Convex Optimization

Networks

Today

- Duality at work: network applications. . .

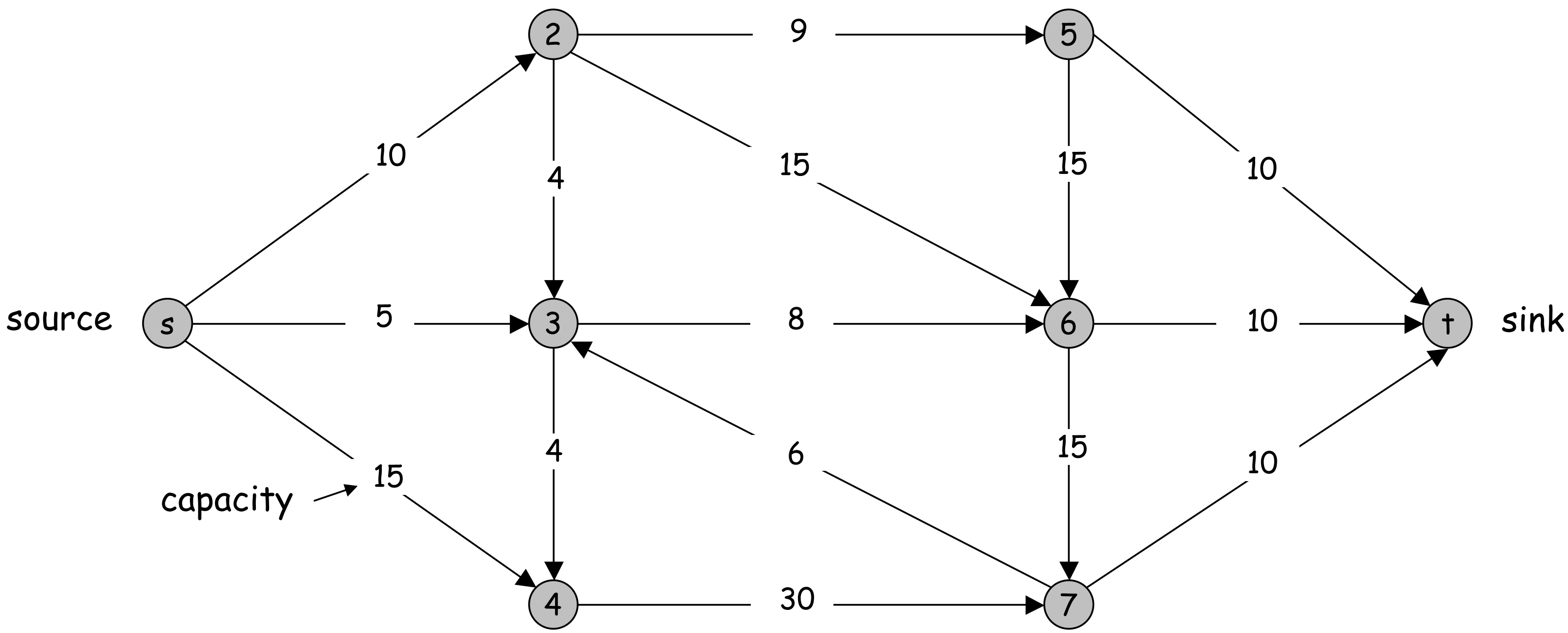
Convex Optimization

- Most duals have a very natural interpretation
- Numerical software generally solve both at the same time (more later)
- Provide a lot of information beyond sensitivity
- Also give a definitive proof of convergence
- Many duals for one problem

Duality: applications

Duality: network flow problems

Let start with a simple network:



Network flow problems

Network characteristics:

- Flow through each arc in one direction only
- Source s, sink in t.
- Each link has a fixed capacity
- No parallel edges, self-loops, etc
- No edges leading to s, no edges leaving t

Simple question: What is the maximum throughput in this network?

Network flow problems

Model formulation:

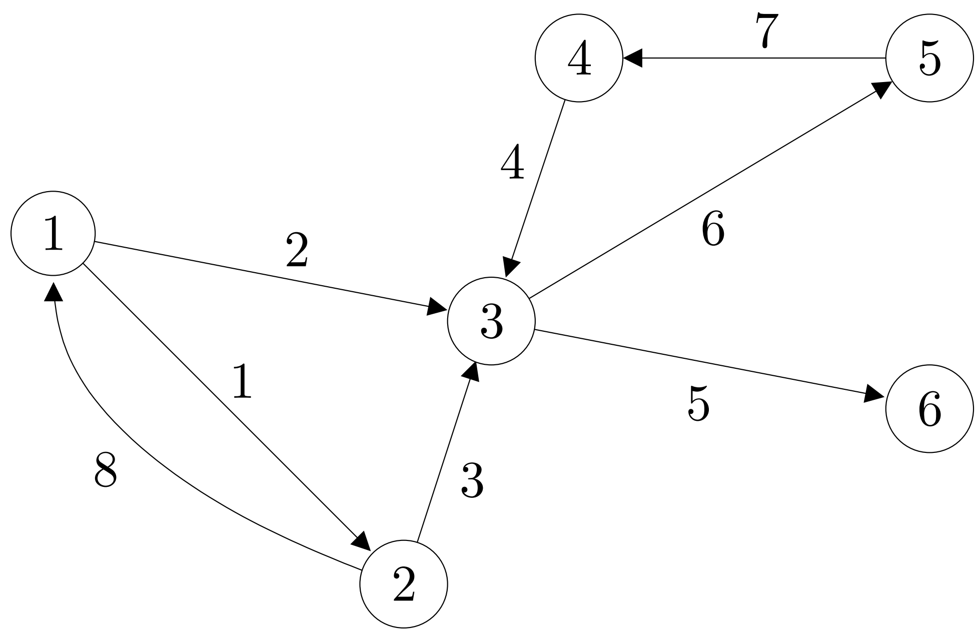
- We can define the network's **incidence matrix**:

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

- By construction, we have $\mathbf{1}^T A = 0$.
- We note x_i the **flow** through arc i. Could be negative if the flow is going against the direction of the arc.

Network flow problems

example ($m = 6$, $n = 8$)

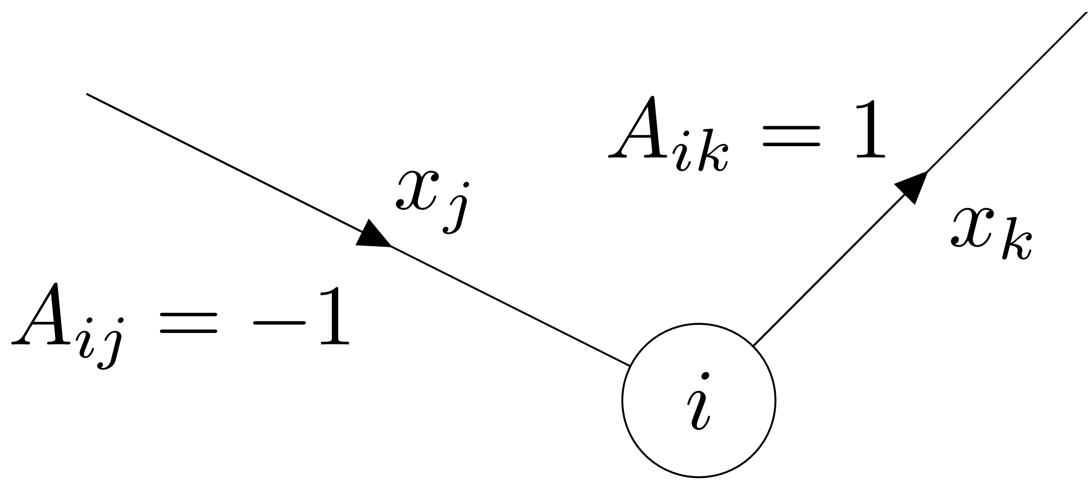


$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

Network flow problems

We can compute the **total flow** leaving node i as:

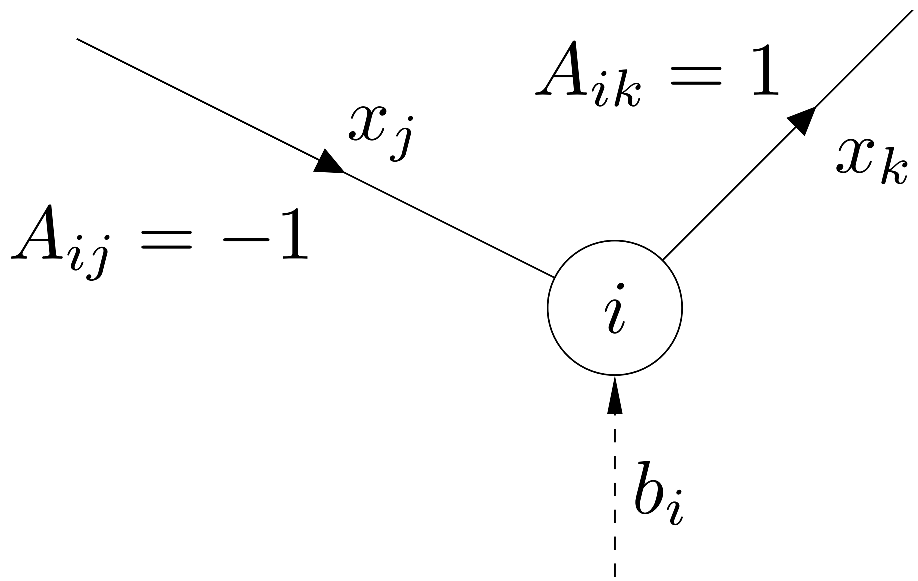
$$\sum_{j=1}^{n} A_{ij} x_j = (Ax)_i$$



Network flow problems

We define the **supply vector** $b \in \mathbf{R}^m$:

- $b_i > 0$: external flow entering the network at node i
- $b_i < 0$: flow leaving the network at node i
- We have a balanced flow: $\mathbf{1}^T b = 0$ (inflow = outflow)



The **balance equations** are written: $Ax = b$

Network flow problems

We consider **minimum cost network flow** problems:

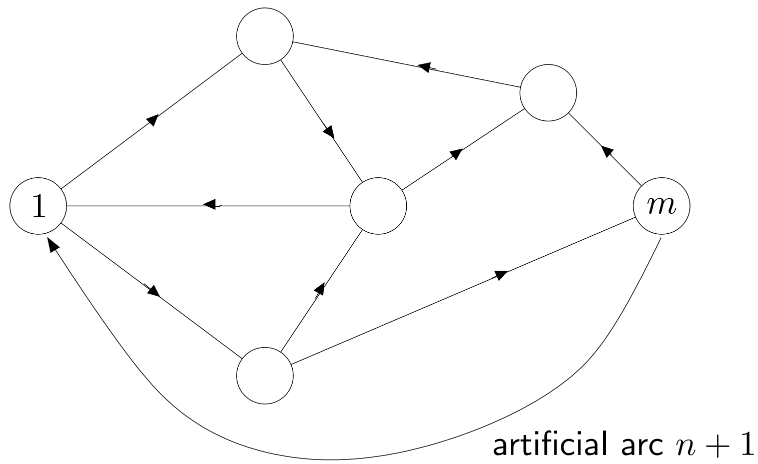
$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & l \leq x \leq u \end{array}$$

- c_i is the cost of one unit of flow going through node i
- l_j and u_j are upper and lower bounds on thee flow through arc j

This problem class includes maximum flow problems, and many others. . .

Network flow problems

We introduce an artificial arc in the network, from the sink to the source:



To maximize the flow from 1 to m, we simply attach a negative cost to this artificial arc, and solve the following minimum cost network flow problem:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & [A\ , -e] \begin{bmatrix} x \\ t \end{bmatrix} = 0 \\ & 0 \leq x \leq u \end{array}$$

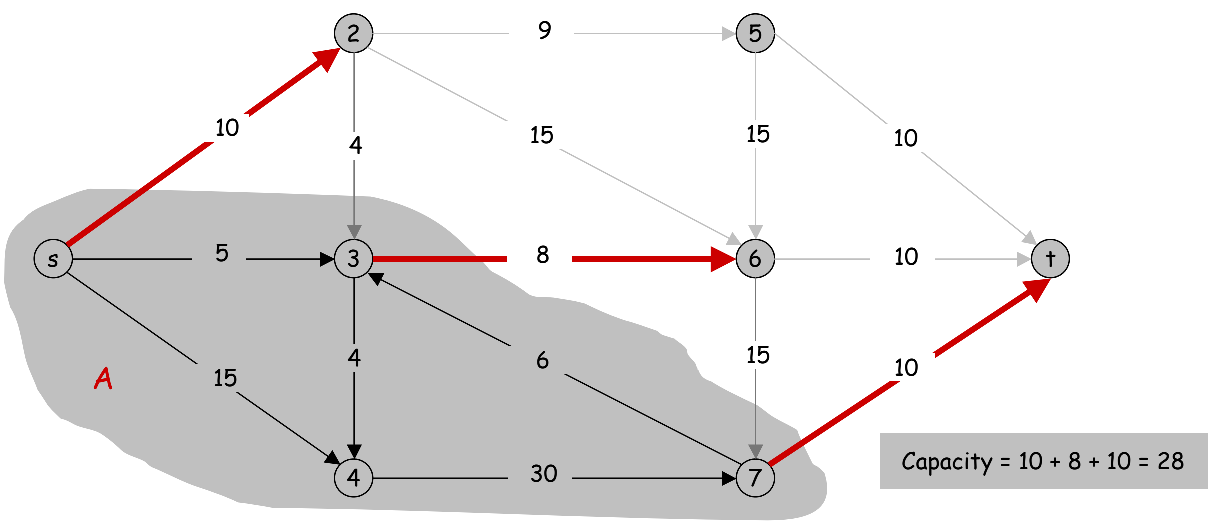
with $e = (1, 0, \ldots, 0, -1)$. This is a **maximum flow problem**.

Network flow problems

We can also define cuts in the network:

- An (s,t) **cut** of the network is a partition of the nodes in two sets U and V such that $s \in U$ and $t \in V$.
- The **capacity** of a cut (U,V) is computed as:

$$cap(U,V) = \sum_{\{\text{arc } j \text{ leaves } U\}} u_j$$



Network flow problems

In this problem, an admissible **flow** satisfies:

- Capacity constraints: $0 \leq x_j \leq u_j$
- Conservation constraints: $(Ax)_i = 0$, when $i \neq s, t$

The **value** of a flow x is the total flow coming out of the source node s:

$$val(x) = \sum_{\{\text{arc } j \text{ leaves } s\}} x_j$$

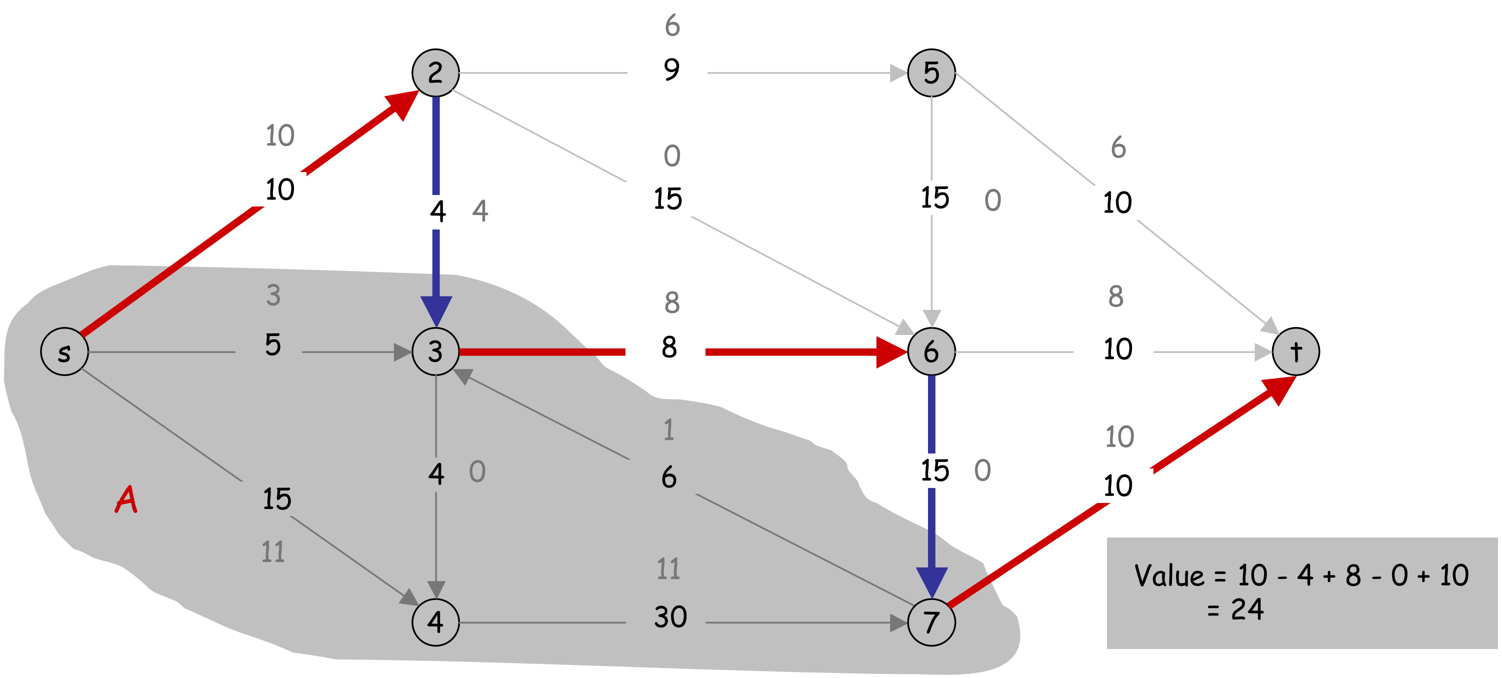
We write $cut(U, V)$ the **net flow** coming out of a cut (U, V):

$$cut(U, V) = \sum_{\{\text{arc } j \text{ leaves } U\}} x_j - \sum_{\{\text{arc } j \text{ enters } U\}} x_j$$

We have the following **flow value lemma**. If $s \in U$ and $t \in V$ then

$$val(x) = cut(U, V)$$

which means that the net flow across the cut is equal to the flow leaving s



Network flow problems

Proof is easy. . . By conservation (only the terms below with $i = s$ are nonzero) we have:

$$val(x) = \sum_{\{\text{arc } j \text{ leaves } s\}} x_j$$

$$= \sum_{\{\text{node } i \text{ in } U\}} \left(\sum_{\{\text{arc } j \text{ leaves } i\}} x_j - \sum_{\{\text{arc } j \text{ enters } i\}} x_j \right)$$

Which is, after simplification:

$$= \sum_{\{\text{arc } j \text{ leaves } U\}} x_j - \sum_{\{\text{arc } j \text{ enters } U\}} x_j$$

$$= cut(U, V)$$

Network flow problems

We can get another result: $val(x) \leq cap(U,V)$ which says that the **value** of the flow x cannot exceed the **capacity** of the cut

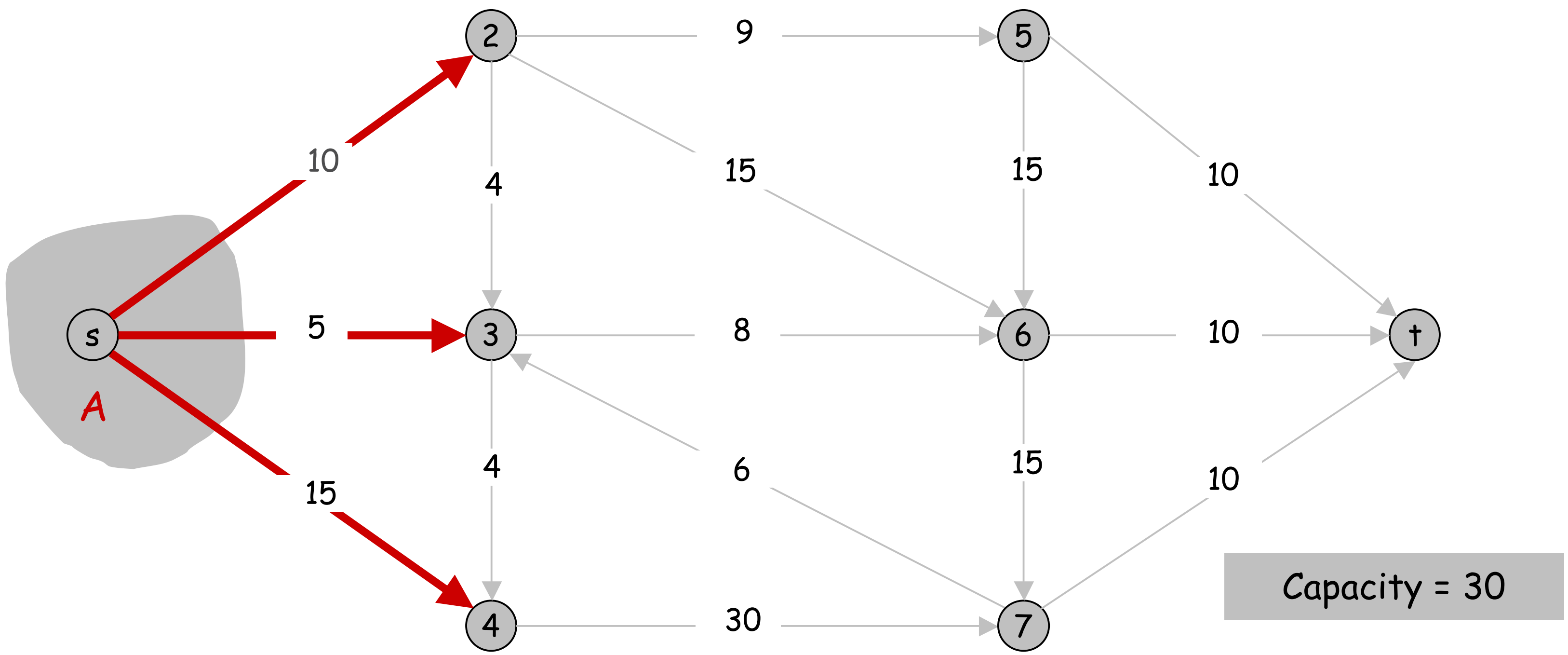
Proof also simple:

$$\begin{aligned} val(x) &= \sum_{\{\text{arc } j \text{ leaves } U\}} x_j - \sum_{\{\text{arc } j \text{ enters } U\}} x_j \\ &\leq \sum_{\{\text{arc } j \text{ leaves } U\}} x_j \\ &\leq \sum_{\{\text{arc } j \text{ leaves } U\}} u_j \\ &= cap(U,V) \end{aligned}$$

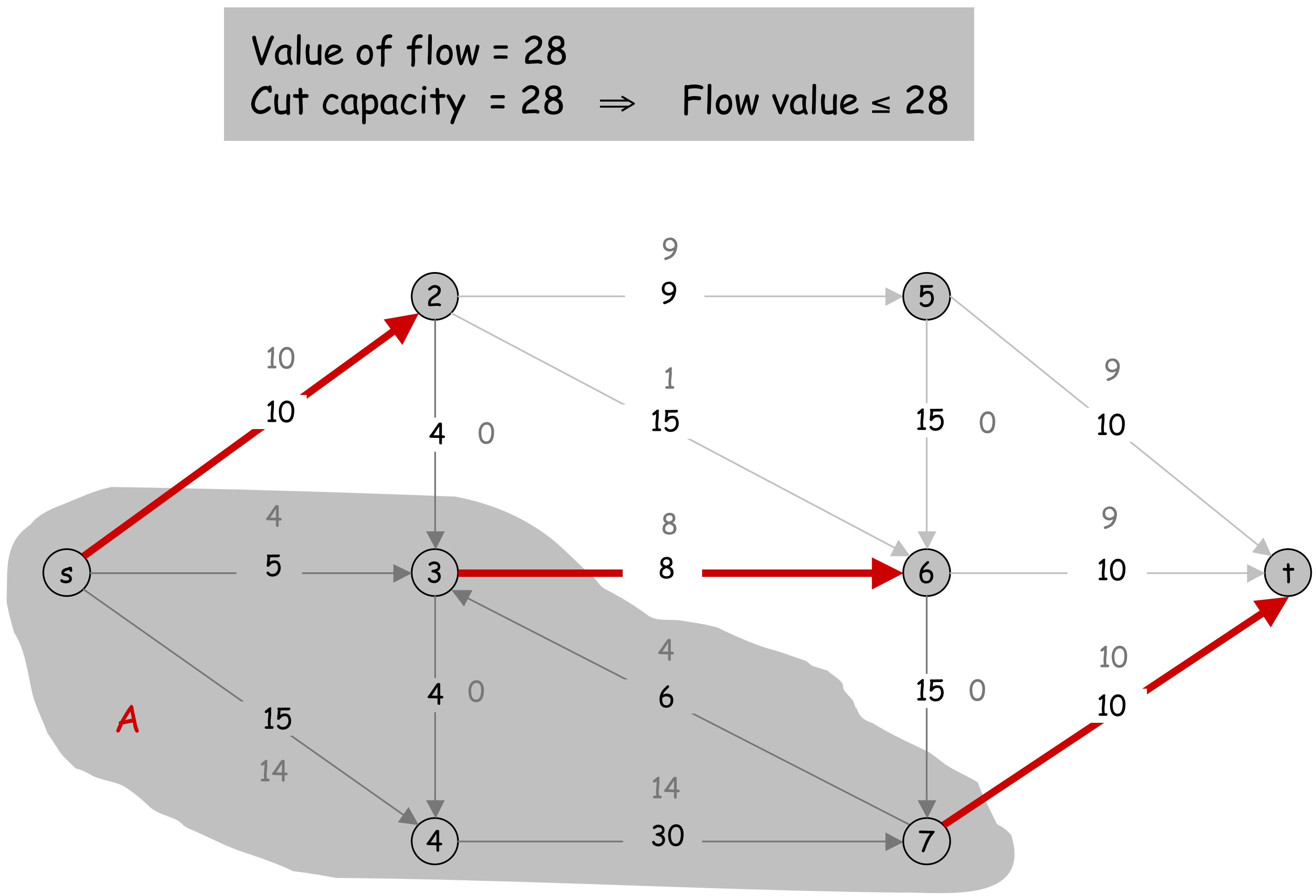
Network flow problems

Illustration:

Cut capacity = 30 ⇒ Flow value ≤ 30



Theorem (Max Flow - Min Cut): The value of the maximum flow is equal to the capacity of the minimum cut.



Network flow problems

Intuition:

- Each cut (U, V) such that $s \in U$ and $t \in V$ gives an **upper bound** on the maximum flow through the network
- Similarly, each flow through the network gives a lower bound on the capacity of such cuts (U, V)
- If we find a flow x and a cut (U, V) such that $val(x) = cap(U, V)$ we know that both are necessarily **optimal**

Network flow problems

This means that the two following problems are closely related:

Maximum Flow:

$$\begin{array}{ll} \text{maximize} & val(x) \\ \text{subject to} & Ax = 0 \\ & 0 \leq x \leq u \end{array}$$

Minimum Cut:

$$\begin{array}{ll} \text{minimize} & cap(U, V) \\ \text{subject to} & s \in U,\ t \in V \\ & U + V = [1, m] \end{array}$$

In particular, both problems have the same **optimal value**

Network flow problems

Can we write the **minimum cut** as a linear program? Consider:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in\mathcal{V}} y_{ij}u_{ij} \\
\text{subject to} \quad & y_{ij} + z_j - z_i \geq 0 \qquad (i,j)\in\mathcal{V} \\
& y_{ij} \geq 0
\end{aligned}
$$

in the variables y and z, where $(i,j)\in\mathcal{V}$ means that there is a link going from i to j, with capacity given by u_{ij}.

Using y and z we define the following cut (U,V) with $s\in U$ and $t\in V$:

$$
\begin{cases}
\text{node } i \text{ in } U \text{ if } z_i > 0 \\
\text{node } i \text{ in } V \text{ if } z_i = 0
\end{cases}
$$

We have of course $z_s = 1$ and $z_t = 0$.

Network flow problems

- By construction $z_s = 1$ so the first constraints are:

$$y_{sj} + z_j \geq 1, \quad (s, j) \in \mathcal{V}$$

- Then, two things can happen at a solution:
  - $y_{sj} = 1$ with $z_j = 0$ and all the following y_{jk} and z_k can be zero
  - $y_{sj} = 0$ with $z_j = 1$ and we get the same equation for the next node:

$$y_{jk} + z_k \geq 1, \quad (j, k) \in \mathcal{V}$$

- This means that the set of nodes such that $z_j = 1$ defines a **cut**.
- Because of the objective, it will be the minimum cut.

Max flow - min cut

The **maximum flow** problem was:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & [A\ , -e] \left[\begin{array}{c} x \\ t \end{array} \right] = 0 \\ & 0 \leq x \leq u \end{array}$$

with $e = (1, 0, \ldots, 0, -1)$. Its **Lagrangian** was:

$$L(x, y, z) = c^T x + z^T [A\ \ -e] \left[\begin{array}{c} x \\ t \end{array} \right] + y^T (x - u)$$

for $x \geq 0$. The Lagrange **dual** function is then defined as

$$\begin{array}{rl} g(y, z) & = \inf_{x \geq 0} L(x, y, z) \\ & = \inf_{x \geq 0} x^T \left(c + y + \left[\begin{array}{c} A^T \\ -e \end{array} \right] z \right) - u^T y \end{array}$$

This minimization yields either $-\infty$ or $-u^T y$, so:

$$g(y,z) = \begin{cases} -u^T y & \text{if } \left(c + y + \begin{bmatrix} A^T \\ -e \end{bmatrix} z\right) \geq 0 \\ -\infty & \text{otherwise} \end{cases}$$

This means that the **dual** of the maximum flow problem is written:

$$\begin{array}{rll} - & \text{minimize} & u^T y \\ & \text{subject to} & c + y + \begin{bmatrix} A^T \\ -e \end{bmatrix} z \geq 0 \end{array}$$

Compare to the **minimum cut problem**:

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in\mathcal{V}} y_{ij} u_{ij} \\ \text{subject to} & y_{ij} + z_j - z_i \geq 0, \quad (i,j) \in \mathcal{V} \\ & y_{ij} \geq 0 \end{array}$$

The two problems are **identical**. . .

Duality: examples

- The **max flow - min cut** result is a particular case of linear programming duality
- Both primal and dual solutions have direct interpretations