# CS/ECE/ISYE524: Introduction to Optimization – Linear Optimization Models

Jeff Linderoth

Department of Industrial and Systems Engineering
University of Wisconsin-Madison

March 4, 2024

## Duality review

Every LP has a dual, which is also an LP.

- Every primal constraint corresponds to a dual variable
- Every primal variable corresponds to a dual constraint

| Minimization LP | Maximization LP |
|---|---|
| Nonnegative variable $\geq$ | Inequality constraint $\leq$ |
| Nonpositive variable $\leq$ | Inequality constraint $\geq$ |
| Free variable | Equality constraint $=$ |
| Inequality constraint $\geq$ | Nonnegative variable $\geq$ |
| Inequality constraint $\leq$ | Nonpositive variable $\leq$ |
| Equality constraint $=$ | Free Variable |

## Duality review

$$\begin{aligned} \max_{x} \quad & c^{\mathsf{T}} x && \text{(maximization)} \\ \text{s.t.} \quad & Ax \le b && \text{(constraint } \le\text{)} \\ & x \ge 0 && \text{(variable } \ge\text{)} \end{aligned}$$

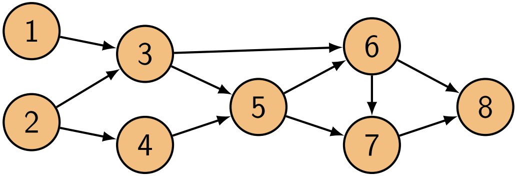
$$\begin{aligned} \min_{\lambda} \quad & b^{\mathsf{T}} \lambda && \text{(minimization)} \\ \text{s.t.} \quad & \lambda \ge 0 && \text{(variable } \ge\text{)} \\ & A^{\mathsf{T}} \lambda \ge c && \text{(constraint } \ge\text{)} \end{aligned}$$

**LP with every possible variable and constraint:**

$$\begin{aligned} \max_{x,y,z} \quad & c^{\mathsf{T}} x + d^{\mathsf{T}} y + f^{\mathsf{T}} z \\ \text{s.t.} \quad & Ax + By + Cz \le p \\ & Dx + Ey + Fz \ge q \\ & Gx + Hy + Jz = r \\ & x \ge 0 \\ & y \le 0 \\ & z \text{ free} \end{aligned}$$

$$\begin{aligned} \min_{\lambda,\eta,\mu} \quad & p^{\mathsf{T}} \lambda + q^{\mathsf{T}} \eta + r^{\mathsf{T}} \mu \\ & \lambda \ge 0 \\ & \eta \le 0 \\ & \mu \text{ free} \\ & A^{\mathsf{T}} \lambda + D^{\mathsf{T}} \eta + G^{\mathsf{T}} \mu \ge c \\ & B^{\mathsf{T}} \lambda + E^{\mathsf{T}} \eta + H^{\mathsf{T}} \mu \le d \\ & C^{\mathsf{T}} \lambda + F^{\mathsf{T}} \eta + J^{\mathsf{T}} \mu = f \end{aligned}$$

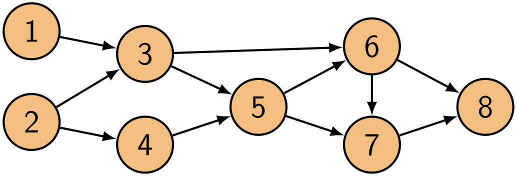
## Minimum-cost flow problems (review)



- **Decision variables**: $x_{ij}$ is the flow on edge $(i,j) \in \mathcal{E}$.
- **Capacity constraints**: $p_{ij} \le x_{ij} \le q_{ij} \qquad \forall (i,j) \in \mathcal{E}$.
- **Conservation**: $\sum_{j:(k,j)\in\mathcal{E}} x_{kj} - \sum_{i:(i,k)\in\mathcal{E}} x_{ik} = b_k \qquad \forall k \in \mathcal{N}$.
- **Total cost**: $\sum_{(i,j)\in\mathcal{E}} c_{ij} x_{ij}$.

Either $b_i > 0$ (source), $b_i < 0$ (sink), or $b_i = 0$ (relay). Also, assume $\sum_{i\in\mathcal{N}} b_i = 0$ (model is balanced).

## Minimum-cost flow problems (review)



The entire model (compact form):

$$
\begin{aligned}
\underset{x \in \mathbb{R}^{|\mathcal{E}|}}{\text{minimize}} \quad & c^{\mathsf{T}} x \\
\text{subject to:} \quad & Ax = b \\
& 0 \le x \le q
\end{aligned}
$$

**Note:** We assume here a lower bound of $0$ on the flows (the usual case).

## Dual of minimum-cost flow problems

$$\begin{array}{lll} \min\limits_{x} & c^\mathsf{T} x & \text{(minimization)} \\ \text{s.t.} & Ax = b & \text{(constraint $=$)} \\ & x \le q & \text{(constraint $\le$)} \\ & x \ge 0 & \text{(variable $\ge$)} \end{array}$$

- balance constraints (at nodes)
- capacity constraints (on edges)
- flow variables $x$ (on edges)

$$\begin{array}{lll} \max\limits_{\mu,\eta} & b^\mathsf{T}\mu + q^\mathsf{T}\eta & \text{(maximization)} \\ \text{s.t.} & \mu \text{ free} & \text{(variable free)} \\ & \eta \le 0 & \text{(variable $\le$)} \\ & A^\mathsf{T}\mu + \eta \le c & \text{(constraint $\le$)} \end{array}$$

- dual variables $\mu$ (at nodes)
- dual variables $\eta$ (each edge)
- dual constraints (each edge)

**Next**: what does the dual mean for: transportation? planning? max-flow?

## Transportation

$$\begin{array}{lll} \min\limits_{x} & c^\top x & \text{(minimization)} \\ \text{s.t.} & Ax = b & \text{(constraint $=$)} \\ & \cancel{x \le q} & \text{(constraint $\le$)} \\ & x \ge 0 & \text{(variable $\ge$)} \end{array}$$

- balance constraints (at nodes)
- ~~capacity constraints (on edges)~~
- flow variables (on edges)

$$\begin{array}{lll} \max\limits_{\mu,\eta} & b^\top \mu + \cancel{q^\top \eta} & \text{(maximization)} \\ \text{s.t.} & \mu \text{ free} & \text{(variable free)} \\ & \cancel{\eta \le 0} & \text{(variable $\le$)} \\ & A^\top \mu + \cancel{\eta} \le c & \text{(constraint $\le$)} \end{array}$$

- dual variables (at nodes)
- ~~dual variables (each edge)~~
- dual constraints (each edge)

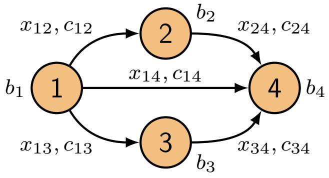
Transportation/transshipment/assignment problems: no capacity constraints on edges

## Transportation: Interpretation of Primal

$$\begin{aligned} \min_x \quad & c^\mathsf{T} x & & \text{(minimization)} \\ \text{s.t.} \quad & Ax = b & & \text{(constraint } =\text{)} \\ & x \geq 0 & & \text{(variable } \geq\text{)} \end{aligned}$$

$$\begin{aligned} \max_\mu \quad & b^\mathsf{T} \mu & & \text{(maximization)} \\ \text{s.t.} \quad & \mu \text{ free} & & \text{(variable free)} \\ & A^\mathsf{T} \mu \leq c & & \text{(constraint } \leq\text{)} \end{aligned}$$

$$\begin{aligned} \min_x \quad & c_{12}x_{12} + c_{13}x_{13} + c_{14}x_{14} + c_{23}x_{23} + c_{34}x_{34} \\ \text{s.t.} \quad & x_{12} + x_{13} + x_{14} = b_1 \\ & -x_{12} + x_{24} = b_2 \\ & -x_{13} + x_{34} = b_3 \\ & -x_{14} - x_{24} - x_{34} = b_4 \\ & x_{ij} \geq 0 \quad \forall i,j \end{aligned}$$



- $x_{ij}$ are flow amounts along edges.
- Node constraints: flow is conserved and supply/demand is met.
- Edges have transportation cost. Pick $x_{ij}$ to minimize total cost.

## I Wanna Make Some Moola

- I think the transportation company is inefficient, and I can make money by getting into the bizness.
- I will buy "extra" commodity at location $i$ (that has excess) and sell it at location $j$ (that has demand)
- At each node $i$, I want to set my purchase and sell prices $(\pi_i)$ in order to maximize my profit.

$$\max_{\pi} \underbrace{\sum_{i:b_i<0} (-\pi_i) b_i}_{\text{Revenue (since selling)}} - \underbrace{\sum_{i:b_i>0} (\pi_i) b_i}_{\text{Cost (since buying)}} = \max_{\pi} -\sum_i b_i \pi_i$$

## Gonna Be Rich!

- Can I set my prices $\pi$ arbitrarily?
- Suppose, $\pi_2 = 10$ and $\pi_1 = 4$.
    - I can buy the extra at node $1$ for \$4 and sell it at node $2$ for \$10 dollars, making $\$6$ profit!
    - If, however, the shipping cost from $1 \to 2$ $c_{12} \leq 6$, no rational person would accept my prices, since it would be cheaper to ship it themselves.
- So in order for my prices to be rational, I need to set $\pi_2 - \pi_1 \leq c_{12}$

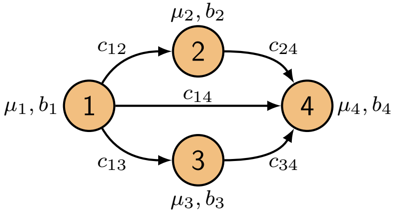
---

- Now let's look at the dual.

## Transportation: Interpretation of Dual

$$\begin{array}{lll} \min\limits_x & c^\mathsf{T} x & \text{(minimization)} \\ \text{s.t.} & Ax = b & \text{(constraint } =\text{)} \\ & x \geq 0 & \text{(variable } \geq\text{)} \end{array} \qquad \begin{array}{lll} \max\limits_\mu & b^\mathsf{T} \mu & \text{(maximization)} \\ \text{s.t.} & \mu \text{ free} & \text{(variable free)} \\ & A^\mathsf{T} \mu \leq c & \text{(constraint } \leq\text{)} \end{array}$$

$$\begin{array}{ll} \max\limits_\mu & b_1\mu_1 + b_2\mu_2 + b_3\mu_3 + b_4\mu_4 \\ \text{s.t.} & \mu_1 - \mu_2 \leq c_{12} \\ & \mu_1 - \mu_3 \leq c_{13} \\ & \mu_1 - \mu_4 \leq c_{14} \\ & \mu_2 - \mu_4 \leq c_{24} \\ & \mu_3 - \mu_4 \leq c_{34} \end{array}$$



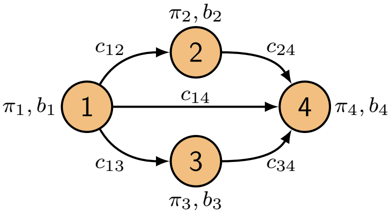
- A shipping company wants to get in on this business.
  - will buy commodity from sources (to alleviate supply)
  - will sell commodity to destinations (to satisfy demand)
  - at node $i$, the buy/sell price will be $\pi_i = -\mu_i$.

## Transportation: Interpretation of Dual

$$\begin{aligned} \min_x \quad & c^\mathsf{T} x & \text{(minimization)} \\ \text{s.t.} \quad & Ax = b & \text{(constraint $=$)} \\ & x \geq 0 & \text{(variable $\geq$)} \end{aligned}$$

$$\begin{aligned} \max_\pi \quad & -b^\mathsf{T} \pi & \text{(maximization)} \\ \text{s.t.} \quad & \pi \text{ free} & \text{(variable free)} \\ & -A^\mathsf{T} \pi \leq c & \text{(constraint $\leq$)} \end{aligned}$$

$$\begin{aligned} \max_\pi \quad & -(b_1\pi_1 + b_2\pi_2 + b_3\pi_3 + b_4\pi_4) \\ \text{s.t.} \quad & \pi_2 - \pi_1 \leq c_{12} \\ & \pi_3 - \pi_1 \leq c_{13} \\ & \pi_4 - \pi_1 \leq c_{14} \\ & \pi_4 - \pi_2 \leq c_{24} \\ & \pi_4 - \pi_3 \leq c_{34} \end{aligned}$$



- $\pi_i$ is buy/sell price of commodity at node $i$
- Edge constraints: ensures the prices are competitive. e.g. if we had $\pi_2 - \pi_1 > c_{12}$, it would be cheaper to transport it ourselves!
- Pick prices $\pi_i$ to maximize total profit.

## Transportation summary

**Primal problem:**

- Pick how much commodity flows along each edge of the network to minimize the total transportation cost while satisfying supply/demand constraints.
- If each supply/demand $b_i$ is integral, flows will be integral.

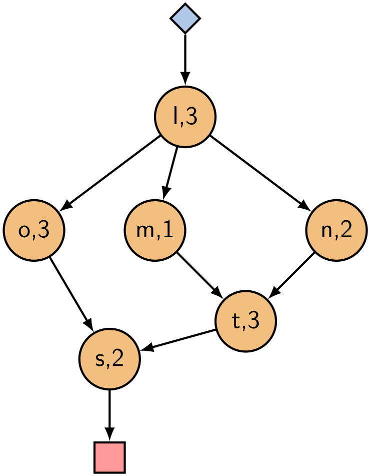
**Dual problem:**

- Pick the buy/sell price for the commodity at each node of the network to maximize the total profit while ensuring that the prices are competitive.
- If each edge cost $c_{ij}$ is integral, prices will be integral.

## Longest path

Recall the house-building example, a longest-path problem.

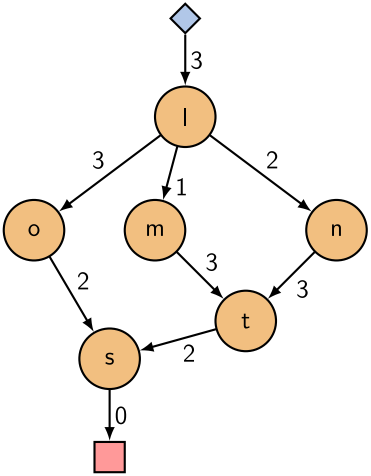
- Add source and sink nodes



## Longest path

Recall the house-building example, a longest-path problem.

- Add source and sink nodes
- Move times out of nodes and onto preceding edges
- Solve longest-path problem

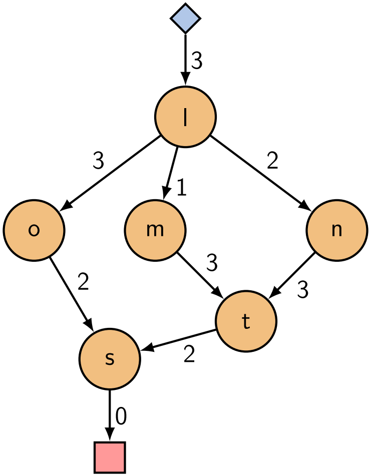


## Longest path (primal)

Recall the house-building example, a longest-path problem.

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^\mathsf{T} x \\ \text{subject to:} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

**unit flow:** $b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}$

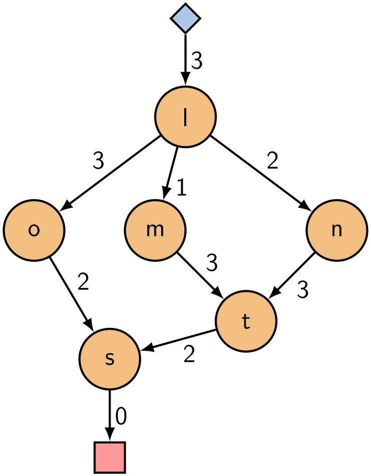


## Longest path (dual)

Recall the house-building example, a longest-path problem.

$$\begin{aligned} \underset{\mu}{\text{minimize}} \quad & b^{\mathsf{T}}\mu \\ \text{subject to:} \quad & A^{\mathsf{T}}\mu \geq c \end{aligned}$$

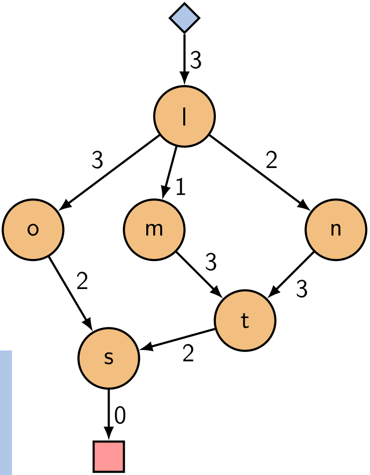
- change variables to $t = -\mu$, to obtain an equivalent min problem.



## Longest path (dual)

Recall the house-building example, a longest-path problem.

$$\begin{aligned} \underset{t}{\text{minimize}} \quad & t_{\text{end}} - t_{\text{start}} \\ \text{subject to:} \quad & t_l - t_{\text{start}} \geq 3 \\ & t_o - t_l \geq 3 \\ & t_m - t_l \geq 1 \\ & \ldots \end{aligned}$$



Precisely the original problem formulation we deduced in class (before we learned about networks)!

## Longest path (dual)

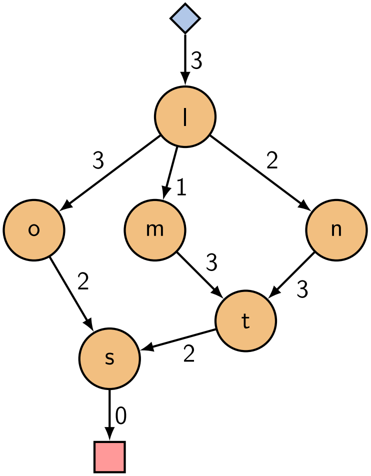
Recall the house-building example, a longest-path problem.

Key players:

- Primal variables: $x_{ij} \in \{0, 1\}$
- Dual constraints: $t_j - t_i \geq c_{ij}$

Complementary slackness:

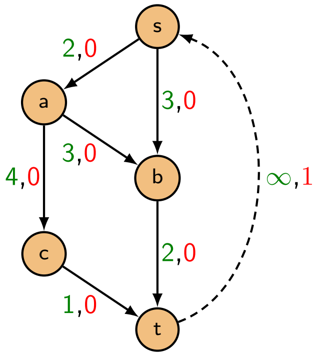
- if $x_{ij} = 1$ then $t_j - t_i = c_{ij}$ (longest path corresponds to tight time constraints)
- if $t_j - t_i > c_{ij}$ then $x_{ij} = 0$ (this path has slack)



## Max-flow

We are given a directed graph and edge capacities. Find the maximum flow that we can push from source to sink.

- Edges have max capacities
- Edges have zero cost except feedback edge, with cost $-1$.
- Finding max flow is equivalent to finding the minimum cost flow.



(edge capacity)
(cost)

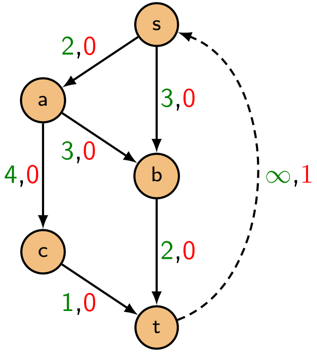
## Max-flow (primal)

- **Primal problem**:

$$\max_{x_{ij}} \quad x_{ts}$$

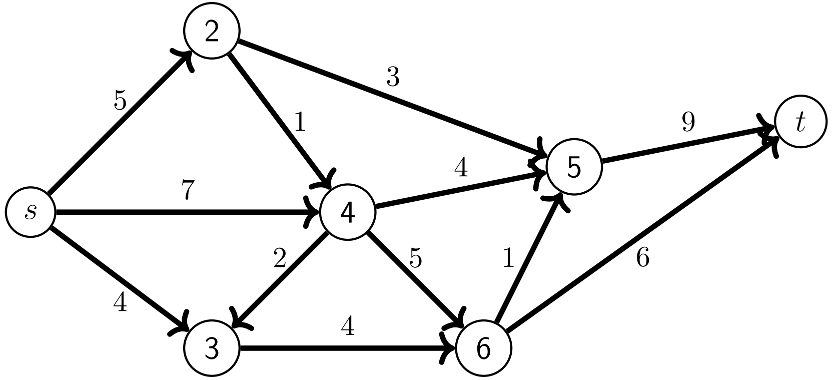
$$\text{s.t.} \quad \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_{sa} \\ x_{sb} \\ x_{ab} \\ x_{ac} \\ x_{bt} \\ x_{ct} \\ x_{ts} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \le \begin{bmatrix} x_{sa} \\ x_{sb} \\ x_{ab} \\ x_{ac} \\ x_{bt} \\ x_{ct} \end{bmatrix} \le \begin{bmatrix} 2 \\ 3 \\ 3 \\ 4 \\ 2 \\ 1 \end{bmatrix}$$



(edge capacity)
(cost)

## Let's Get an Upper Bound on Max Flow



## Cuts

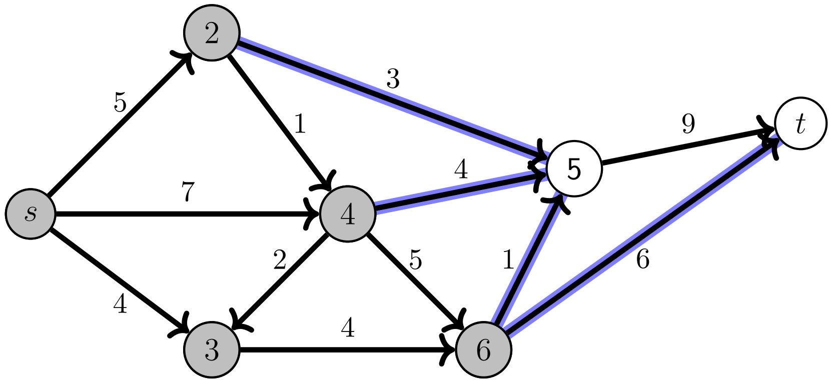
Let $S$ be *any* set of nodes containing the source but not containing the sink

- A cut is the set of arcs from nodes in $S$ to nodes not in $S$
- The cut capacity is the sum of the arc capacities.
- Suppose I found a cut whose value is 14. What can you say about the maximum flow value?
  - Capacity of *any* cut provides an upper bound on maximum flow
- Suppose I have also found feasible flow whose value is 14. What can you say about the maximum flow value now?

**Max-Flow Min-Cut Theorem**

For any network, the maximum feasible flow from the source to the sink *equals* the minimum cut capacity for all cuts in the network.

## An example of a cut



Let $S = \{s, 2, 3, 4, 6\}$

- Arcs in cut are $(2, 5)$, $(4, 5)$, $(6, 5)$, $(6, t)$

## Max Flow LP – With additional variable $v = x_{ts}$

$$\max \quad v$$

s.t. $$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0 \quad \forall i \in N, i \neq s, i \neq t$$

$$\sum_{j:(s,j)\in A} x_{sj} - \sum_{j:(j,s)\in A} x_{js} - v = 0$$

$$\sum_{j:(t,j)\in A} x_{tj} - \sum_{j:(j,t)\in A} x_{jt} + v = 0$$

$$x_{ij} \leq u_{ij} \quad \forall (i,j) \in A$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in A$$

## Max Flow – Dual

$$\begin{array}{lrll}
\min & \displaystyle\sum_{(i,j)\in A} u_{ij}\lambda_{ij} & & \\
\text{s.t.} & \mu_i - \mu_j + \lambda_{ij} \geq 0 & \forall (i,j) \in A \\
& -\mu_s + \mu_t \geq 1 & \\
& \mu_i \text{ free} & \forall i \in N \\
& \lambda_{ij} \geq 0 & \forall (i,j) \in A
\end{array}$$

- We can assume $\mu_s = 0$ in any dual solution
- Integrality property holds for max flow dual.
  - If $A$ is TU, then so is $A^{\mathsf{T}}$
- $\Rightarrow$ There is an optimal dual solution with $\mu_i$ is $0$ or $1$ and $\lambda_{ij}$ is $0$ or $1$

## Max-Flow Min Cut Theorem

- For any cut $(S, \overline{S})$, consider the values

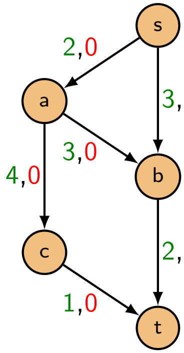
$$\mu'_i = \begin{cases} 1 & i \in \overline{S} \\ 0 & \text{otherwise} \end{cases} \qquad \lambda'_{ij} = \begin{cases} 1 & (i,j) \in A, i \in S, j \in \overline{S} \\ 0 & \text{otherwise} \end{cases}$$

- This "dual solution" is feasible. (check)
- Its objective value sums $u_{ij}$ for $(i,j) \in A$ with $i \in S$ and $j \in \overline{S}$: The capacity of the cut $(S, \overline{S})$.
- The value of any cut (any dual feasible solution) provides an upper bound on the optimal solution value
- By using strong duality, this proves the max flow min cut theorem.

## Max-flow (dual)

- **Dual problem**:

$$
\begin{aligned}
\min_{\lambda_{ij},\mu_i} \quad & 2\lambda_{sa} + 3\lambda_{sb} + 3\lambda_{ab} + 4\lambda_{ac} + 2\lambda_{bt} + \lambda_{ct} \\
\text{s.t.} \quad & \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_s \\ \mu_a \\ \mu_b \\ \mu_c \\ \mu_t \end{bmatrix} + \begin{bmatrix} \lambda_{sa} \\ \lambda_{sb} \\ \lambda_{ab} \\ \lambda_{ac} \\ \lambda_{bt} \\ \lambda_{ct} \\ 0 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
& \lambda_{ij} \geq 0, \quad \mu_i \text{ free}
\end{aligned}
$$



(edge capacity)
(cost)

  - $\mu_i$ are shift-invariant - replacing $\mu_i \leftarrow \mu_i + \alpha$ for all $i$ and any $\alpha$ does not change the constraints. Therefore we may assume $\mu_s = 0$.
  - We want each $\lambda_{ij}$ small; no slack!

# Max-flow (dual)

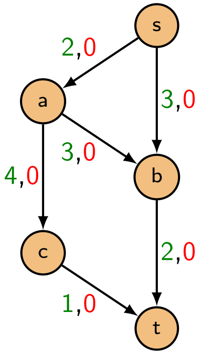
- **Dual problem**:

$$\min_{\lambda_{ij},\mu_i} \quad 2\lambda_{sa} + 3\lambda_{sb} + 3\lambda_{ab} + 4\lambda_{ac} + 2\lambda_{bt} + \lambda_{ct}$$

$$\text{s.t.} \quad \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_a \\ \mu_b \\ \mu_c \\ \mu_t \end{bmatrix} + \begin{bmatrix} \lambda_{sa} \\ \lambda_{sb} \\ \lambda_{ab} \\ \lambda_{ac} \\ \lambda_{bt} \\ \lambda_{ct} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\lambda_{ij} \geq 0, \quad \mu_i \text{ free}$$

  - Rearrange constraints, isolate $\lambda_{ij}$.



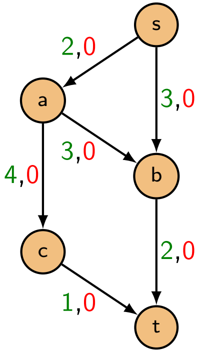
(edge capacity)
(cost)

## Max-flow (dual)

$\mu_i \in \{0,1\}$ for all $i$.

- **Dual problem**:

$$\begin{aligned}
\min_{\lambda_{ij},\mu_i} \quad & 2\lambda_{sa} + 3\lambda_{sb} + 3\lambda_{ab} + 4\lambda_{ac} + 2\lambda_{bt} + \lambda_{ct} \\
\text{s.t.} \quad & \mu_s = 0 \\
& \mu_a - \mu_s = \lambda_{sa} \\
& \mu_b - \mu_s = \lambda_{sb} \\
& \mu_b - \mu_a = \lambda_{ab} \\
& \mu_c - \mu_a = \lambda_{ac} \\
& \mu_t - \mu_b = \lambda_{bt} \\
& \mu_t - \mu_c = \lambda_{ct} \\
& \mu_t = 1 \\
& \lambda_{ij} \geq 0, \quad \mu_i \text{ free}
\end{aligned}$$



(edge capacity)
(cost)

- Each path, e.g. $s \to a \to c \to t$ has: $0 = \mu_s \leq \mu_a \leq \mu_c \leq \mu_t = 1$

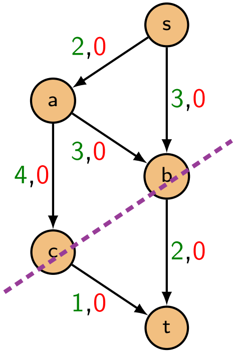
## Max-flow (dual)

- **Dual problem**:

$$\begin{aligned} \min_{\lambda_{ij},\mu_i} \quad & 2\lambda_{sa} + 3\lambda_{sb} + 3\lambda_{ab} + 4\lambda_{ac} + 2\lambda_{bt} + \lambda_{ct} \\ \text{s.t.} \quad & \text{Along each path } s \to i \to \cdots \to j \to t \\ & \text{exactly one edge } p \to q \text{ is chosen.} \\ & \lambda_{pq} = 1, \text{ with } \lambda_{ij} = 0 \text{ for all other edges.} \end{aligned}$$

  - Each path is broken by *cutting* edges. We are choosing the cut with lowest total cost (*min-cut*).

Max flow = Min cut



(edge capacity)
(cost)

## Max-flow summary

**Primal problem:**

- Each edge of the network has a maximum capacity.
- Pick how much commodity flows along each edge to maximize the total amount transported from the start node to the end node while obeying conservation constraints. This total amount of flow is called the *max flow*.

**Dual problem:**

- Find a partition of the nodes into two subsets where the first subset includes the start node and the second subset includes the end node.
- Choose the partition that minimizes the sum of capacities of all edges that connect starting subset to ending subset. This total capacity is called the *min cut*.

## LP solvers

Modern LP solvers are very efficient. Problems with millions of variables and/or constraints are routinely solved. Three main categories of algorithms are used in practice for solving LPs:

- **Simplex algorithms**: traverse the *surface* of the feasible polyhedron looking for the best vertex.
- **Interior point**: traverse the *interior* of the polyhedron and move toward the center of the face of optimal solutions. (`GLPK`, `SCS`, `ECOS`, `Ipopt`)
    - For large problems, often start with an interior-point approach, then "cross over" to simplex in order to give a corner point optimal solution.

## Simplex method

- Invented by George Dantzig in 1947.
- Named one of the "Top 10 algorithms of the 20th century" by *Computing in Science & Engineering* Magazine.
- The basic idea:
  - We know the solution is a vertex of the feasible polyhedron.
  - Each vertex is characterized by the subset of the constraints that have no slack; it's just a system of linear equations!
  - Start at a vertex, then *pivot*: swap out one of the constraints in the no-slack subset, moving to an adjacent vertex with better objective.
  - Do this in a systematic way that avoids cycles. When we can no longer improve, we are optimal!

## Simplex method

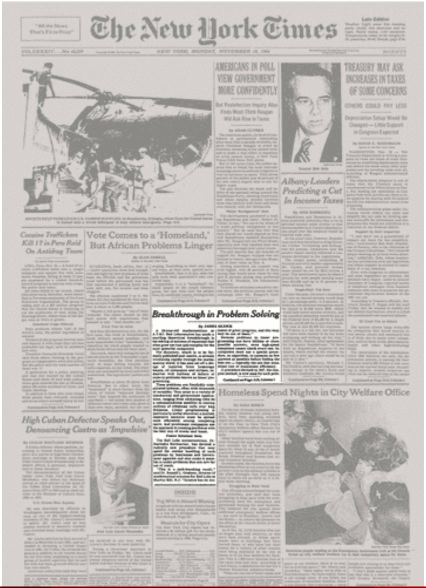
- With $m$ constraints in $n$ variables, the feasible polyhedron can have roughly up to $\binom{m}{n}$ vertices, a very large number!
- A cube in $n$ dimensions has $2^n$ vertices.
- By carefully designing the problem, the simplex method may visit *all the vertices*! Look up the Klee–Minty cube.
- It is not known whether there is a more clever version of simplex that is sub-exponential in the worst case.

Despite these difficulties, the simplex method works **very well** in practice. For typical problems, its performance scales linearly with $m$ and $n$.

## Interior point methods

- Big family of optimization algorithms, dating back to mid-1950s. Can be used for solving convex nonlinear optimization problems.
- Ellipsoid method when applied to LPs achieves polynomial-time convergence (Khachiyan, 1979), but typically much slower than simplex in practice.
- An alternative interior-point approach proposed by Karmarkar (1984) showed promise of good performance on practical problems.
- Karmarkar's paper spurred developments that led to *primal-dual* algorithms in the late 1980s, which are the basis of modern practical polynomial-time procedures that are competitive with simplex, esp. on large problems.
- Primal-dual: Solve the primal and dual problems simultaneously!

## LP Hits the Big Time



"All the News That's Fit to Print"

The New York Times

Late Edition

[illegible]

VOL. CXXXIV . . . No. [illegible] NEW YORK, MONDAY, NOVEMBER 19, 1984 [illegible] CENTS

AMERICANS IN POLL VIEW GOVERNMENT MORE CONFIDENTLY

But Postelection Inquiry Also Finds Most Think Reagan Will Ask Rise in Taxes

[illegible]

TREASURY MAY ASK INCREASES IN TAXES OF SOME CONCERNS

OTHERS COULD PAY LESS

Depreciation Setup Would Be Changed — Little Support in Congress Expected

[illegible]

*Albany Leaders Predicting a Cut In Income Taxes*

[illegible]

*Cocaine Traffickers Kill 17 in Peru Raid On Antidrug Team*

[illegible]

Vote Comes to a 'Homeland,' But African Problems Linger

[illegible]

***Breakthrough in Problem Solving***

By JAMES GLEICK

[illegible]

*High Cuban Defector Speaks Out, Denouncing Castro as 'Impulsive'*

[illegible]

INSIDE

[illegible]

Homeless Spend Nights in City Welfare Office

[illegible]

## Specialized algorithms

If the LP has a special form, specialized algorithms are often vastly superior to generic simplex or interior point solvers.

- **Network simplex method**: Special version of simplex method for solving minimum-cost flow problems. Can be 10s of times faster than using ordinary simplex method. Polynomial worst-case behavior.
- **Graph searches**: Djikstra's algorithm, A* search, etc. Can be used for example to find the shortest path in a graph.
- Assignment problems: Kuhn–Munkres, Auction algorithm.
- Max-flow problems: Ford–Fulkerson, Preflow-Push algorithm.

## LP wrap-up

- Relevant courses at UW–Madison
  - CS 525: linear programming methods
  - CS 730: graduate course in nonlinear optimization, but we usually discuss interior-point methods for LP.
  - CS 577: introduction to algorithms

  525 and 577 prove major results (e.g. zero duality gap), give detailed explanations/analyses of simplex/graph algorithms.