# Robust combinatorial optimization

Presentation of *Robust discrete optimization and network flows*
by Dimitris Bertsimas and Melvyn Sim
and a Linear Programming Lecture by Michel Goemans

Seminar 'Optimization under Uncertainty'

22.05.2024

Leon Höpfl

# Motivation

How to handle uncertainty?

- Stochastic
- Absolute Robust

Problems?

# Motivation

How to handle uncertainty?

- Stochastic
- Absolute Robust

Problems?

Need to know distribution
Hard to calculate

All or Nothing

# Motivation

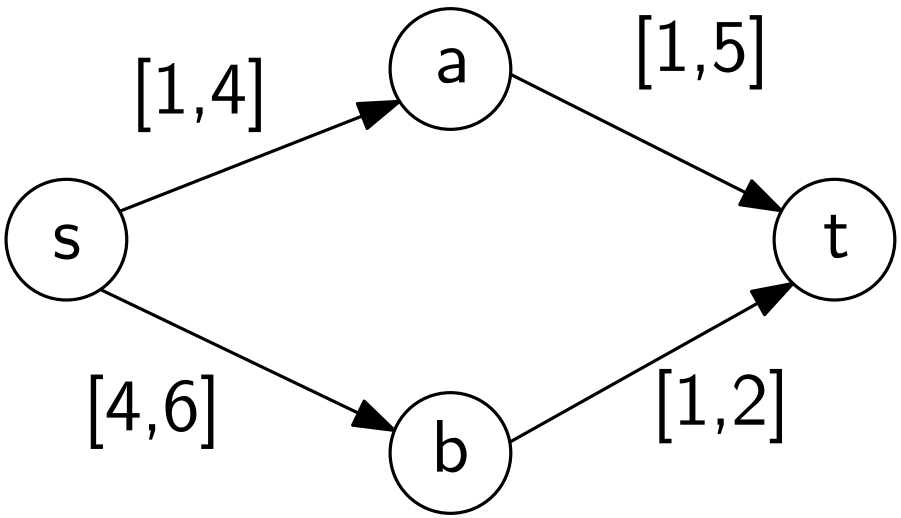
How to handle uncertainty?

- Stochastic
- Absolute Robust

Problems?

Need to know distribution
Hard to calculate

All or Nothing



# Motivation

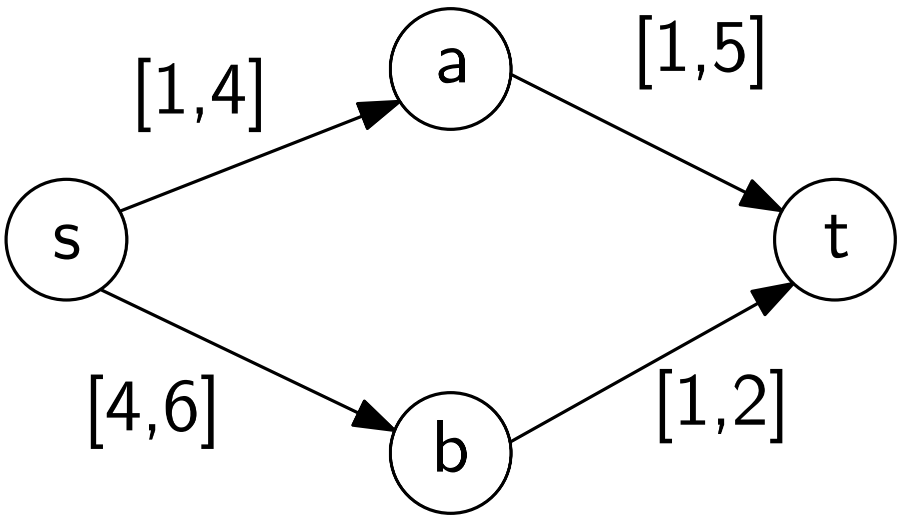
How to handle uncertainty?

- Stochastic
- Absolute Robust

Problems?

Need to know distribution
Hard to calculate

All or Nothing




Solve later!

$\Gamma$-Robust, with at most $\Gamma$ deviations

# Motivation

How to handle uncertainty?

- Stochastic
- Absolute Robust

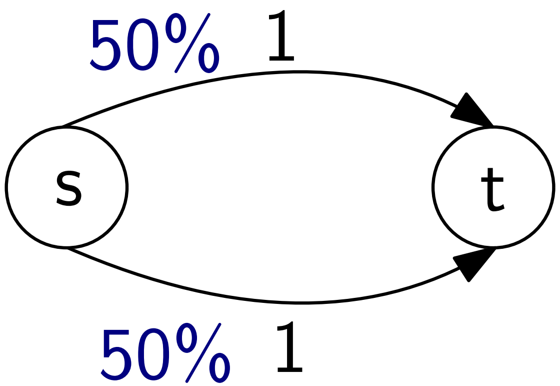
Why combinatorial (Integer)? Isn't Shortest Path relaxable?

# Motivation

How to handle uncertainty?

- Stochastic
- Absolute Robust

Why combinatorial (Integer)? Isn't Shortest Path relaxable?

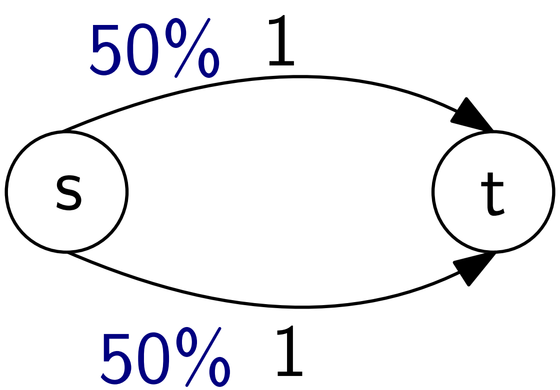


# Motivation

How to handle uncertainty?

- Stochastic
- Absolute Robust

Why combinatorial (Integer)? Isn't Shortest Path relaxable?




Can be relaxed!

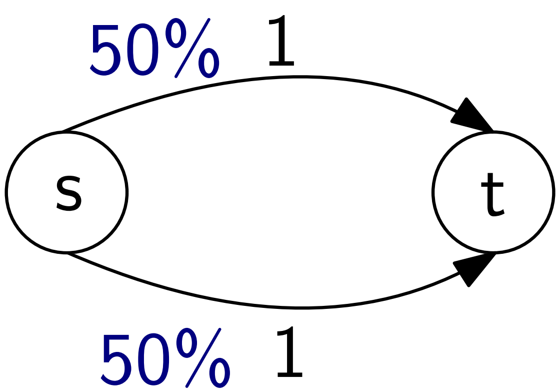
# Motivation

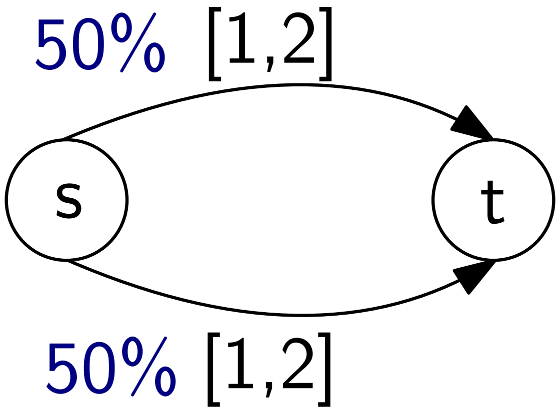
How to handle uncertainty?

- Stochastic
- Absolute Robust

Why combinatorial (Integer)? Isn't Shortest Path relaxable?

$\Gamma = 1$, so at most one deviation




Relaxable?

Can be relaxed!

# Motivation

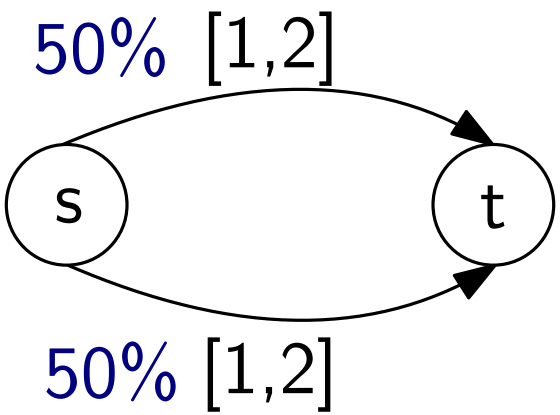
How to handle uncertainty?

- Stochastic
- Absolute Robust

Why combinatorial (Integer)? Isn't Shortest Path relaxable?

$\Gamma = 1$, so at most one deviation

LP not sufficient,
need ILP strategy!




Relaxable?

No! $\Gamma = 1$ makes other edge better

# Linear Programs: Duality

$$\begin{aligned} \text{maximize} \quad & z = 5x_1 + 4x_2 \\ \text{subject to} \quad & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Def.: A set of variables $x_1, \ldots, x_n$ is called *feasible* for an LP, if it satisfies all of its constraints

Any feasible solution is a lower bound for $z$

$x_1 = 4, x_2 = 2 \rightarrow z = 5 \cdot 4 + 4 \cdot 2 = 28$

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Def.: A set of variables $x_1, \ldots, x_n$ is called *feasible* for an LP, if it satisfies all of its constraints

Any feasible solution is a lower bound for $z$

$x_1 = 4, x_2 = 2 \rightarrow z = 5 \cdot 4 + 4 \cdot 2 = 28$

Upper bound?

# Linear Programs: Duality

$$\begin{aligned} \text{maximize} \quad & z = 5x_1 + 4x_2 \\ \text{subject to} \quad & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

Find upper bound of z!

# Linear Programs: Duality

$$\begin{aligned} \text{maximize} \quad & z = 5x_1 + 4x_2 \\ \text{subject to} \quad & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

Find upper bound of z!

Idea: Combine constraints for an inequality

# Linear Programs: Duality

$$\begin{aligned} \text{maximize} \quad & z = 5x_1 + 4x_2 \\ \text{subject to} \quad & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

Find upper bound of z!

Idea: Combine constraints for an inequality

$$3x_1 + 2x_2 \leq 16$$

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Find upper bound of z!

Idea: Combine constraints for an inequality

$$3x_1 + 2x_2 \leq 16$$

$$6x_1 + 4x_2 \leq 32$$

# Linear Programs: Duality

$$
\begin{array}{ll}
\text{maximize} & z = 5x_1 + 4x_2 \\
\text{subject to} & x_1 \leq 4 \\
& x_1 + 2x_2 \leq 10 \\
& 3x_1 + 2x_2 \leq 16 \\
& x_1 \geq 0, x_2 \geq 0
\end{array}
$$

Find upper bound of z!

Idea: Combine constraints for an inequality

$$3x_1 + 2x_2 \leq 16$$

$$6x_1 + 4x_2 \leq 32$$

$$z = 5x_1 + 4x_2 \leq 6x_1 + 4x_2 \leq 32$$

Non-negativity

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Find upper bound of z!

Idea: Combine constraints for an inequality

$$(x_1) + (x_1 + 2x_2) + (3x_1 + 2x_2) \leq (4) + (10) + (16)$$

$$z = 5x_1 + 4x_2 \leq 30$$

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Find upper bound of z!

Idea: Combine constraints for an inequality

$$(x_1) + (x_1 + 2x_2) + (3x_1 + 2x_2) \leq (4) + (10) + (16)$$

$$z = 5x_1 + 4x_2 \leq 30$$

Idea: Generalize to find minimum upper bound?

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Generalization: finding the minimum upper bound

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \le 4 \\ & x_1 + 2x_2 \le 10 \\ & 3x_1 + 2x_2 \le 16 \\ & x_1 \ge 0, x_2 \ge 0 \end{array}$$

Generalization: finding the minimum upper bound

$$y_1(x_1) + y_2(x_1 + 2x_2) + y_3(3x_1 + 2x_2) \le y_1(4) + y_2(10) + y_3(16)$$

$$(y_1 + y_2 + 3y_3)x_1 + (2y_2 + 2y_3)x_2 \le 4y_1 + 10y_2 + 16y_3$$

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Generalization: finding the minimum upper bound

$$y_1(x_1) + y_2(x_1 + 2x_2) + y_3(3x_1 + 2x_2) \leq y_1(4) + y_2(10) + y_3(16)$$

$$(y_1 + y_2 + 3y_3)x_1 + (2y_2 + 2y_3)x_2 \leq 4y_1 + 10y_2 + 16y_3$$

Constraints?

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Generalization: finding the minimum upper bound

$$y_1(x_1) + y_2(x_1 + 2x_2) + y_3(3x_1 + 2x_2) \leq y_1(4) + y_2(10) + y_3(16)$$

$$(y_1 + y_2 + 3y_3)x_1 + (2y_2 + 2y_3)x_2 \leq 4y_1 + 10y_2 + 16y_3$$

Constraints?

$$y_1 + y_2 + 3y_3 \geq 5$$

$$0y_1 + 2y_2 + 2y_3 \geq 4$$

# Linear Programs: Duality

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Generalization: finding the minimum upper bound

$$\begin{array}{ll} \text{minimize} & z = 4y_1 + 10y_2 + 16y_3 \\ \text{subject to} & y_1 + y_2 + 3y_3 \geq 5 \\ & 2y_2 + 2y_3 \geq 4 \\ & y_1 \geq 0, y_2 \geq 0, y_3 \geq 0 \end{array}$$

# Linear Programs: Duality

Primal

$$\begin{array}{ll} \text{maximize} & z = 5x_1 + 4x_2 \\ \text{subject to} & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

Generalization: finding the minimum upper bound

Dual

$$\begin{array}{ll} \text{minimize} & z = 4y_1 + 10y_2 + 16y_3 \\ \text{subject to} & y_1 + y_2 + 3y_3 \geq 5 \\ & 2y_2 + 2y_3 \geq 4 \\ & y_1 \geq 0, y_2 \geq 0, y_3 \geq 0 \end{array}$$

# Linear Programs: Duality

Primal

$$\begin{aligned} \text{maximize} \quad & z = 5x_1 + 4x_2 \\ \text{subject to} \quad & x_1 \leq 4 \\ & x_1 + 2x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

Generalization: finding the minimum upper bound

Dual

$$\begin{aligned} \text{minimize} \quad & z = 4y_1 + 10y_2 + 16y_3 \\ \text{subject to} \quad & y_1 + y_2 + 3y_3 \geq 5 \\ & 2y_2 + 2y_3 \geq 4 \\ & y_1 \geq 0, y_2 \geq 0, y_3 \geq 0 \end{aligned}$$

If the optimal value of the primal $z^*$ is finite, then so is the optimal value of the dual $w^*$, and $z^* = w^*$

# Combinatorial Optimization

Def.: Combinatorial Optimization problems are a subclass of discrete (integer) optimization problems where the decision variables can only be 0 or 1

$$\begin{array}{ll} \text{minimize} & c'x \\ \text{subject to} & x \in X \end{array}$$

with $x \in X \subseteq \{0,1\}^n, c' \in (\mathbb{R}^n)^T$

# Combinatorial Optimization

Def.: Combinatorial Optimization problems are a subclass of discrete (integer) optimization problems where the decision variables can only be 0 or 1

$$\begin{array}{ll} \text{minimize} & c'x \\ \text{subject to} & x \in X \end{array}$$

with $x \in X \subseteq \{0,1\}^n, c' \in (\mathbb{R}^n)^T$

Generalized case, also called the 'nominal problem'
Constraints are encoded into $X$

# Combinatorial Optimization

Def.: Combinatorial Optimization problems are a subclass of discrete (integer) optimization problems where the decision variables can only be 0 or 1

$$\begin{array}{ll} \text{minimize} & c'x \\ \text{subject to} & x \in X \end{array}$$

with $x \in X \subseteq \{0,1\}^n, c' \in (\mathbb{R}^n)^T$

Generalized case, also called the 'nominal problem'
Constraints are encoded into $X$

However, not all variables have to be integer, we consider Mixed Integer Linear Programs here

# Combinatorial Optimization - Robust?

Deviations?

$$\begin{array}{ll} \text{minimize} & c'x \\ \text{subject to} & x \in X \end{array}$$

# Combinatorial Optimization - Robust?

Deviations?

$$\begin{array}{ll} \text{minimize} & c'x \\ \text{subject to} & x \in X \end{array}$$

Range based deviations, not scenarios
Each modified entry $\tilde{c}_j$ with $j \in N = \{1, 2, \ldots, n\}$ of $c'$ takes value in $[c_j, c_j + d_j]$ with $d_j \geq 0, d_j$ finite

Set $X$ is fixed and does not deviate - there are other problems, where constraints can deviate too

# Combinatorial Optimization - Robust?

Deviations?

$$\begin{array}{ll} \text{minimize} & c'x \\ \text{subject to} & x \in X \end{array}$$

Range based deviations, not scenarios
Each modified entry $\tilde{c}_j$ with $j \in N = \{1, 2, \ldots, n\}$ of $c'$ takes value in $[c_j, c_j + d_j]$ with $d_j \geq 0, d_j$ finite

Set $X$ is fixed and does not deviate - there are other problems, where constraints can deviate too

All or nothing approach - Robustness Parameter?

# Combinatorial Optimization - Robust?

Deviations?

$$\begin{array}{ll} \text{minimize} & c'x \\ \text{subject to} & x \in X \end{array}$$

Range based deviations, not scenarios
Each modified entry $\tilde{c}_j$ with $j \in N = \{1, 2, \ldots, n\}$ of $c'$ takes value in $[c_j, c_j + d_j]$ with $d_j \geq 0, d_j$ finite

Set $X$ is fixed and does not deviate - there are other problems, where constraints can deviate too

All or nothing approach - Robustness Parameter?

Maximum number $\Gamma$ of cost entries
that we assume can deviate

# Combinatorial Optimization - $\Gamma$ Robust!

$$\begin{array}{ll} \text{minimize} & c'x + \max\limits_{\{S \mid S \subseteq N, |S| \leq \Gamma\}} \sum\limits_{j \in S} d_j x_j \\ \text{subject to} & x \in X \end{array}$$

# Combinatorial Optimization - $\Gamma$ Robust!

$$\begin{array}{ll} \text{minimize} & c'x + \max\limits_{\{S \mid S \subseteq N, |S| \leq \Gamma\}} \sum\limits_{j \in S} d_j x_j \\ \text{subject to} & x \in X \end{array}$$

At most $\Gamma$ cost entries can deviate!

We will show: There is an algorithm such that we can calculate $Z^*$ by just solving $n+1$ times the nominal instance!

# Combinatorial Optimization - $\Gamma$ Robust!

$$\begin{aligned} \text{minimize} \quad & c'x + \max_{\{S \mid S \subseteq N, |S| \leq \Gamma\}} \sum_{j \in S} d_j x_j \\ \text{subject to} \quad & x \in X \end{aligned}$$

$$\begin{aligned} \text{minimize} \quad & c'x + \max \sum_{j \in N} d_j x_j u_j \\ \text{subject to} \quad & u_j \in \{0,1\}, \qquad j \in N \\ & \sum_{j \in N} u_j \leq \Gamma \qquad j \in N \\ & x \in X \end{aligned}$$

$u_j$ can be relaxed

# Combinatorial Optimization - $\Gamma$ Robust!

$$\begin{array}{ll} \text{minimize} & c'x + \max_{\{S \mid S \subseteq N, |S| \leq \Gamma\}} \sum_{j \in S} d_j x_j \\ \text{subject to} & x \in X \end{array}$$

$$\begin{array}{lll} \text{minimize} & c'x + \max \sum_{j \in N} d_j x_j u_j & \\ \text{subject to} & u_j \in \{0,1\}, & j \in N \\ & \sum_{j \in N} u_j \leq \Gamma & j \in N \\ & x \in X & \end{array}$$

$u_j$ can be relaxed

Using Strong Duality!

$$\begin{array}{lll} \text{minimize} & c'x + \min(\Gamma\theta + \sum_{j \in N} y_j) & \\ \text{subject to} & y_j + \theta \geq d_j x_j, & j \in N \\ & y_j, \theta \geq 0, & j \in N \\ & x \in X & \end{array}$$

# Combinatorial Optimization - $\Gamma$ Robust!

$$\begin{aligned} \text{minimize} \quad & c'x + \max_{\{S \mid S \subseteq N, |S| \le \Gamma\}} \sum_{j \in S} d_j x_j \\ \text{subject to} \quad & x \in X \end{aligned}$$

$$\begin{aligned} \text{minimize} \quad & c'x + \max \sum_{j \in N} d_j x_j u_j \\ \text{subject to} \quad & u_j \in \{0, 1\}, \qquad j \in N \\ & \sum_{j \in N} u_j \le \Gamma \qquad j \in N \\ & x \in X \end{aligned}$$

$u_j$ can be relaxed

Using Strong Duality!

$$\begin{aligned} \text{minimize} \quad & c'x + \min(\Gamma\theta + \sum_{j \in N} y_j) \\ \text{subject to} \quad & y_j + \theta \ge d_j x_j, \qquad j \in N \\ & y_j, \theta \ge 0, \qquad j \in N \\ & x \in X \end{aligned}$$

Group min

# Combinatorial Optimization - $\Gamma$ Robust!

$$
\begin{aligned}
\text{minimize} \quad & c'x + \Gamma\theta + \sum_{j \in N} y_j \\
\text{subject to} \quad & y_j + \theta \geq d_j x_j, \quad j \in N \\
& y_j, \theta \geq 0, \quad j \in N \\
& x \in X
\end{aligned}
$$

# Combinatorial Optimization - $\Gamma$ Robust!

$$
\begin{array}{lll}
\text{minimize} & c'x + \Gamma\theta + \sum_{j \in N} y_j & \\
\text{subject to} & y_j + \theta \geq d_j x_j, & j \in N \\
 & y_j, \theta \geq 0, & j \in N \\
 & x \in X &
\end{array}
$$

Optimal solution $(x^*, y^*, \theta^*)$ satisfies
$y_j^* = \max(d_j x_j^* - \theta^*, 0)$

# Combinatorial Optimization - $\Gamma$ Robust!

$$\begin{array}{lll} \text{minimize} & c'x + \Gamma\theta + \sum_{j \in N} y_j & \\ \text{subject to} & y_j + \theta \geq d_j x_j, & j \in N \\ & y_j, \theta \geq 0, & j \in N \\ & x \in X & \end{array}$$

Optimal solution $(x^*, y^*, \theta^*)$ satisfies

$y_j^* = \max(d_j x_j^* - \theta^*, 0)$ Why?

Both

$y_j \geq d_j x_j - \theta$

$y_j \geq 0$

# Combinatorial Optimization - $\Gamma$ Robust!

$$\begin{array}{lll} \text{minimize} & c'x + \Gamma\theta + \sum_{j \in N} y_j & \\ \text{subject to} & y_j + \theta \geq d_j x_j, & j \in N \\ & y_j, \theta \geq 0, & j \in N \\ & x \in X & \end{array}$$

Optimal solution $(x^*, y^*, \theta^*)$ satisfies
$y_j^* = \max(d_j x_j^* - \theta^*, 0)$

Use $x_j \in \{0, 1\}$?

# Combinatorial Optimization - $\Gamma$ Robust!

$$
\begin{array}{lll}
\text{minimize} & c'x + \Gamma\theta + \sum_{j \in N} y_j & \\
\text{subject to} & y_j + \theta \geq d_j x_j, & j \in N \\
 & y_j, \theta \geq 0, & j \in N \\
 & x \in X &
\end{array}
$$

Optimal solution $(x^*, y^*, \theta^*)$ satisfies

$y_j^* = \max(d_j x_j^* - \theta^*, 0) = x_j^* \max(d_j - \theta^*, 0)$

Use $x_j \in \{0, 1\}$?

# Combinatorial Optimization - $\Gamma$ Robust!

$$
\begin{aligned}
\text{minimize} \quad & c'x + \Gamma\theta + \sum_{j\in N} y_j \\
\text{subject to} \quad & y_j + \theta \geq d_j x_j, \quad j \in N \\
& y_j, \theta \geq 0, \quad j \in N \\
& x \in X
\end{aligned}
$$

Optimal solution $(x^*, y^*, \theta^*)$ satisfies

$y_j^* = \max(d_j x_j^* - \theta^*, 0) = x_j^* \max(d_j - \theta^*, 0)$ $\quad 0 \cdot x_j = 0$

Use $x_j \in \{0, 1\}$?

$x_j^* = 0 \rightarrow d_j x_j^* - \theta = 0 - \theta \leq 0 \rightarrow y_j^* = 0$

$x_j^* = 1 \rightarrow d_j x_j^* - \theta = 1 \cdot d_j - \theta = 1(d_j - \theta)$

# Combinatorial Optimization - $\Gamma$ Robust!

$$\begin{array}{lll} \text{minimize} & c'x + \Gamma\theta + \sum_{j \in N} y_j & \\ \text{subject to} & y_j + \theta \geq d_j x_j, & j \in N \\ & y_j, \theta \geq 0, & j \in N \\ & x \in X & \end{array}$$

Optimal solution $(x^*, y^*, \theta^*)$ satisfies
$y_j^* = \max(d_j x_j^* - \theta^*, 0) = x_j^* \max(d_j - \theta^*, 0)$

$$Z^* = \min_{x \in X, \theta \geq 0} \Big( c'x + \Gamma\theta + \sum_{j \in N} x_j \cdot max(d_j - \theta, 0) \Big)$$

# Combinatorial Optimization - $\Gamma$ Robust!

$$
\begin{array}{lll}
\text{minimize} & c'x + \Gamma\theta + \sum_{j \in N} y_j & \\
\text{subject to} & y_j + \theta \geq d_j x_j, & j \in N \\
 & y_j, \theta \geq 0, & j \in N \\
 & x \in X &
\end{array}
$$

Optimal solution $(x^*, y^*, \theta^*)$ satisfies

$y_j^* = \max(d_j x_j^* - \theta^*, 0) = x_j^* \max(d_j - \theta^*, 0)$

replace $y_j$

$$Z^* = \min_{x \in X, \theta \geq 0} \Big( c'x + \Gamma\theta + \sum_{j \in N} x_j \cdot max(d_j - \theta, 0) \Big)$$

constraints

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min_{x \in X, \theta \geq 0} \left( c'x + \Gamma\theta + \sum_{j \in N} x_j \cdot max(d_j - \theta, 0) \right)$$

Idea: Partition $\theta$

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min_{x \in X, \theta \geq 0} \Big( c'x + \Gamma\theta + \sum_{j \in N} x_j \cdot max(d_j - \theta, 0) \Big)$$

Idea: Partition $\theta$

Indizes are ordered such that $d_1 \geq d_2 \geq \cdots \geq d_n$
Define $d_{n+1} = 0$

Partition $\mathbb{R}^+$ into $[0 = d_{n+1}, d_n], [d_n, d_{n-1}]. \ldots, [d_2, d_1], [d_1, \infty[$

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min_{x \in X, \theta \geq 0} \left( c'x + \Gamma\theta + \sum_{j \in N} x_j \cdot max(d_j - \theta, 0) \right)$$

Idea: Partition $\theta$

Indizes are ordered such that $d_1 \geq d_2 \geq \cdots \geq d_n$
Define $d_{n+1} = 0$

Partition $\mathbb{R}^+$ into $[0 = d_{n+1}, d_n], [d_n, d_{n-1}] \ldots, [d_2, d_1], [d_1, \infty[$

$$\sum_{j \in N} x_j max(d_j - \theta, 0) = \begin{cases} \sum_{j=1}^{l-1} x_j(d_j - \theta), & \text{if } \theta \in [d_l, d_{l-1}] \text{ for some } l \in [n+1, 2] \\ 0, & \text{if } \theta \in [d_1, \infty[ \end{cases}$$

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min_{x \in X, \theta \geq 0} \Big( c'x + \Gamma\theta + \sum_{j \in N} x_j \cdot max(d_j - \theta, 0) \Big)$$

Idea: Partition $\theta$

Indizes are ordered such that $d_1 \geq d_2 \geq \cdots \geq d_n$
Define $d_{n+1} = 0$

Partition $\mathbb{R}^+$ into $[0 = d_{n+1}, d_n], [d_n, d_{n-1}], \ldots, [d_2, d_1], [d_1, \infty[$

$$d_n \leq \cdots \leq d_l \leq \theta \rightarrow \forall j \geq l : (d_j - \theta) \leq 0$$

$$\sum_{j \in N} x_j max(d_j - \theta, 0) = \begin{cases} \sum_{j=1}^{l-1} x_j (d_j - \theta), & \text{if } \theta \in [d_l, d_{l-1}] \text{ for some } l \in [n+1, 2] \\ 0, & \text{if } \theta \in [d_1, \infty[ \end{cases}$$

$$\theta \geq d_1 \rightarrow \forall j \in N : d_j - \theta \leq 0$$

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min_{x \in X, \theta \geq 0} \Big( c'x + \Gamma\theta + \sum_{j \in N} x_j \cdot max(d_j - \theta, 0) \Big)$$

Idea: Partition $\theta$

Use partioning!

$$Z^* = \min_{l=1}^{n+1} Z^l$$

with $Z^l = \min_{x \in X, \theta \in [d_l, d_{l-1}]} \Big( c'x + \Gamma\theta + \sum_{j=1}^{l-1} x_j(d_j - \theta) \Big)$

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min_{x \in X, \theta \geq 0} \left( c'x + \Gamma\theta + \sum_{j \in N} x_j \cdot max(d_j - \theta, 0) \right)$$

Idea: Partition $\theta$

$$Z^* = \min_{l=1}^{n+1} Z^l$$

Use partioning!

0 iterations

with $Z^l = \min_{x \in X, \theta \in [d_l, d_{l-1}]} \left( c'x + \Gamma\theta + \sum_{j=1}^{l-1} x_j(d_j - \theta) \right)$

Special case $l = 1$, $d_0 = \infty$

Only lower bound worth considering

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min_{x \in X, \theta \geq 0} \Big( c'x + \Gamma\theta + \sum_{j \in N} x_j \cdot max(d_j - \theta, 0) \Big)$$

Idea: Partition $\theta$

$$Z^* = \min_{l=1}^{n+1} Z^l$$

Use partioning!

with $Z^l = \min_{x \in X, \theta \in [d_l, d_{l-1}]} \Big( c'x + \Gamma\theta + \sum_{j=1}^{l-1} x_j (d_j - \theta) \Big)$

$$Z^l = \min \Bigg( \Gamma d_l + \min_{x \in X} \Big( c'x + \sum_{j=1}^{l-1} x_j (d_j - d_l) \Big),$$

Minimum at either limit of $\theta$

$$\Gamma d_{l-1} + \min_{x \in X} \Big( c'x + \sum_{j=1}^{l-1} x_j (d_j - d_{l-1}) \Big) \Bigg)$$

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min\Bigg(\Gamma d_1 + \min_{x\in X}\Big(c'x\Big),$$

$$\ldots,$$

$$\Gamma d_l + \min_{x\in X}\Big(c'x + \sum_{j=1}^{l-1} x_j(d_j - d_l)\Big),$$

$$\ldots,$$

$$\min_{x\in X}\Big(c'x + \sum_{j=1}^{n} x_j d_j\Big)\Bigg)$$

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min\Bigg(\Gamma d_1 + \min_{x \in X}\Big(c'x\Big), \quad l = 1, \text{ sum is always } 0$$

$$\ldots,$$

$$\Gamma d_l + \min_{x \in X}\Bigg(c'x + \sum_{j=1}^{l-1} x_j(d_j - d_l)\Bigg),$$

$$\ldots,$$

$$\min_{x \in X}\Bigg(c'x + \sum_{j=1}^{n} x_j d_j\Bigg)\Bigg)$$

$l = n + 1$ with $d_{n+1} = 0$

# Combinatorial Optimization - $\Gamma$ Robust!

$$Z^* = \min\Bigg(\Gamma d_1 + \min_{x\in X}\Big(c'x\Big), \\ \dots, \\ \Gamma d_l + \min_{x\in X}\Big(c'x + \sum_{j=1}^{l-1} x_j(d_j - d_l)\Big), \\ \dots, \\ \min_{x\in X}\Big(c'x + \sum_{j=1}^{n} x_j d_j\Big)\Bigg)$$

Solve $Z^*$ by solving at most $n+1$ modified base instances!
Easy problems stay easy! Woo

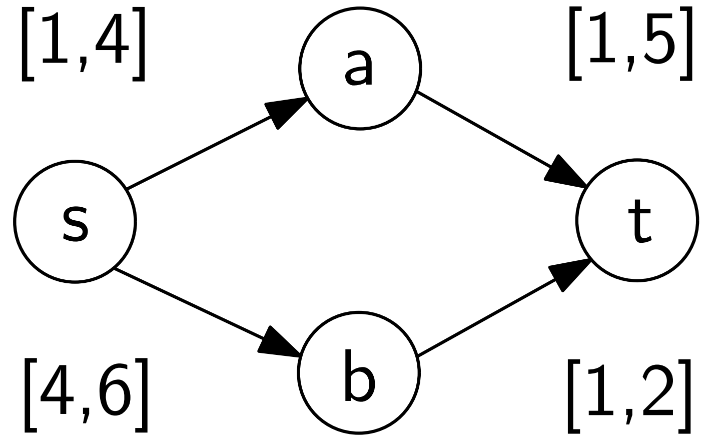
# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

Simplification

$$z^l = \Gamma d_l + \min_{x \in X}\Big(c'x + \sum_{j=1}^{l-1} x_j(d_j - d_l)\Big)$$

$\Gamma = 1$




Deviations?

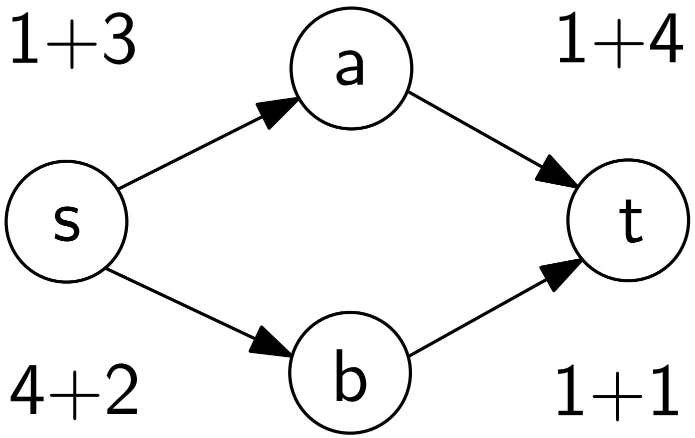
# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

Simplification

$$z^l = \Gamma d_l + \min_{x \in X} \Big( c'x + \sum_{j=1}^{l-1} x_j (d_j - d_l) \Big)$$

$\Gamma = 1$




Ordering?

Reminder: $d_1 \geq \cdots \geq d_n$

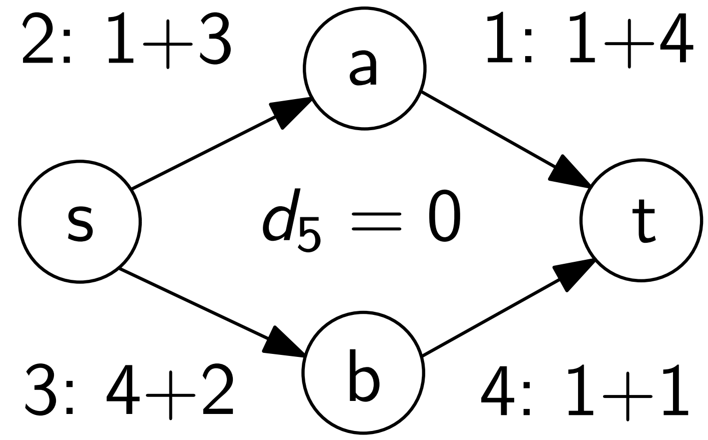
# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

Simplification

$$z^l = \Gamma d_l + \min_{x \in X}\left(c'x + \sum_{j=1}^{l-1} x_j(d_j - d_l)\right)$$

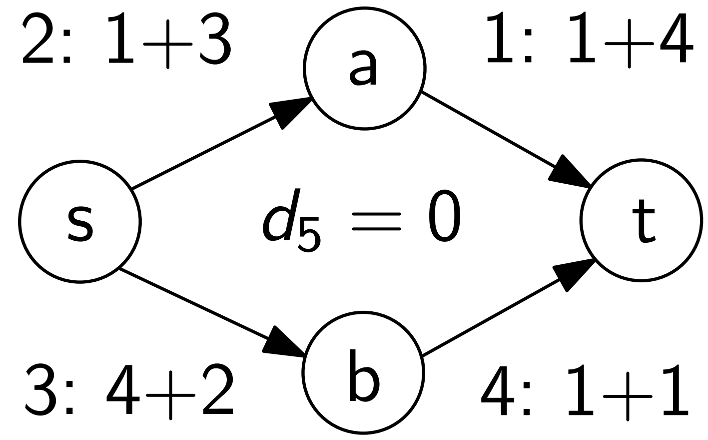
$\Gamma = 1$



# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

$\Gamma = 1$




## Simplification

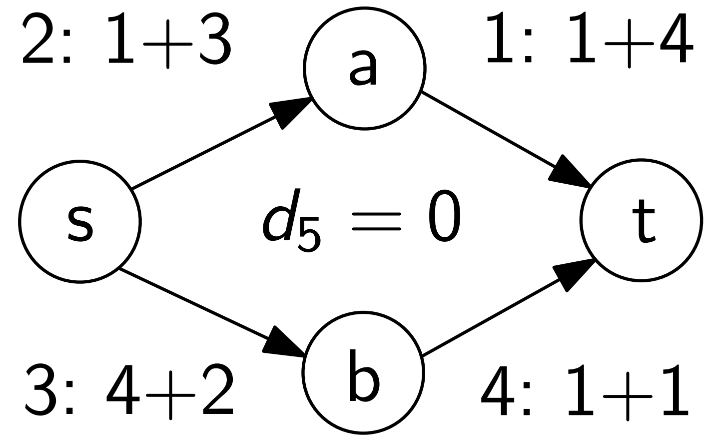
$$z^l = \Gamma d_l + \min_{x \in X}\left(c'x + \sum_{j=1}^{l-1} x_j (d_j - d_l)\right)$$

$$z^1 = \Gamma d_1 + \min_{x \in X}\left(c'x + \sum_{j=1}^{1-1} x_j (d_j - d_1)\right)$$

# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

$\Gamma = 1$




## Simplification

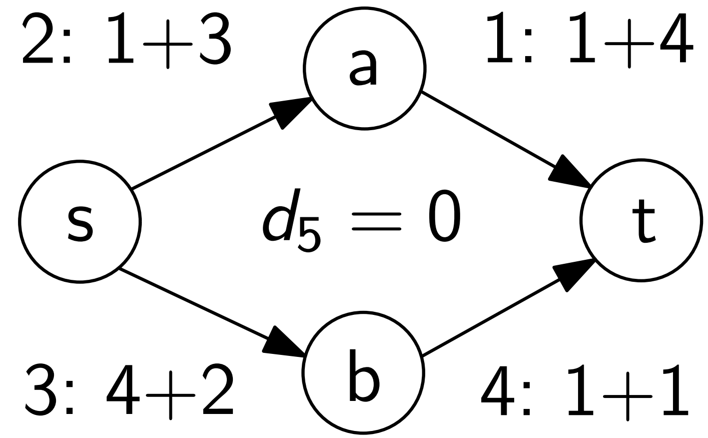
$$z^l = \Gamma d_l + \min_{x \in X} \left( c'x + \sum_{j=1}^{l-1} x_j (d_j - d_l) \right)$$

$$z^1 = \Gamma d_1 + \min_{x \in X} \left( c'x + \sum_{j=1}^{1-1} x_j (d_j - d_1) \right) = 4 \cdot 1 + \min_{x \in X} \left( c'x \right)$$

# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

$\Gamma = 1$




## Simplification

$$z^l = \Gamma d_l + \min_{x \in X}\Big(c'x + \sum_{j=1}^{l-1} x_j (d_j - d_l)\Big)$$

$$z^1 = \Gamma d_1 + \min_{x \in X}\Big(c'x + \sum_{j=1}^{1-1} x_j (d_j - d_1)\Big) = 4 \cdot 1 + \min_{x \in X}\Big(c'x\Big) = 6$$

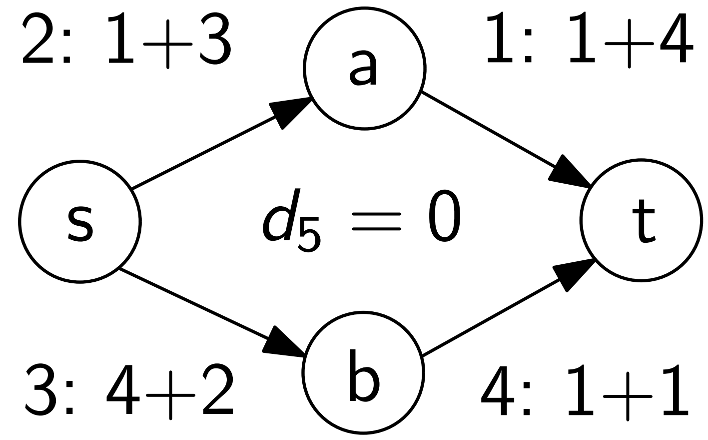
# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

## Simplification

$\Gamma = 1$

$$z^l = \Gamma d_l + \min_{x \in X}\Big(c'x + \sum_{j=1}^{l-1} x_j(d_j - d_l)\Big)$$

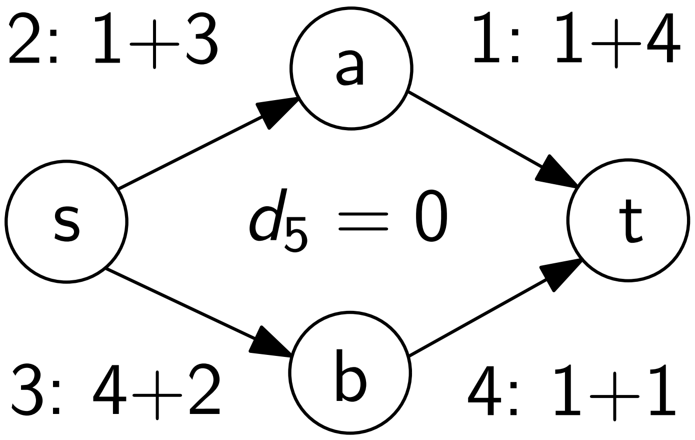



$z^1 = 6$ (over a)
$z^2 = 6$ (over a)
$z^3 = 7$ (over a or b)

# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

$\Gamma = 1$




## Simplification

$$z^l = \Gamma d_l + \min_{x \in X}\left(c'x + \sum_{j=1}^{l-1} x_j(d_j - d_l)\right)$$

$z^1 = 6$ (over a)
$z^2 = 6$ (over a)
$z^3 = 7$ (over a or b)

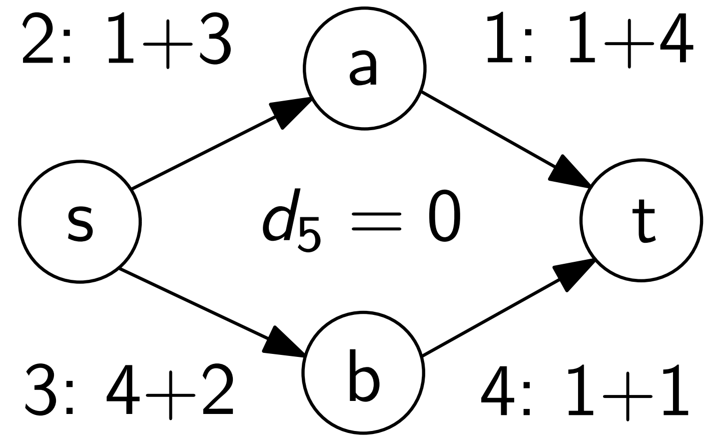
$$z^4 = \Gamma d_4 + \min_{x \in X}\left(c'x + \sum_{j=1}^{4-1} x_j(d_j - d_4)\right) =$$

$$1 \cdot 1 + \min_{x \in X}\left(c'x + 3x_1 + 2x_2 + 1x_3\right)$$

# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

$\Gamma = 1$




## Simplification

$$z^l = \Gamma d_l + \min_{x \in X} \left( c'x + \sum_{j=1}^{l-1} x_j (d_j - d_l) \right)$$

$z^1 = 6$ (over a)
$z^2 = 6$ (over a)
$z^3 = 7$ (over a or b)

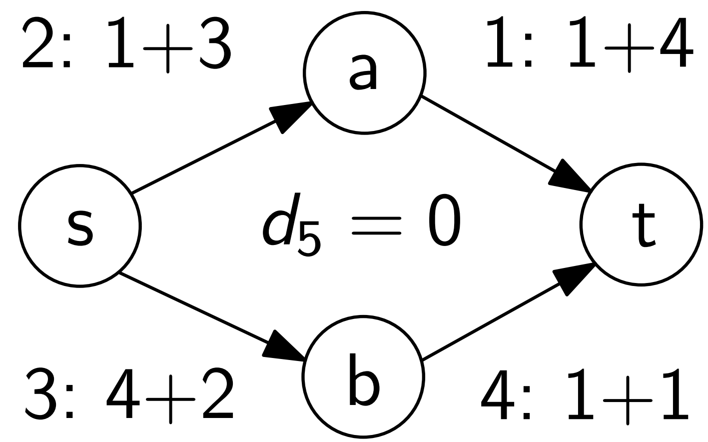
$$z^4 = \Gamma d_4 + \min_{x \in X} \left( c'x + \sum_{j=1}^{4-1} x_j (d_j - d_4) \right) =$$

$$1 \cdot 1 + \min_{x \in X} \left( c'x + 3x_1 + 2x_2 + 1x_3 \right) = 1 + \min(7, 6) = 7$$

# Example

$$Z^* = \min_{l=1}^{n+1} z^l$$

$\Gamma = 1$




Simplification

$$z^l = \Gamma d_l + \min_{x \in X}\Big(c'x + \sum_{j=1}^{l-1} x_j(d_j - d_l)\Big)$$

$z^1 = 6$ (over a)
$z^2 = 6$ (over a)
$z^3 = 7$ (over a or b)
$z^4 = 7$ (over b)
$z^5 = 7$ (over b)

Selection $x_1, x_2$ (routing over a) is optimal with $Z^* = 6$

# Further Results

- Similar strategy for general discrete problems, albeit with worse runtime (but still polynomial factor)
- Similar strategy for $\alpha$ approximable combinatorial problems, that stay $\alpha$ approximable by solving similar $n+1$ modified nominal instances
- Through proper construction, can be applied to cost flow networks too

# Sources

- Robust discrete optimization and network flows by Dimitris Bertsimas and Melvyn Sim
- Linear Programming: Lecture Notes by Michel Goemans