Modeling, Analysis and Optimization of Networks

Alberto Ceselli
alberto.ceselli@unimi.it

Università degli Studi di Milano
Dipartimento di Informatica

Doctoral School in Computer Science

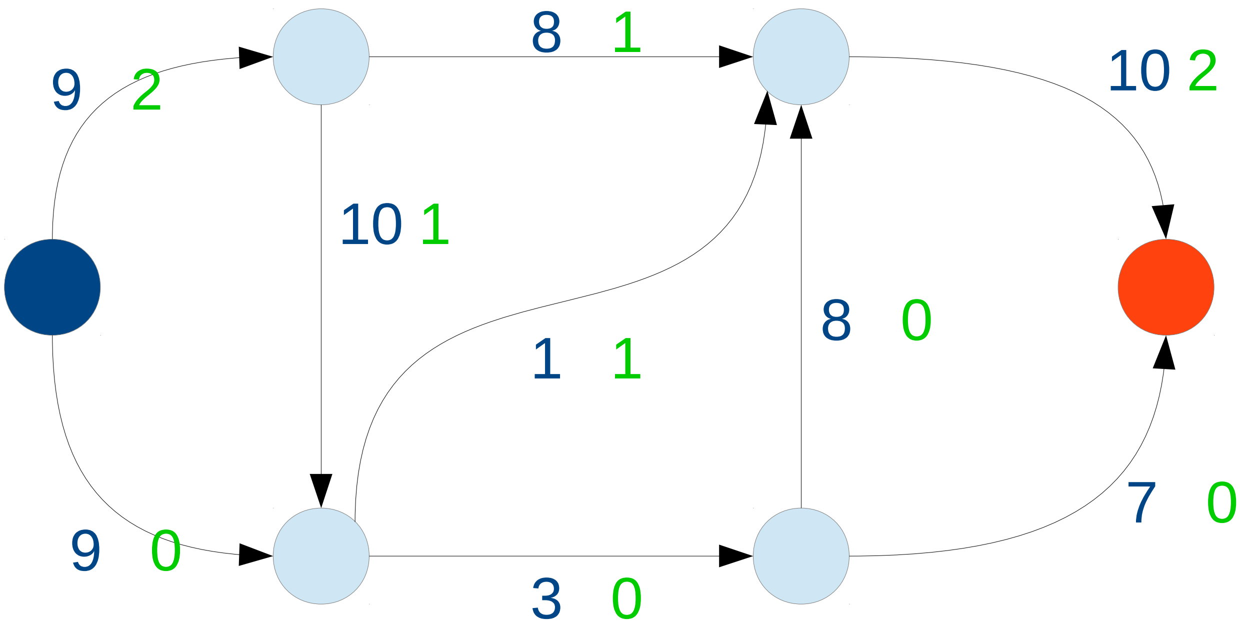
A.A. 2015/2016

Summary of Lecture 1

- Networks are pervasive
- Networks are too complex in features and size to be managed without advanced tools
- A suitable formal framework is known to model network flows that
  - is « compact »
  - is flexible
  - is tractable

Summary of Lecture 1

- Given :
  - a graph G(V,A), with two special nodes s and t
  - capacity on each arc
- Find: an optimal flow from s to t

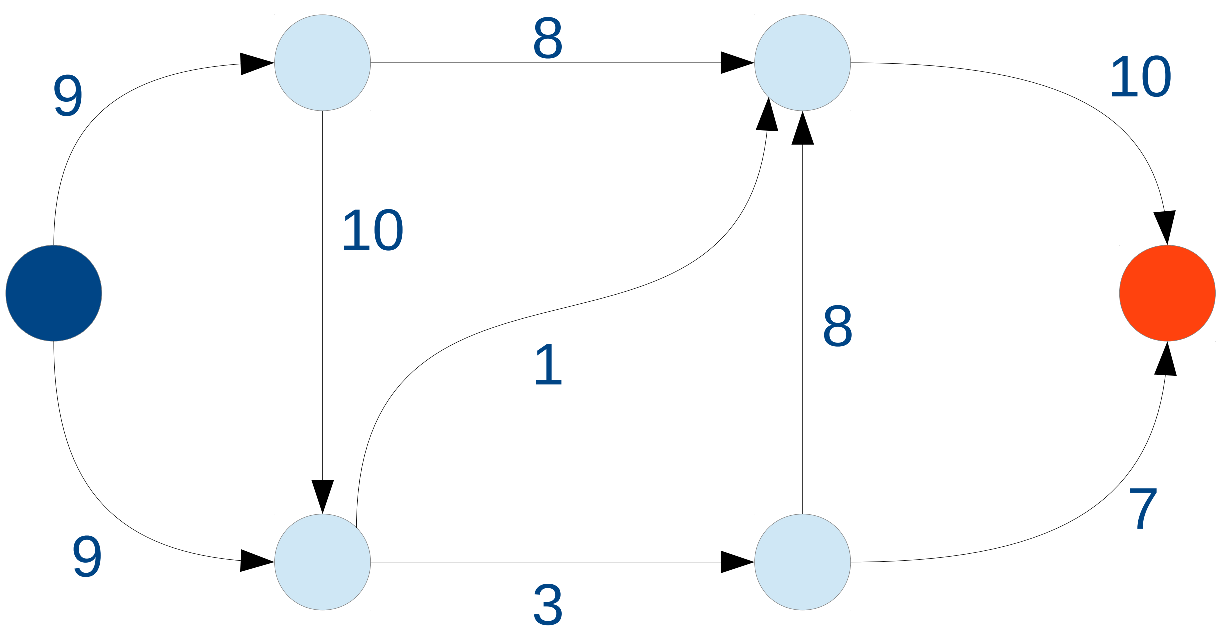


Summary of Lecture 1

- The framework allows also to
  - Take into account circulations instead of s-t flows
  - Have additional capacities on nodes
  - Have flow lower bounds (on both arcs and nodes)
  - Have multiple sources and multiple sinks
  - Have source and demand points
- by simple modeling gadgets ...
- … and therefore without any increase in the complexity of the resolution algorithms.

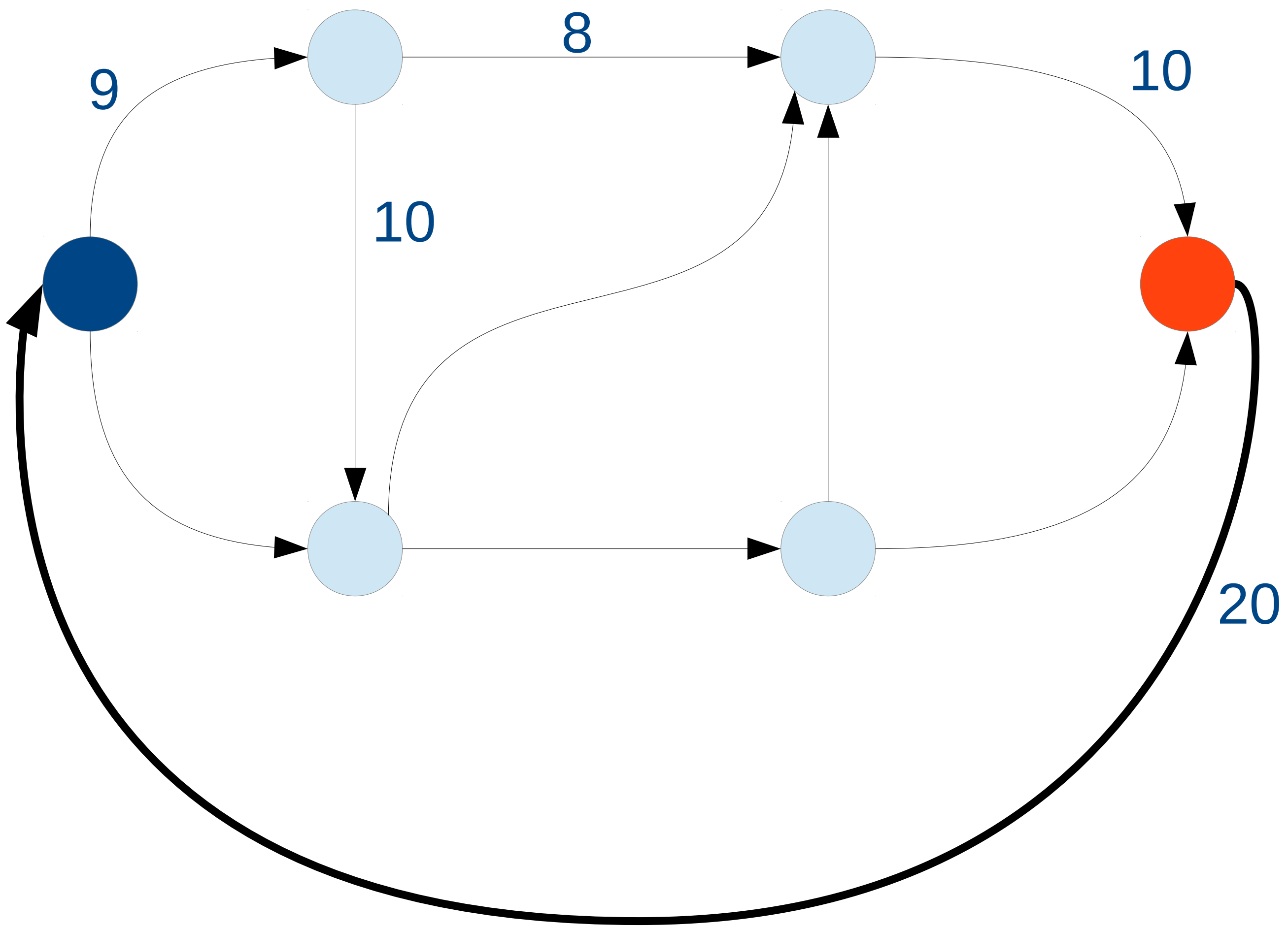
Summary

- Basic model



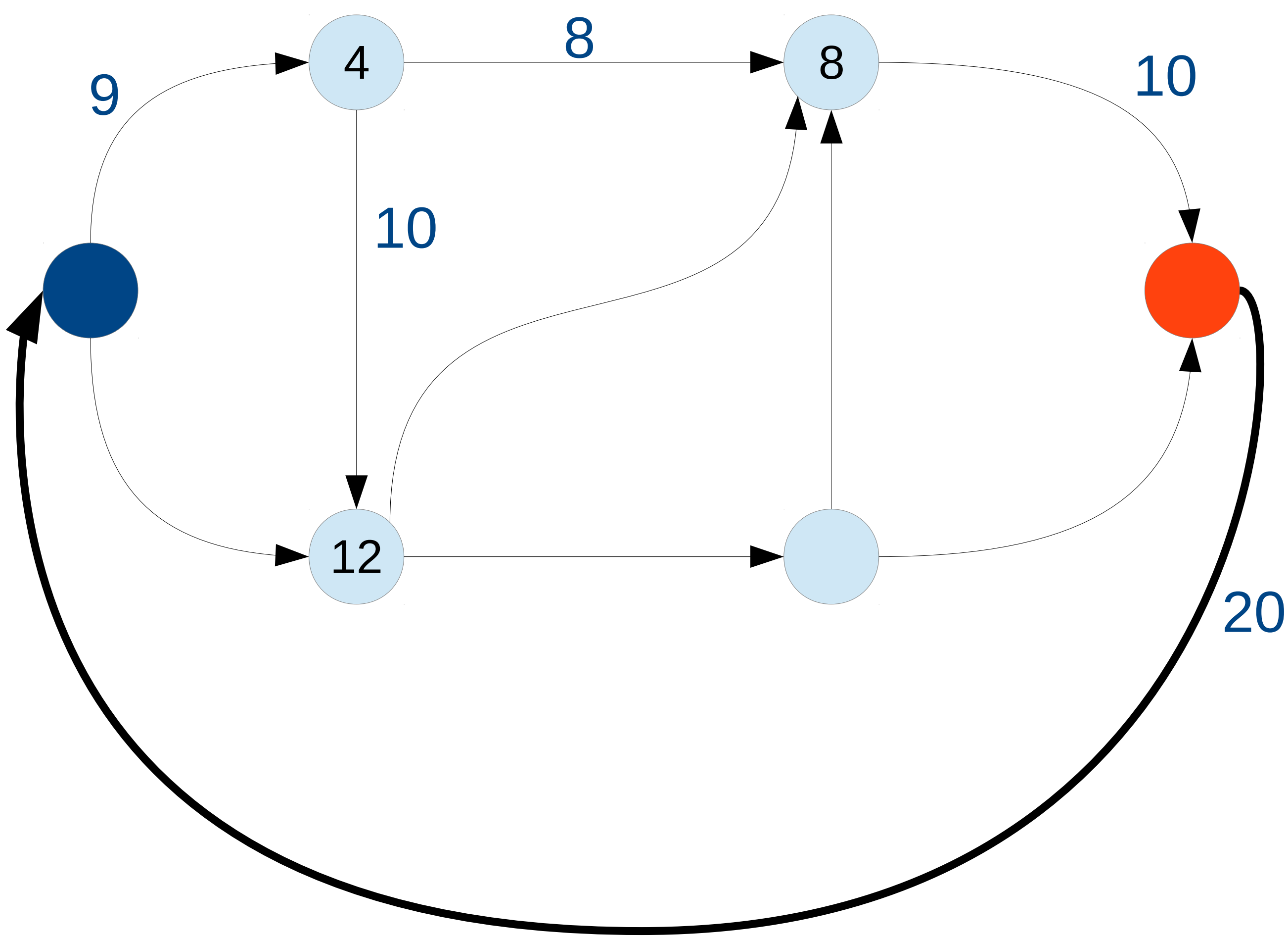
Summary

- Flow circulation



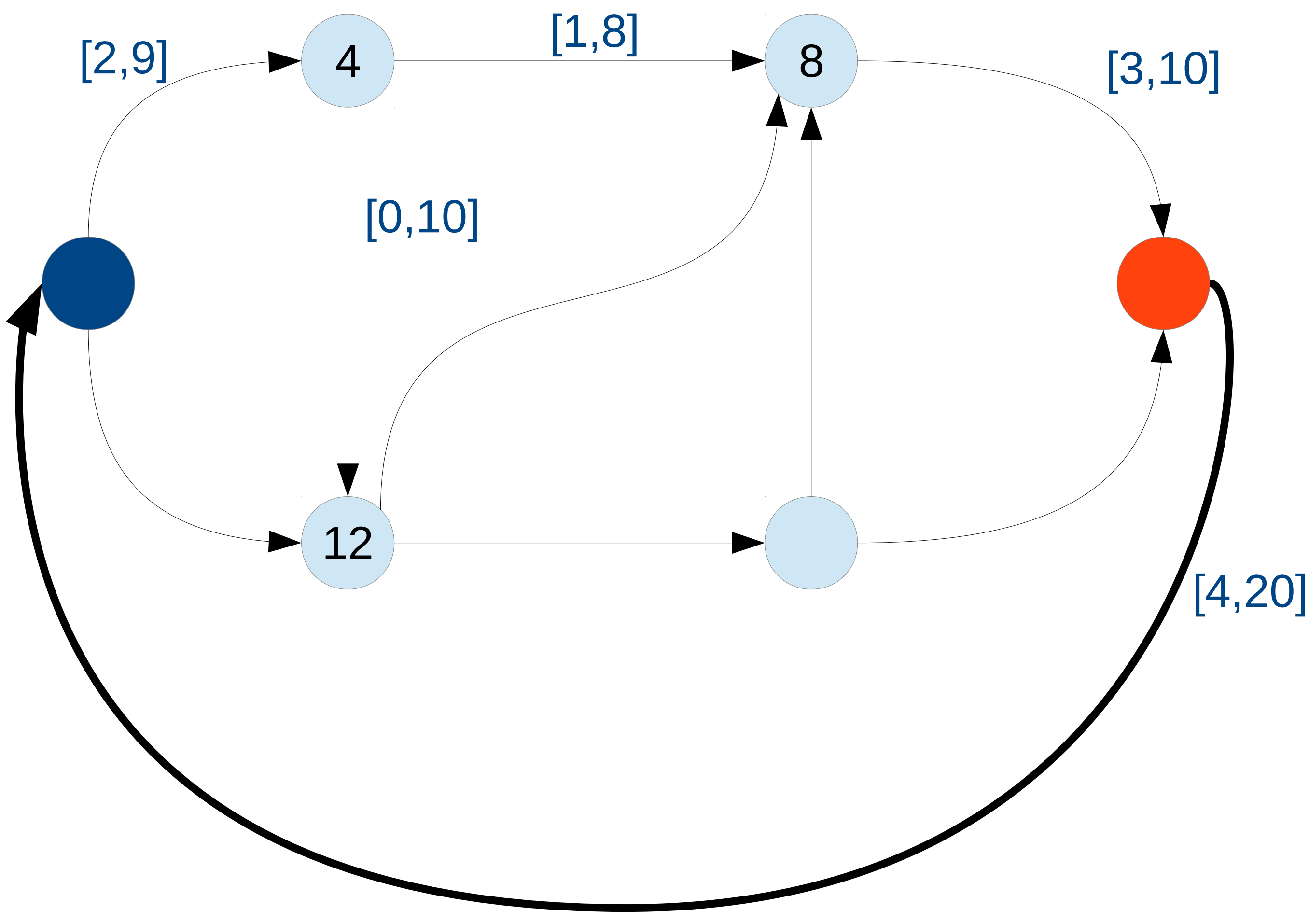
Summary

- Capacities on nodes



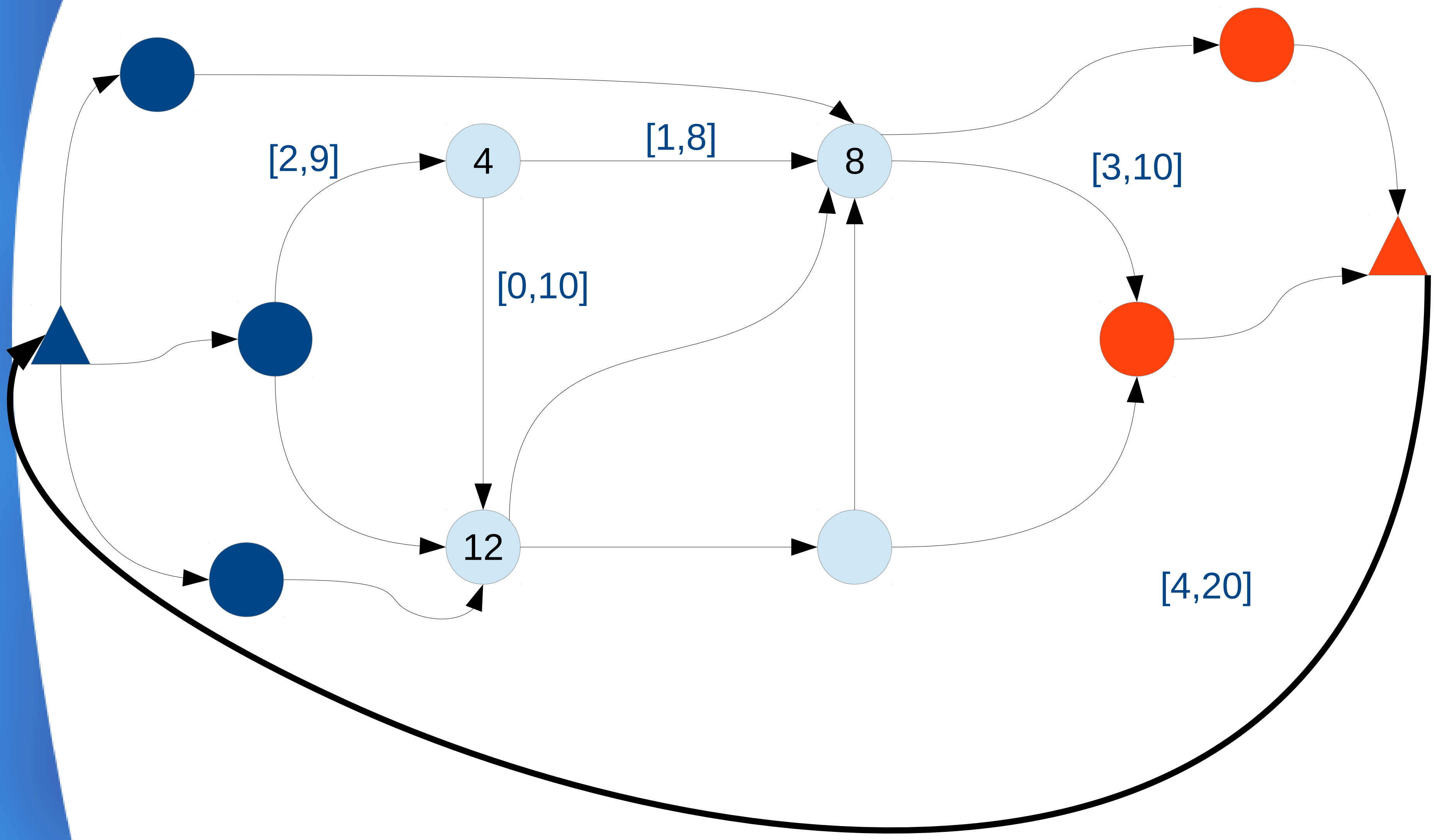
Summary

- Flow lower bounds



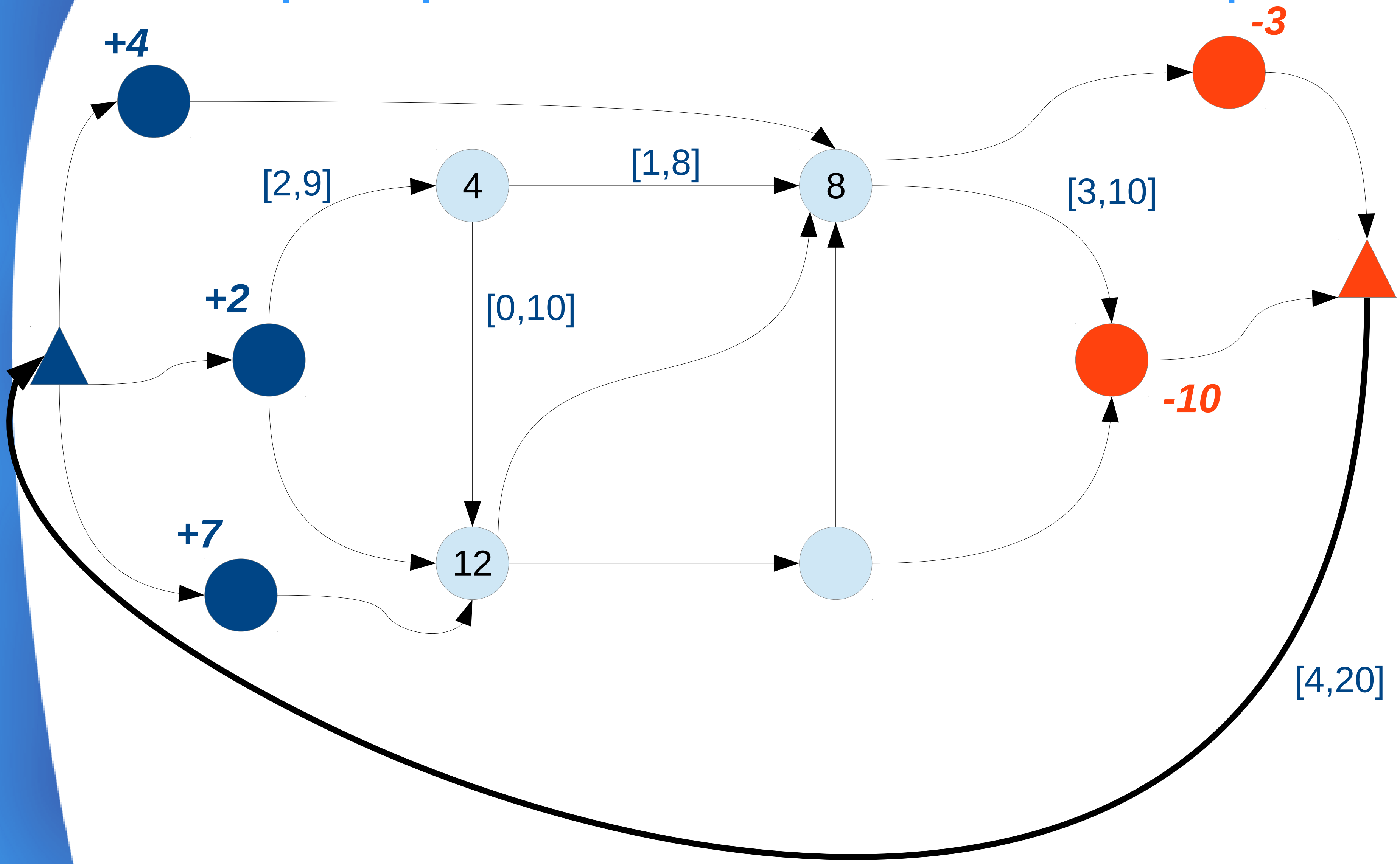
Summary

- Multiple sources and multiple sinks



Summary

Multiple «production» and «demand» points



Examples

- Analysis of (telecom) network connectivity
  - Single path
  - Link disjoint paths
  - «robustness» against link failures
- Tango dancing problem (max matching)
- Matrix rounding problem

Example 4: survey design

Taken from J. Kleinberg, E. Tardos, « Algorithm Design»

- We would like to design a survey of products used by consumers (i.e., « Consumer i: what did you think of product j? »).
- The ith consumer agreed in advance to answer a certain number of questions in the range $[c_i, d_i]$
- Similarly, for each product j we would like to have at least p_j opinions about it, but not more than q_j
- Each consumer can be asked about a subset of the products which they consumed.
- The question is how to assign questions to consumers, so that we get all the informations we need, and each consumer is being asked a valid number of questions.

Part II – Properties of Flows

- Another good feature of Network Flows :
  - structural properties can be proved by the design of efficient algorithms!
- … i.e. theory and computation fit nicely!

Lecture Plan

- Shortest path problems and algorithms
- Flows and Cuts
- Max flow algortihms
- Combinatorial properties of max flows
- Application examples

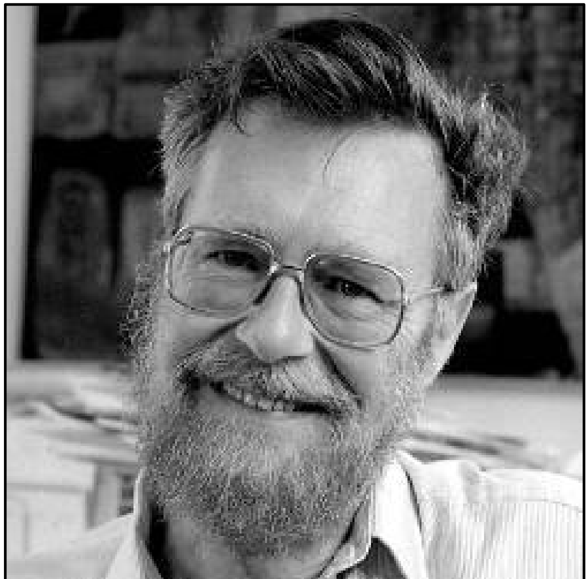
Shortest Path Problems

- Given a graph G(V,A)
- Given a source node s and a destination node t
- Given a length function $l: A \rightarrow \mathbb{R}_+$
- Find a path from s to t of minimum overall length
- Path: sequence of adjacent nodes
- Length of a path: sum of lengths of arcs connecting pairs of subsequent nodes

Shortest Path Algorithms

- Most famous one: Dijkstra
  - Keep a set of « labels » d(i), one for each i in V
  - … encoding « tentative » distances from s to i
  - Iteratively pick one node, and try to improve the labels of its neighbors
  - Until no more improvements are possible

```
Procedure update(i) {
    foreach j in ∂+(i) do
        if d(j) > d(i) + l(i,j) then
            d(j) = d(i) + l(i,j)
}
```



- Slides 8 – 12 from

  « Network Flows », by J. Orlin
- Dijkstra Algorithm example

  by J. Orlin
- Proof of correctness (blackboard)
- Summary: we have an efficient way for finding (special) paths between nodes of networks

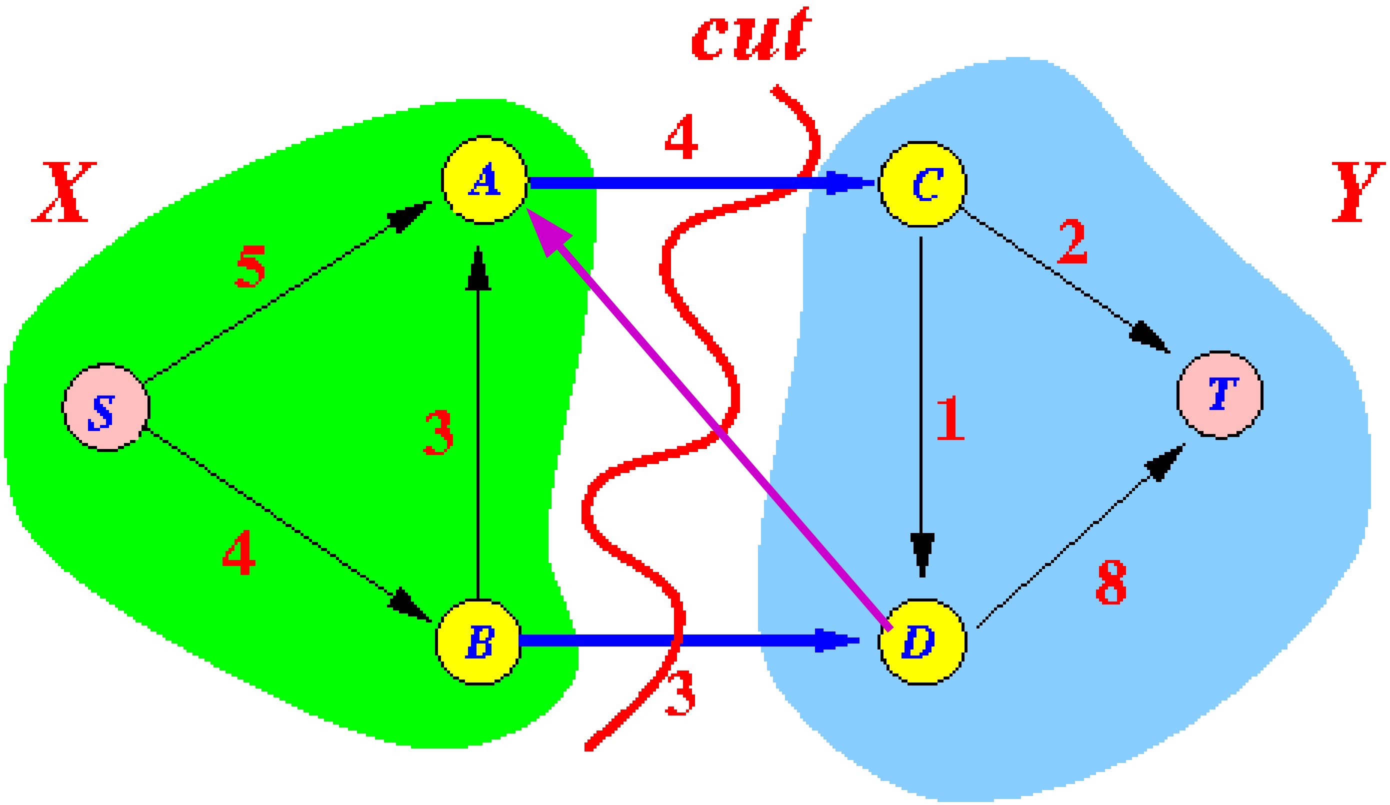
Max Flow algorithms

- Working with residual networks, slides
  « Network Flows », by J. Orlin
- Ford Fulkerson Algorithm example
  by J. Orlin

Flows and Cuts

- Intuitively (s-t) **Cut**: a set of arcs whose removal makes sink unreachable from source
  (no more path exists going from s to t)
- Formally, a cut is a **partition** [S,V \ S] of the nodes of the graph (s-t cut if s is in S and t in V \ S)
- Arcs of the cut are those having one endpoint in S and the other in V \ S;
  - forward arc: (i,j) with i in S and j in V \ S
  - backward arc: (i,j) with i in V \ S and j in S
- The **capacity** of a cut is the sum of capacities of its **forward** arcs
- A cut is **minimum** if its capacity is minimum

Flows and cuts



Cut = { (A,C), (B,D) }

Properties of max flows

- Obs. 1: if caps are **integral**, I always increment flows by **integers**, and therefore obtain only **integral flows!**
- Obs. 2: Intuitively, how can I detect if a flow is maximum? No more augm. path!
- Formally, I need to compare flows and cuts !
  - Weak duality theorem for flows and cuts
  - Optimality conditions for max flows

Weak duality for max flow

- Def. **Flow across a cut** [S,T] is the sum of flow on forward arcs minus the sum of flow on backward arcs
- Claim 1:
  - If f() is a feasible flow and [S,T] is an (s-t) cut
  - The the flow across the cut = flow from s = flow into t
  - Proof: blackboard
- Claim 2:
  - If f() is a feasible flow and [S,T] is an (s-t) cut
  - Then the **flow** across the cut is at most equal to the **capacity** of the cut [S,T]
  - Proof: blackboard
- → If f() is a feasible flow and [S,T] is an (s-t) cut
  - Then the **flow** from s to t is **at most** equal to the **capacity** of the cut [S,T]

Optimality conditions for max flows

- The following are equivalent :

  1 - A flow f() is maximum

  2 - There is no augmenting path in the residual network

  3 - There is an (s-t) cut whose capacity equals the s-t flow of f()

- Proof: blackboard

Cuts and Flows

- Recall: find **how many** paths are there from source to destination, having **no common arc?**
- **Menger's Theorem (1927).** The *max* number of *edge*-disjoint s-t paths is equal to the *min* number of *edges* whose removal disconnects destination from source.

K. Menger "Zur allgemeinen Kurventheorie". Fund. Math. 10: 96–115 (1927)

Cuts and Flows

Let F be a subset of A

- Proof ←
  (suppose the removal of F disconnects t from s, and |F| = k)
  - All s-t paths use at least one edge of F.
  - Hence, the number of edge-disjoint paths is **at most k**.
- Proof →
  (suppose max number of edge-disjoint paths is k)
  - Then max flow value is k.
  - A cut (S, T) of capacity k exists (max flow min cut duality)
  - Let F be set of edges going from S to T.
  - |F| = k, and F disconnects t from s.

Example 6: image segmentation

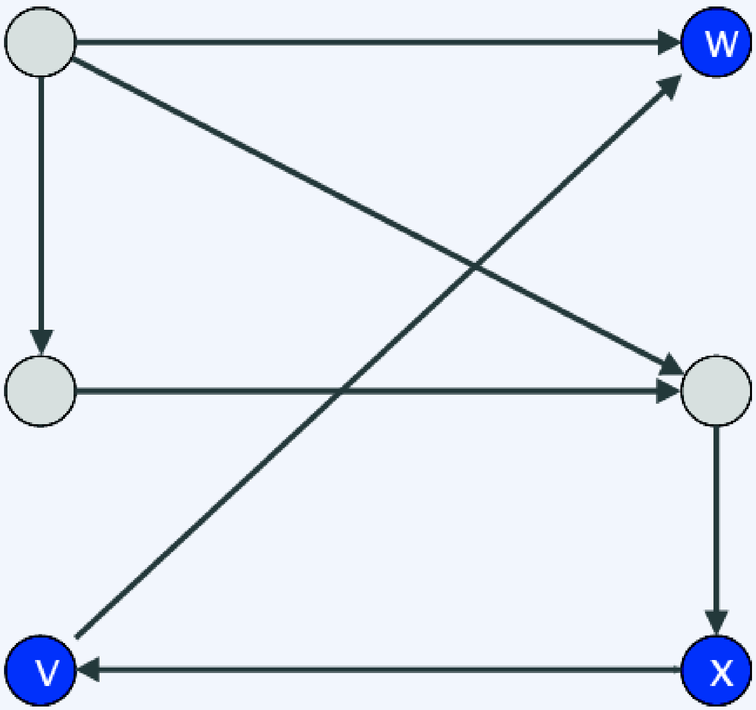
Taken from J. Kleinberg, E. Tardos, « Algorithm Design»

- **You are given an image, say a set V of pixels**
- **For each pixel i in V :**
  - **a_i is the likelihood that i is foreground**
  - **b_i is the likelihood that i is background**
  - **p_{ij} is the similarity between i and j**
- **Goal: find a partition of pixels in foreground and background, optimizing accuracy and smoothness**

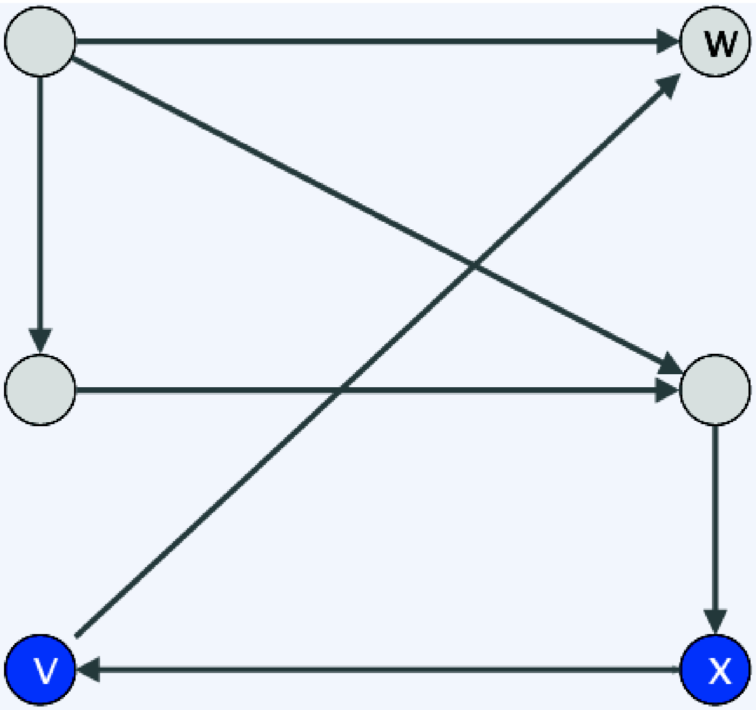
Example 7: project management

Taken from J. Kleinberg, E. Tardos, « Algorithm Design»

- Set of possible projects P
- Each project i has an associated coefficient p_i
  - Profit if $p_i > 0$
  - Cost if $p_i < 0$
- Set of prerequisites E: if (i,j) in E, i cannot be done unless also j is done
- A subset A of projects is feasible if each prerequisite of a project in A is also in A
- Goal: find a feasible subset of projects of maximum revenue (maximum weight closure problem)



{ v, w, x } is feasible



{ v, x } is infeasible