ORF307 – Optimization

16. Network optimization

Bartolomeo Stellato — Spring 2022

Ed Forum

- In the adding variable example, we checked whether (x*, 0) was still primal and dual feasible, from which we concluded that having 0 of the new product was still optimal. **Why did we not need to check whether the duality gap was still 0 in order to conclude this?**

- In the case that something is dual feasible but not primal feasible, **we can run dual simplex starting from the feasible dual solution to find the optimal solution.** Is this because if we get to the end of dual simplex and have an optimal solution, then this solution is also the optimal solution for the primal solution because of strong duality?

- I know that the primal and the dual have a special relationship, where one as basically the 'flip' of the other. However, **I wasn't exactly sure how the feasibility of one affected the other;** I was wondering if you could explain that a bit more.

Recap

Primal and dual basic feasible solutions

Primal problem

minimize $c^T x$
subject to $Ax = b$
$x \geq 0$

Dual problem

maximize $-b^T y$
subject to $A^T y + c \geq 0$

Given a **basis** matrix B

Reduced costs

Primal feasible: $Ax = b,\ x \geq 0 \quad \Rightarrow \quad x_B = A_B^{-1} b \geq 0$

Dual feasible: $A^T y + c \geq 0$. Set $y = -A_B^{-T} c_B$. Dual feasible if $\bar{c} = c + A^T y \geq 0$

Zero duality gap: $c^T x + b^T y = c_B^T x_B - b^T A_B^{-T} c_B = c_B x_B - c_B^T A_B^{-1} b = 0$

(by construction)

The primal (dual) simplex method

Primal problem

minimize $c^T x$

subject to $Ax = b$

$x \geq 0$

Dual problem

maximize $-b^T y$

subject to $A^T y + c \geq 0$

Primal simplex

- Primal feasibility
- Zero duality gap

↓

Dual feasibility

Dual simplex
(solve dual instead)

- Dual feasibility
- Zero duality gap

↓

Primal feasibility

Adding new variables

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array} \longrightarrow \begin{array}{ll} \text{minimize} & c^T x + c_{n+1} x_{n+1} \\ \text{subject to} & Ax + A_{n+1} x_{n+1} = b \\ & x, x_{n+1} \geq 0 \end{array}$$

Solution $x^\star, y^\star$

Is the solution $(x^\star, 0), y^\star$ **optimal** for the new problem?

Adding new variables

Optimality conditions

minimize $c^T x + c_{n+1} x_{n+1}$

subject to $Ax + A_{n+1} x_{n+1} = b$ ⟶ Solution $(x^\star, 0)$ is still **primal feasible**

$x, x_{n+1} \geq 0$

Is $y^\star$ still **dual feasible**?

$$A_{n+1}^T y^\star + c_{n+1} \geq 0$$

Yes

$(x^\star, 0)$ still **optimal** for new problem

Otherwise

Primal simplex

Today's lecture

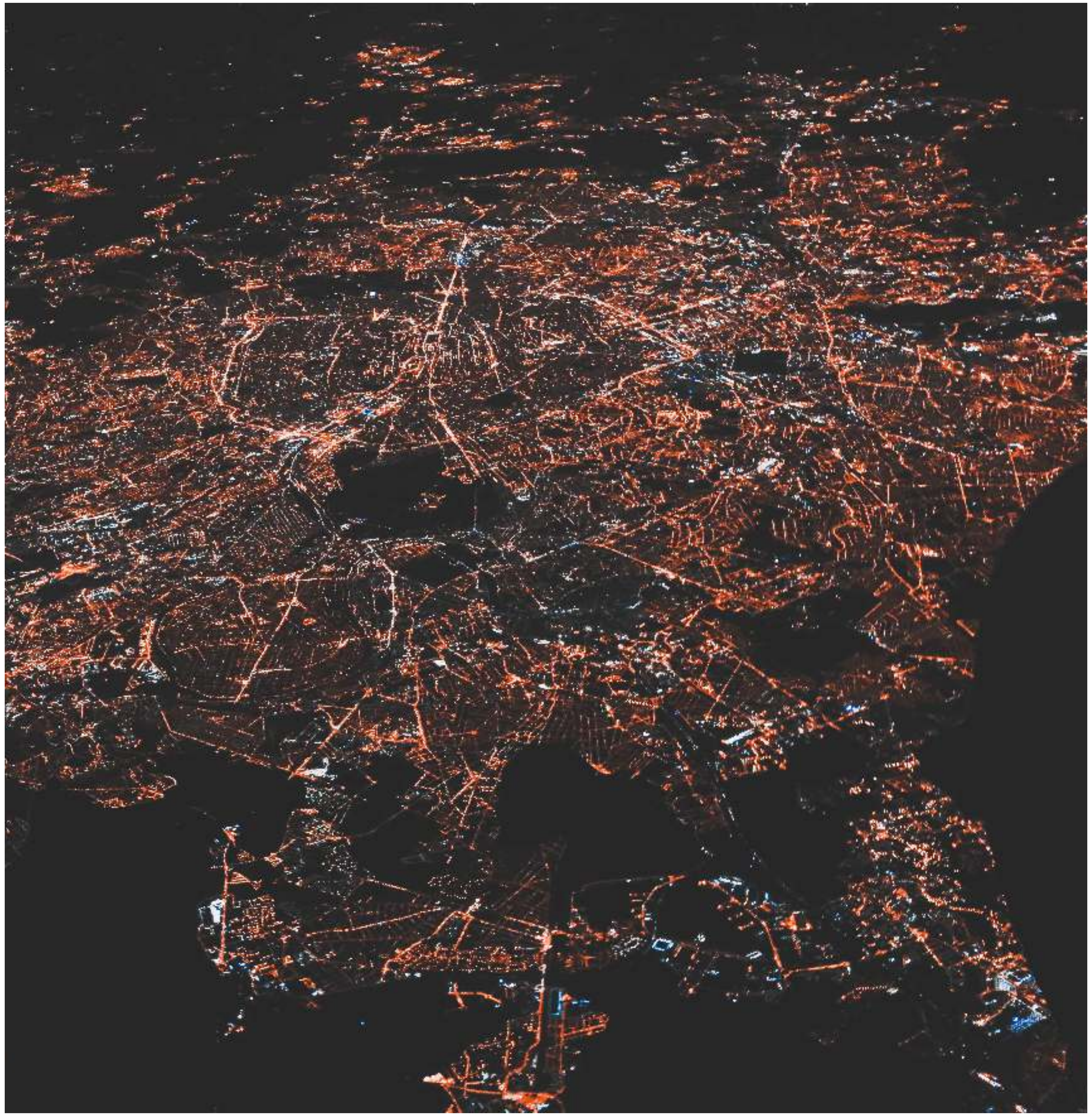
Network optimization

- Network flows
- Minimum cost network flow problem
- Network flow solutions
- Examples: maximum flow, shortest path, assignment

Network flows

Networks

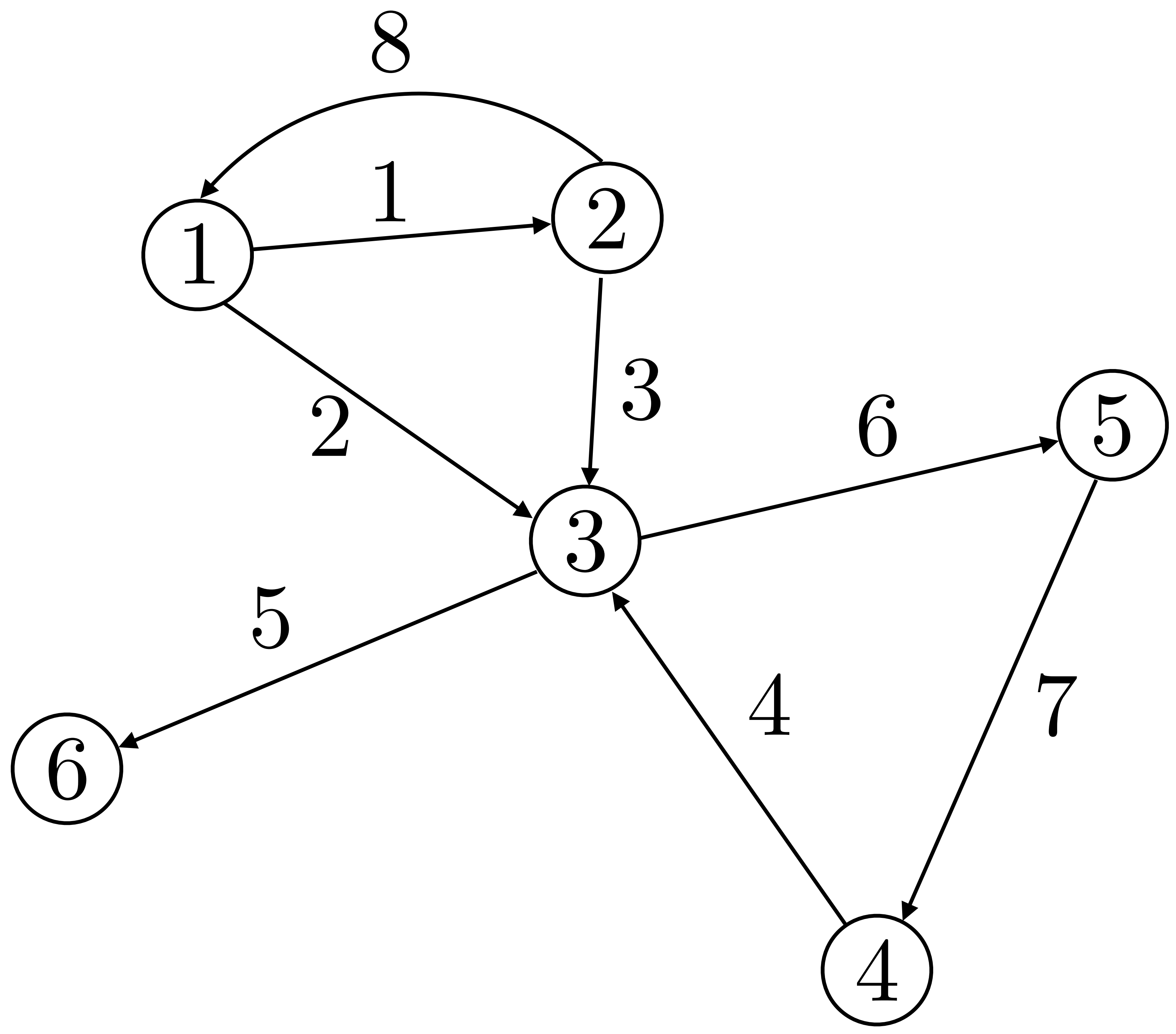
- Electrical and power networks
- Road networks
- Airline routes
- Printed circuit boards
- Social networks



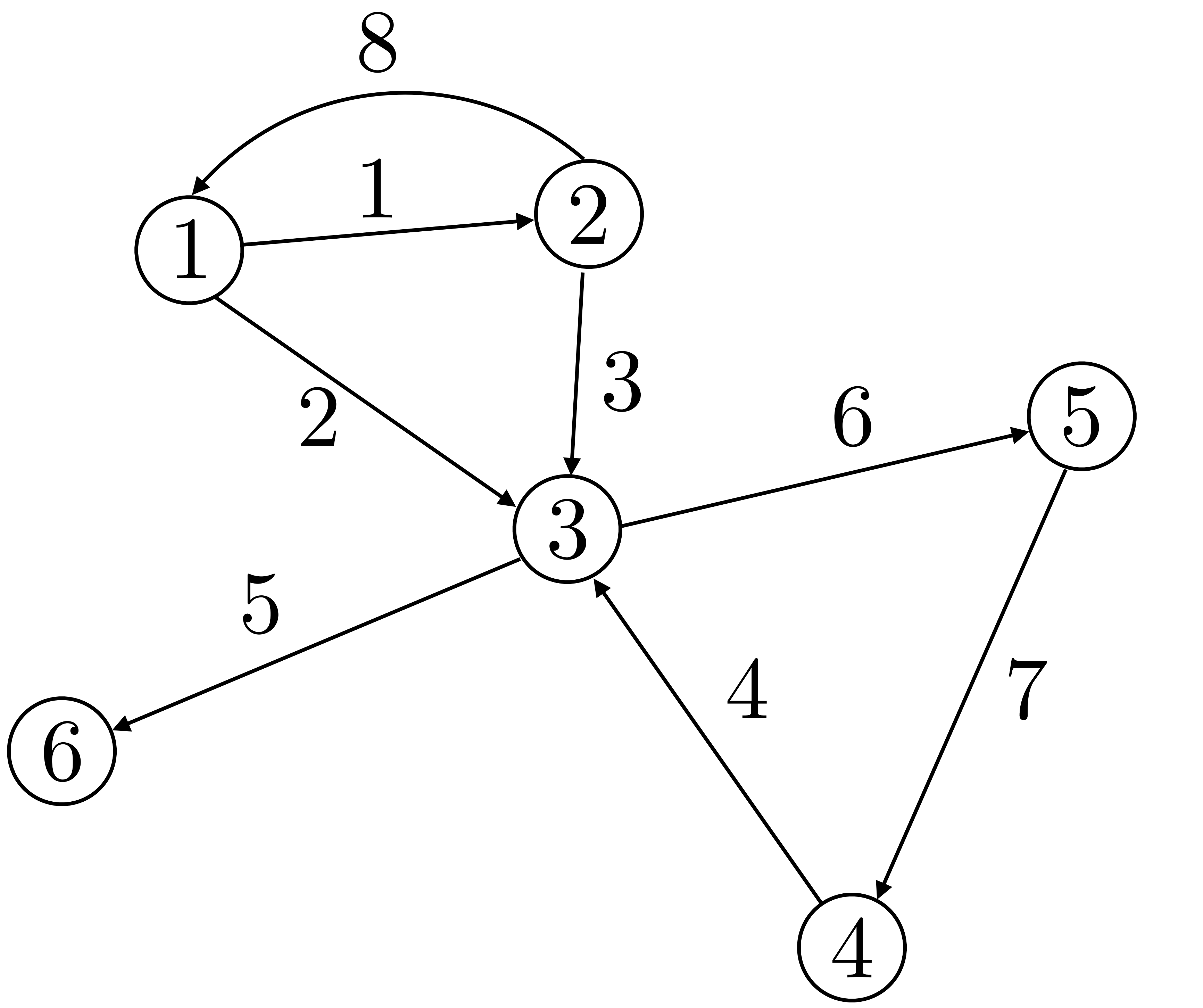
Network modelling

A **network** (or *directed graph*, or *digraph*) is a set of m nodes and n directed arcs

- Arcs are ordered pairs of nodes (a, b) (leaves a, enters b)
- **Assumption** there is at most one arc from node a to node b
- There are no loops (arcs from a to a)



Arc-node incidence matrix



$m \times n$ matrix A with entries

$$A_{ij} = \begin{cases} 1 & \text{if arc } j \text{ starts at node } i \\ -1 & \text{if arc } j \text{ ends at node } i \\ 0 & \text{otherwise} \end{cases}$$

Note Each column has one -1 and one 1

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

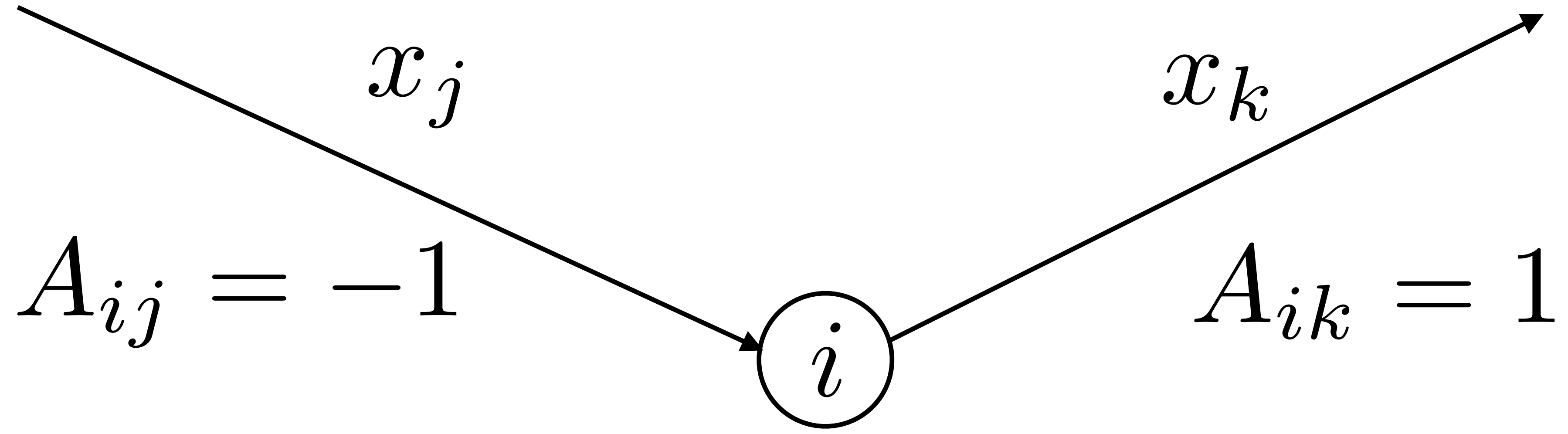
Network flow

flow vector $x \in \mathbf{R}^n$

x_j: flow (of material, traffic, information, electricity, etc) through arc j

total flow leaving node i

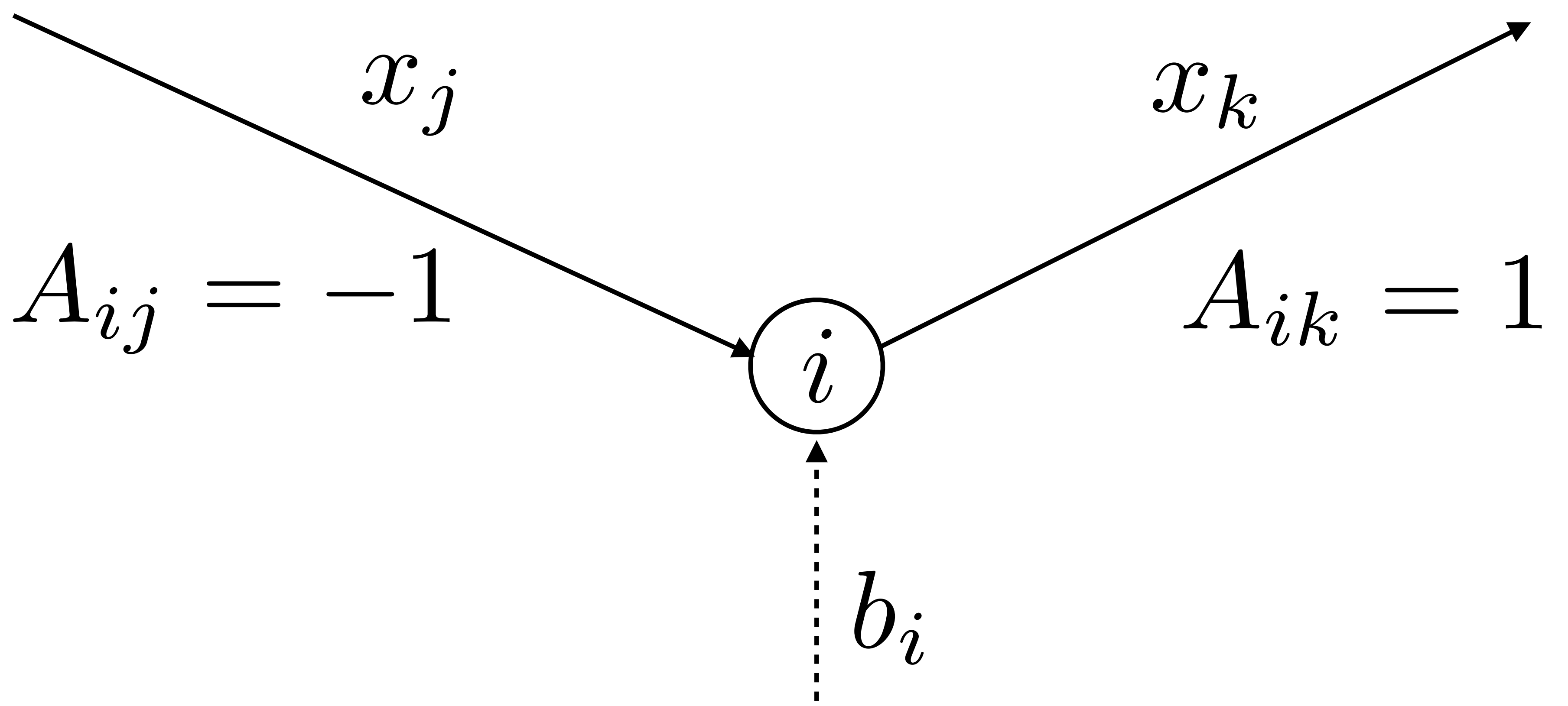
$$\sum_{j=1}^{n} A_{ij} x_j = (Ax)_i$$



External supply

supply vector $b \in \mathbf{R}^m$

- b_i is the external supply at node i (if $b_i < 0$, it represents demand)
- We must have $\mathbf{1}^T b = 0$ (total supply = total demand)



Balance equations

$$\sum_{j=1}^{n} A_{ij} x_j = (Ax)_i = b_i, \quad \text{for all } i \longrightarrow Ax = b$$

Total leaving flow: $(Ax)_i$

Supply: b_i

Minimum cost network flow problem

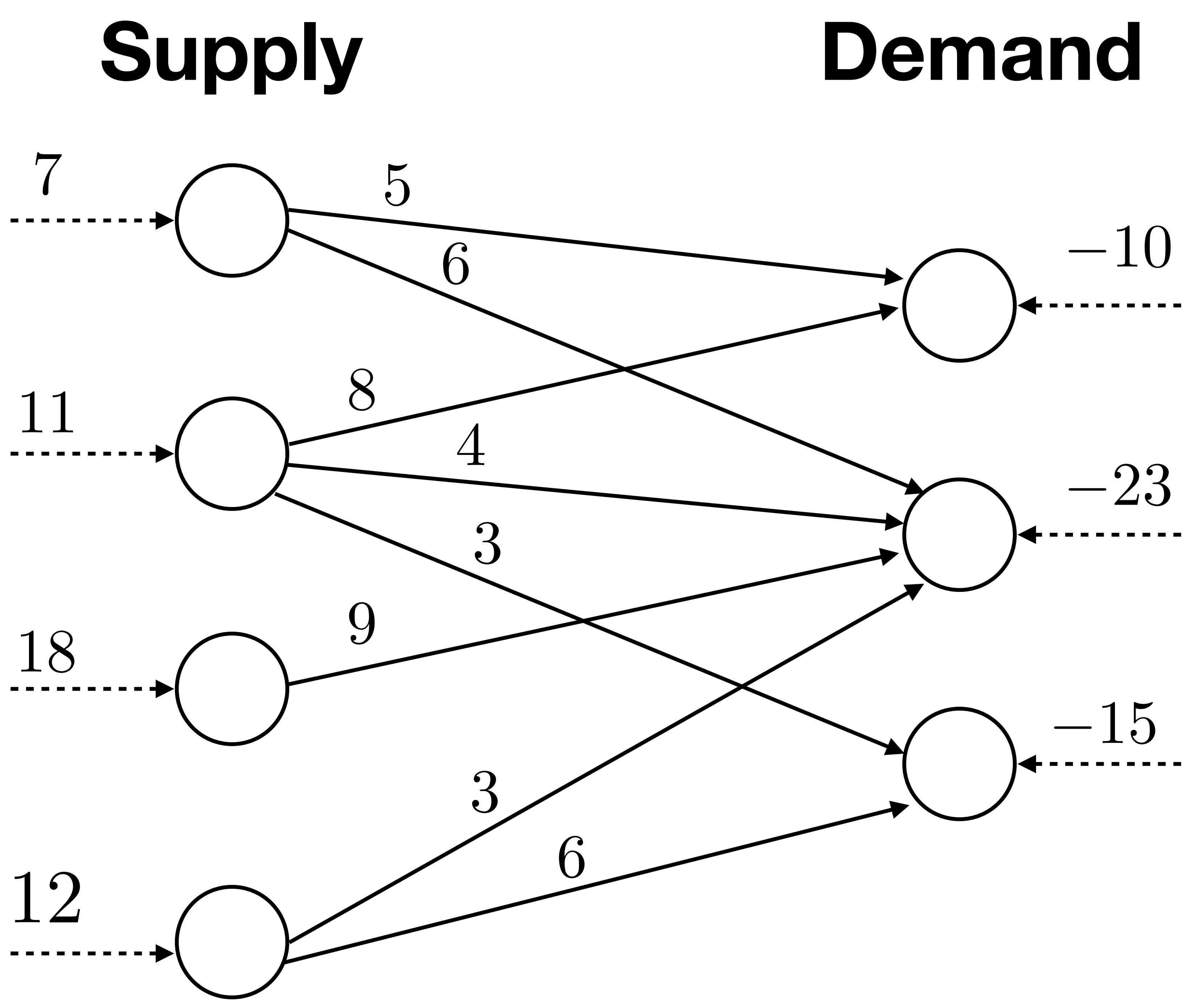
Minimum cost network flow problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & 0 \leq x \leq u \end{array}$$

- c_i is unit cost of flow through arc i
- Flow x_i must be nonnegative
- u_i is the maximum flow capacity of arc i
- Many network optimization problems are just special cases

Example
Transportation

Goal ship $x \in \mathbf{R}^n$ to satisfy demand



(arc costs shown)
All capacities 20

$c = (5, 6, 8, 4, 3, 9, 3, 6)$

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \end{bmatrix}$$

$b = (7, 11, 18, 12, -10, -23, -15)$

$u = 20\,\mathbf{1}$

Minimum cost network flow

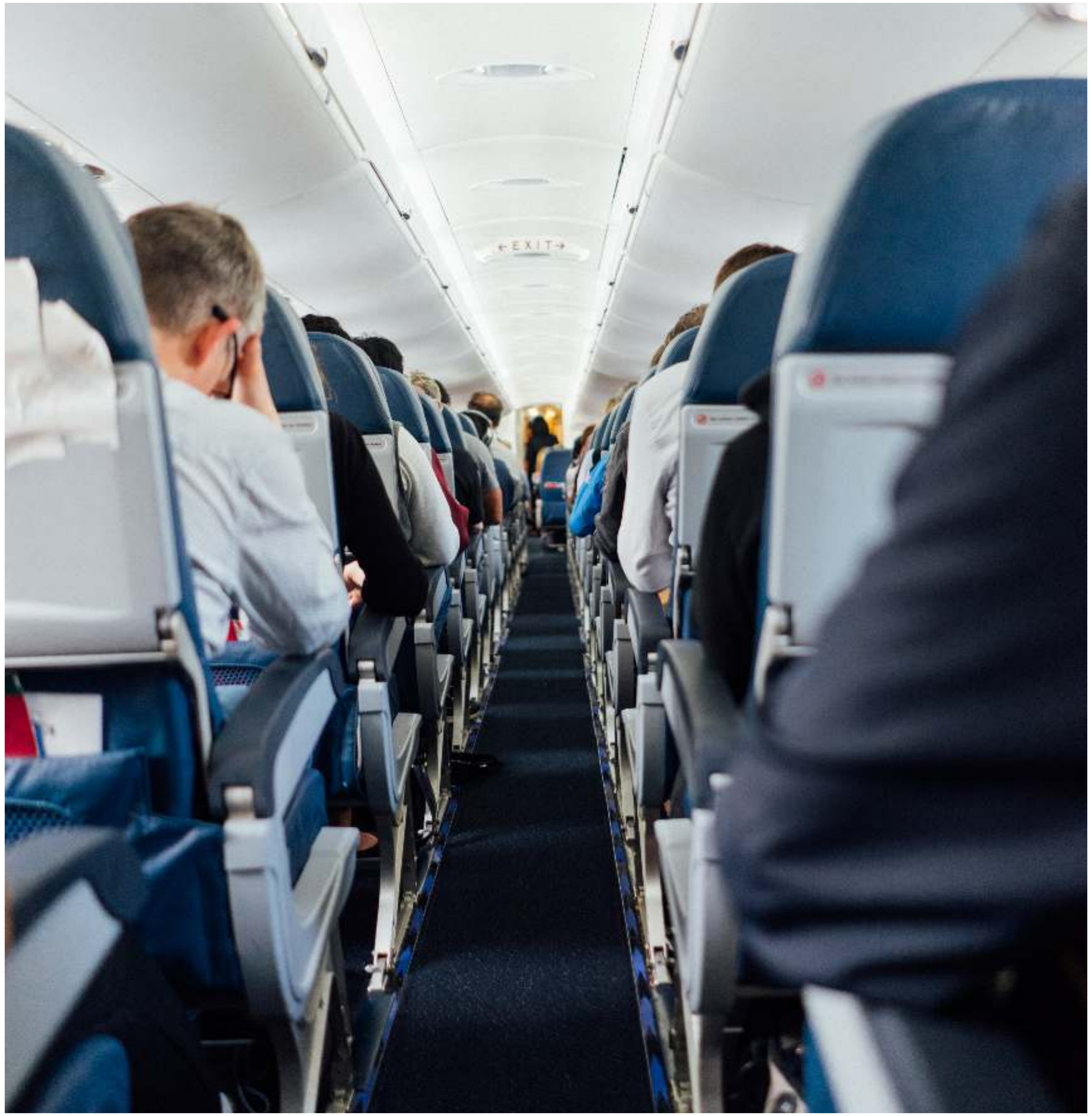
$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & 0 \le x \le u \end{array}$$

$x^\star = (7, 0, 3, 0, 8, 18, 5, 7)$

Example

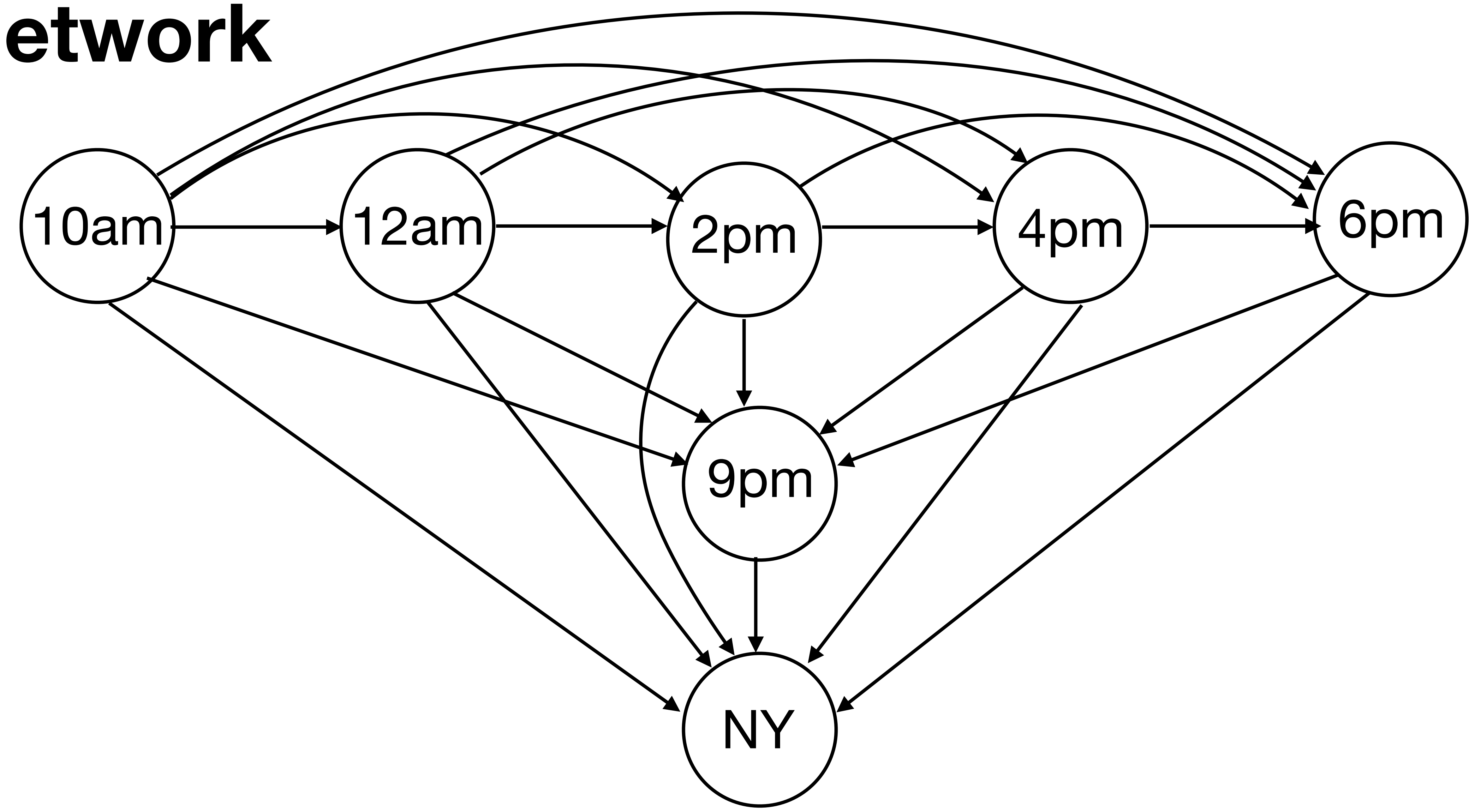
Airline passenger routing

- United Airlines has 5 flights per day from BOS to NY
(10am, 12pm, 2pm, 4pm, 6pm)
- Flight capacities
(100, 100, 100, 150, 150)
- Costs: $50/hour of delay
- Last option: 9pm flight with other company (additional cost $75)
- Today’s reservations
(110, 118, 103, 161, 140)



Airline passenger routing

Network



Decisions

x_j: passengers flowing on arc j

Costs

c_j: cost of moving passenger on arc j

- Between flights: $\$50$/hour
- To 9pm flight: $\$75$ additional
- To NY: $\$0$ (as scheduled)

Supplies

b_i reserved passengers for flight i

- 9pm flight: $b_i = 0$
- NY supply: - total reserved passeng.

Capacities

u_j maximum passengers over arc j

- Between flights: $u_j = \infty$
- To NY: $u_i =$ flight capacity

Network flow formulation

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & 0 \leq x \leq u \end{array}$$

Network flow solutions

Remove arc capacities

Goal: create equivalent network without arc capacities

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & 0 \le x \le u \end{array} \longrightarrow \begin{array}{ll} \text{minimize} & \tilde{c}^T \tilde{x} \\ \text{subject to} & \tilde{A}\tilde{x} = \tilde{b} \\ & \tilde{x} \ge 0 \end{array}$$

Standard form LP with arc-node incidence matrix

Remove arc capacities

Idea: slack variables

$$x_j \le u_j \quad \Rightarrow \quad x_j + s_j = u_j,\ s_j \ge 0$$

$$\cdots + x_j \ldots = b_p$$
$$\cdots - x_j \ldots = b_q$$
$$x_j + s_j = u_j$$

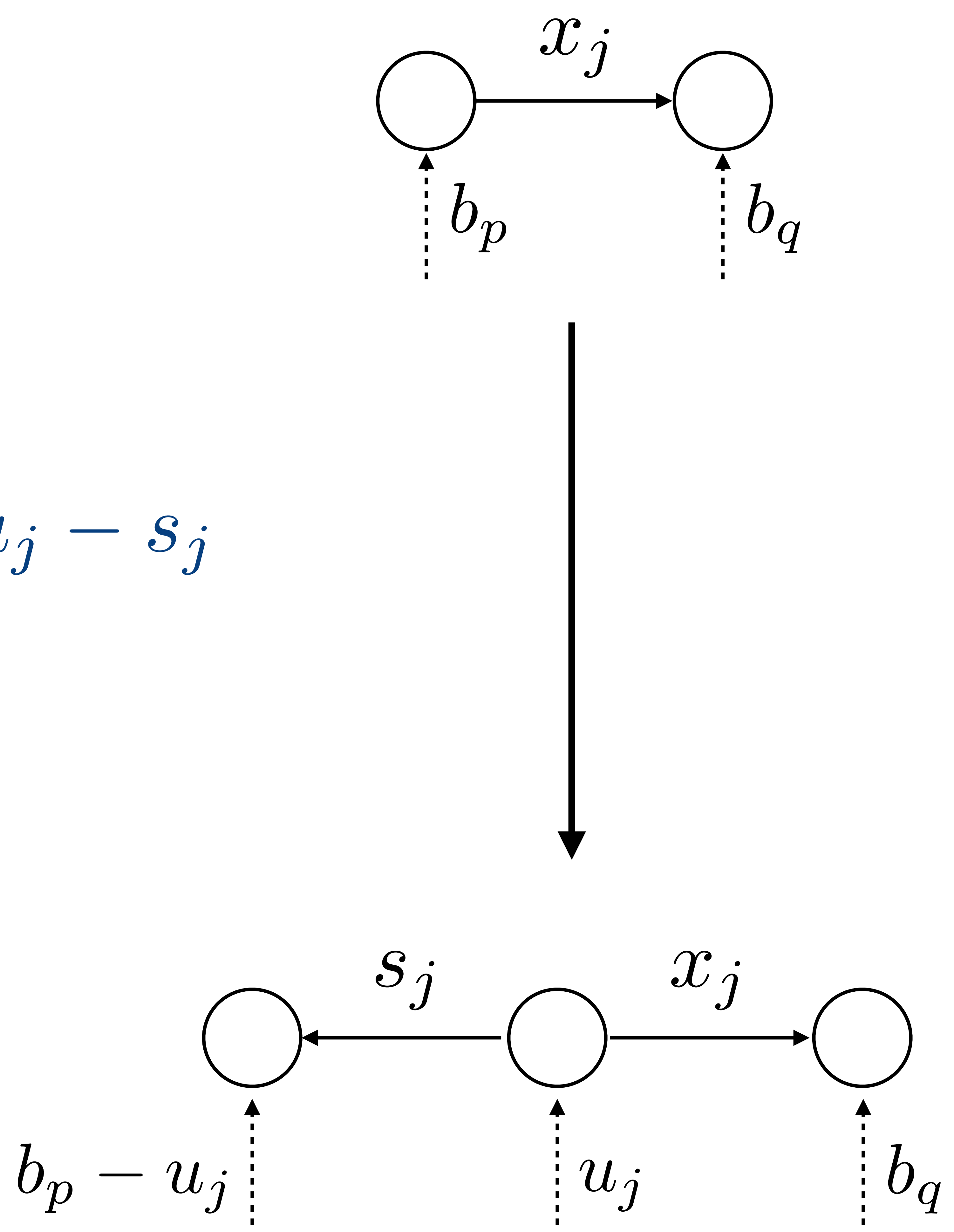
$x_j = u_j - s_j$

Network structure lost
no longer one -1
and one 1 per column

$$\cdots - s_j = b_p - u_j$$
$$\cdots - x_j \ldots = b_q$$
$$x_j + s_j = u_j$$

Network structure
recovered
(new node and new arc)

Nodes/arcs interpretation



Equivalent uncapacitated network flow

$$
\begin{array}{ll}
\text{minimize} & c^T x \\
\text{subject to} & Ax = b \\
& x \geq 0
\end{array}
$$

- A still an arc-node incidence matrix
- Can we say something about the extreme points?

Total unimodularity

A matrix is **totally unimodular** if all its minors are $-1, 0$ or 1
(minor is the determinant of a square submatrix of A)

example: a node-arc incidence matrix of a directed graph

$$A = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 \\ -1 & -1 & 0 & 1 & 0 & -1 \end{bmatrix}$$

properties

- the entries of A_{ij} (i.e., its minors of order 1) are $-1, 0$, or 1
- The inverse of any nonsingular square submatrix of A has entries $+1, -1$, or 0

Integrality theorem

Given a polyhedron $P = \{x \in \mathbf{R}^n \mid Ax = b, \quad x \geq 0\}$

where

- A is totally unimodular
- b is an integer vector

$\longrightarrow$ all the extreme points of P are integer vectors.

Proof

- All extreme points are basic feasible solutions with $x_B = A_B^{-1} b$ and $x_i = 0, \ i \neq B$
- A_B^{-1} has integer components because of total unimodularity of A
- b has also integer components
- Therefore, also x is integral

∎

Implications for network and combinatorial optimization

Minimum cost network flow

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & 0 \leq x \leq u \end{array}$$

→ If b and u are integral solutions $x^\star$ are integral

Integer linear programs

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & 0 \leq x \leq u \\ & x \in \mathbf{Z}^n \end{array}$$

Very difficult in general
(more on this in a few weeks)

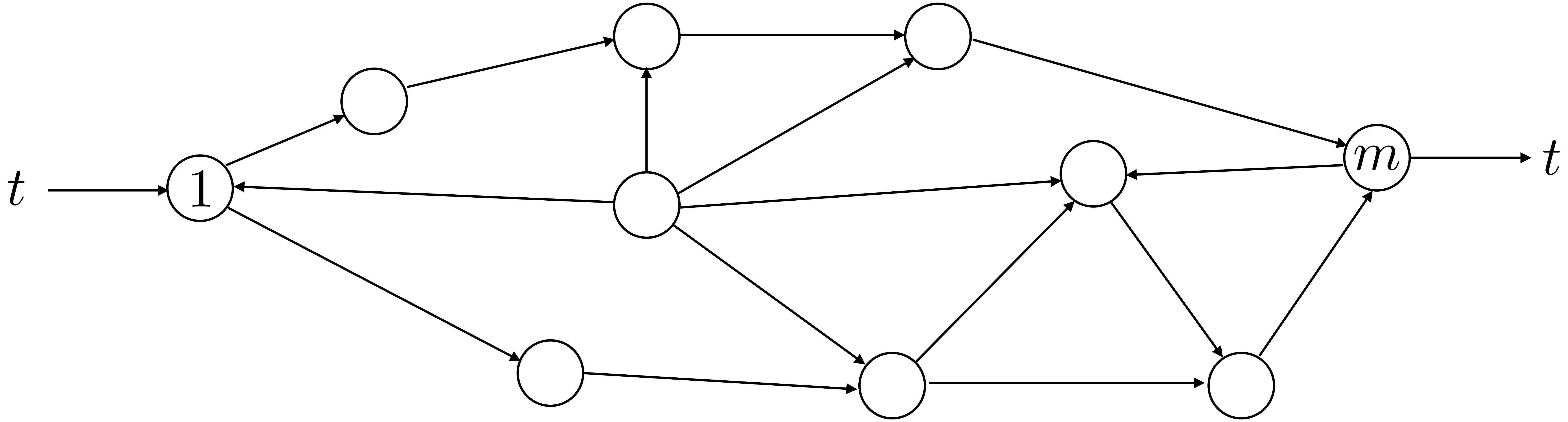
↓

If A totally unimodular and b, u integral, we can relax integrality and solve a fast LP instead

Examples

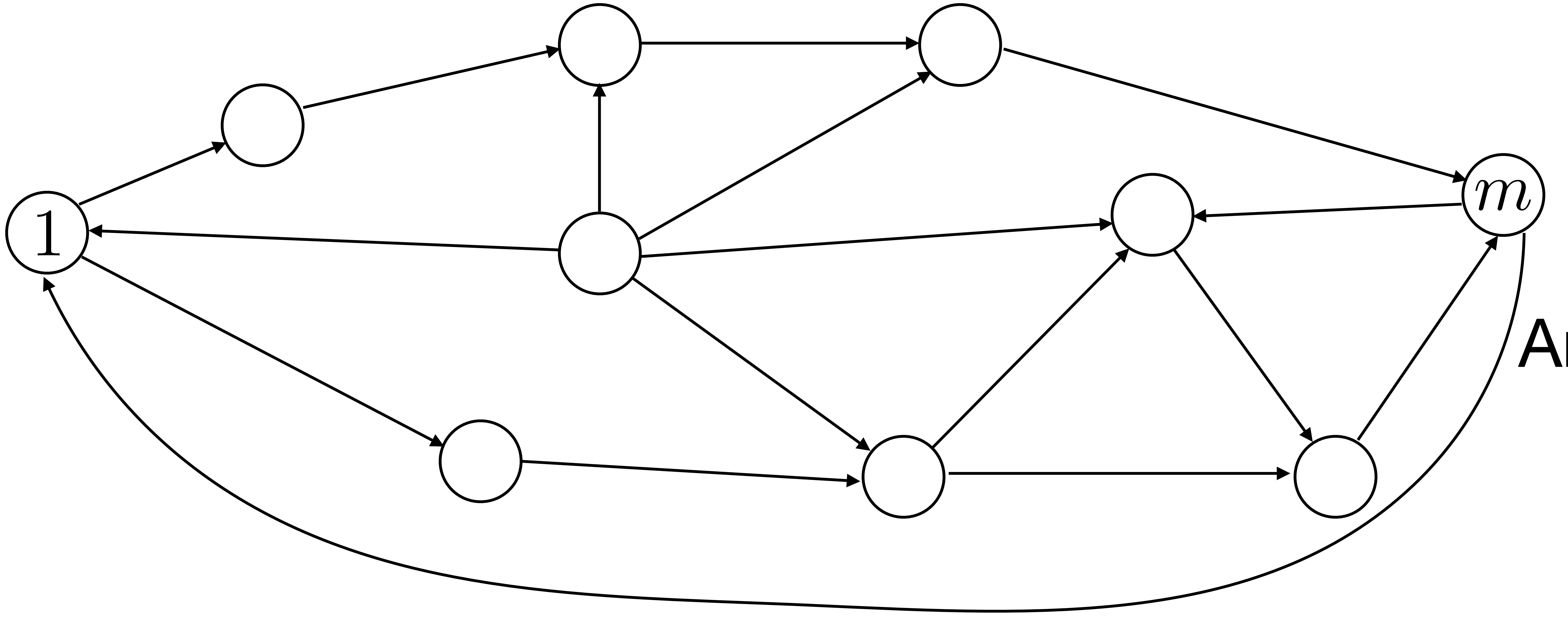
Maximum flow problem

Goal maximize flow from node 1 (source) to node m (sink) through the network



$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & Ax = te \\ & 0 \le x \le u \end{array} \qquad e = (1, 0, \dots, 0, -1)$$

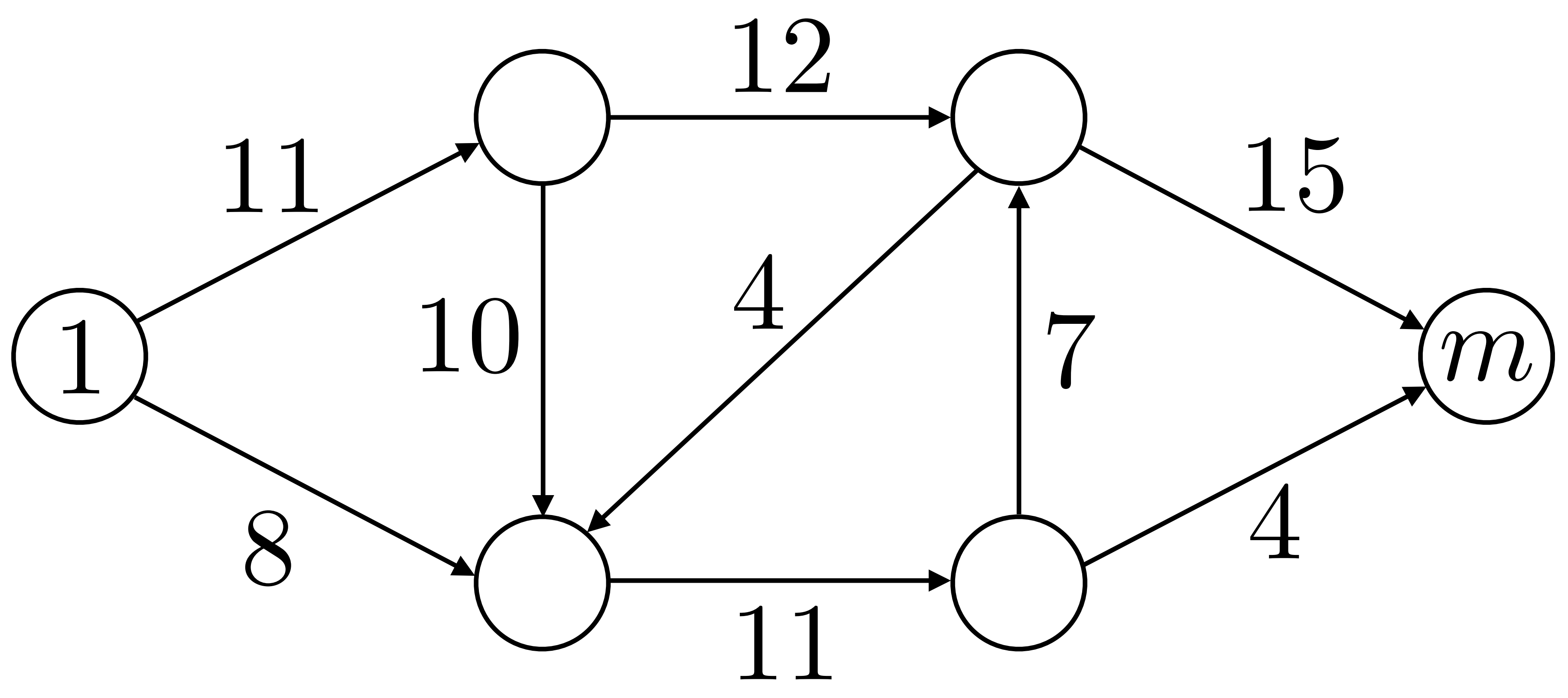
Maximum flow as minimum cost flow



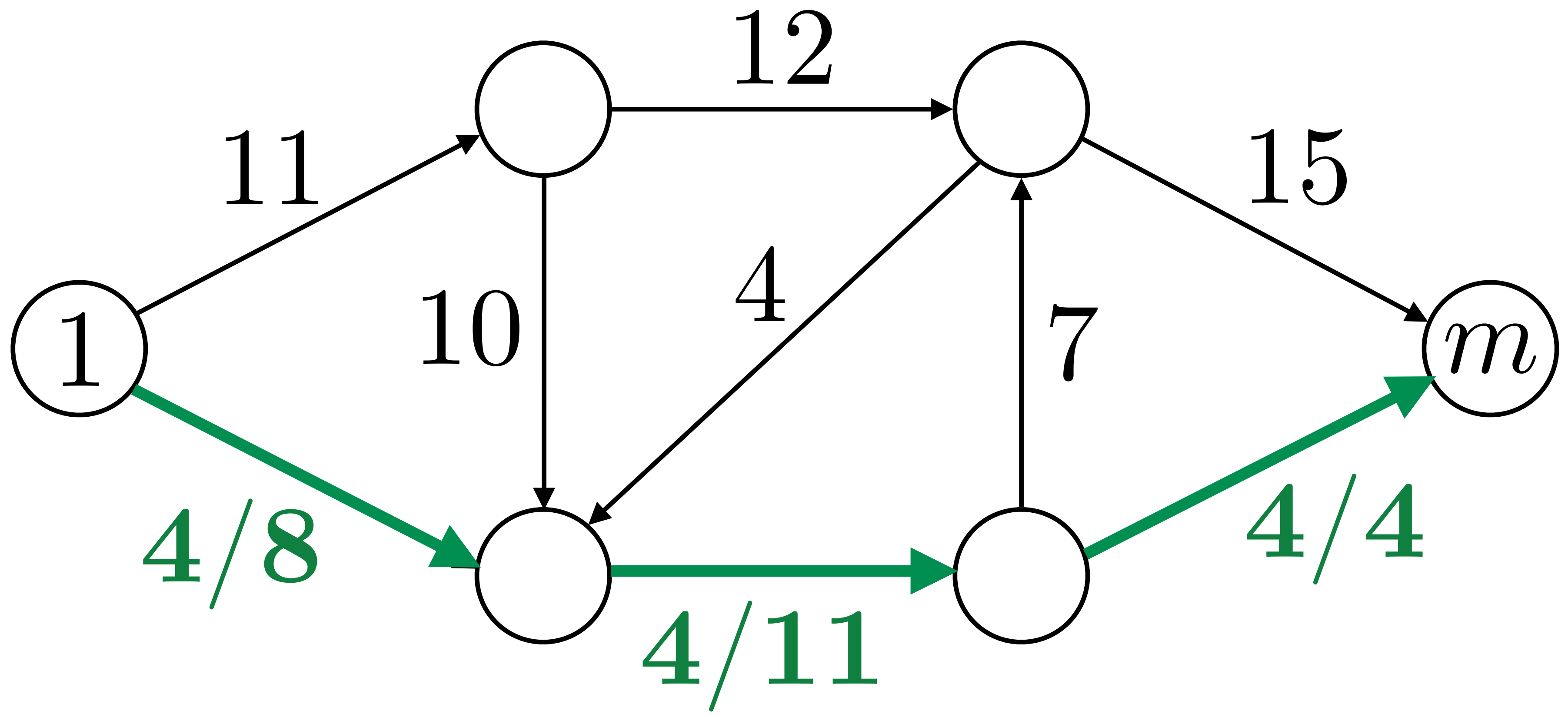
$$\begin{array}{ll} \text{minimize} & -t \\ \text{subject to} & \begin{bmatrix} A & -e \end{bmatrix} \begin{bmatrix} x \\ t \end{bmatrix} = 0 \\ & 0 \leq \begin{bmatrix} x \\ t \end{bmatrix} \leq \begin{bmatrix} u \\ \infty \end{bmatrix} \end{array}$$

Maximum flow example

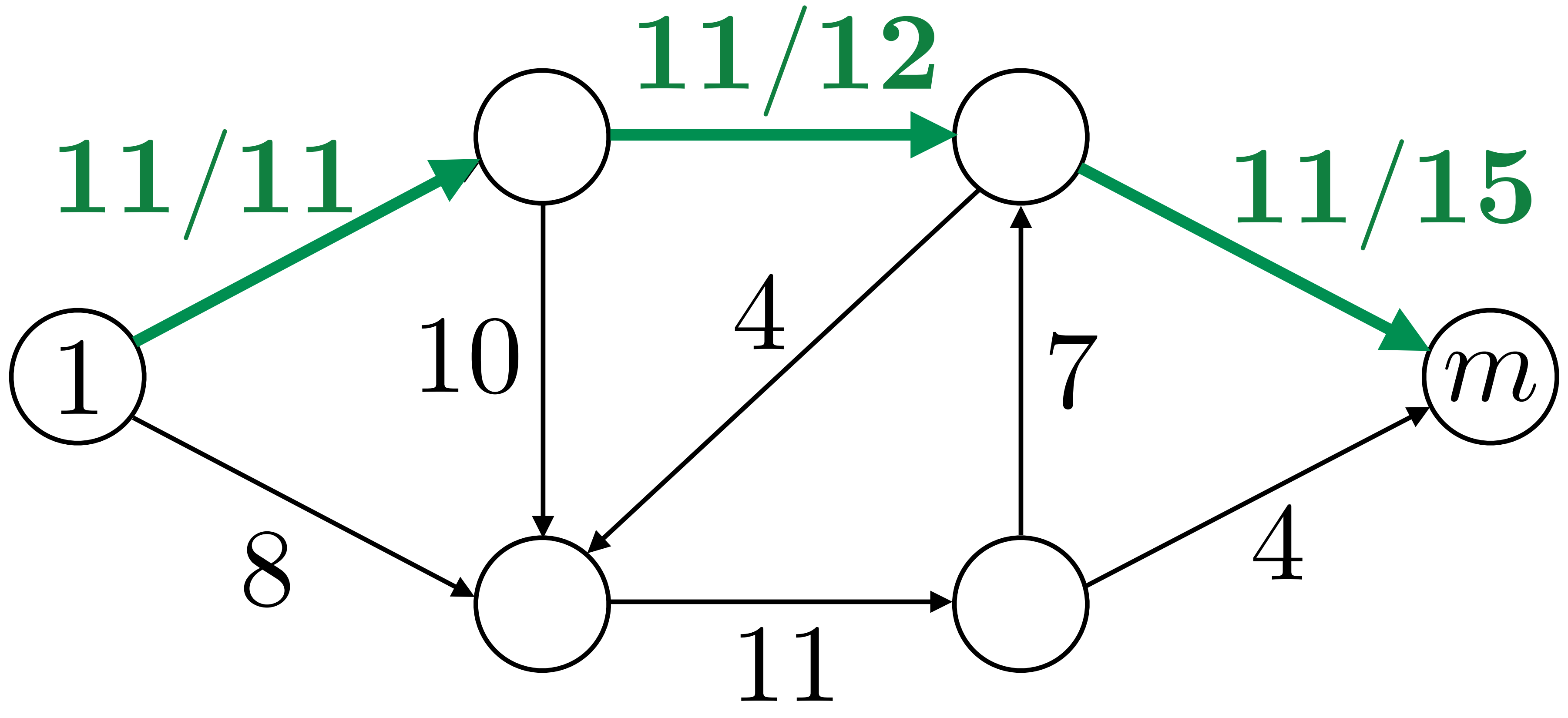
(arc capacities shown)



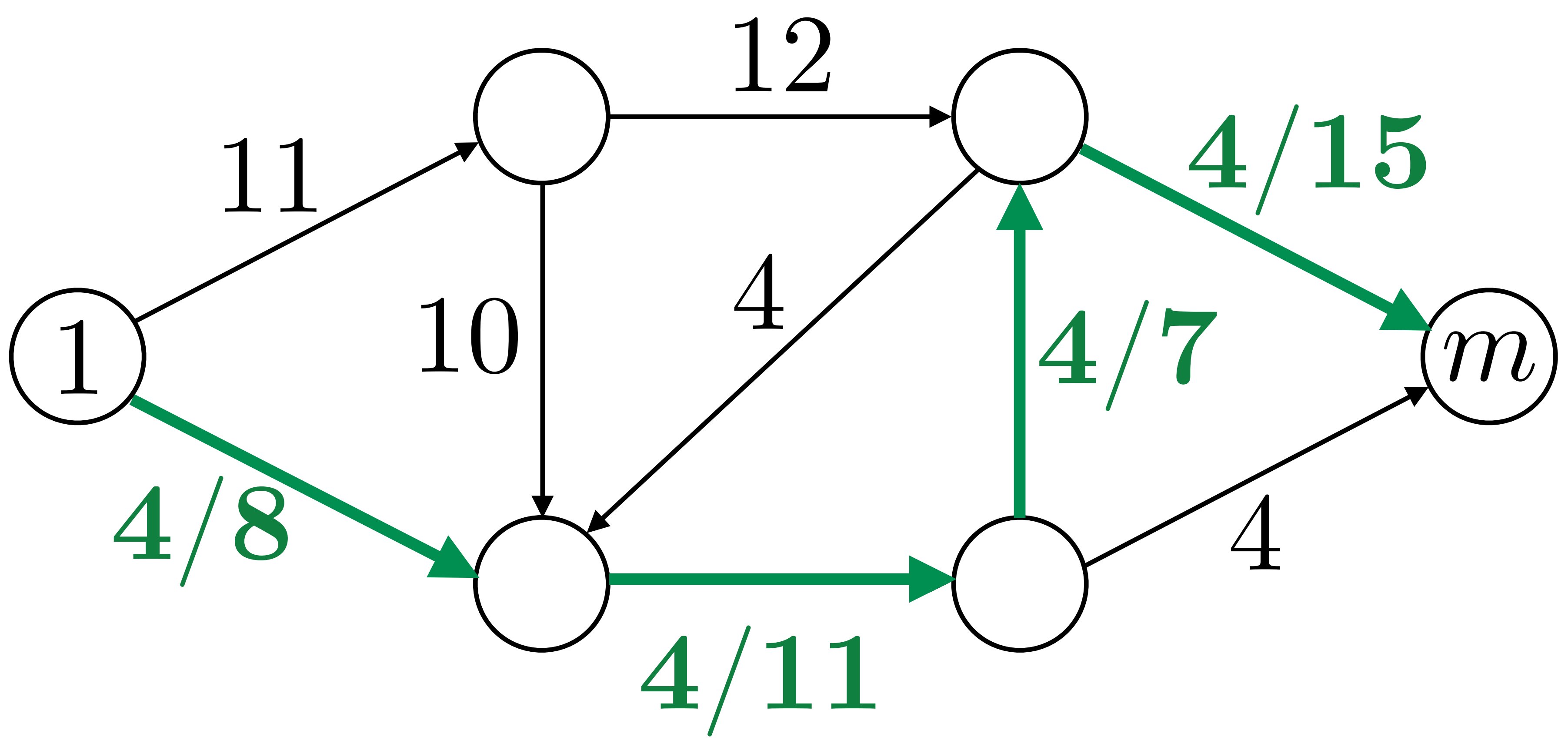
First flow



Second flow



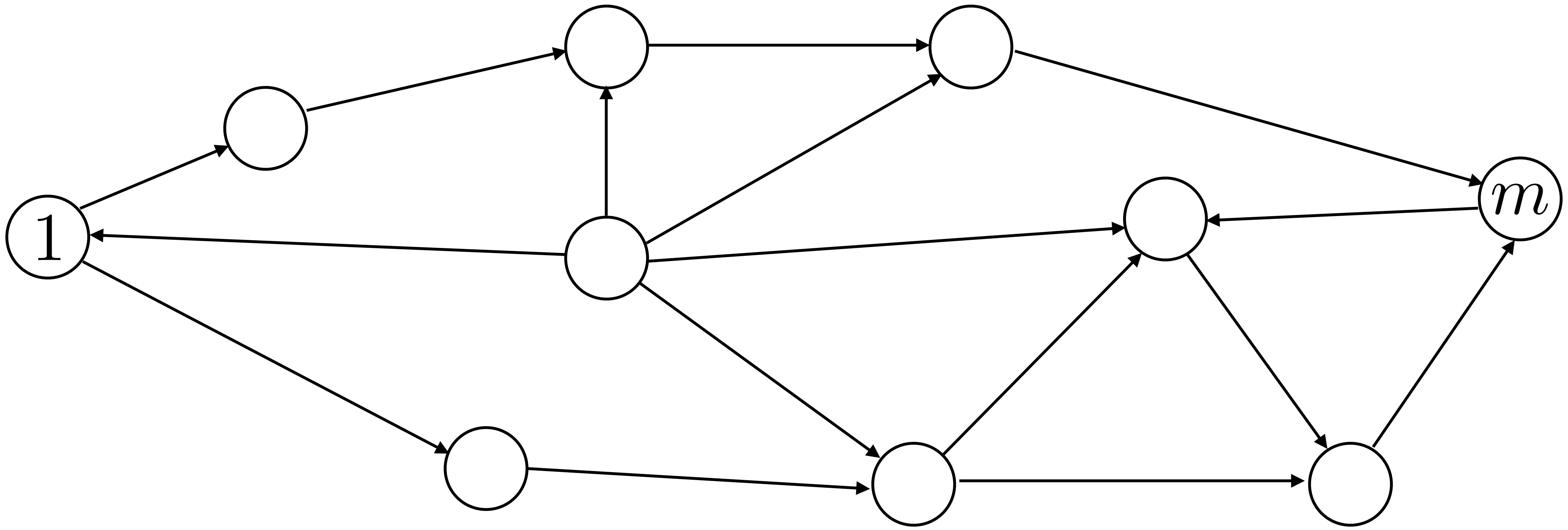
Third flow



Total flow: 19

Shortest path problem

Goal Find the shortest path between nodes 1 and m



paths can be represented as vectors $x \in \{0,1\}^n$

Formulation

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = e \\ & x \in \{0,1\}^n \end{array}$$

- c_j is the "length" of arc j
- $e = (1, 0, \ldots, 0, -1)$
- Variables are binary (include or not arc in path)

Shortest path as minimum cost flow

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = e \\ & x \in \{0,1\}^n \end{array}$$

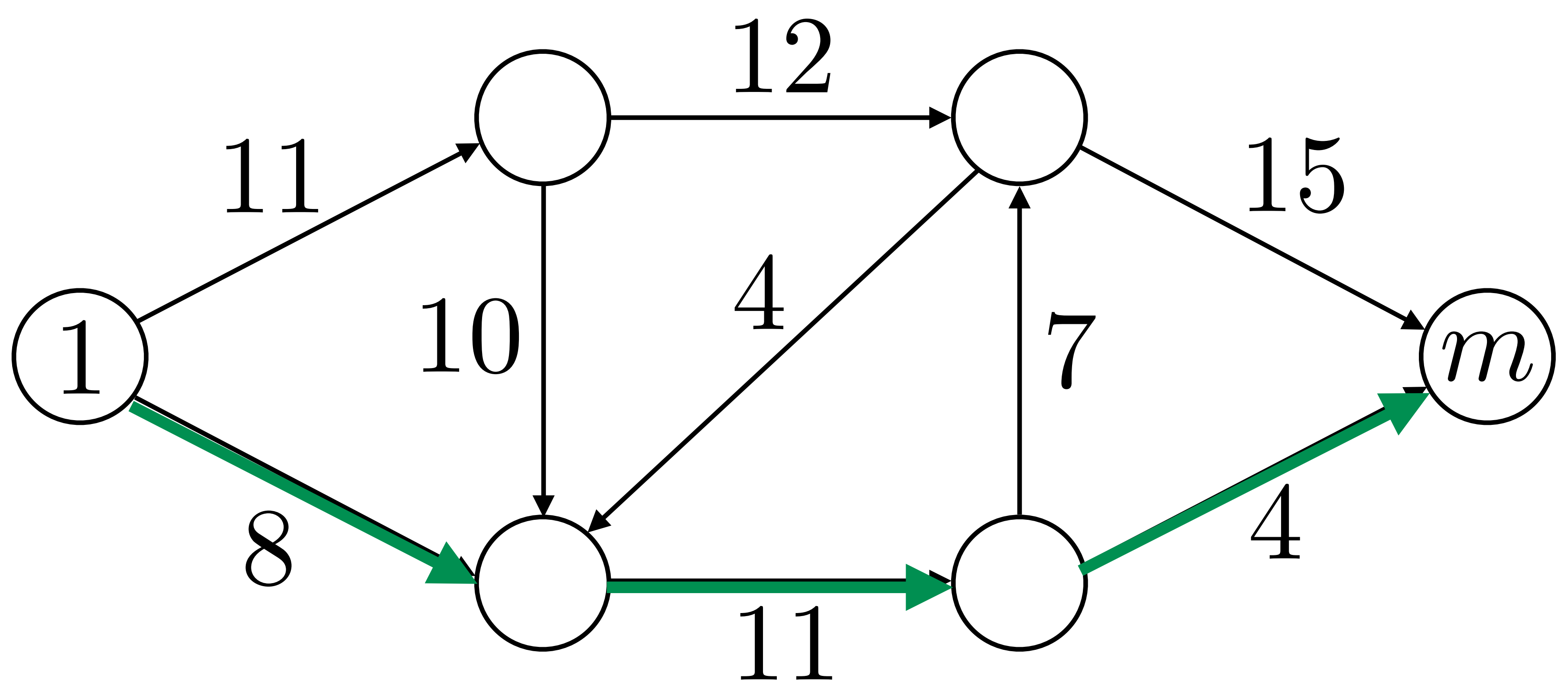
→

Relaxation

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = e \\ & 0 \leq x \leq 1 \end{array}$$

Extreme points satisfy $x_i \in \{0, 1\}$

Example (arc costs shown)



$$c = (11, 8, 10, 12, 4, 11, 7, 15, 4)$$

$$x^\star = (0, 1, 0, 0, 0, 1, 0, 0, 1)$$

$$c^T x^\star = 24$$

Assignment problem

Goal match N persons to N tasks

- Each person assigned to one task, each task to one person
- C_{ij} Cost of matching person i to task j

LP formulation

minimize $$\sum_{i,j=1}^{N} C_{ij} X_{ij}$$

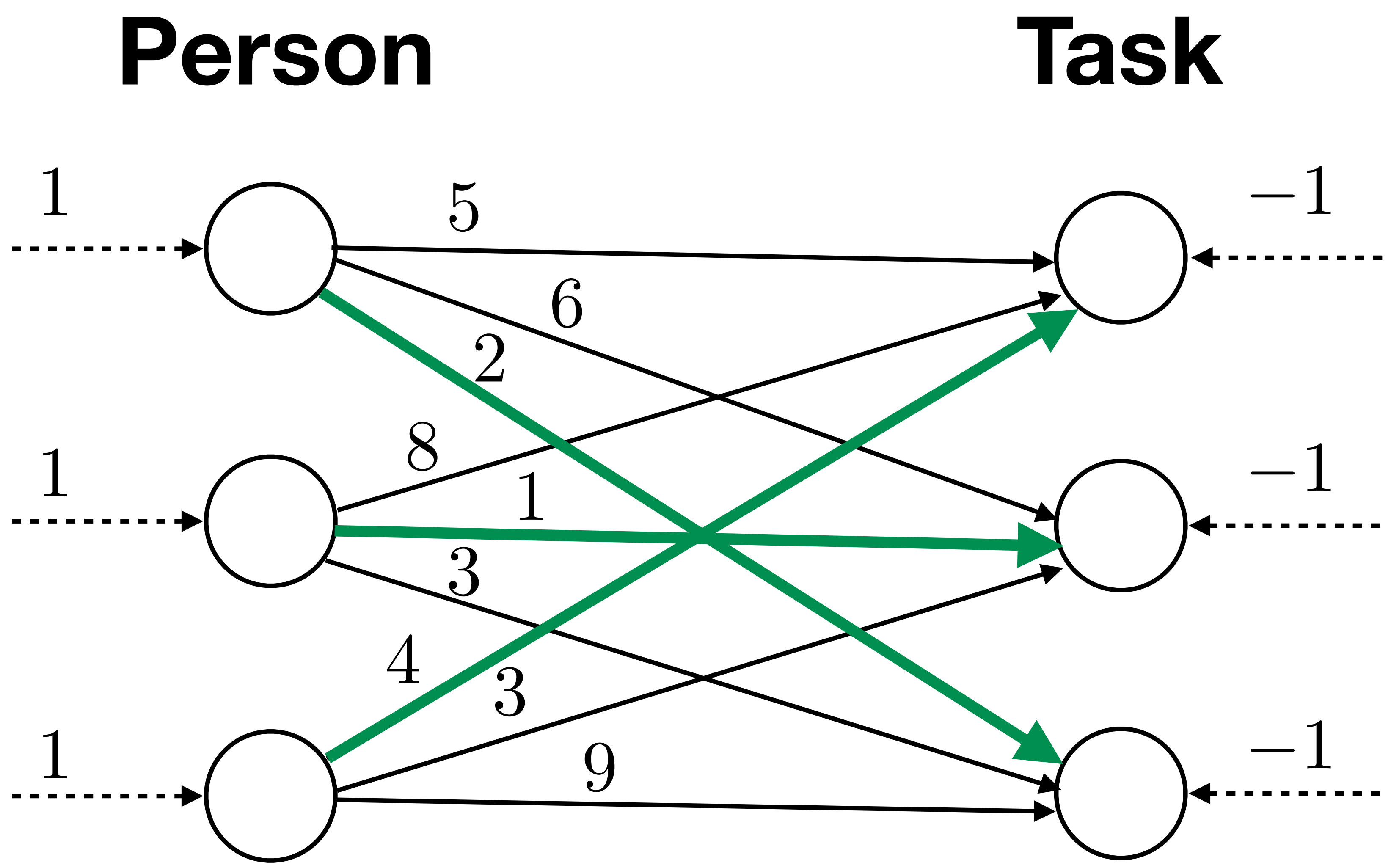
subject to $$\sum_{i=1}^{N} X_{ij} = 1, \quad j = 1, \ldots, N$$

$$\sum_{j=1}^{N} X_{ij} = 1, \quad i = 1, \ldots, N$$

$$X_{ij} \in \{0, 1\}$$

How do you define the network?

Task assignment as minimum cost network flow



(arc costs shown)

$$c = (5, 6, 2, 8, 1, 3, 4, 3, 9)$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \end{bmatrix}$$

$$b = (1, 1, 1, -1, -1, -1)$$

Minimum cost network flow

minimize $c^T x$

subject to $Ax = b$

$0 \leq x \leq \mathbf{1}$

Extreme points satisfy $x_i \in \{0, 1\}$ ⟶

Optimal solution

$$x^\star = (0, 0, 1, 0, 1, 0, 0, 0, 1)$$
$$c^T x^\star = 7$$

Network optimization

Today, we learned to:

- **Model** flows across networks
- **Formulate** minimum cost network flow problems
- **Analyze** network flow problem solutions (integrality theorem)
- **Formulate** maximum-flow, shortest path, and assignment problems as minimum cost network flows

References

- D. Bertsimas and J. Tsitsiklis: Introduction to Linear Optimization
  - Chapter 7: Network flow problems
- R. Vanderbei: Linear Programming
  - Chapter 14: Network Flow Problems
  - Chapter 15: Applications

Next lecture

- Interior point algorithms