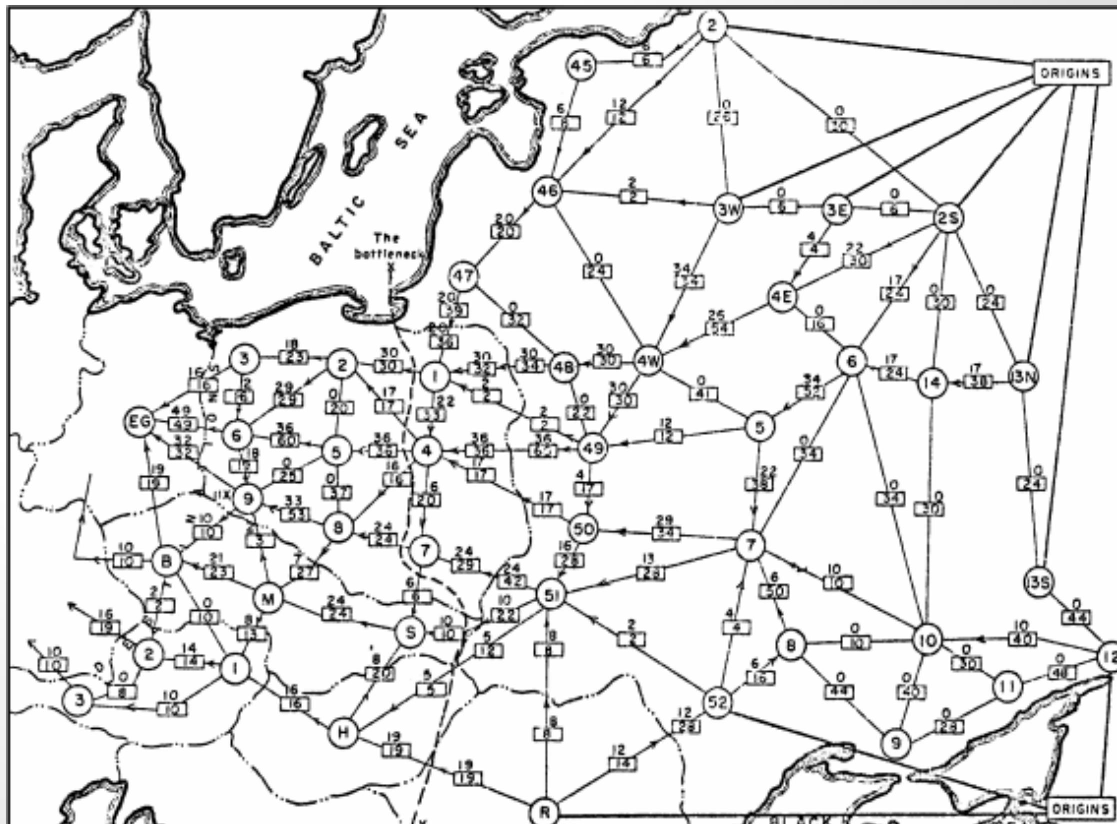
Max-Flow, Min-Cut

History and Concepts Behind the Max-Flow, Min-Cut Theorem in Graph Theory

Greg Eusden
April 6, 2013

Some Brief Context
(Why do we care?)

- **Scene**: The Cold War in the 1950's
- **Question**: What is the optimal way to disrupt the rail network between the Soviet Union and Eastern Germany?



[1] Image due to Harris and Ross (1954)

Agenda

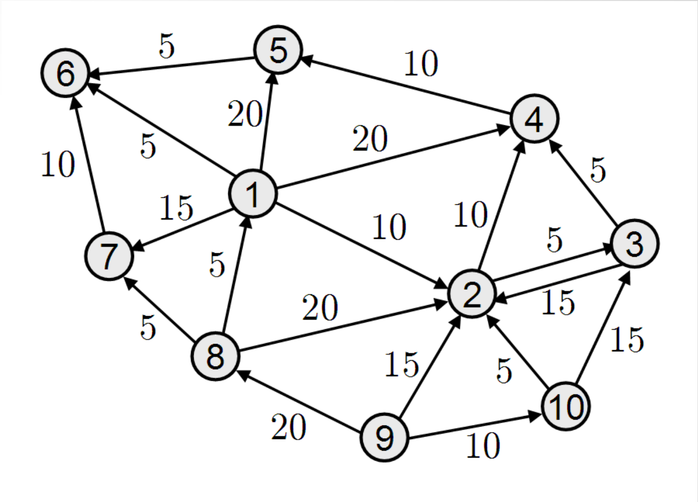
1. Basic Background
2. Cuts
3. Flows
4. The Max-Flow, Min-Cut Theorem
5. Conclusion

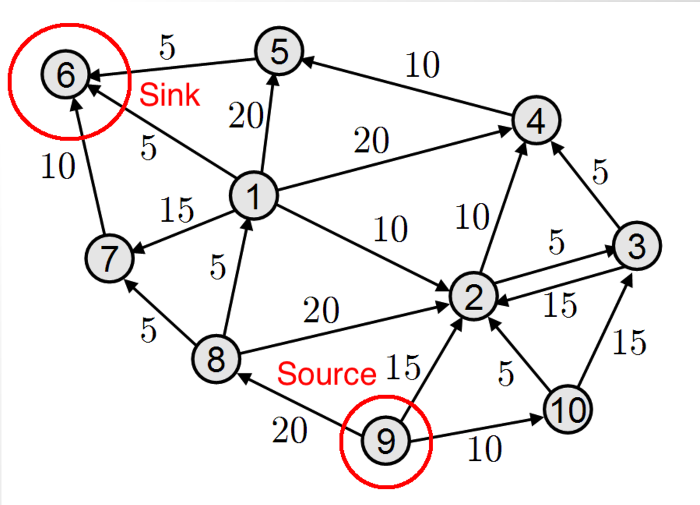
Basic Background

- A directed, weighted graph is called a (flow) **network**.

- Each edge has a weight and direction.

- We assume there exists a **source** and a **sink**.





Agenda

1. Basic Background
2. Cuts
3. Flows
4. The Max-Flow, Min-Cut Theorem
5. Conclusion

Cuts

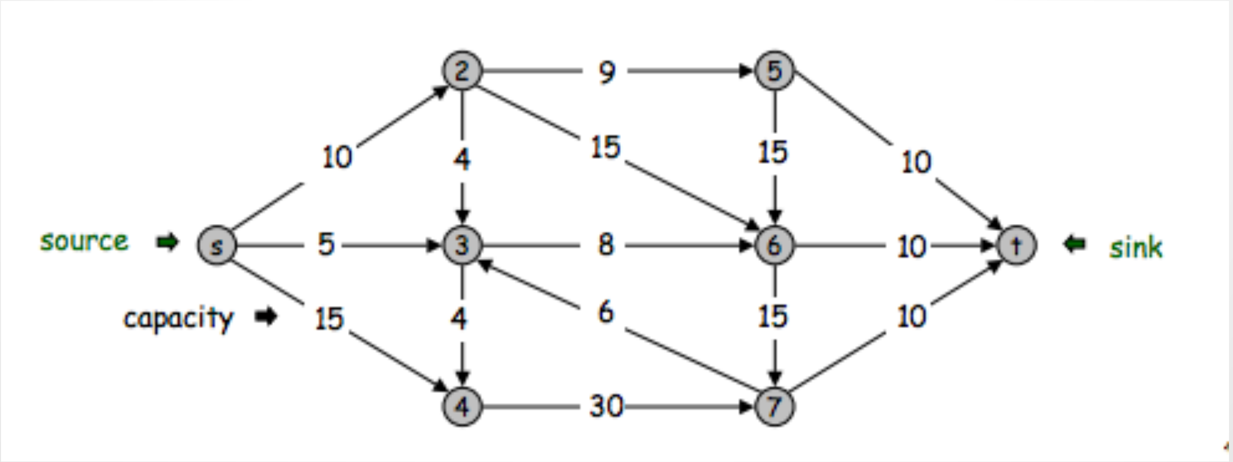
- A **cut** is a partition of the vertices into disjoint subsets S and T. In a flow network, the source is located in S, and the sink is located in T.

- The **cut-set** of a cut is the set of edges that begin in S and end in T.

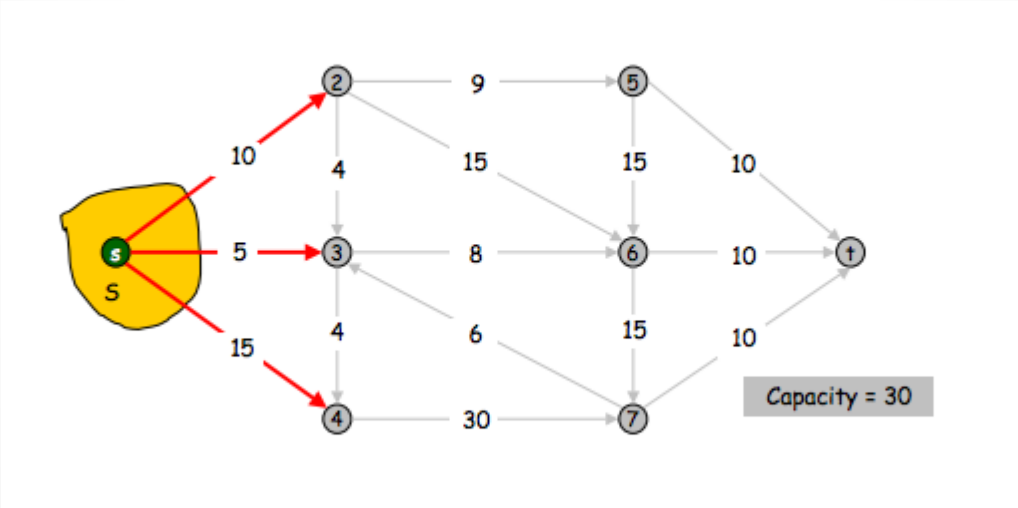
- The **capacity** of a cut is sum of the weights of the edges beginning in S and ending in T.

Cut Examples

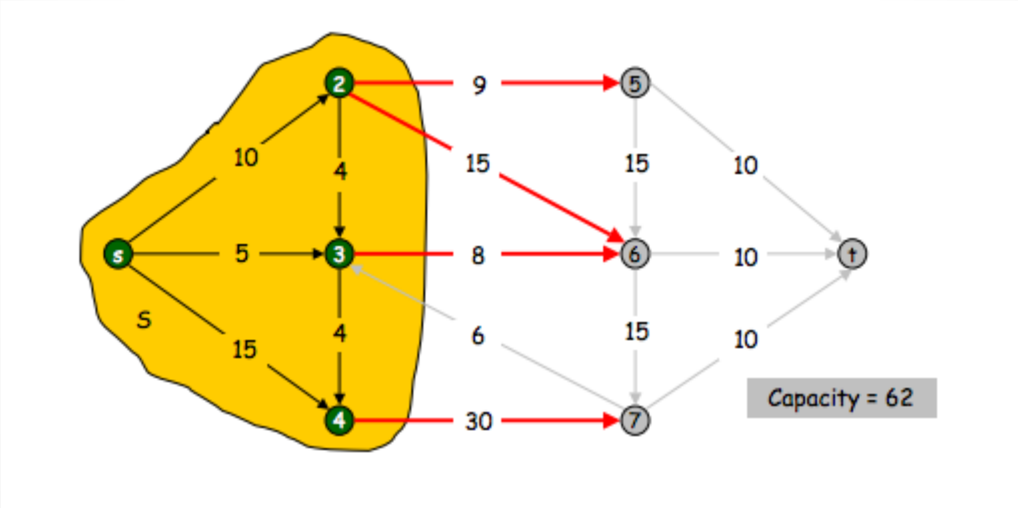
Consider a basic flow network.



Cut Examples (cont'd)



Cut Examples (cont'd)



Agenda

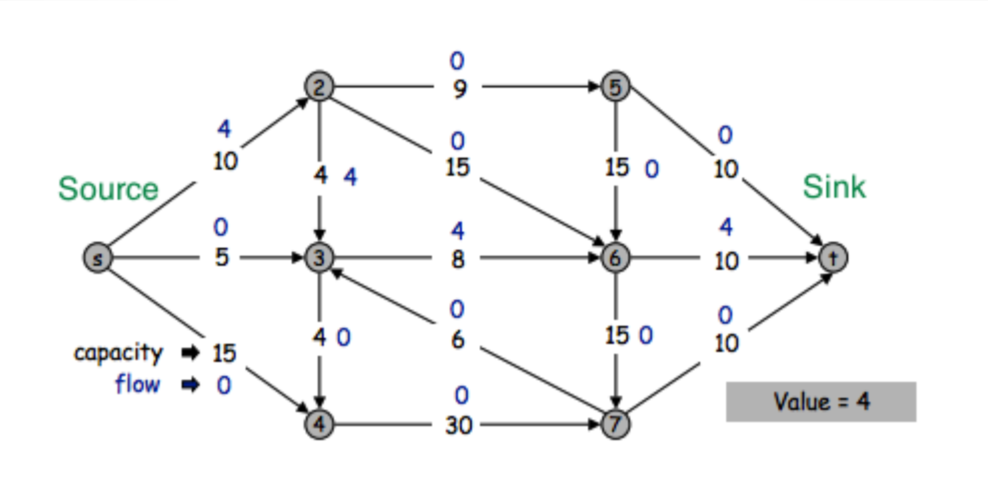
1. Basic Background
2. Cuts
3. Flows
4. The Max-Flow, Min-Cut Theorem
5. Conclusion

Flow

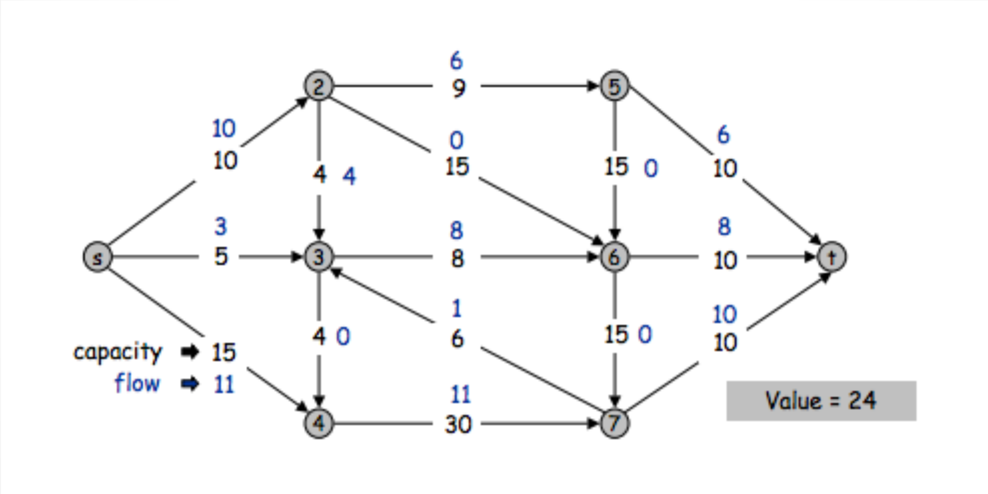
The **flow** over a network is a function f: E -> R, assigning values to each of the edges in the network which are nonnegative and less than the capacity of that edge. For each intermediate vertex, the outflow and inflow must be equal.

The value of this flow is the total amount leaving the source (and thus entering the sink).

Flow Example



Flow Example (cont'd)



Agenda

1. Basic Background
2. Cuts
3. Flows
4. The Max-Flow, Min-Cut Theorem
5. Conclusion

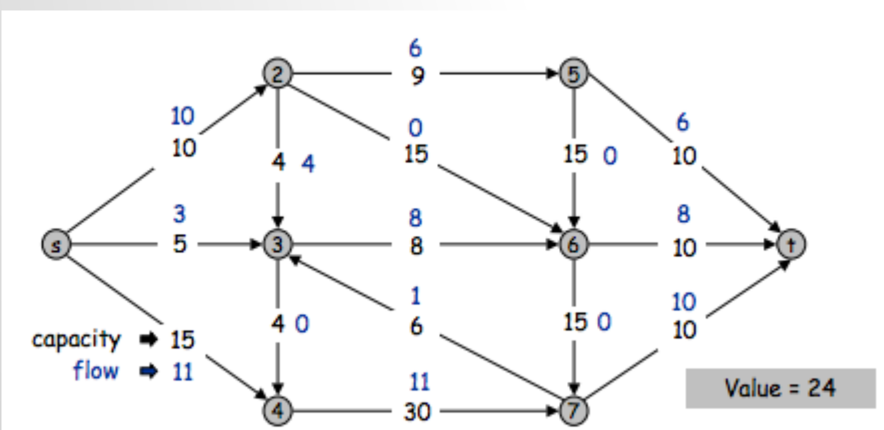
The Max-Flow, Min-Cut Theorem[1]

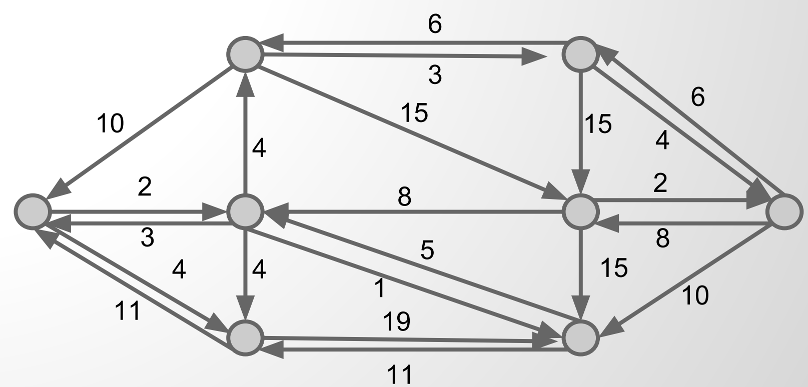
Theorem: For any network, the value of the maximum flow is equal to the capacity of the minimum cut.

[1] The **max-flow min-cut theorem** was proven by Ford and Fulkerson in 1954 for undirected graphs and 1955 for directed graphs.

Ford-Fulkerson Algorithm
Residual Graphs

$$c_f(u\to v) = \begin{cases} c(u\to v) - f(u\to v) & \text{if } u\to v \in E \\ f(v\to u) & \text{if } v\to u \in E\,. \\ 0 & \text{otherwise} \end{cases}$$





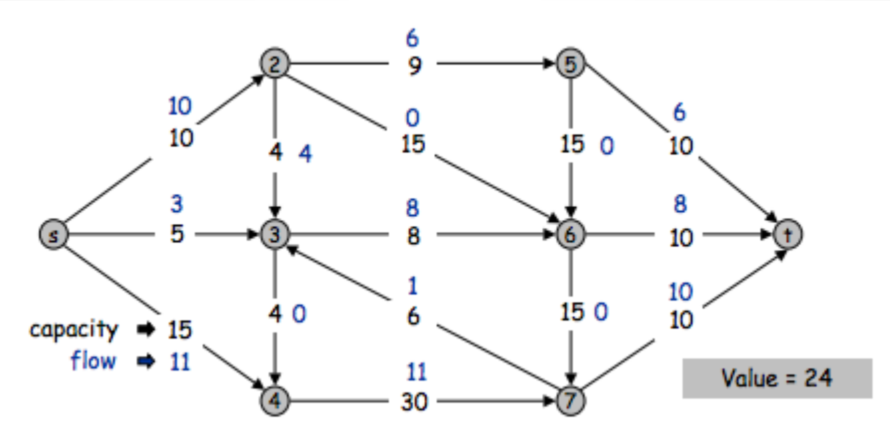
The Ford-Fulkerson Algorithm

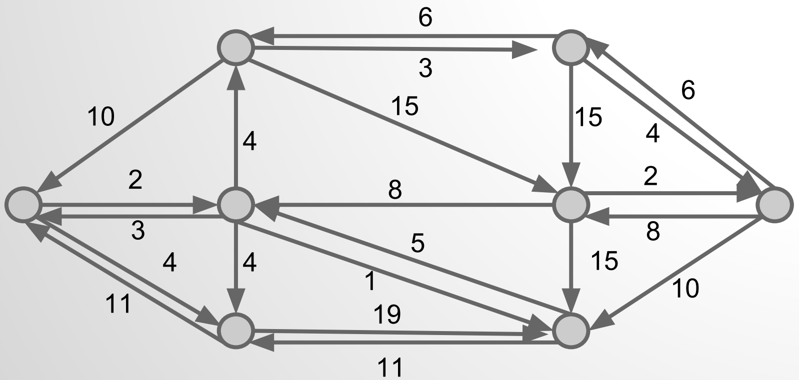
The Ford-Fulkerson algorithm for finding the maximum flow:

a. Construct the Residual Graph
b. Find a path from the source to the sink with strictly positive flow.
c. If this path exists, update flow to include it. Go to Step a.
d. Else, the flow is maximal.
e. The (s,t)-cut has as S all vertices reachable from the source, and T as V - S.

Ford-Fulkerson Algorithm

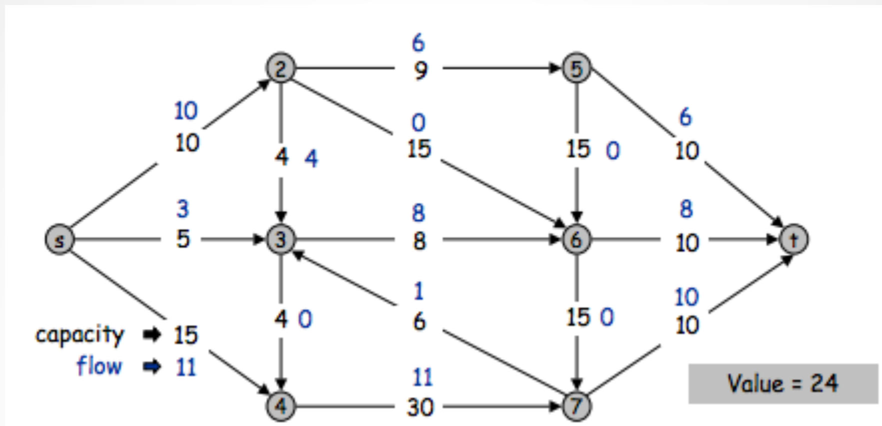
Residual Graphs

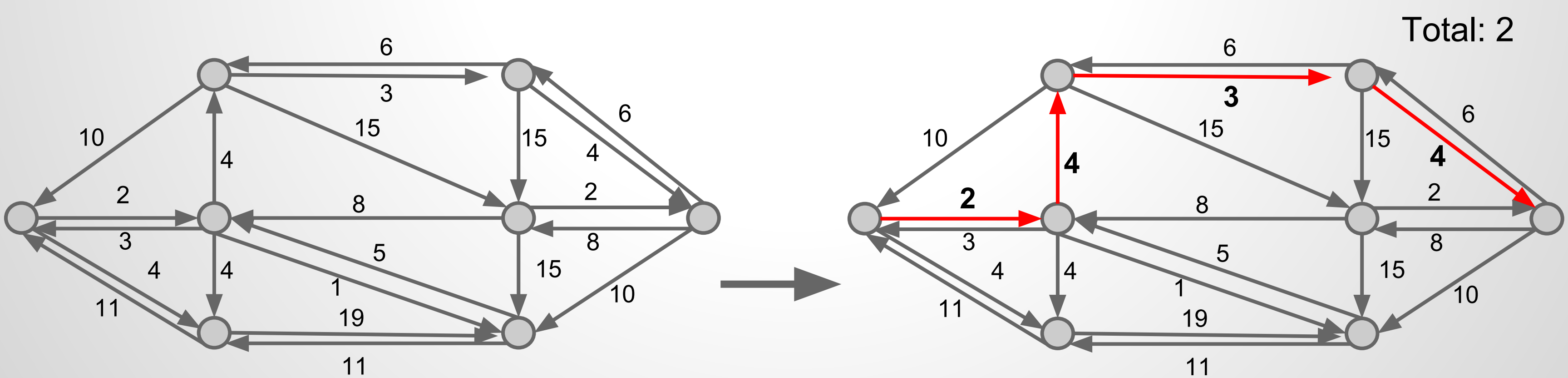




Ford-Fulkerson Algorithm

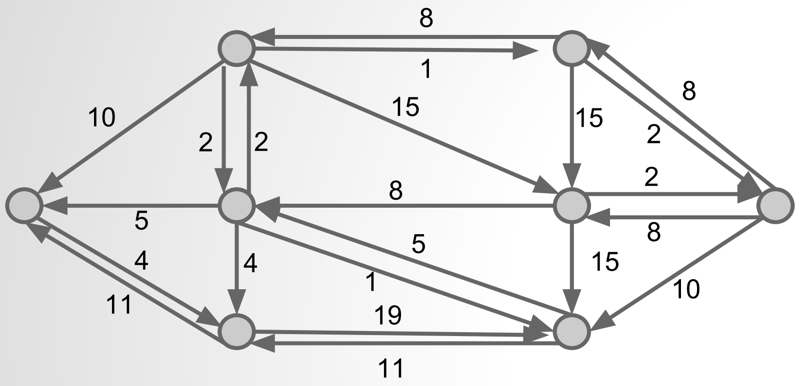
Residual Graphs





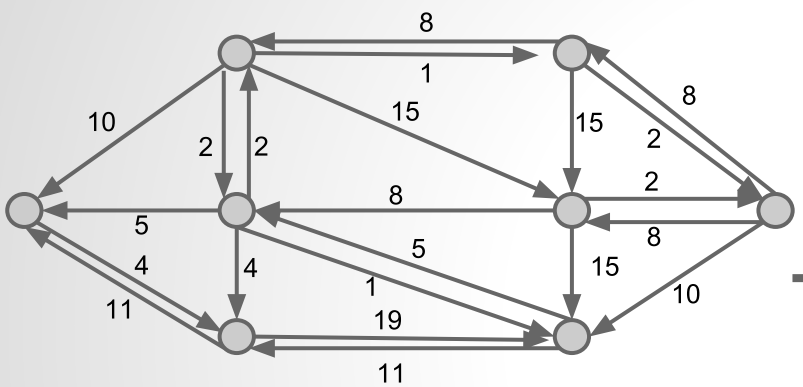
Ford-Fulkerson Algorithm

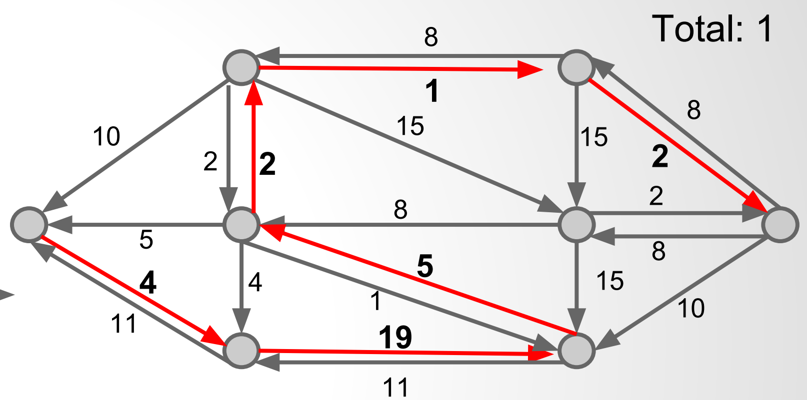
Residual Graphs



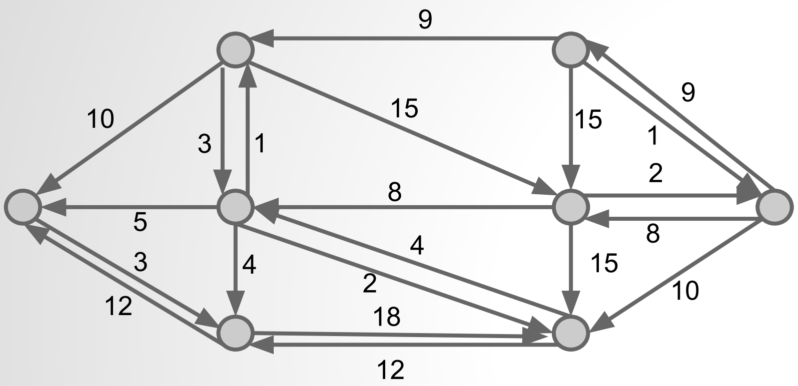
Ford-Fulkerson Algorithm

Residual Graphs



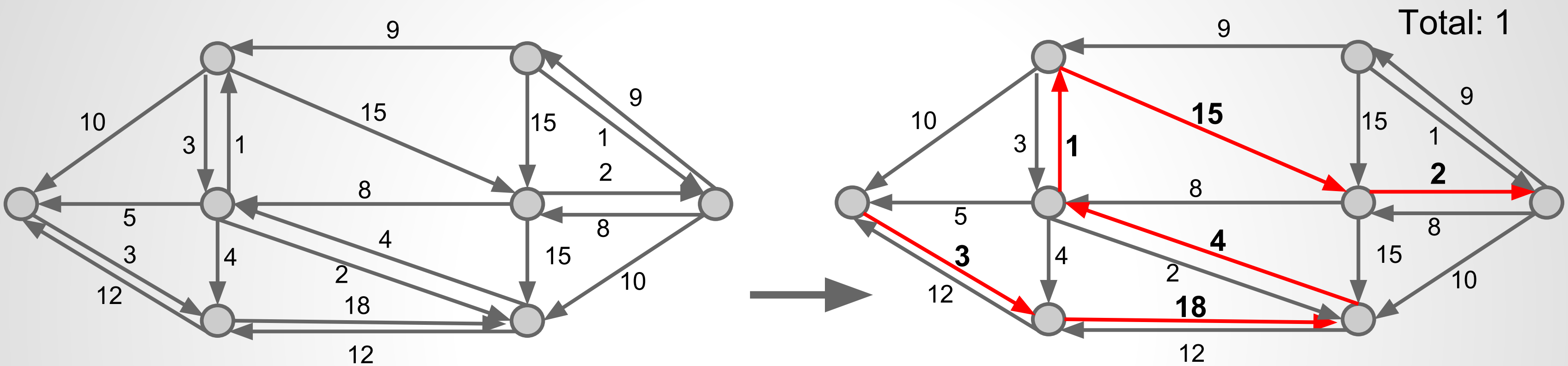


Ford-Fulkerson Algorithm
Residual Graphs



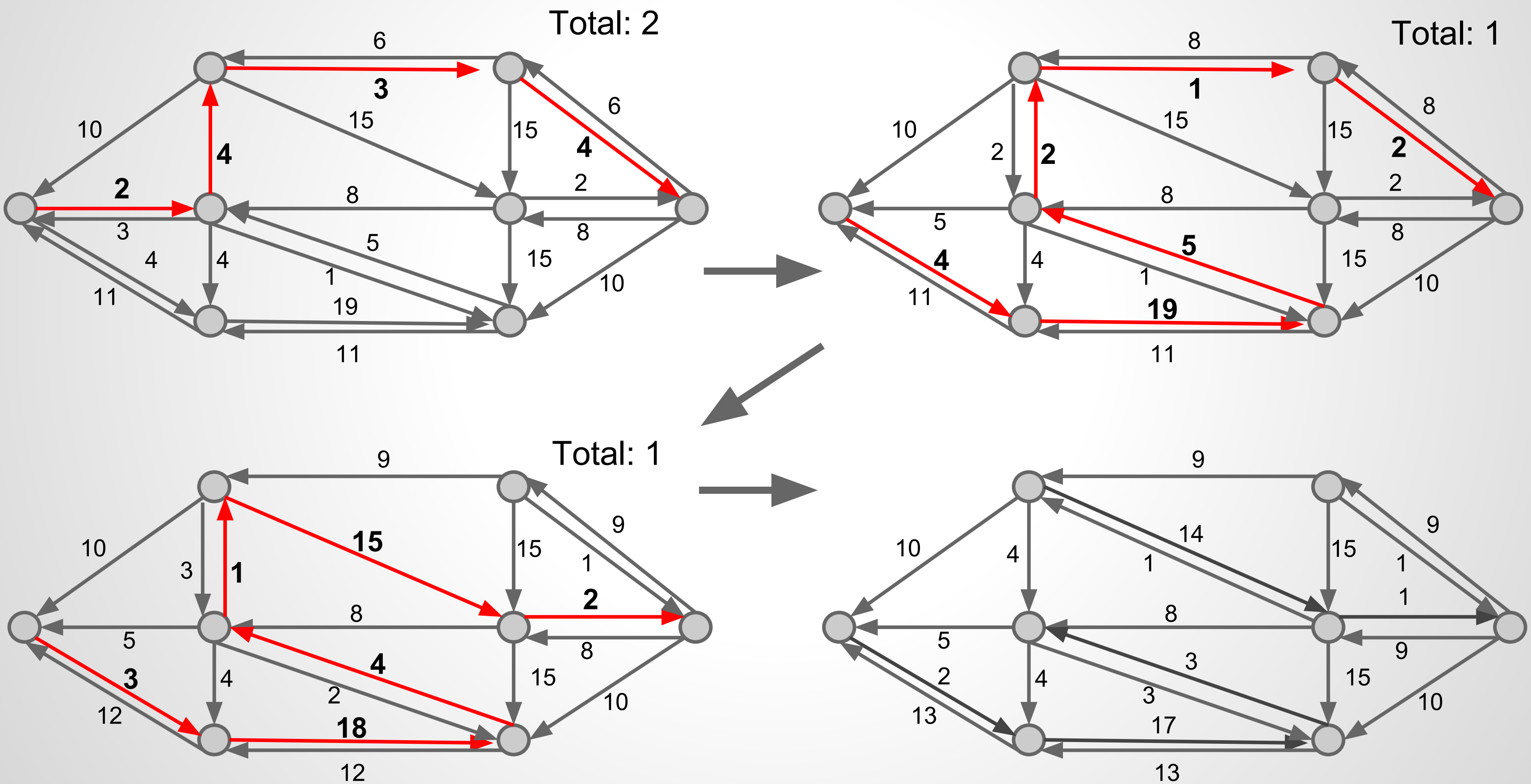
Ford-Fulkerson Algorithm

Residual Graphs



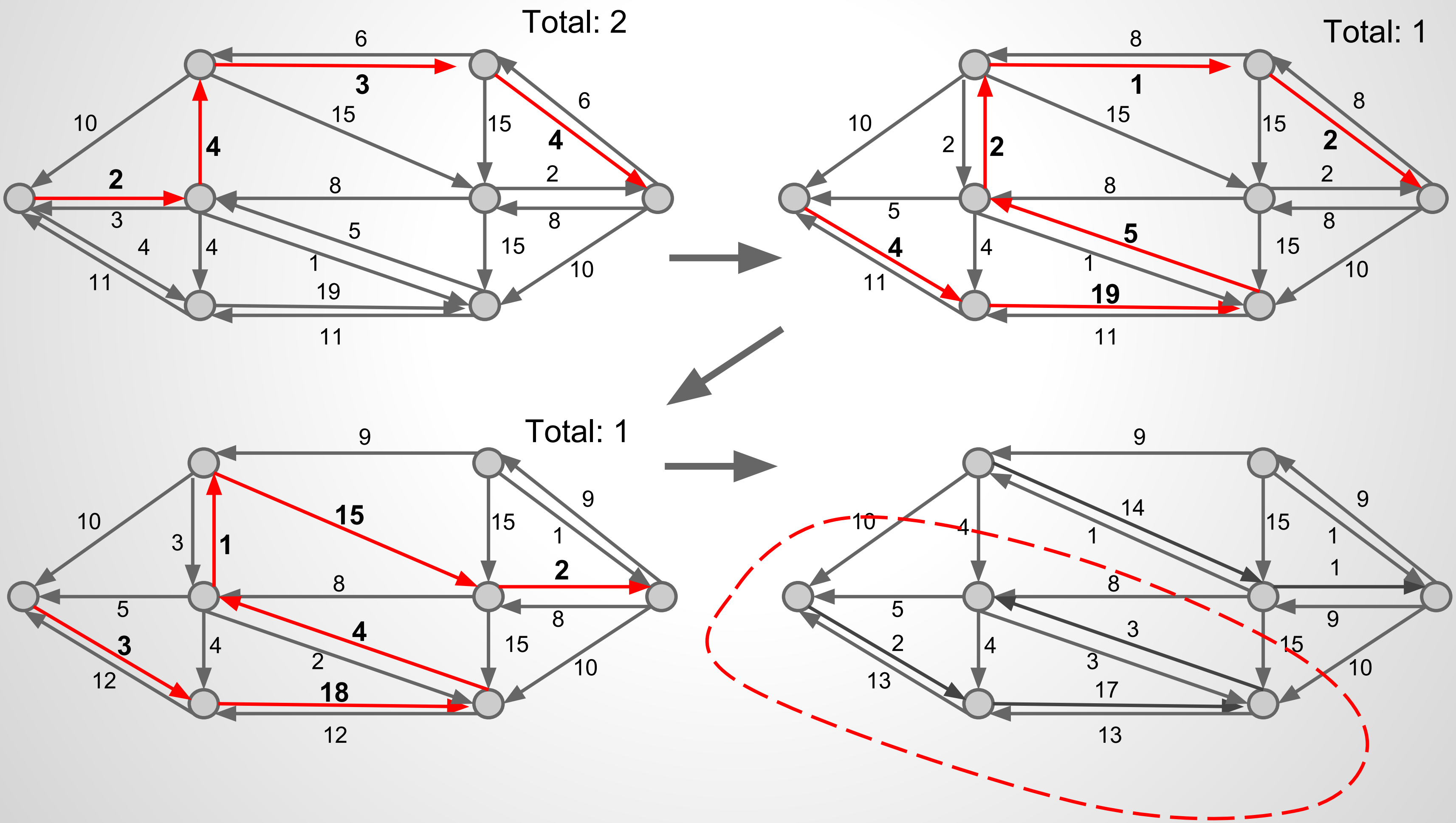
Ford-Fulkerson Algorithm

Residual Graphs



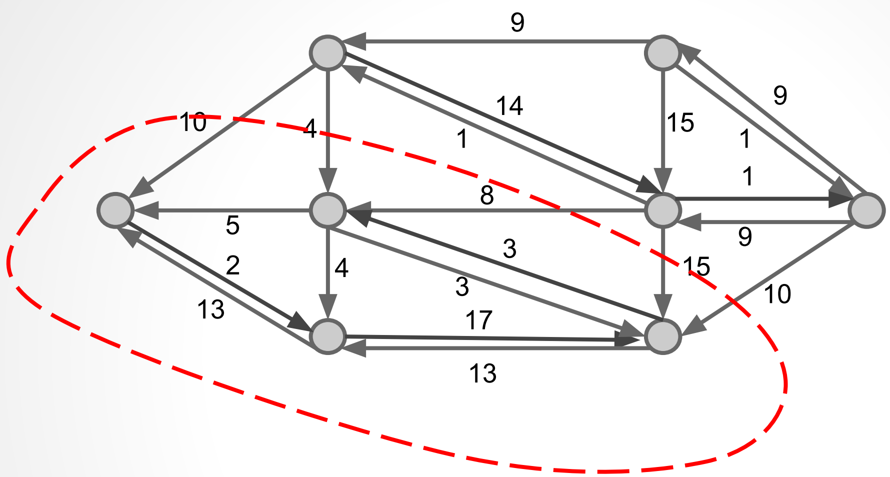
Ford-Fulkerson Algorithm

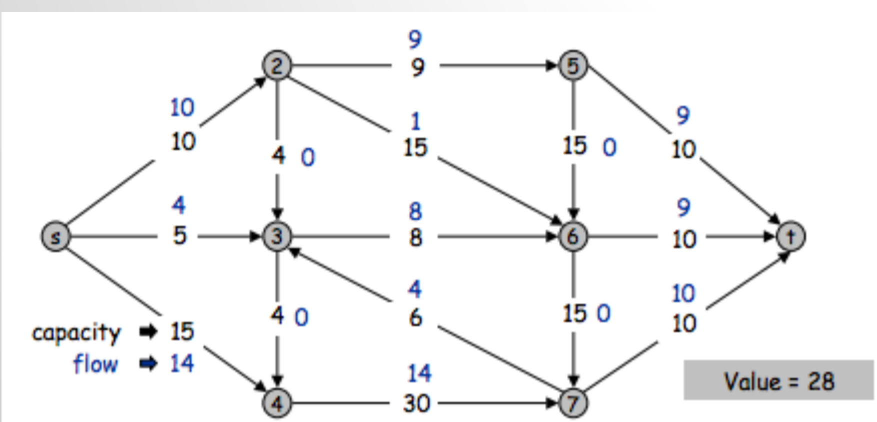
Residual Graphs



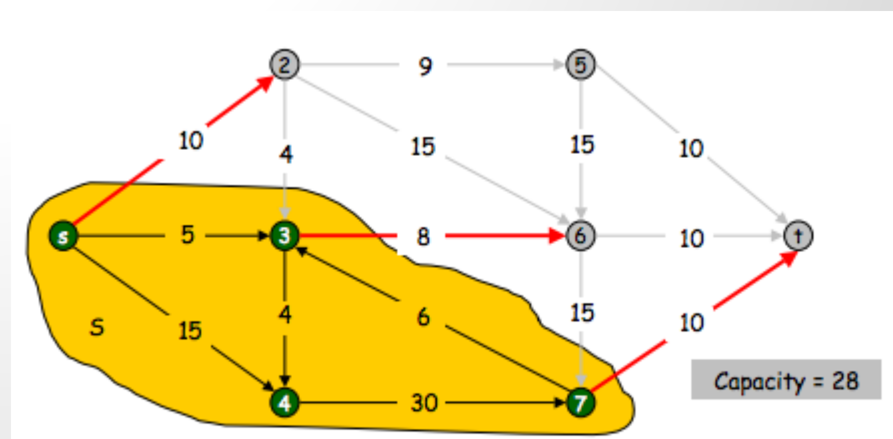
Ford-Fulkerson Algorithm

Residual Graphs





Max Flow

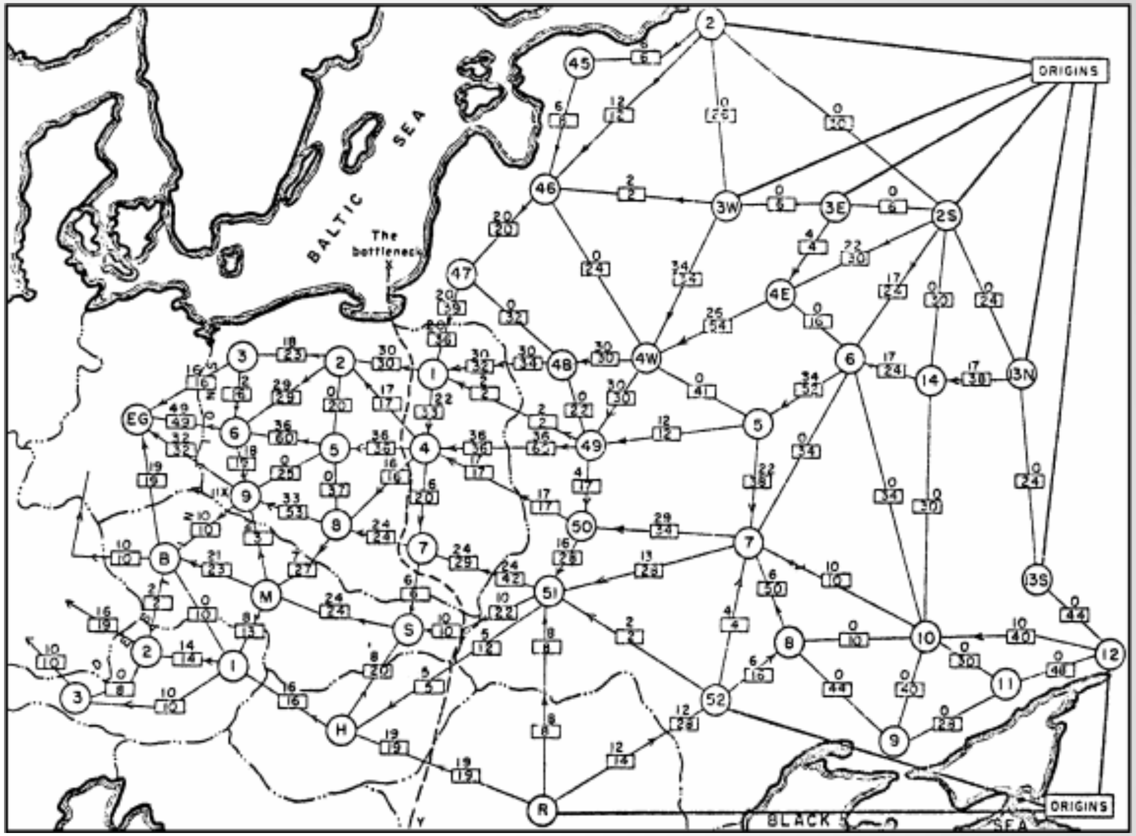


Min Cut

Agenda

1. Basic Background
2. Cuts
3. Flows
4. The Max-Flow, Min-Cut Theorem
5. Conclusion

Conclusion



Widely used in computer science.

Bibliography

"On the history of combinatorial optimization (till 1960)" by Alexander Schrijver

- http://homepages.cwi.nl/~lex/files/histco.pdf

Cut and Flow Examples

- http://www.cs.princeton.edu/courses/archive/spr04/cos226/lectures/maxflow.4up.pdf

Flow Network Image

- http://stochastix.wordpress.com/2009/07/28/