# *Linear Programming and Network Optimization*

Zongpeng Li

Department of Computer Science

University of Calgary

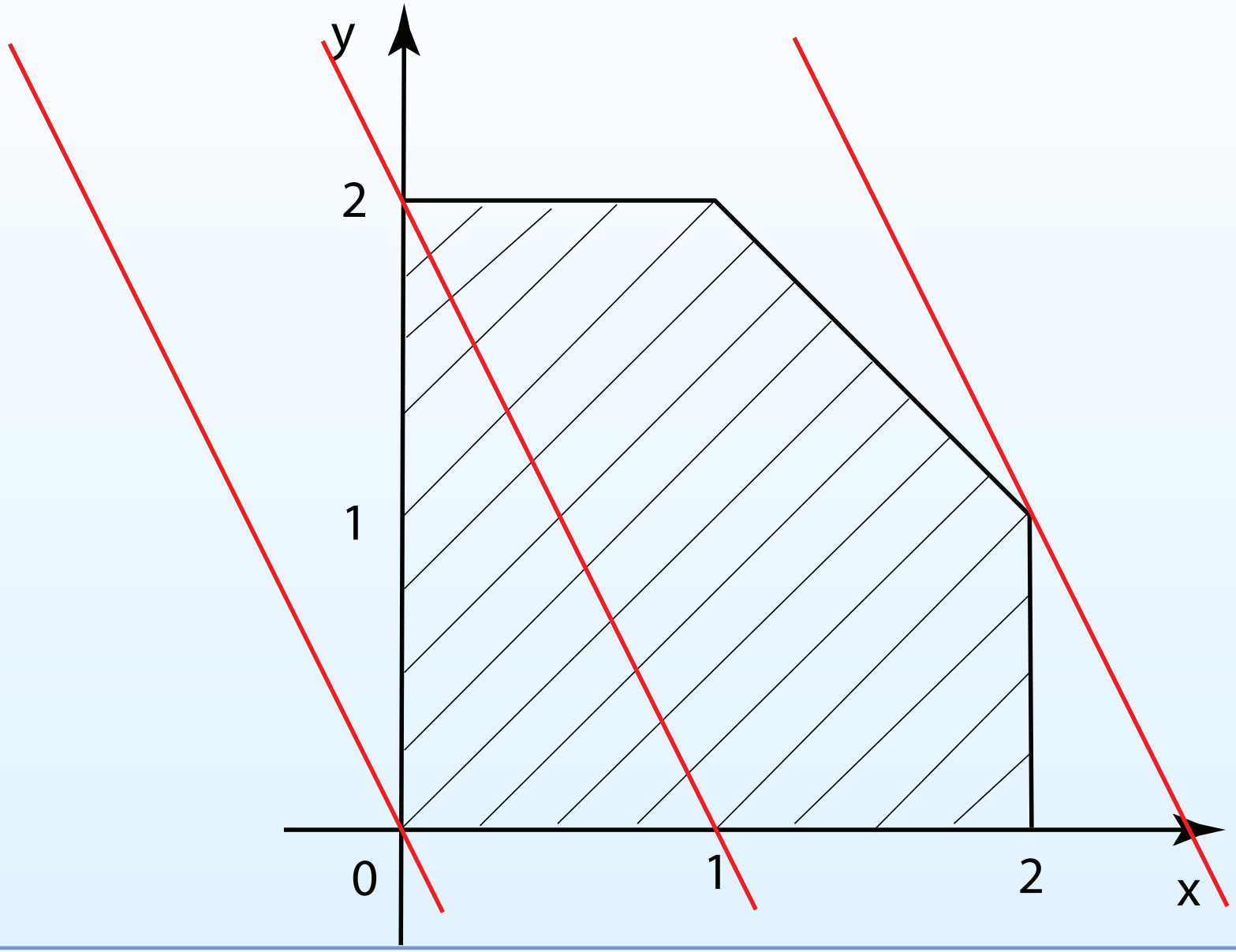
# Outline

- Hello World linear program
- The power of LP
- LP models in network optimization
- LP duality
- Solving LPs
- Beyond LP

# Hello World

$$
\begin{aligned}
\text{maximize} \quad & 2x + y \\
\text{s.t. :} \quad & \\
& x \leq 2 \\
& y \leq 2 \\
& x + y \leq 3 \\
& \\
& x, y \geq 0
\end{aligned}
$$

# Hello World



## The power of LP

*But we all know the world is nonlinear.*

— Harold Hotelling, 1948

# The power of LP

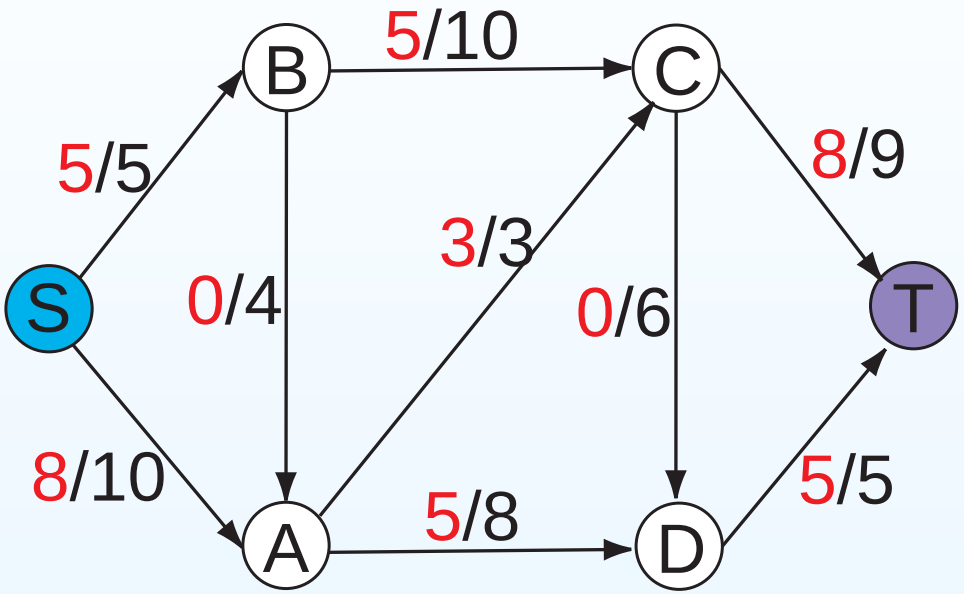
*But we all know the world is nonlinear.*

1. *If you have a problem that satisfies the axioms (of LP), then use it. If it does not, then don't.*

   — John von Neumann, 1948

2. Much more problems can be modelled using LPs than suggested by intuition.
3. LP constitutes building blocks for nonlinear programming

# LP model: max-flow




- Maximum rate we can push flows from $S$ to $T$ in a given capacitied flow network.
- flow-rate/link-capacity

# LP model: max-flow

Maximize $\chi = f(\overrightarrow{TS})$

Subject to:

$$\begin{cases} f(\overrightarrow{uv}) \leq C(uv) & \forall\ \overrightarrow{uv} \neq \overrightarrow{TS} \\ \sum_{v \in N(u)} f(\overrightarrow{uv}) = \sum_{v \in N(u)} f(\overrightarrow{vu}) & \forall u \end{cases}$$

$$f(\overrightarrow{uv}) \geq 0 \qquad \forall\ \overrightarrow{uv}$$

# Totally unimodular LPs

- Totally unimodular: every square sub-matrix of the coefficient matrix has determinant of 1 or -1.
- Totally unimodular LPs always have integral optimal solutions.
- The node-arc incidence matrix of a directed network is totally unimodular!

# LP model: min-cut

Minimize $\sum_{\vec{uv}} C(uv)y(\vec{uv})$

Subject to:

$$\begin{cases} y(\vec{uv}) + p(v) \geq p(u) & \forall\ \vec{uv} \neq \vec{TS} \\ p(T) - p(S) \geq 1 \end{cases}$$

$$y(\vec{uv}) \geq 0 \qquad \forall\ \vec{uv}$$

- Max-cut cannot be modelled as a simple LP; it is NP-hard.
- Elegant approximation algorithm of max-cut based on semidefinite programming.

# LP model: min-cost flow

Minimize $\sum_{\vec{uv}} w(\vec{uv}) f(\vec{uv})$

Subject to:

$$\begin{cases} f(\vec{TS}) = d & \\ f(\vec{uv}) \leq C(uv) & \forall\ \vec{uv} \neq \vec{TS} \\ \sum_{v \in N(u)} f(\vec{uv}) = \sum_{v \in N(u)} f(\vec{vu}) & \forall u \end{cases}$$

$$f(\vec{uv}) \geq 0 \qquad \forall\ \vec{uv}$$

# LP model: shortest path

Minimize $\sum_{\vec{uv}} w(\vec{uv}) f(\vec{uv})$

Subject to:

$$\begin{cases} f(\vec{TS}) = 1 \\ \sum_{v \in N(u)} f(\vec{uv}) = \sum_{v \in N(u)} f(\vec{vu}) \quad \forall u \end{cases}$$

$$f(\vec{uv}) \geq 0 \qquad \forall\, \vec{uv}$$

# LP model: the assignment problem

- Assign $n$ objects to $n$ persons, 1-to-1 mapping
- Each object $o$ worths $v(i,o)$ to each person $i$
- Goal: maximize "total happiness"

Maximize $\sum_i \sum_o f(i,o)v(i,o)$

Subject to:

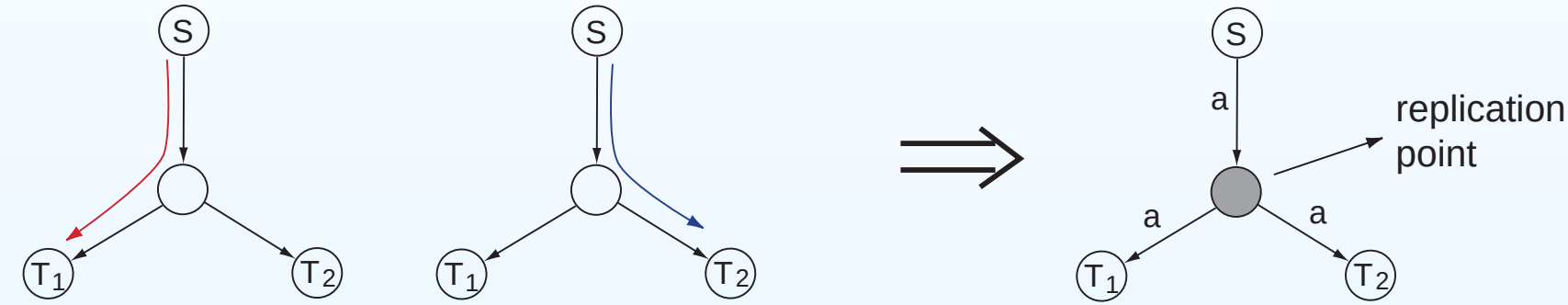
$$\begin{cases} \sum_o f(i,o) = 1 & \forall i \\ \sum_i f(i,o) = 1 & \forall o \end{cases}$$

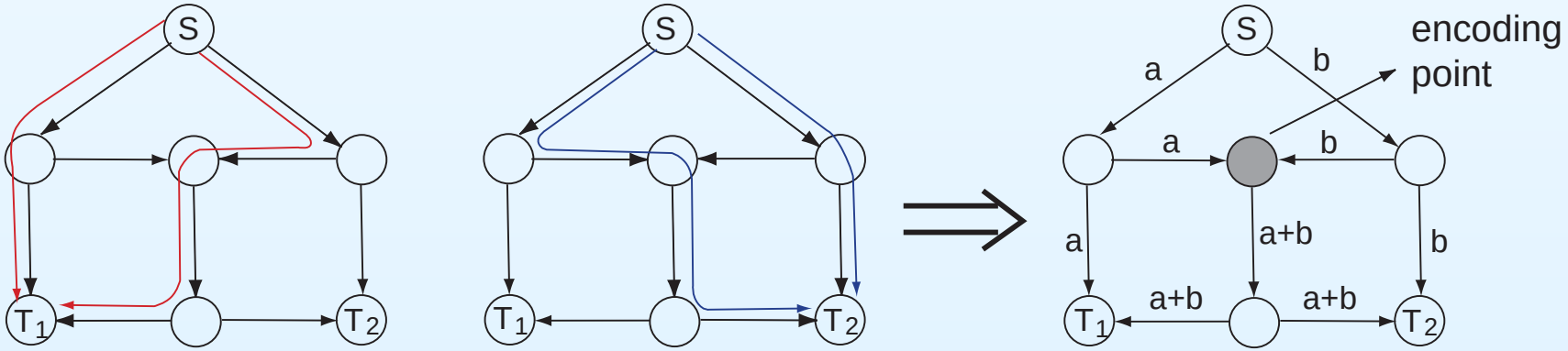
$$f(i,o) \geq 0 \qquad \forall i, \forall o$$

- Totally unimodular LP, integral optimal solution
- primal-dual algorithm design, the celebrated auction algorithm

# LP model: max-rate multicast with network coding

Given network coding, a multicast rate $x$ is feasible in a directed network iff it is feasible as an independent unicast to every receiver. [Ahlswede et al. IT 2000][Koetter and Médard TON 2003]



# LP model: max-rate multicast with network coding

Maximize $\chi$

Subject to:

$$\begin{cases} \chi \le f_i(\overrightarrow{T_iS}) & \forall i & (1) \\ f_i(\overrightarrow{uv}) \le c(\overrightarrow{uv}) & \forall i, \forall\ \overrightarrow{uv} \neq \overrightarrow{T_iS} & (2) \\ \sum_{v \in N(u)} f_i(\overrightarrow{uv}) = \sum_{v \in N(u)} f_i(\overrightarrow{vu}) & \forall i, \forall u & (3) \\ c(\overrightarrow{uv}) + c(\overrightarrow{vu}) \le C(uv) & \forall uv \neq T_iS & (4) \end{cases}$$

$$c(\overrightarrow{uv}), f_i(\overrightarrow{uv}), \chi \ge 0 \qquad \forall i, \forall\ \overrightarrow{uv}$$

# LP model: max-rate multicast without network coding

Minimize $\sum_t f(t)$

Subject to:

$$\sum_{t:e\in t} f(t) \le c(e) \qquad \forall e$$

$$f(t) \ge 0 \qquad \forall t$$

- Don't be misguided by the seeming simplicity of the LP.
- It has exponentially many variables.
- We know a network instance with $16$ nodes only, having $\sim 50$ million different trees.
- But, what else can we do? It's an NP-hard problem.

# Primal and dual LPs

| Minimize $c_1x_1 + c_2x_2 + c_3x_3$ | Maximize $b_1y_1 + b_2y_2 + b_3y_3$ |
|---|---|
| Subject to: | Subject to: |
| $\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \geq b_1 & \leftrightarrow y_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \geq b_2 & \leftrightarrow y_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \geq b_3 & \leftrightarrow y_3 \end{cases}$ | $\begin{cases} a_{11}y_1 + a_{21}y_2 + a_{31}y_3 \leq c_1 & \leftrightarrow x_1 \\ a_{12}y_1 + a_{22}y_2 + a_{32}y_3 \leq c_2 & \leftrightarrow x_2 \\ a_{13}y_1 + a_{23}y_2 + a_{33}y_3 \leq c_3 & \leftrightarrow x_3 \end{cases}$ |
| $x_1, x_2, x_3 \geq 0$ | $y_1, y_2, y_3 \geq 0$ |

- Poor student *vs.* greedy drug store owner
- Student: satisfying vitamin intaking needs with minimal budget
- Store owner: maximizing revenue while maintaining competitiveness

# LP duality

- Every feasible solution in the primal (minimization) provides a lower-bound for the dual (maximization) and vice versa.
- If the primal is feasible and has optimal solutions, then so does the dual; furthermore, their optimal objective function values must be the same.
- Every max-min theorem (that I know of) in graph theory, combinatorial optimization and game theory can be derived as a corollary of the LP duality theorem and/or the matroid union theorem.

## Complementary slackness

- Let $x^*$ and $y^*$ be a pair of corresponding optimal primal and dual solutions
- $y_1^* > 0 \Rightarrow a_{11}x_1^* + a_{12}x_2^* + a_{13}x_3^* = b_1$, and so on
- The shadow price is nonzero only if the resource supply is tight
- Generalization into nonlinear programming: the Karush-Kuhn-Tucker (KKT) conditions

# An example application of LP duality and CS

Enforcing minimum-cost multicast routing, Li and Williamson, 2007.

- Min-cost multicast, flows selfishly route themselves through cheapest paths available
- Formulate primal and dual LPs
- Use shadow prices to allocate edge costs and set edge taxes
- Each optimal flow can be thus enforced; proof of Nash Equilibrium based on CS conditions

# Solving LPs: the simplex method

- Walk along a sequence of vertice, on the polyhedron boundary
- with improved objective value at each step
- multiple “better neighbors”, which to choose?
- The pivot rule

# Solving LPs: the interior-point method

- Walk within the polytope
- Each step, walk towards a new feasible solution in the polytope
- which had better not be too close to the boundary
- being close to the optimum is naturally good
- Model the above concerns using *barrier functions* and *potential functions*

# Solving LPs: the ellipsoid method

- Solve optimization by solving feasibility, through binary search.
- Enclose the feasibility polytope using an ellipsoid
- either verify feasibility using a separation oracle
- or cut the ellipsoid into two halves and enclose the feasible half using a smaller ellipsoid
- Claim infeasibility when the ellipsoid becomes small enough.
- Why ellipsoid? Why not a sphere? What about other geometric shapes?

# Solving LPs: problem specific methods

- Tailor the simplex algorithm: the network simplex algorithm
- Lagrange relaxation and subgradient optimization
  - Assume a network flow LP with an extra side constraint
  - Can relax the side constraint and solve the smaller network flow LP using highly optimized algorithms
  - Trade-off: need to solve a sequence of these
  - Can help in distributed protocol design

# Solving LPs: realworld experiences

- 1000 variables/constraints ? — that's easy
- 1 million variables/constraints ? — that's OK
- 1 billion variables/constraints ? — no way
- For general LPs: simplex and interior-point algorithms can compete with each other
- For LPs with a network background: interior-point algorithms might perform much better (personal experience)
- Ellipsoid algorithms are of theoretical interest mostly

# Solving LPs: software available

- GNU `glpk`, http://www.gnu.org/software/glpk/
  - free
  - simplex, interior-point, branch-and-cut
- CPLEX, http://www.ilog.com/products/cplex/
  - simplex, interior-point, integer programming, quadratic programming
- CVX, http://www.stanford.edu/ boyd/cvx/
  - free
  - Matlab library
  - solves “disciplined” convex programs

# Liner integer programming: layered multicast

Maximize $\sum_i \sum_k l_k.x_k^i$ (9)

Subject to:

$$\begin{cases} \sum_{v \in N(u)} [f_k^i(\overrightarrow{uv}) - f_k^i(\overrightarrow{vu})] = 0 & \forall k, \forall i, \forall u \\ f_k^i(\overrightarrow{uv}) \leq f_k(\overrightarrow{uv}) & \forall k, \forall i, \forall \overrightarrow{uv} \\ \sum_k f_k(\overrightarrow{uv}) \leq C(\overrightarrow{uv}) & \forall \overrightarrow{uv} \\ x_{k+1}^i \leq x_k^i \leq \frac{f_k^i(\overrightarrow{T_i S})}{l_k} & \forall k = 1..L-1, \forall i \end{cases}$$

$$f_k(\overrightarrow{uv}), f_k^i(\overrightarrow{uv}) \geq 0, \;\; x_k^i \in \{0,1\} \quad \forall k, \forall i, \forall \overrightarrow{uv}$$

# Semidefinite/Vector programming: max cut

Quadratic formulation of max cut ($x(u) = 1$ if $u$ is in the source component; otherwise $x(u) = -1$):

Maximize $\sum_{uv \in E} \frac{1}{2}(1 - x(u)x(v))$

Subject to:

$$x(u) \in \{1, -1\}, \forall u$$

Vector programming relaxation:

Maximize $\sum_{uv \in E} \frac{1}{2}(1 - x(u)x(v))$

Subject to:

$$||x(u)||_n = 1, \forall u$$

$$x(u) \in \mathcal{R}^n, \forall u; n \in \mathcal{Z}_+$$