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

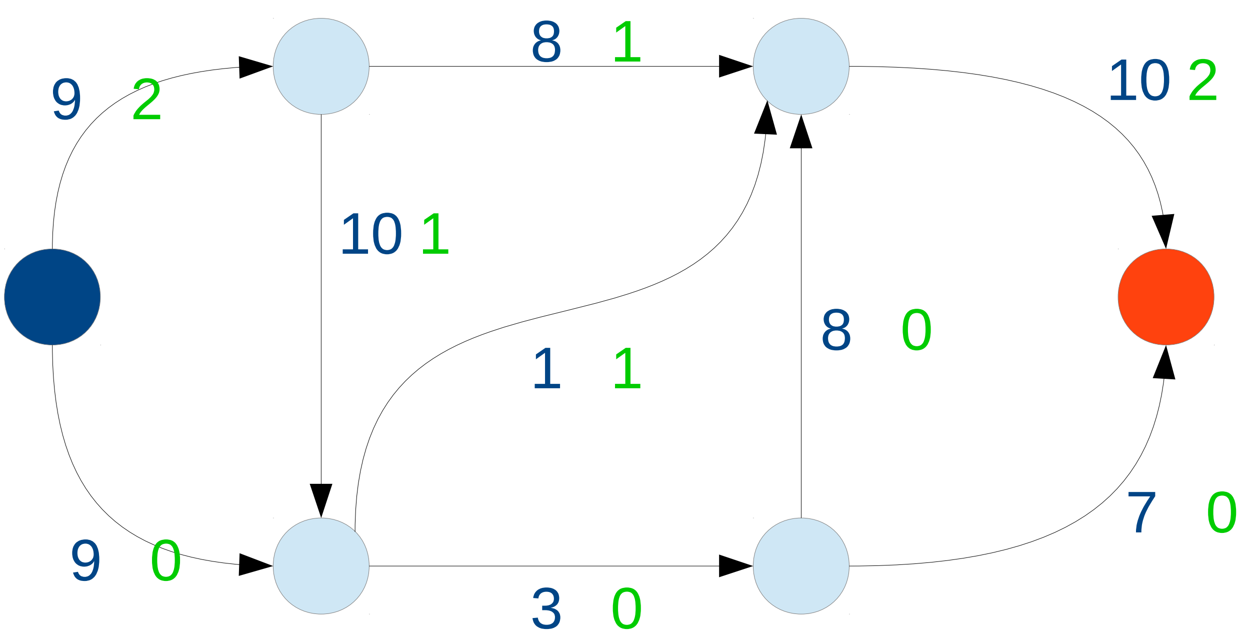
A.A. 2015/2016

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

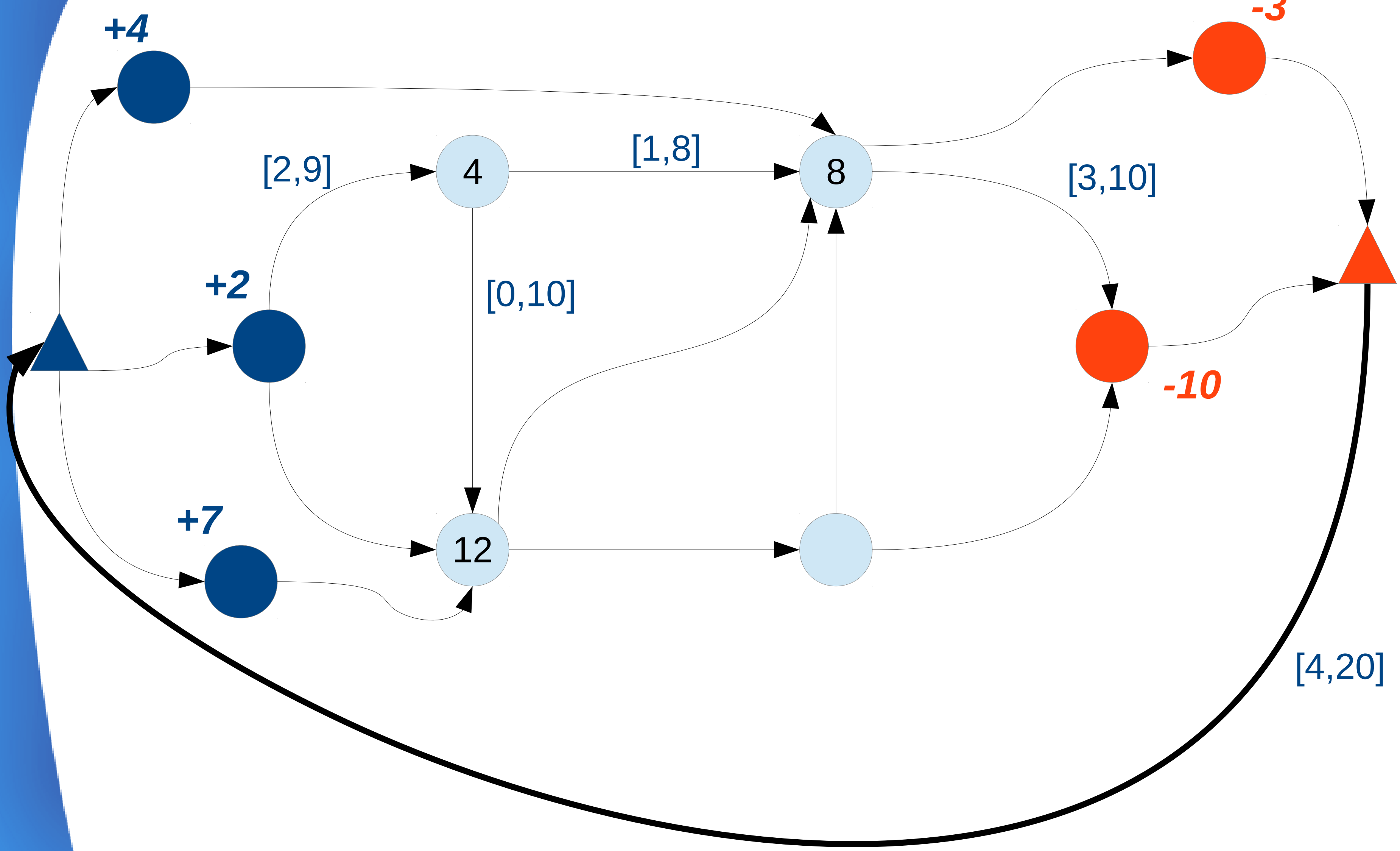
Summary of Lecture 1

- 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

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
- Survey design problem

Summary of Lecture 2
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!

Summary of Lecture 2

- Dijkstra Algorithm with example

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

Lecture 3 Plan

- Max flow algorithms
- Flows and Cuts
- Combinatorial properties of max flows
- Application examples

Max Flow algorithms

- Working with residual networks, slides

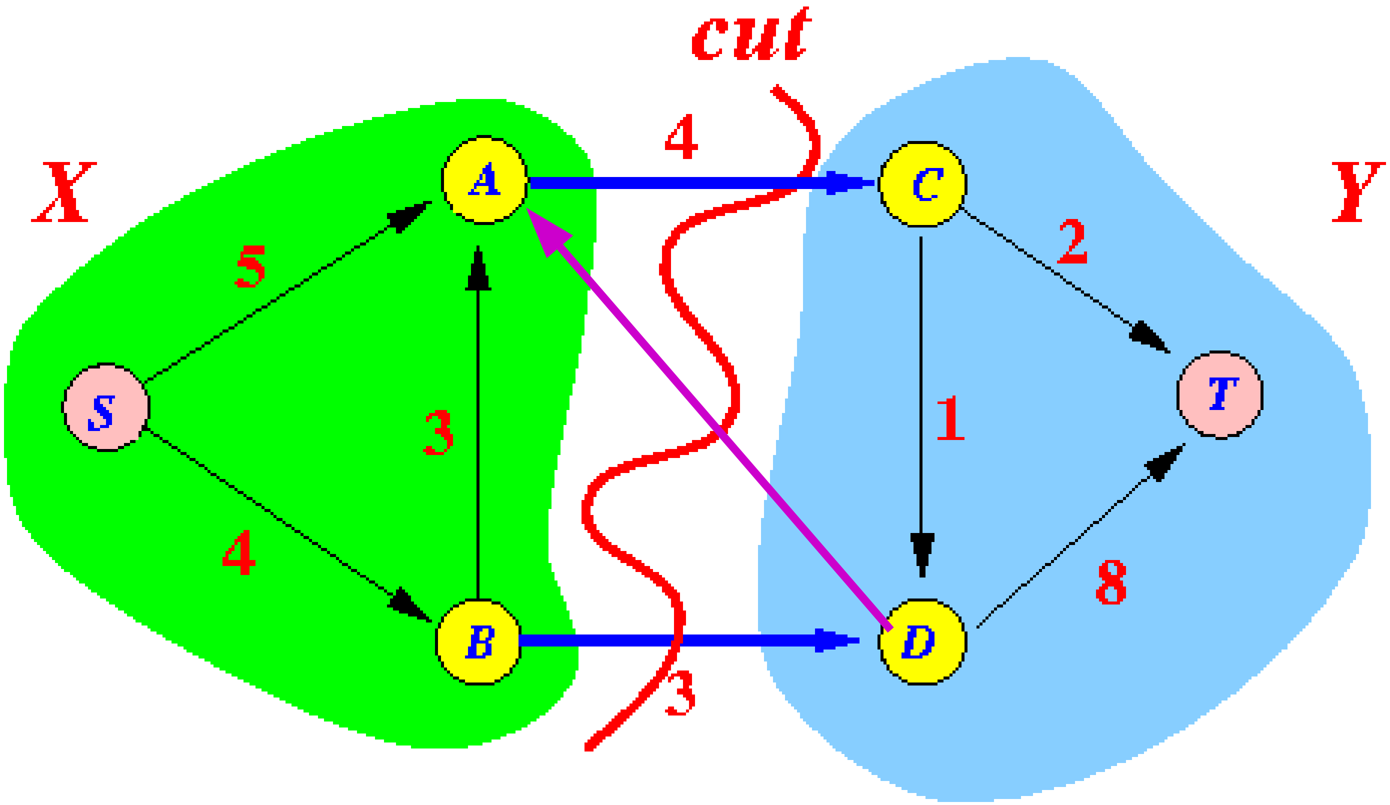
  « Network Flows », by J. Orlin
- Ford Fulkerson Algorithm example

  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

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