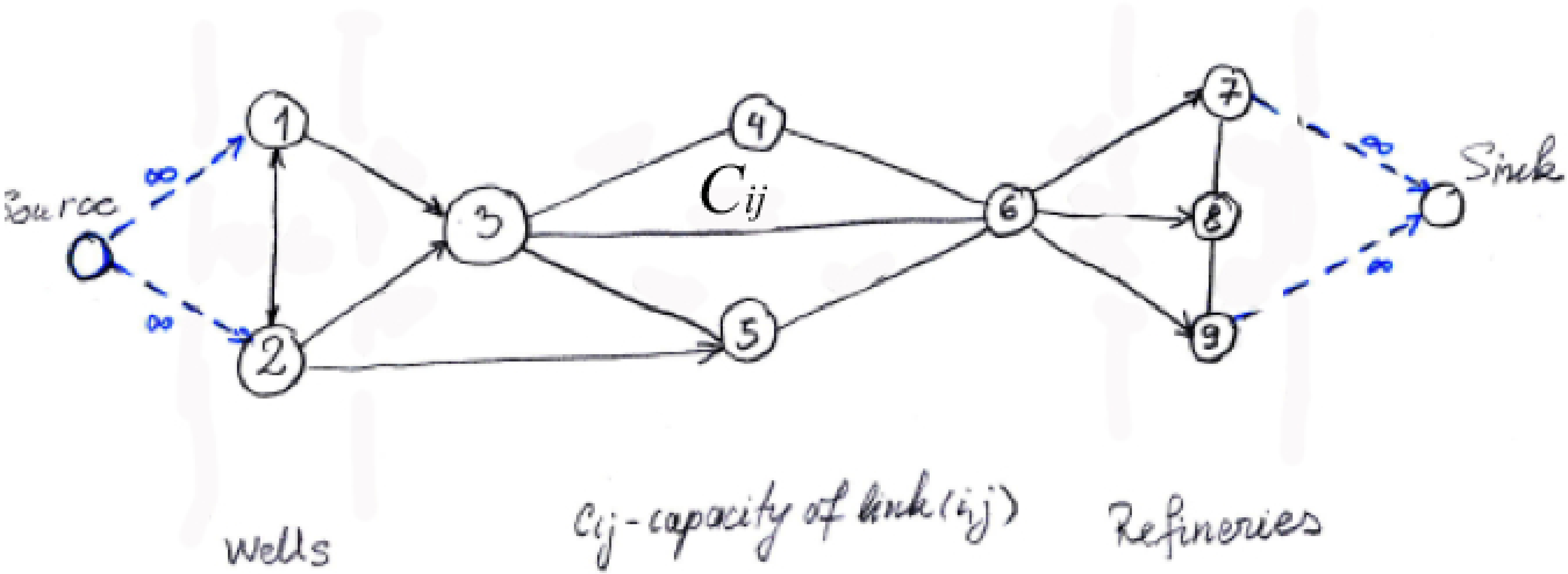


This is **Max-Flow Problem** for multiple-sources and multiple-sinks

Multiple-sources multiple-links problem can be converted to a single-source and single-sink problem by

- Introducing a dummy source node that is connected to the original source nodes with infinite capacity links

- Introducing a dummy sink node that is connected with the original sink nodes with infinite capacity links

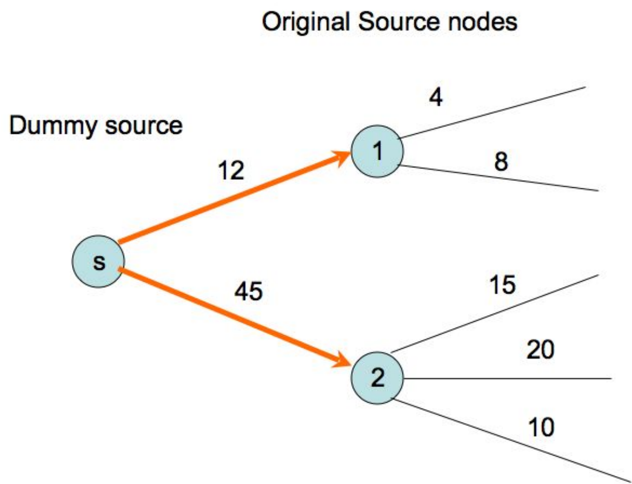


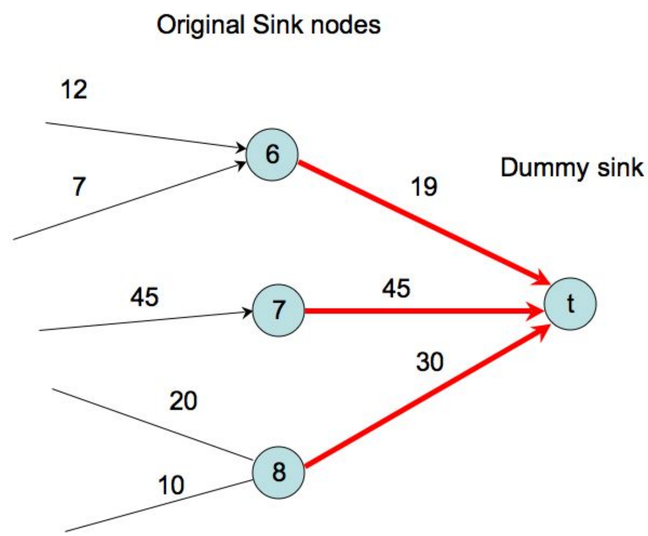
For small scale problems:
Another alternative is to introduce a single "dummy" source node connected with all the original source nodes BUT

- Each outgoing link from the dummy source node to an original source node s gets assigned a capacity that is equal to the total capacity of the outgoing links from s

Similarly, if there are multiple sink nodes, we introduce a single dummy sink node BUT

- Each incoming link from an original sink node t to the dummy sink node gets assigned a capacity that is equal to the total capacity of the incoming links to sink t


Original Source nodes
Dummy source
s
1
2
12
45
4
8
15
20
10


Original Sink nodes
Dummy sink
6
7
8
t
12
7
45
20
10
19
45
30

Maximum Flow Problem: Mathematical Formulation

We are given a directed capacitated network $G = (V, E, C))$ with a single source and a single sink node. We want to formulate the max-flow problem.

- For each link $(i, j) \in E$, let x_{ij} denote the flow sent on link (i, j),
- For each link $(i, j) \in E$, the flow is bounded from above by the capacity c_{ij} of the link: $c_{ij} \geq x_{ij} \geq 0$
- We have to specify the balance equations
  - All the nodes in the network except for the source and the sink node are just "transit" nodes (inflow=outflow)

$$\sum_{\{\ell|(\ell,i)\in E\}} x_{\ell i} - \sum_{\{j|(i,j)\in E\}} x_{ij} = 0 \quad \text{for all } i \neq s, t$$

- The objective is to maximize the outflow from the source node s

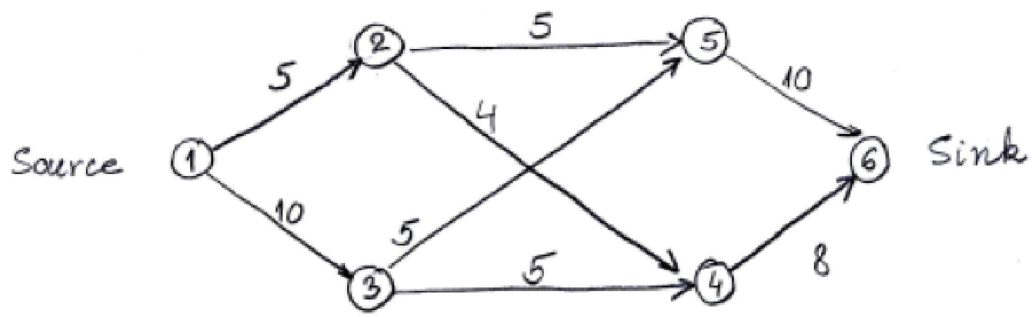
$$\sum_{\{j|(s,j)\in E\}} x_{sj}$$

- Alternatively: to maximize the inflow to the sink node t

$$\sum_{\{\ell|(\ell,t)\in E\}} x_{\ell t}$$

MAX-FLOW FORMULATION

$$\begin{aligned} \text{maximize} \quad & \sum_{\{j:(s,j)\in E\}} x_{sj} \\ \text{subject to} \quad & \sum_{\{\ell|(\ell,i)\in E\}} x_{\ell i} - \sum_{\{j|(i,j)\in E\}} x_{ij} = 0 \quad \text{for all } i \neq s,t \\ & 0 \leq x_{ij} \leq c_{ij} \quad \text{for all } (i,j) \in E. \end{aligned}$$



$$
\begin{array}{ll}
\text{maximize} & x_{12} + x_{13} \\
\\
\text{subject to} & x_{12} - x_{25} - x_{24} = 0 \quad \text{balance for node 2} \\
& x_{13} - x_{35} - x_{34} = 0 \quad \text{balance for node 3} \\
& x_{24} + x_{34} - x_{46} = 0 \quad \text{balance for node 4} \\
& x_{25} + x_{35} - x_{56} = 0 \quad \text{balance for node 5} \\
& 0 \le x_{12} \le 5 \qquad 0 \le x_{13} \le 10 \\
& 0 \le x_{24} \le 4 \qquad 0 \le x_{25} \le 5 \\
& 0 \le x_{34} \le 5 \qquad 0 \le x_{35} \le 5 \\
& 0 \le x_{46} \le 8 \qquad 0 \le x_{56} \le 10
\end{array}
$$

Max-Flow is an LP problem: we could use a simplex method

Max-Flow Algorithms

For a given graph:
n denotes the number of nodes
m denotes the number of edges
$\max |f|$ is the maximum amount of flow

Method	Complexity	Description
Simplex	–	Constrained by legal flow
Ford-Fulkerson algorithm	$O(m \max \lvert f \rvert)$	Weights have to be integers
Edmonds-Karp algorithm	$O(nm^2)$	Based on Ford-Fulkerson
Dinitz blocking flow algorithm	$O(n^2m)$	Builds layered graphs
General push-relabel algorithm	$O(n^2m)$	Uses a preflow

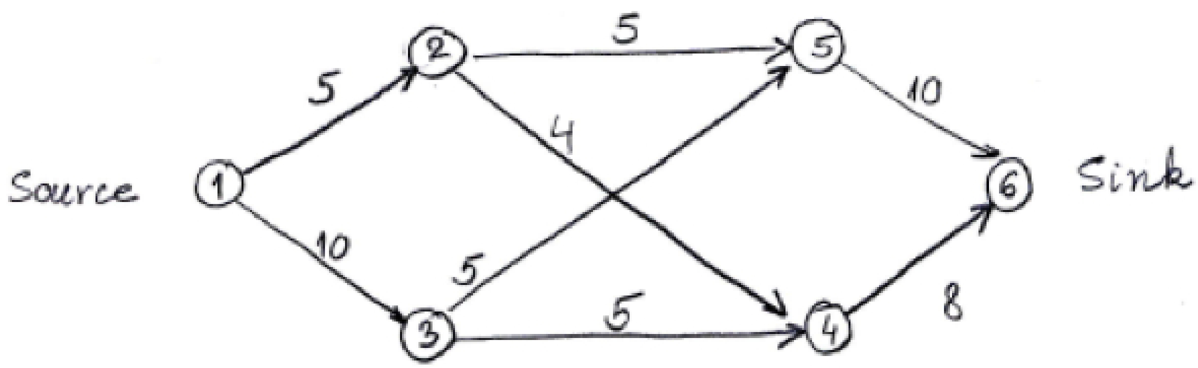
Ford-Fulkerson Algorithm is also known as Augmenting Path algorithm
We will also refer to it as Max-Flow Algorithm

Max-Flow Algorithm

This is an iterative method (operates in stages)

- At each iteration, the algorithm is searching for a path from the source node to the sink node along which it can send a positive flow
  - Such a path is referred to as **augmenting path**
- After a flow is sent along an augmenting path the capacities of the links on that path are adjusted
  - These adjusted capacities are referred to as **residual capacities**
  - Correspondingly, the resulting network is referred to as **residual network**
- The algorithm **terminates** when an augmenting path cannot be found

Example: Augmenting path



Suppose we choose to send the flow along path $1 \to 3 \to 5 \to 6$
Suppose we choose to send 4 units of flow along this path
Then the residual capacities of links (1,2), (3,5) and (5,6)
are 6, 1, and 6 respectively

A better choice is to send 5 units along this path. In this case, the capacity of the link (3,5) is reached (its residual capacity becomes 0)

We say that link (3,5) is saturated (no more flow can be sent)

The flow on a path that saturates at least one of its links is the **capacity of the path**

In other words, the capacity of a path is the maximum possible amount that we can send along the path

The capacity $C(p)$ of the path p is given by

$$C(p) = \min_{(i,j)\in p} u_{ij}$$

where u_{ij} is the capacity of the link (i,j)

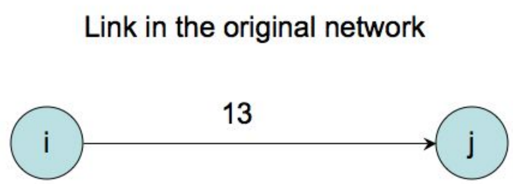
If the path p under consideration is in the residual network then the links $(i,j) \in p$ are the links in the **residual network** and the capacities are **residual capacities**

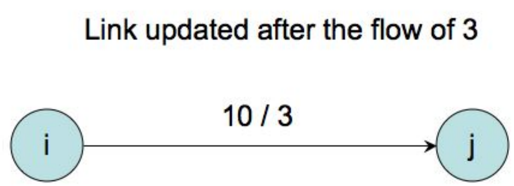
Residual Capacity

The links that have been used to send a flow get updated to reflect the flow push

Every such link (i, j) gets a capacity label of the form a/b where

- a is the remaining capacity of the link and
- b is the total flow sent along that link
- a is viewed as forward capacity of the link
- b is viewed as backward capacity of the link (capacity if we choose to traverse the link in the opposite direction)
- NOTE: $a + b = c_{ij}$ The sum of these numbers is equal to the original capacity of the link



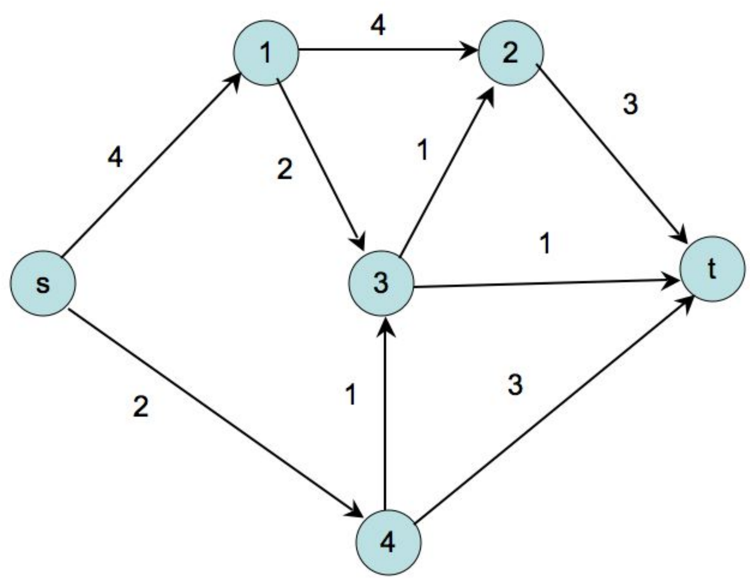


- Link with residual capacity a/b with $b > 0$ can be traversed backward
- Traveling backward means that we are removing the flow from the link

For example the link (i, j) with the capacity $10/3$ can be used backward to send the flow of at most 3 units

If we send 2 units backward [along (j, i)], the resulting residual capacity of the link is (12,1)

Example: Algorithm Application

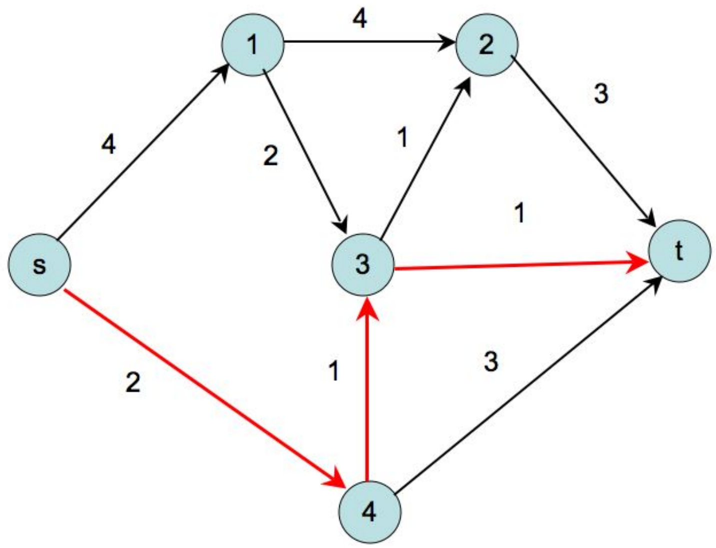


We will use f_i to indicate the amount of flow sent in iteration i

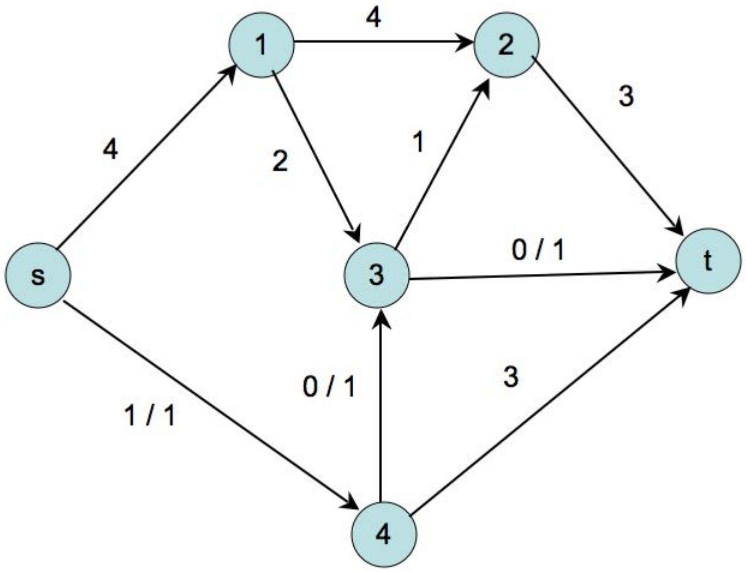
We will apply the algorithm that sends the maximum possible amount of flow at each iteration, i.e., the flow equal to the capacity of the path under consideration

Iteration 1

- We find a path $s \to 4 \to 3 \to t$ that can carry a positive flow
- The maximum flow we can send along this path is $f_1 = \min\{2, 1, 1\}$.
- We send $f_1 = 1$ unit of flow along this path

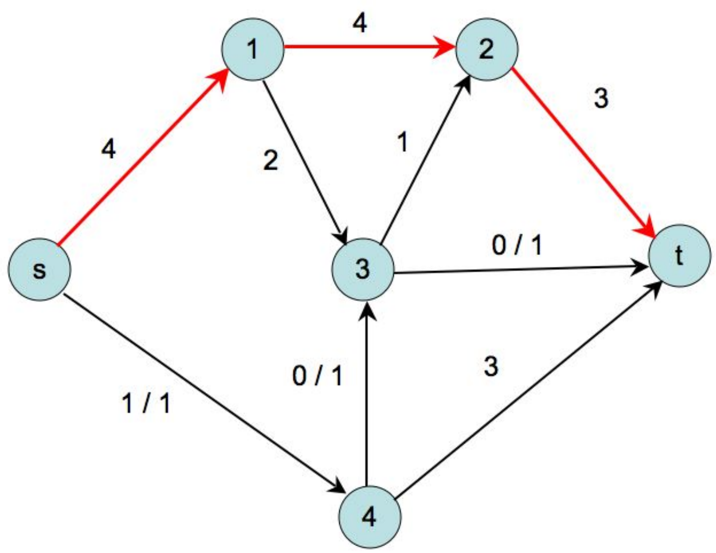


- We obtain a residual network with updated link capacities resulting from pushing the flow along the path

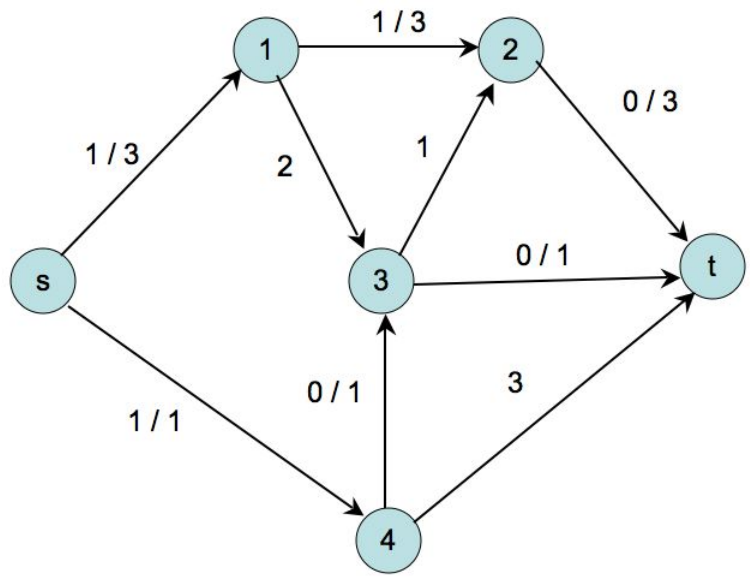


Iteration 2

- We find a path $s \to 1 \to 2 \to t$ that can carry a positive flow
- The maximum flow we can send along this path is $f_2 = \min\{4, 4, 3\}$.
- We send $f_2 = 3$ units of flow along this path

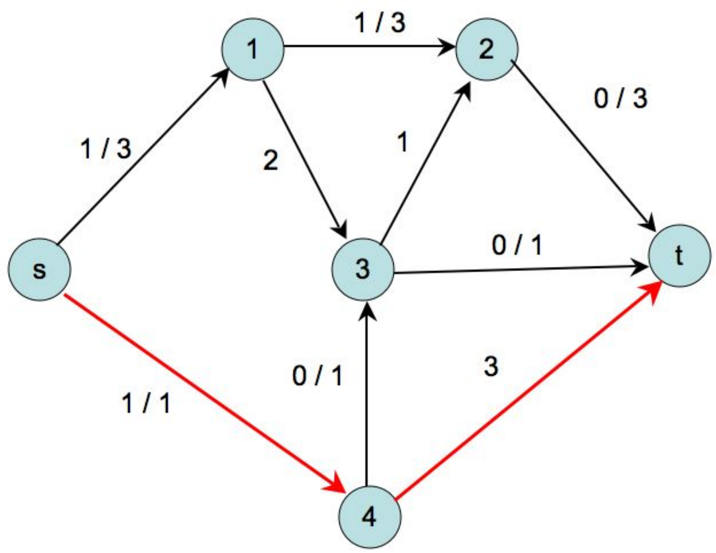


- We obtain a residual network with updated link capacities resulting from pushing the flow along this path

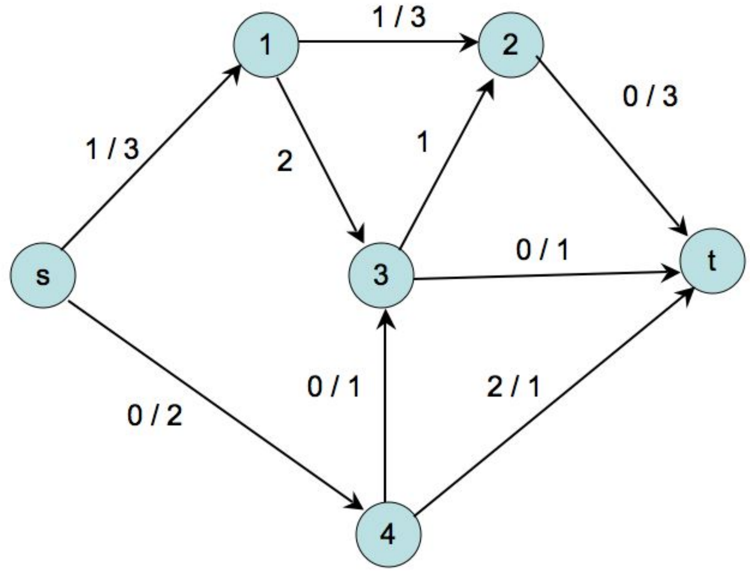


Iteration 3

- We find a path $s \to 4 \to t$ that can carry a positive flow
- The maximum flow we can send along this path is $f_3 = \min\{1, 3\}$.
- We send $f_3 = 1$ unit of flow along this path

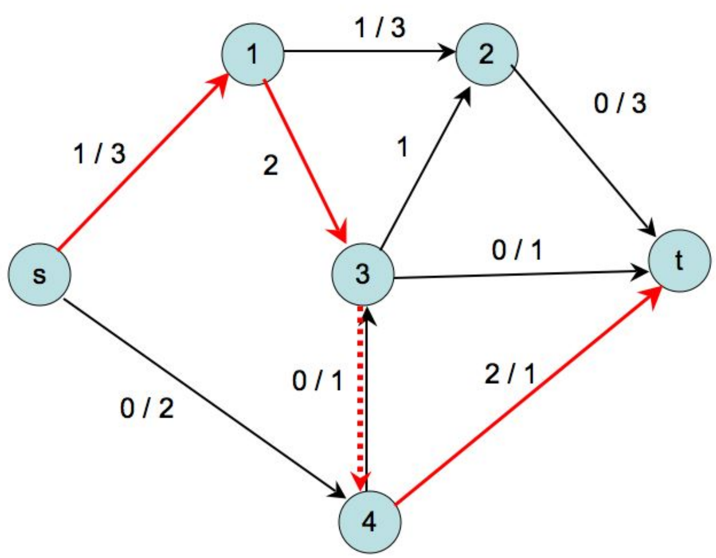


- We obtain a residual network with updated link capacities resulting from pushing the flow along this path

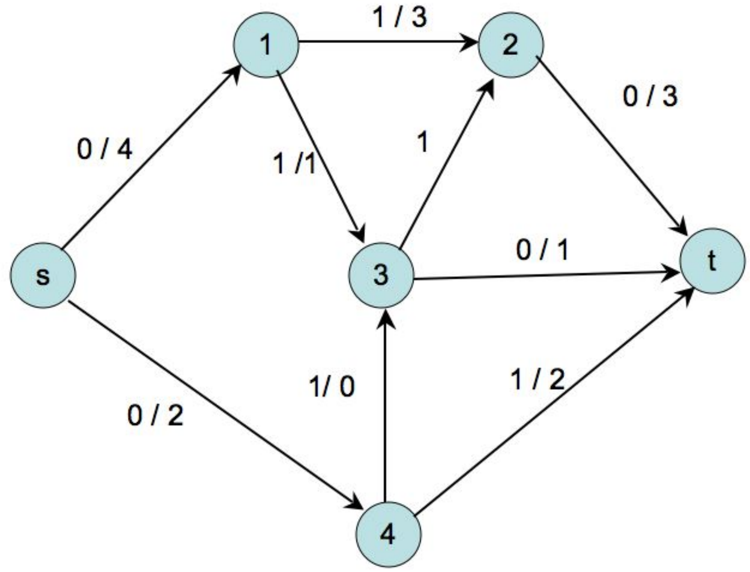


Iteration 4

- We find a path $s \to 1 \to 3 \to 4 \to t$ that can carry a positive flow
- The maximum flow we can send along this path is $f_4 = \min\{1, 2, 1, 2\}$.
- We send $f_4 = 1$ unit of flow along this path



- We obtain a residual network with updated link capacities resulting from pushing the flow along this path



At this point we are done.

The node 2 is disconnected from the rest of the nodes (forward capacity 0 on all outgoing links)

There are no more augmenting paths.

MAX-FLOW VALUE

The maximum flow is the total flow sent:

$$f_1 + f_2 + f_3 + f_4 = 1 + 3 + 1 + 1 = 6.$$