

Encodings in Mixed Integer Linear Programming

Juan Pablo Vielma

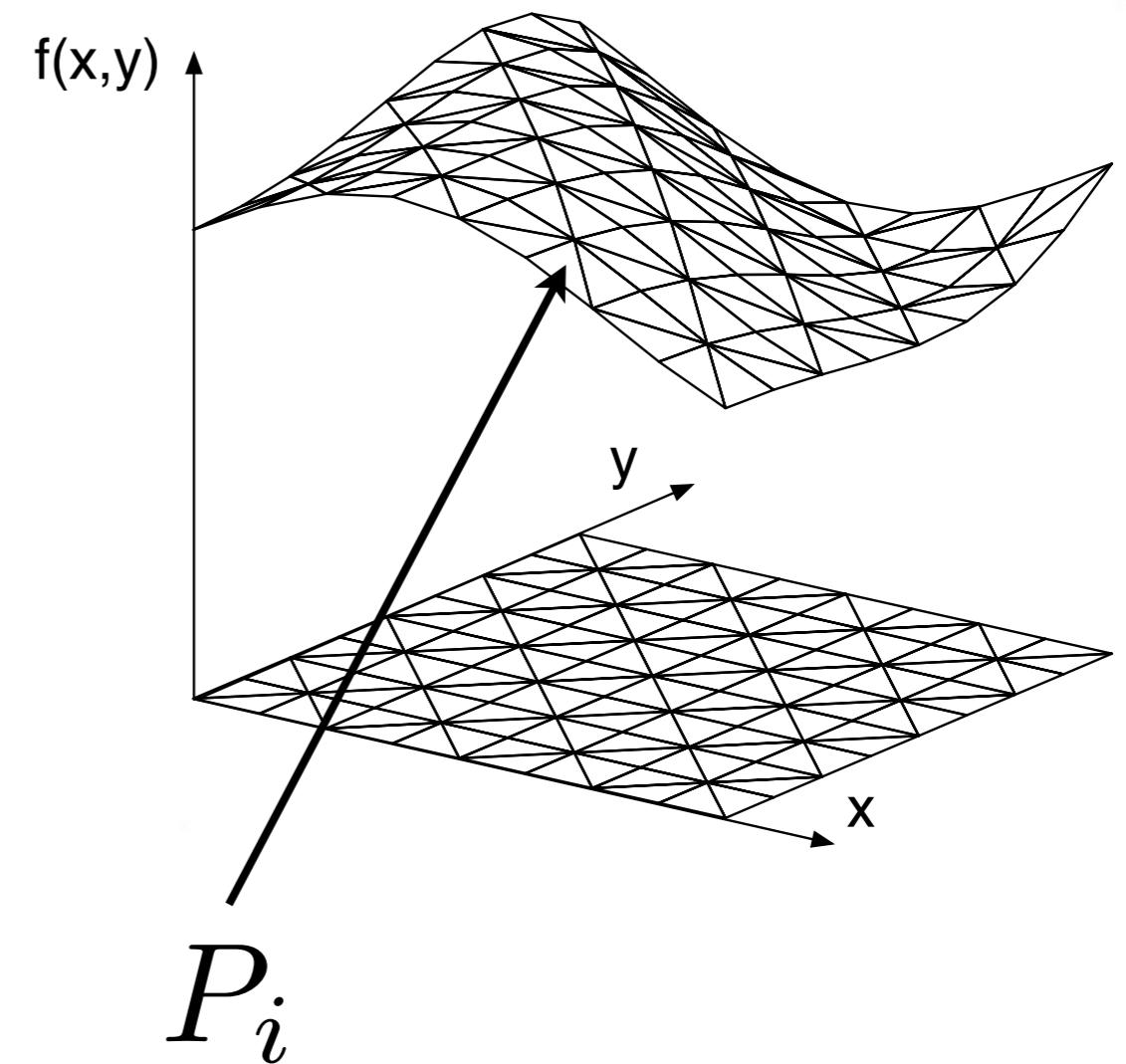
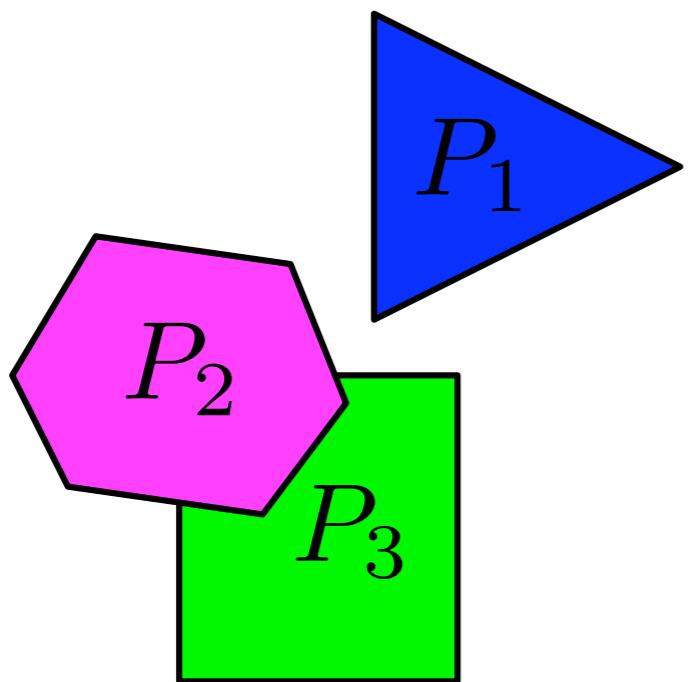
*Sloan School of Business,
Massachusetts Institute of Technology*

Universidad de Chile,
December, 2013 – Santiago, Chile.

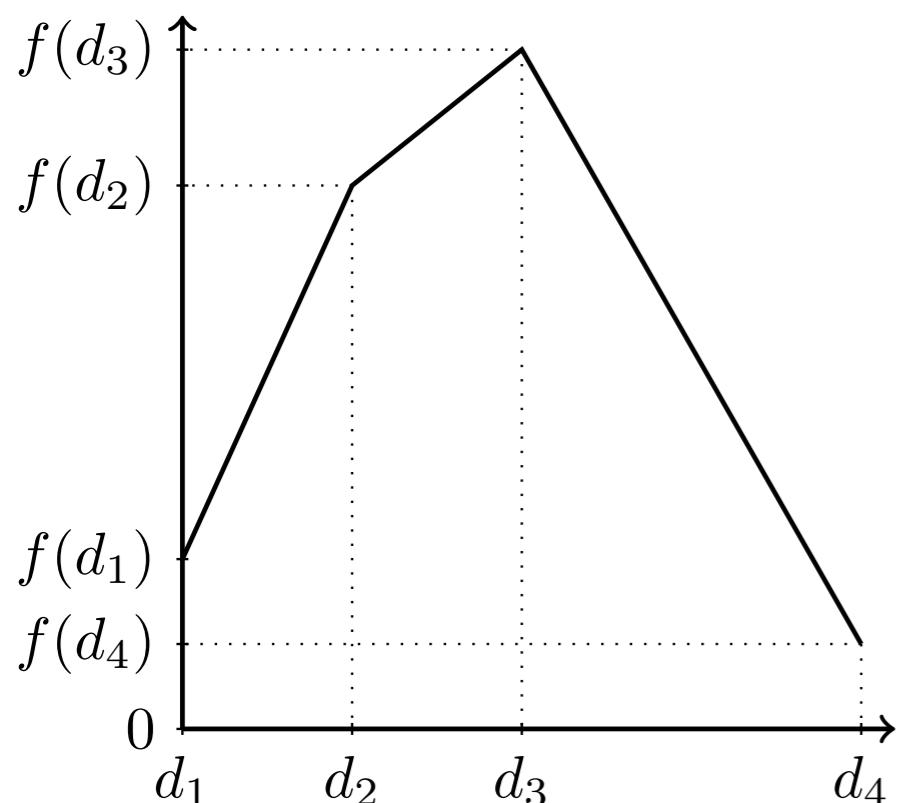
Mixed Integer Binary Formulations

- MIP Formulations = Model Finite Alternatives

$$x \in \bigcup_{i=1}^n P_i \subseteq \mathbb{R}^d$$



Textbook Formulation



Formulation for $f(x)=z$

$$\sum_{i=1}^4 d_i \lambda_i = x,$$

$$\sum_{i=1}^4 \lambda_i = 1,$$

$$\sum_{i=1}^3 y_i = 1,$$

$$\lambda_1 \leq y_1,$$

$$\lambda_3 \leq y_2 + y_3, \quad \lambda_4 \leq y_3$$

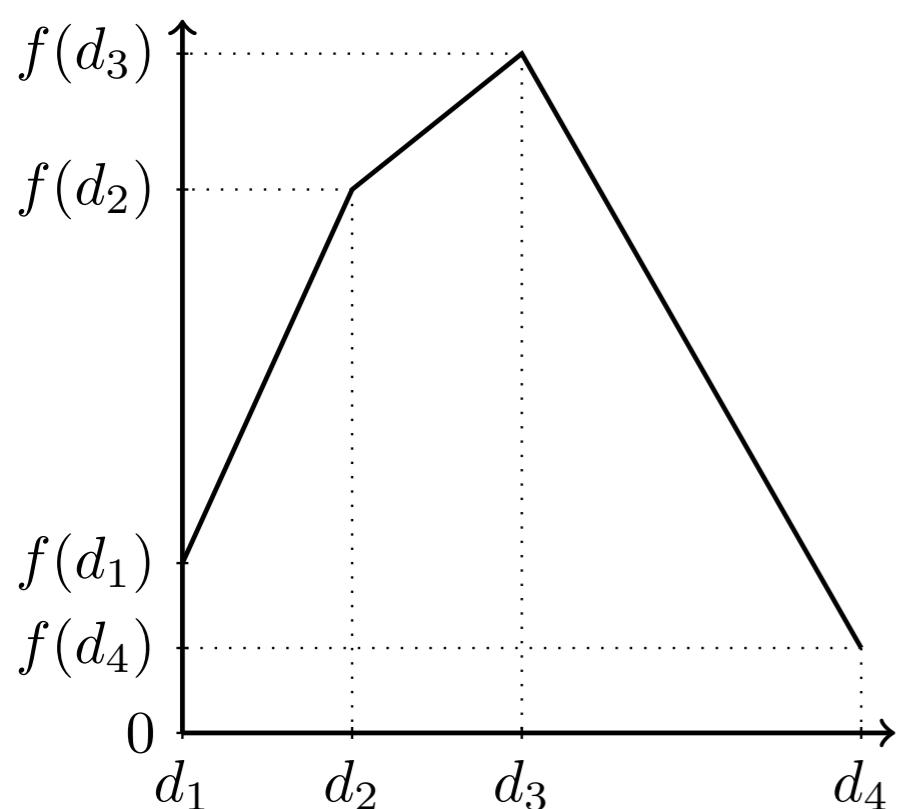
$$\sum_{i=1}^4 f(d_i) \lambda_i = z$$

$$\lambda_i \geq 0$$

$$y_i \in \{0, 1\}$$

$$\lambda_2 \leq y_1 + y_2$$

Textbook Formulation

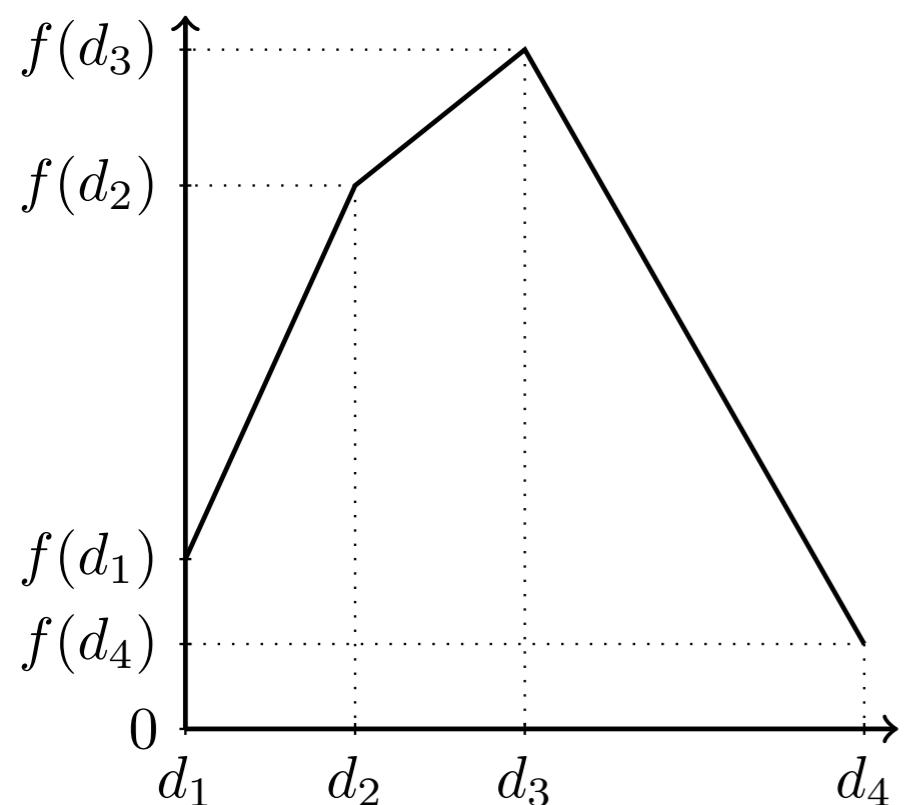


Formulation for $f(x)=z$

$$\begin{aligned} \sum_{i=1}^4 d_i \lambda_i &= x, & \sum_{i=1}^4 f(d_i) \lambda_i &= z \\ \sum_{i=1}^4 \lambda_i &= 1, & \lambda_i &\geq 0 \\ \sum_{i=1}^3 y_i &= 1, & y_i &\in \{0, 1\} \\ \lambda_1 &\leq y_1, & \lambda_2 &\leq y_1 + y_2 \\ \lambda_3 &\leq y_2 + y_3, & \lambda_4 &\leq y_3 \end{aligned}$$

“Weak”

Better Formulation



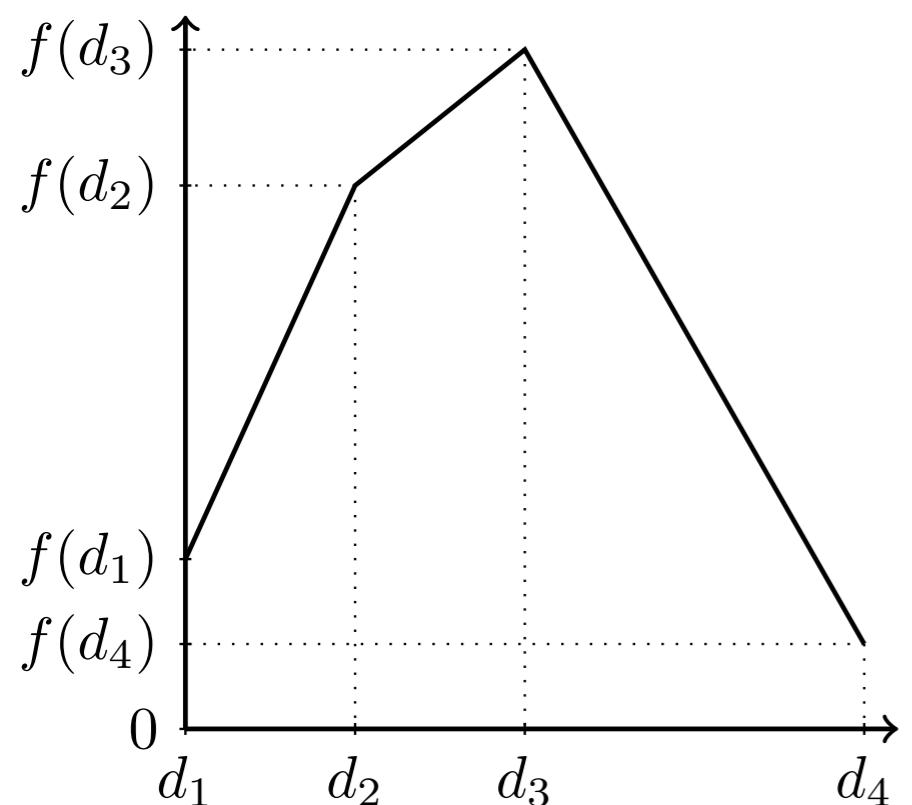
Formulation for $f(x)=z$

$$d_0 + \sum_{i=1}^3 (d_{i+1} - d_i) \delta_i = x,$$

$$f(d_0) + \sum_{i=1}^3 (f(d_{i+1}) - f(d_i)) \delta_i = z$$

$$\delta_3 \leq y_2 \leq \delta_2 \leq y_1 \leq \delta_1$$
$$y_i \in \{0, 1\}$$

Better Formulation



Formulation for $f(x)=z$

$$d_0 + \sum_{i=1}^3 (d_{i+1} - d_i) s_i = x,$$
$$f(d_0) + \sum_{i=1}^3 (f(d_{i+1}) - f(d_i)) \delta_i = z$$

Integral

$$\delta_3 \leq y_2 \leq \delta_2 \leq y_1 \leq \delta_1$$
$$y_i \in \{0, 1\}$$

Solve Times in CPLEX 11



Weak



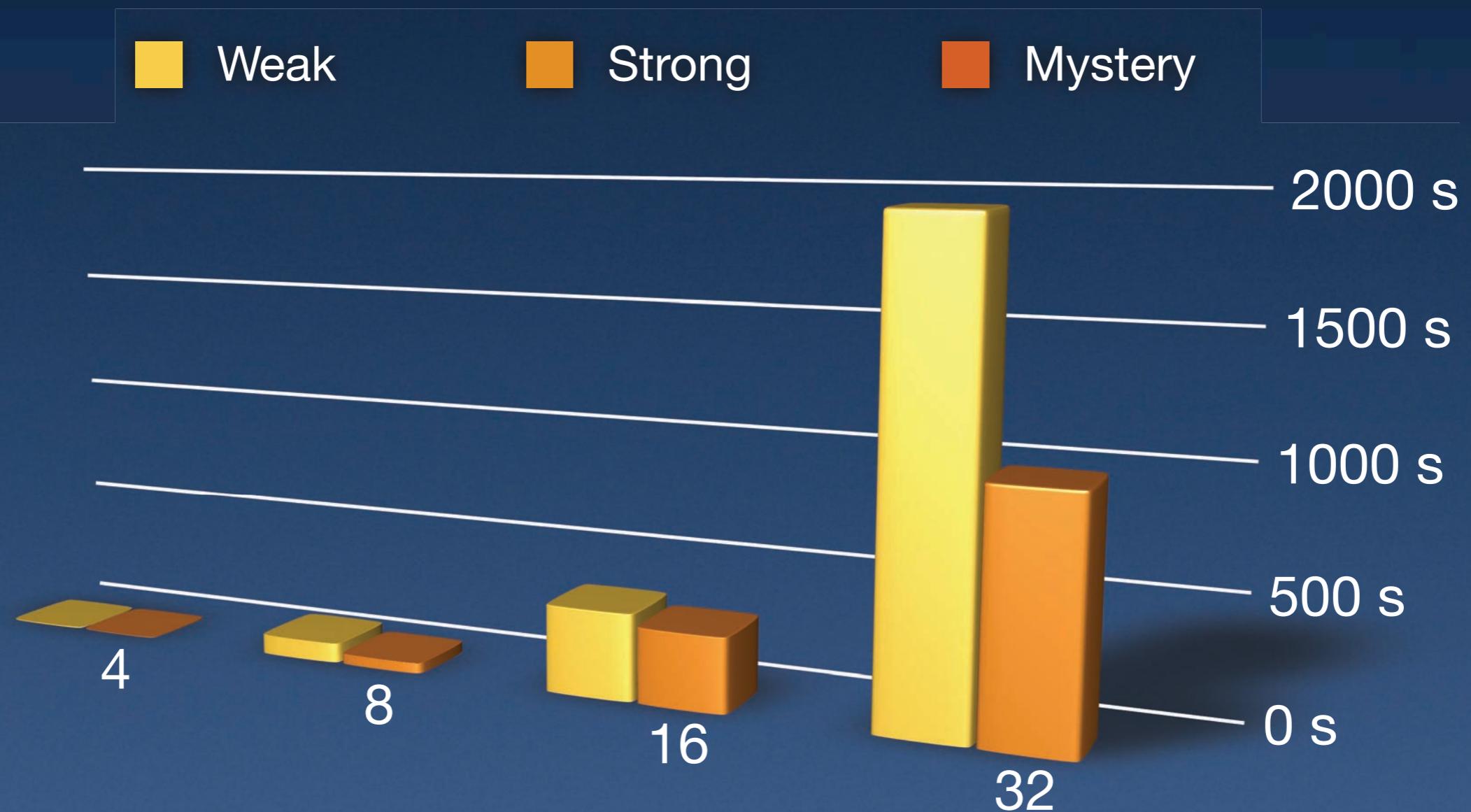
Strong



Mystery

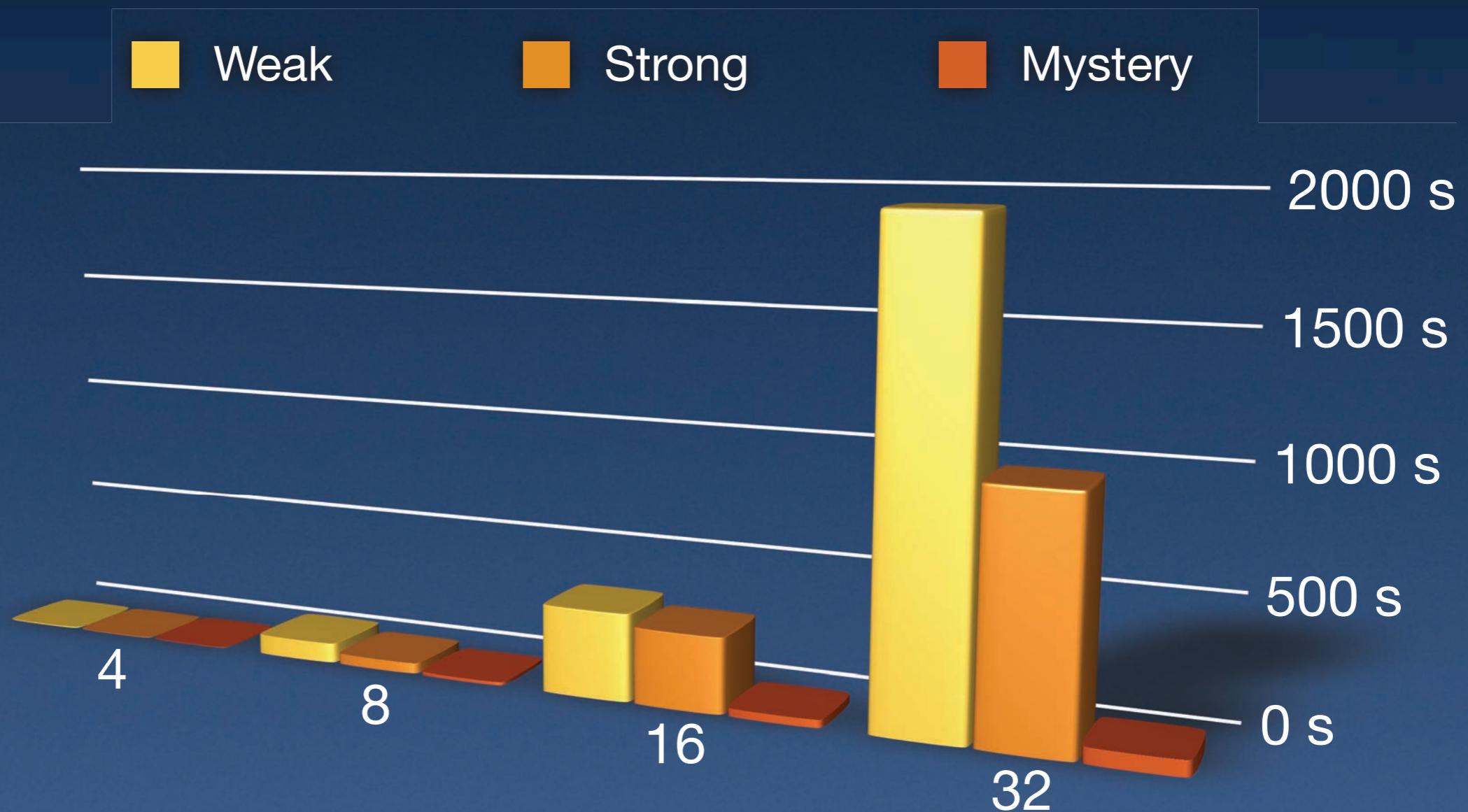
- Transportation Problems in V., Ahmed and Nemhauser '10.

Solve Times in CPLEX 11



- Transportation Problems in V., Ahmed and Nemhauser '10.

Solve Times in CPLEX 11



- Transportation Problems in V., Ahmed and Nemhauser '10.

Outline

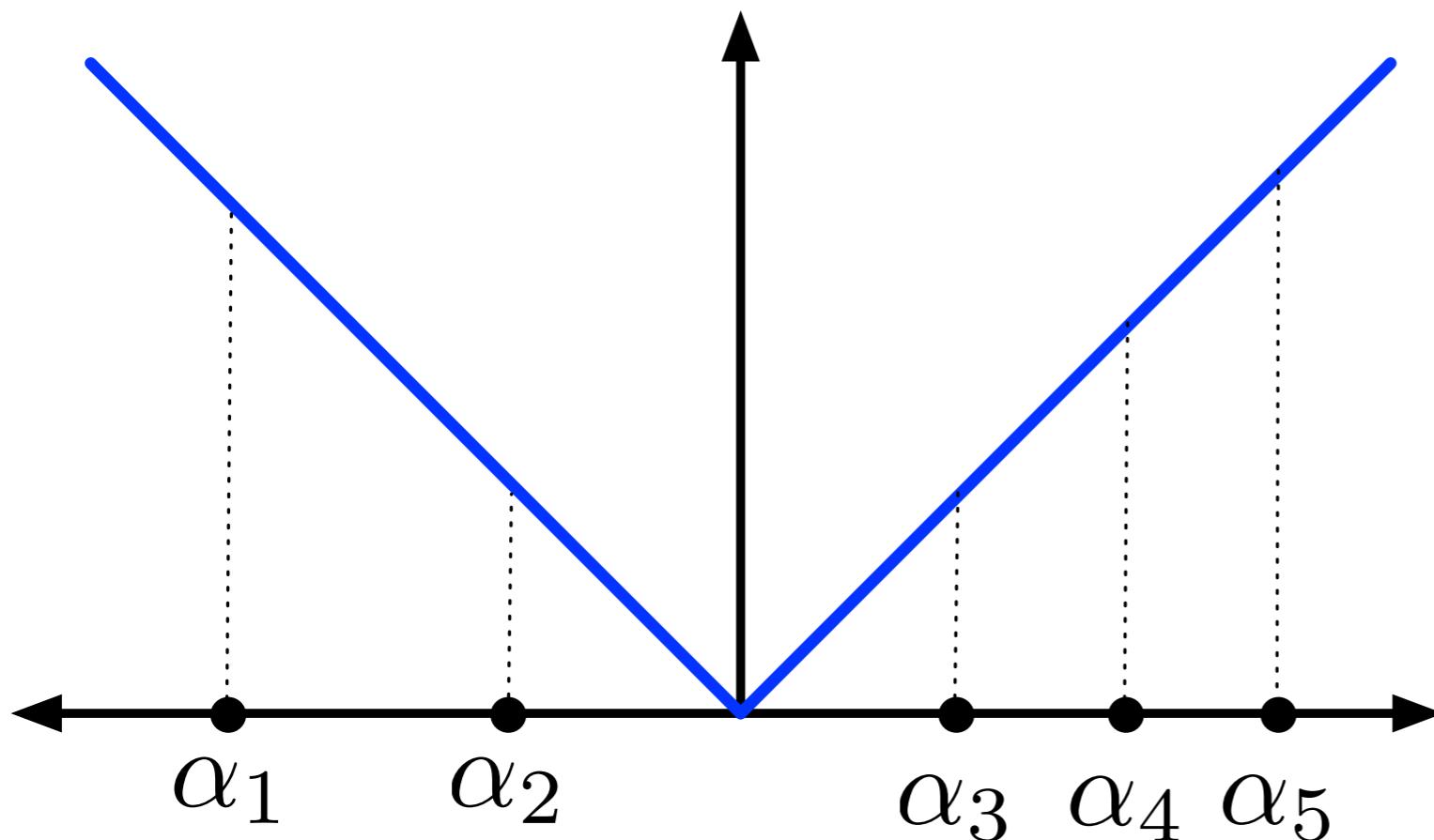
- MIP v/s constraint branching.
- “Have your cake and eat it too” formulation
- Step 1: Encoding alternatives.
- Step 2: Combine with strong “standard” formulation.
- Summary, Extensions and More.

Formulating Discrete Alternatives

$$\min |x|$$

s.t.

$$x \in \{\alpha_i\}_{i=1}^n$$

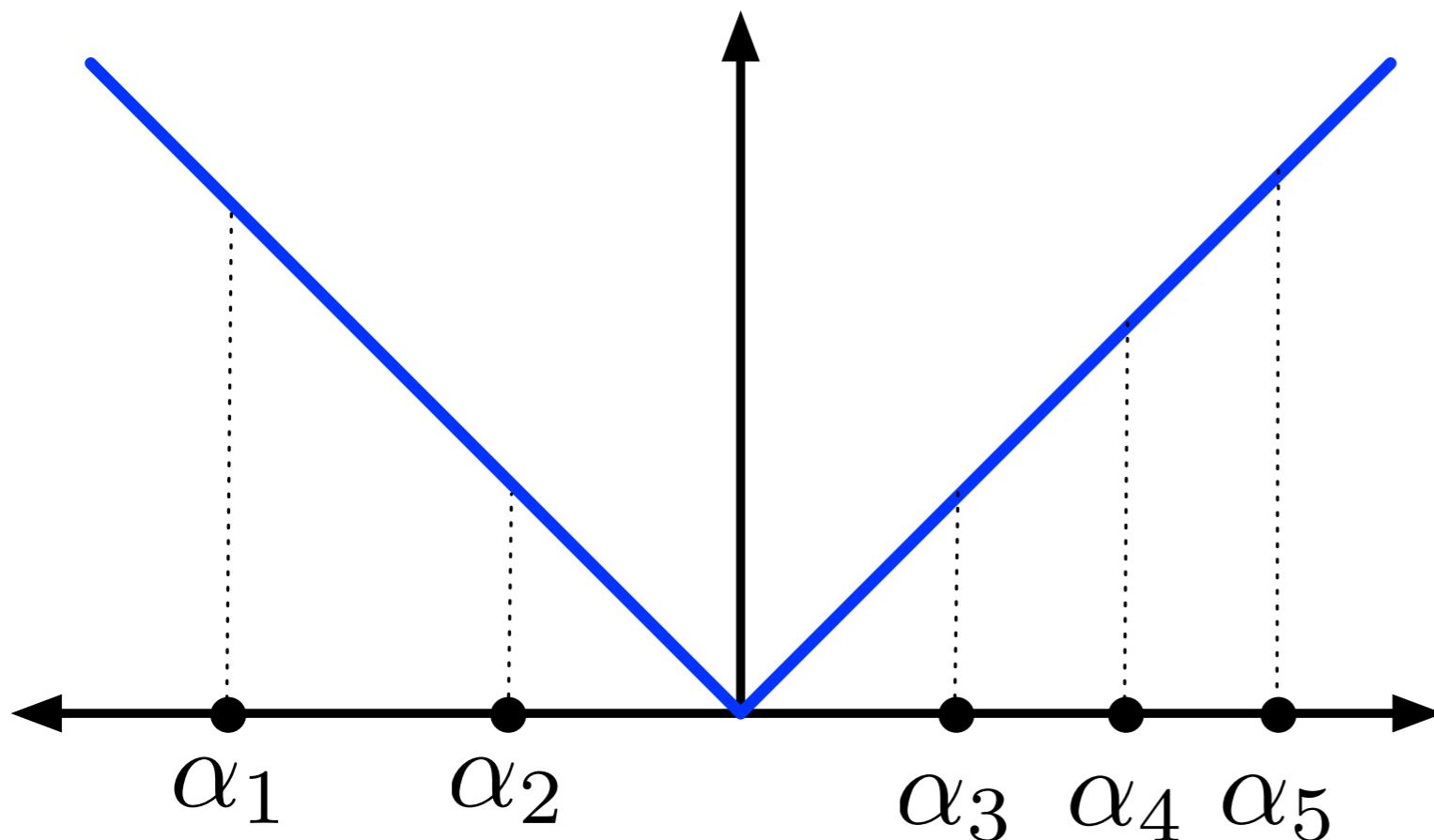


Formulating Discrete Alternatives

$$\min |x|$$

s.t.

$$x \in \{\alpha_i\}_{i=1}^n$$



$$\min |x|$$

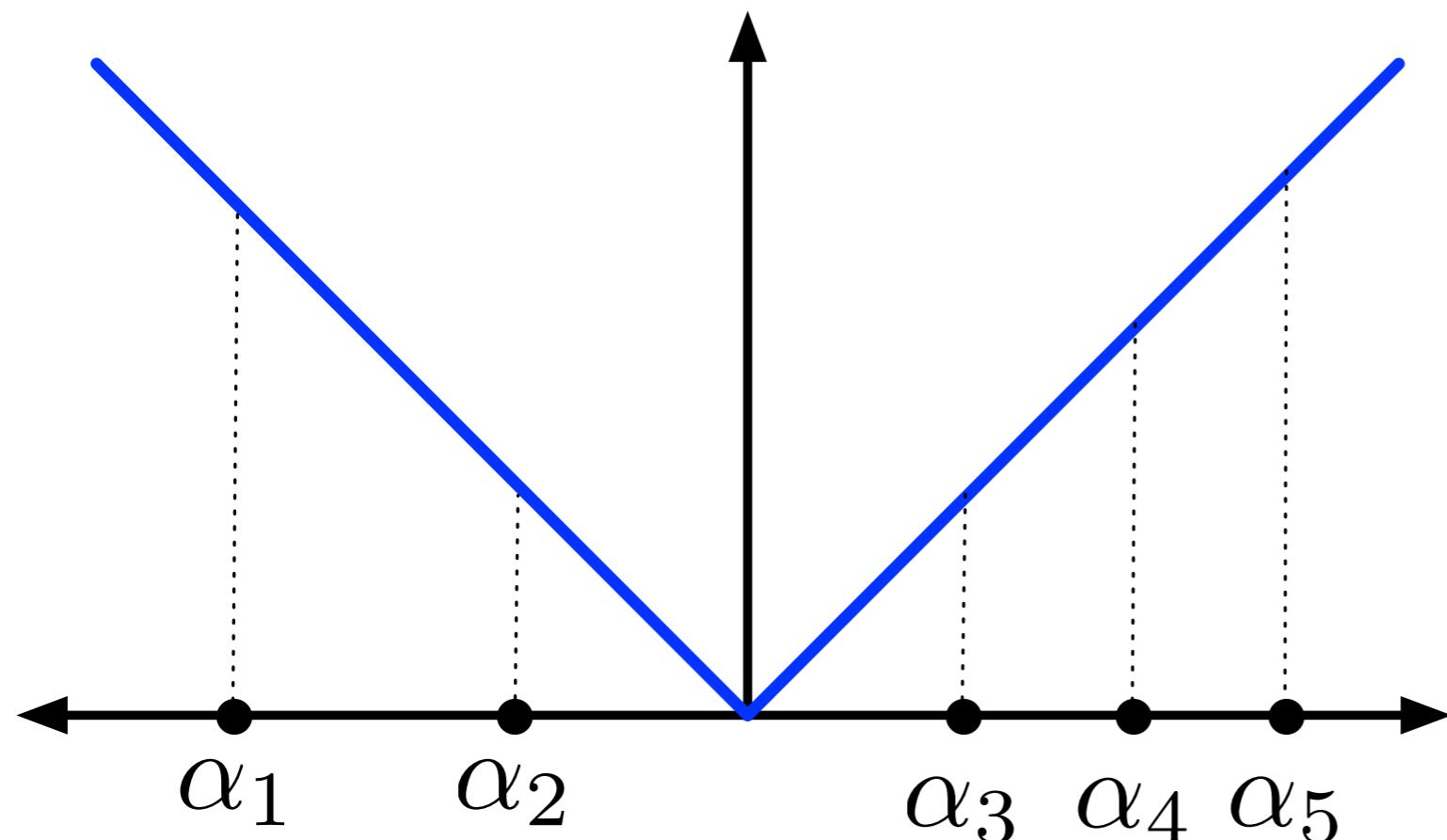
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

$$\sum_{i=1}^n \lambda_i = 1$$

$$\lambda \in \{0, 1\}^n$$

Formulating Discrete Alternatives

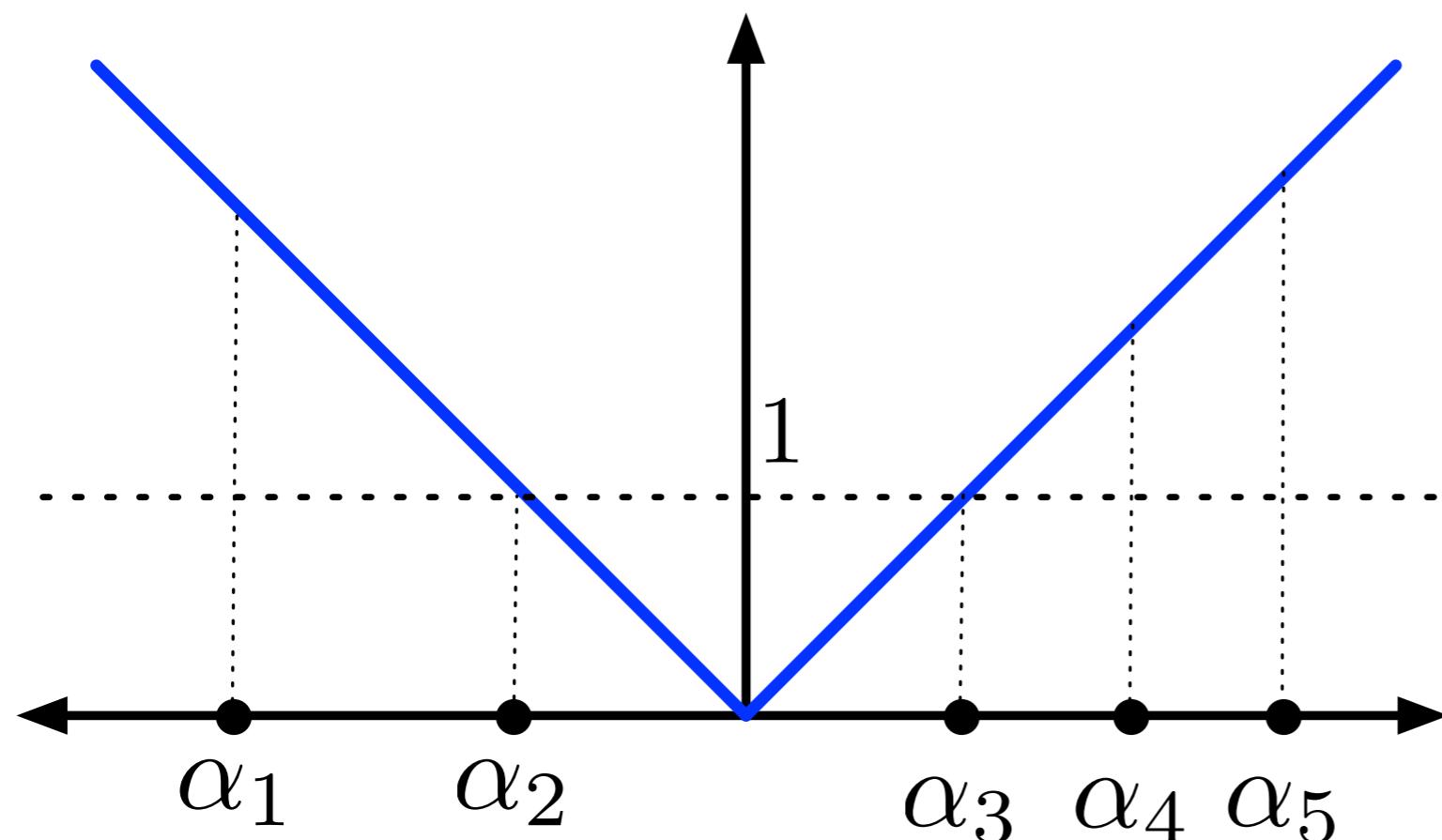

$$\min$$
$$s.t.$$

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

$$\sum_{i=1}^n \lambda_i = 1$$

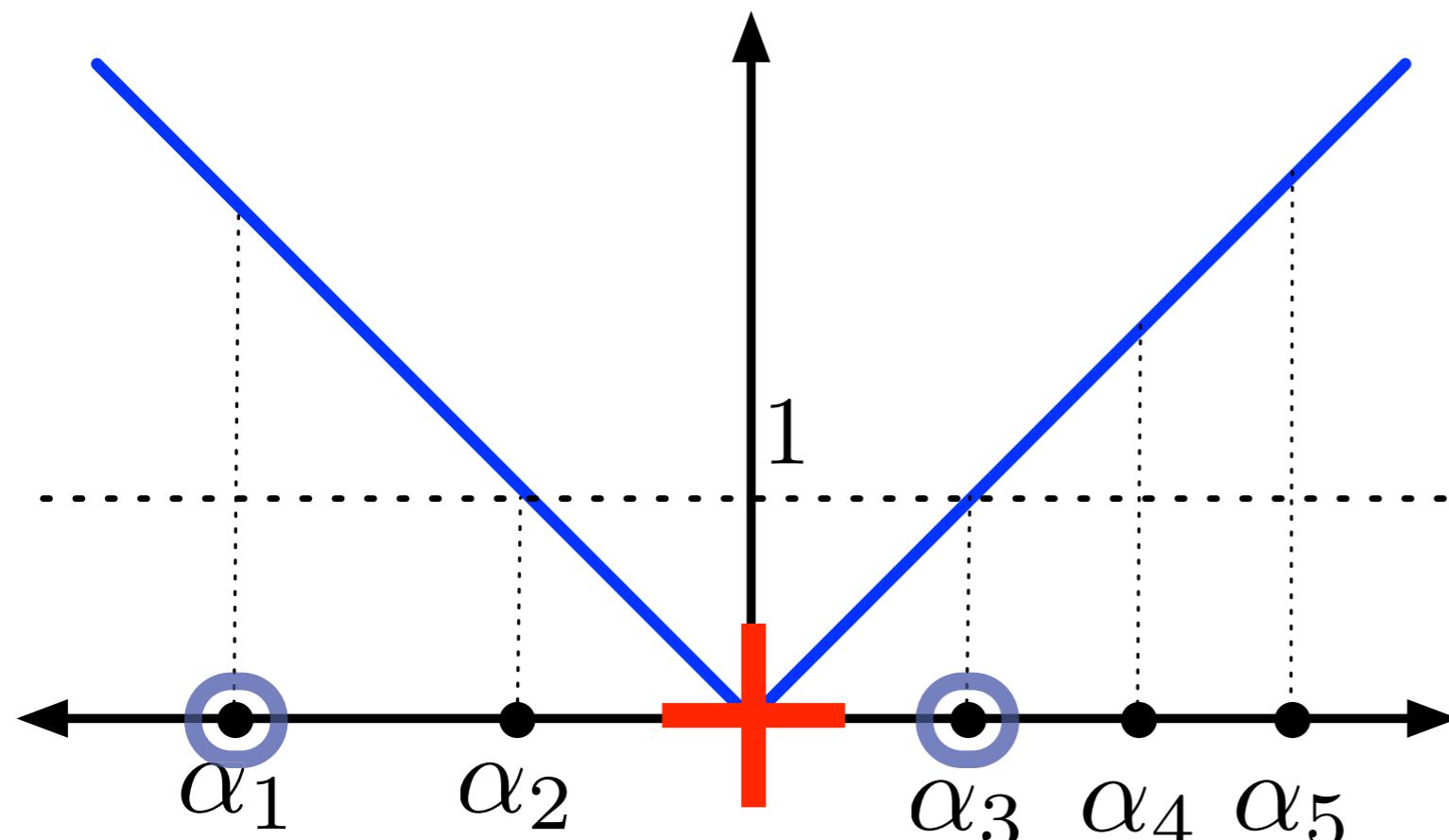
$$\lambda \in \{0, 1\}^n$$

Formulating Discrete Alternatives



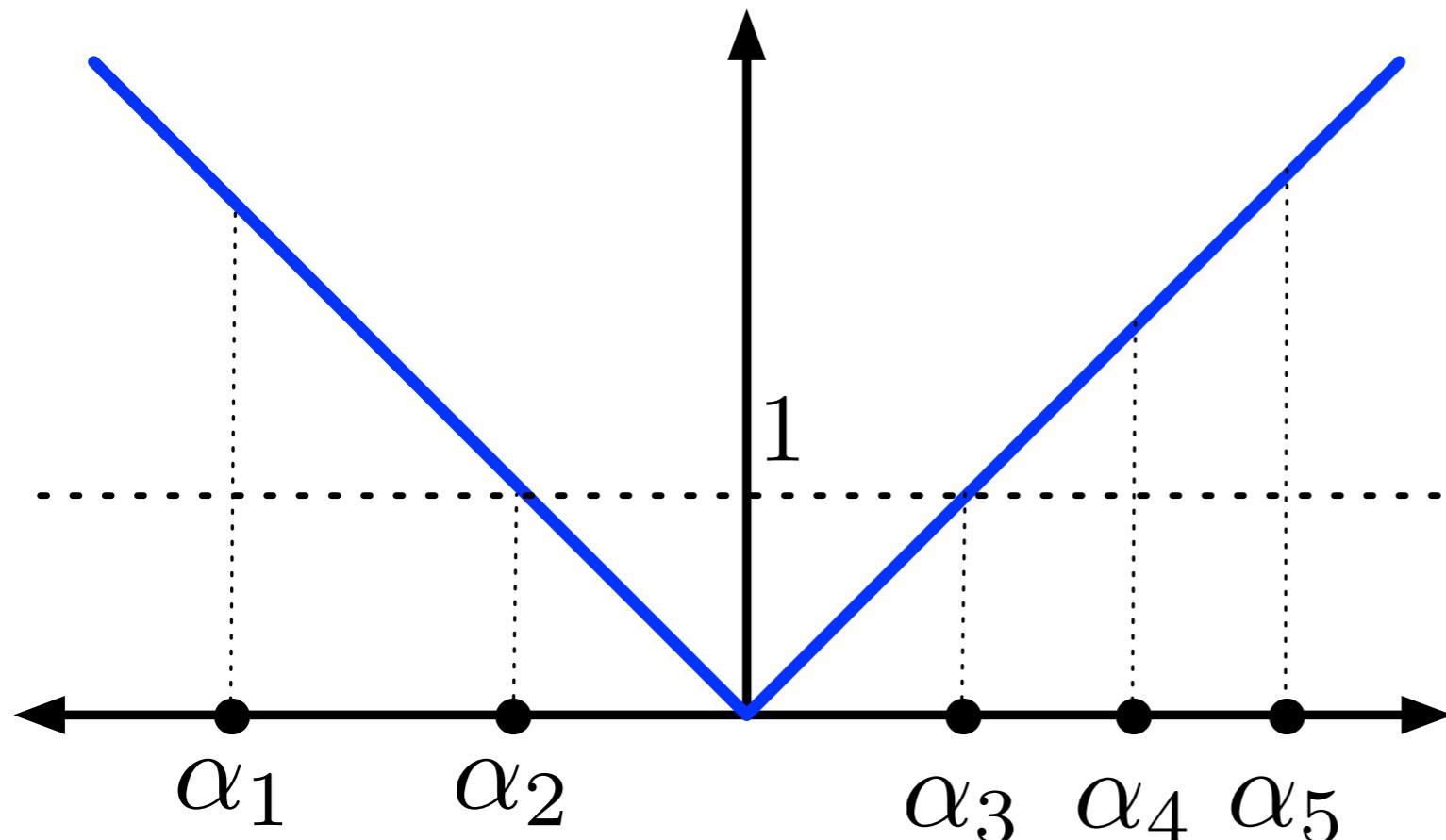
$$\begin{aligned} & \min && |x| \\ & s.t. && \sum_{i=1}^n \lambda_i \alpha_i = x \\ & && \sum_{i=1}^n \lambda_i = 1 \\ & && \lambda \in \{0, 1\}^n \end{aligned}$$

Formulating Discrete Alternatives



$$\begin{aligned} & \min && |x| \\ & s.t. && \sum_{i=1}^n \lambda_i \alpha_i = x \\ & && \sum_{i=1}^n \lambda_i = 1 \\ & && \lambda \in \{0, 1\}^n \\ & && \text{IP}_{\text{opt}} = 1, \text{LP}_{\text{opt}} = 0 \end{aligned}$$

Formulating Discrete Alternatives

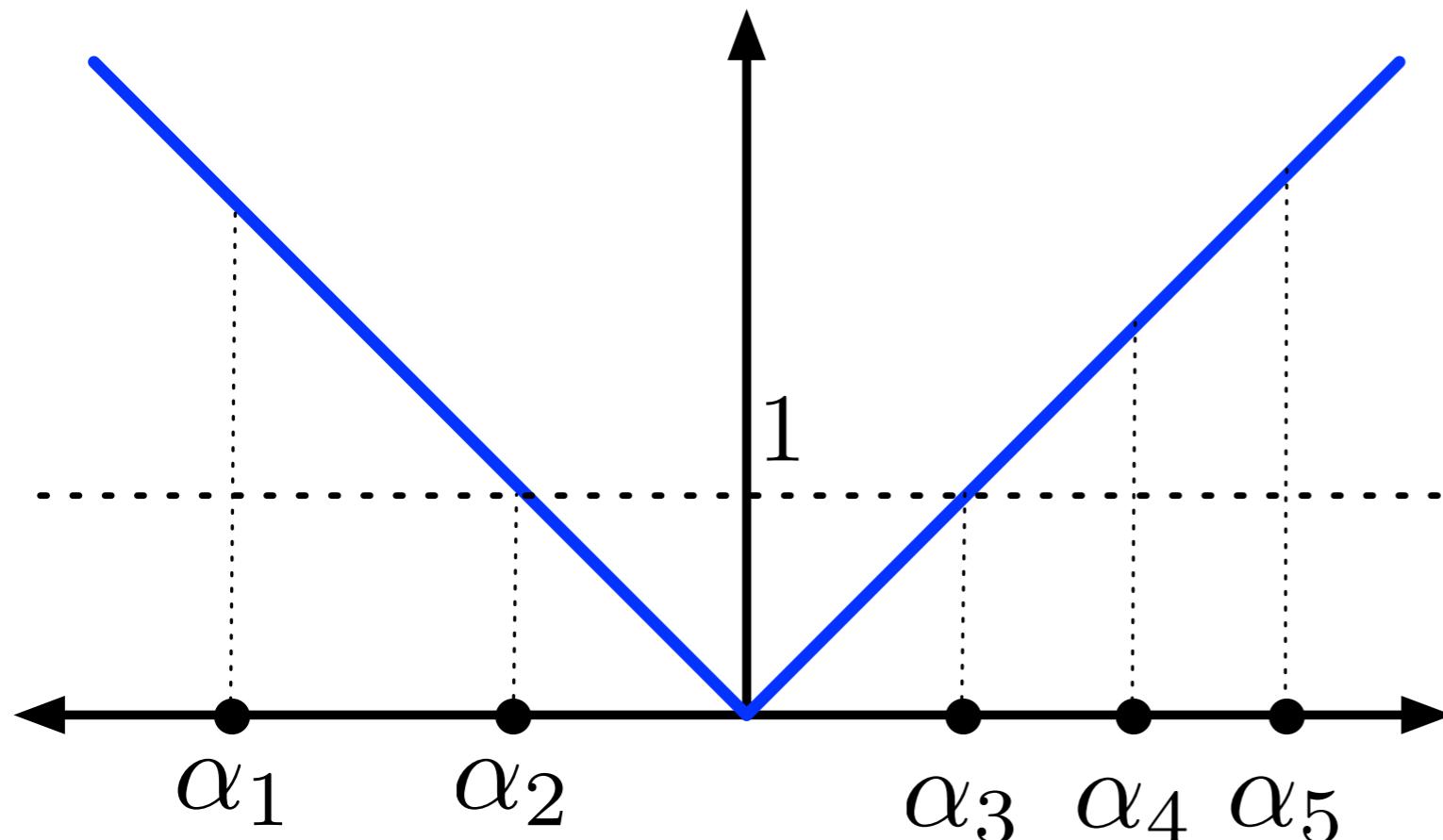


Solve by **binary** Branch-and-Bound:

$$\begin{aligned}
 & \min && |x| \\
 & s.t. && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

$$\text{IP}_{\text{opt}} = 1, \text{LP}_{\text{opt}} = 0$$

Formulating Discrete Alternatives



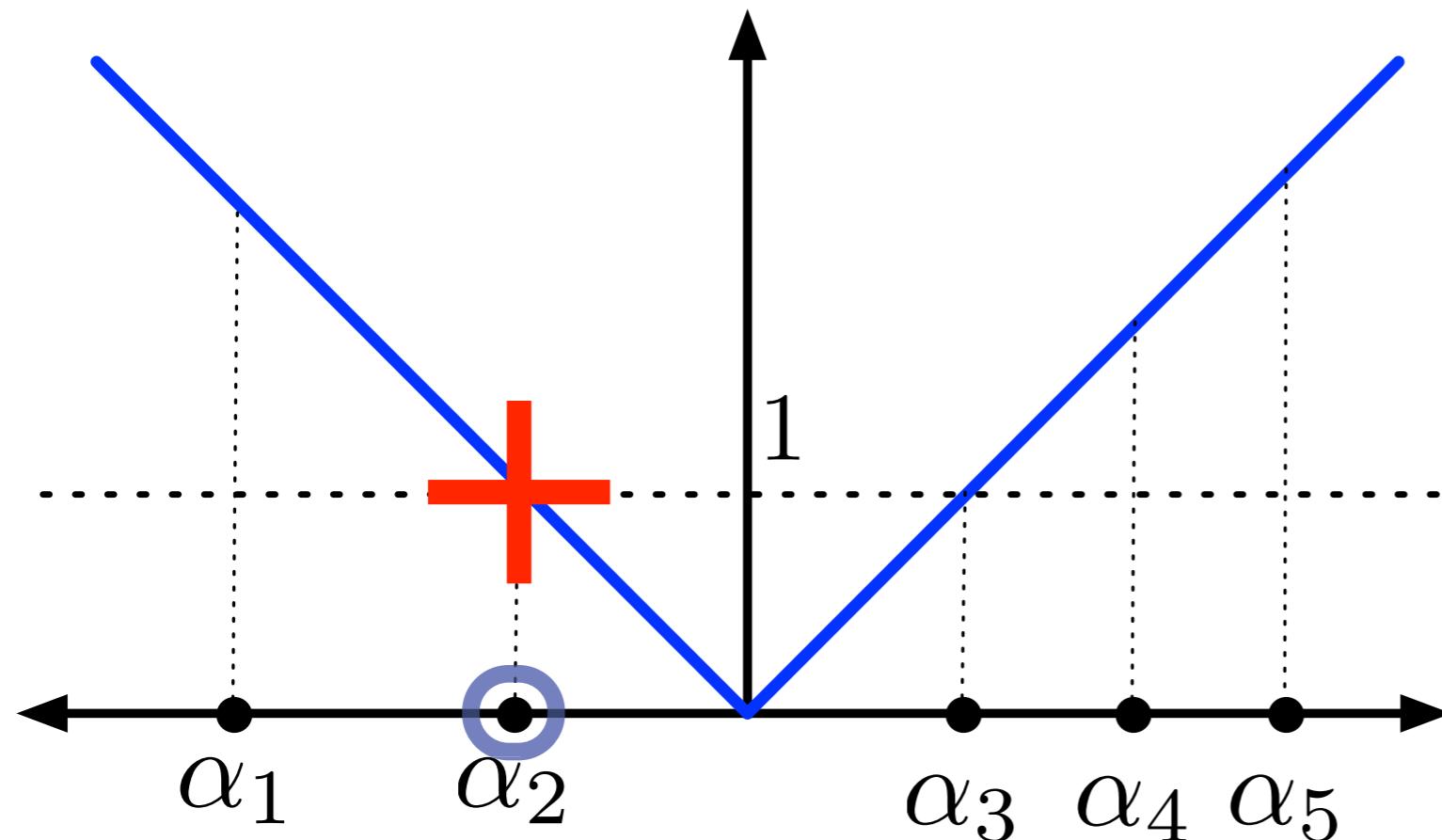
Solve by **binary** Branch-and-Bound:

Branch on λ_2

$$\begin{aligned}
 & \min && |x| \\
 & \text{s.t.} && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

$$\text{IP}_{\text{opt}} = 1, \text{LP}_{\text{opt}} = 0$$

Formulating Discrete Alternatives

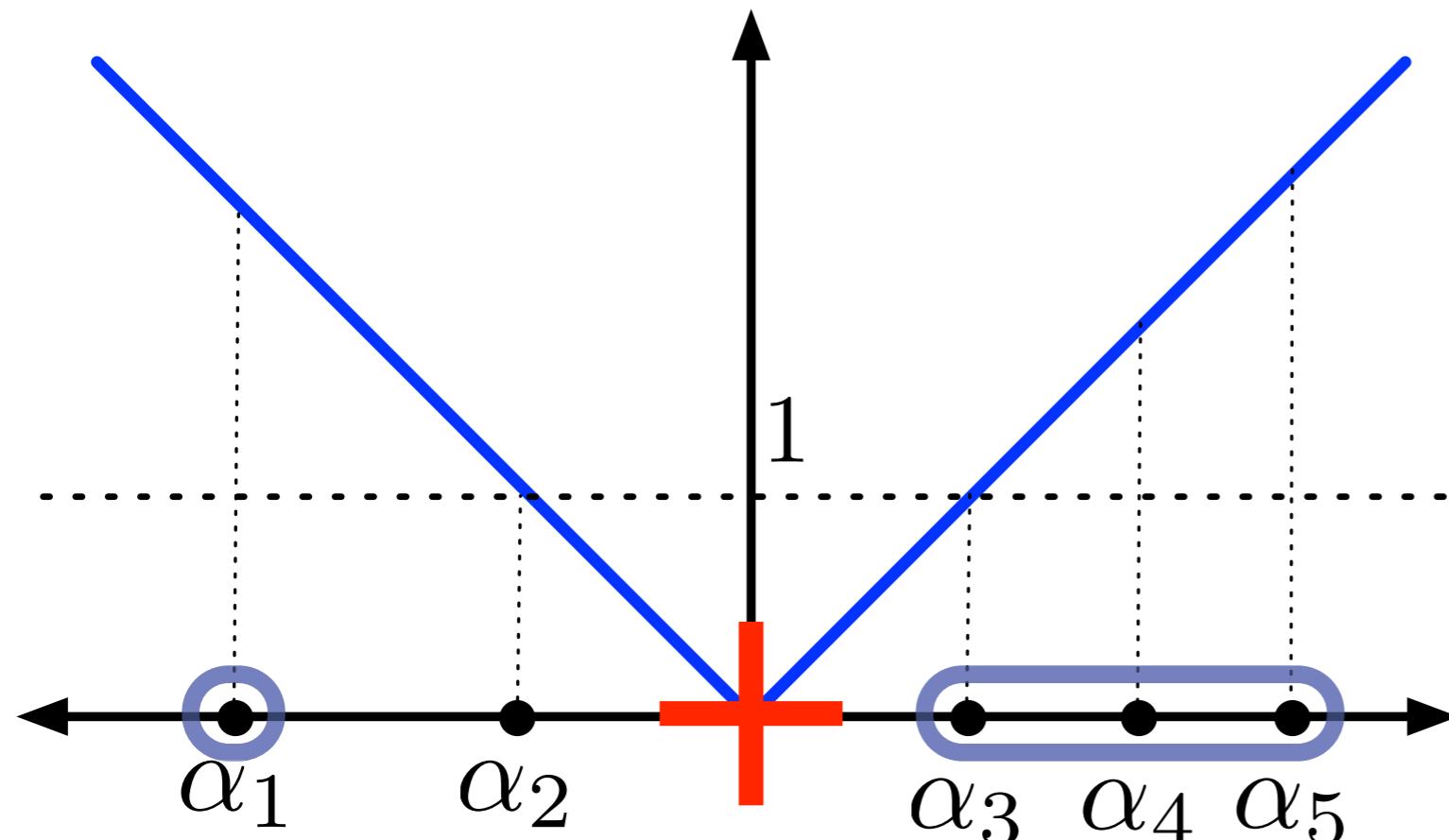


$$\begin{aligned}
 & \min && |x| \\
 & \text{s.t.} && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

Solve by **binary** Branch-and-Bound:

Branch on λ_2 • $\lambda_2 = 1 \rightarrow$ Feasible with $|x| = 1$

Formulating Discrete Alternatives



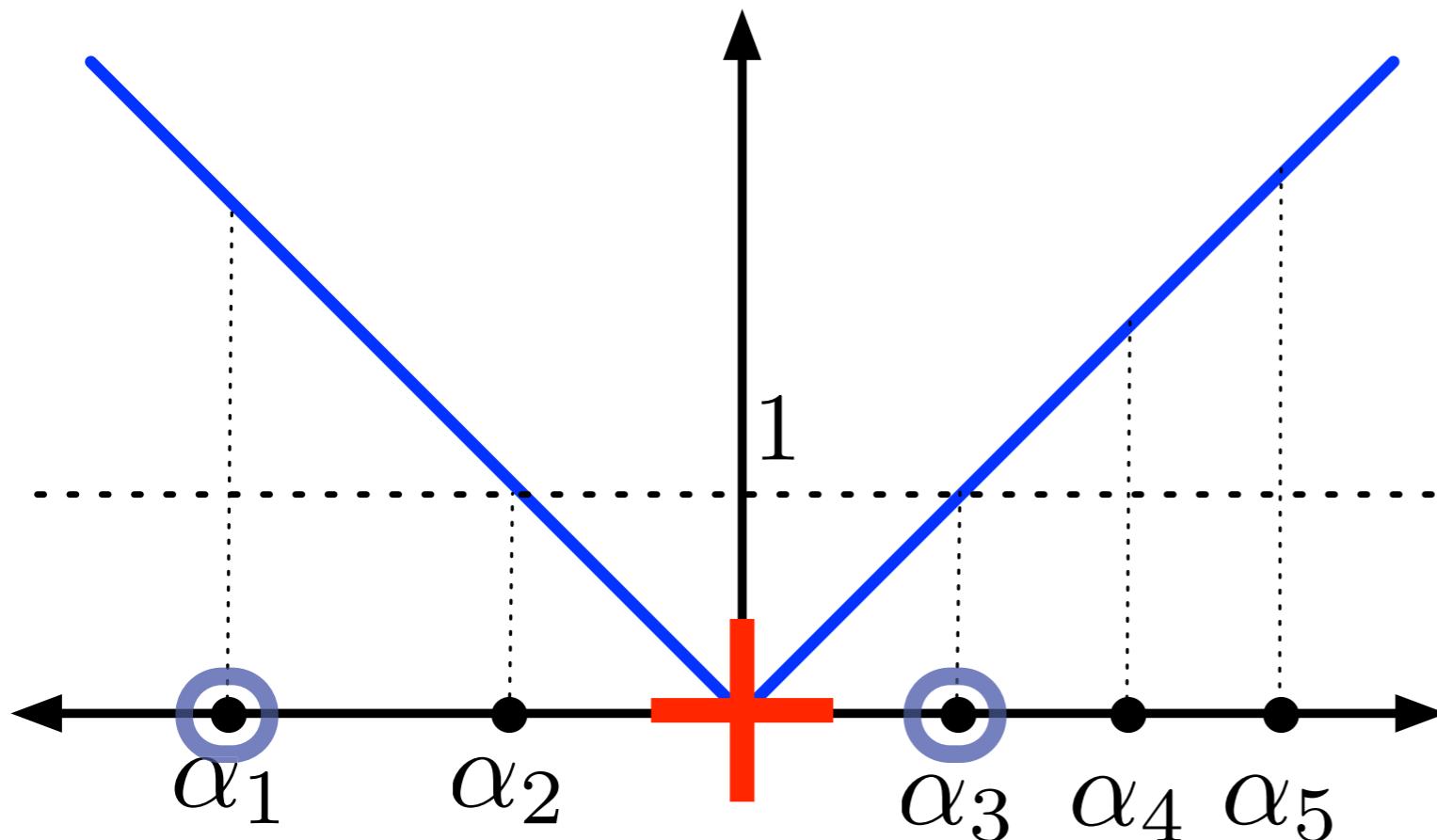
$$\begin{aligned}
 & \min && |x| \\
 & \text{s.t.} && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

Solve by **binary** Branch-and-Bound:

Branch on λ_2

- $\lambda_2 = 1 \rightarrow$ Feasible with $|x| = 1$
- $\lambda_2 = 0 \rightarrow$ Best Bound = 0

Formulating Discrete Alternatives



$$\begin{aligned}
 & \min && |x| \\
 & \text{s.t.} && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

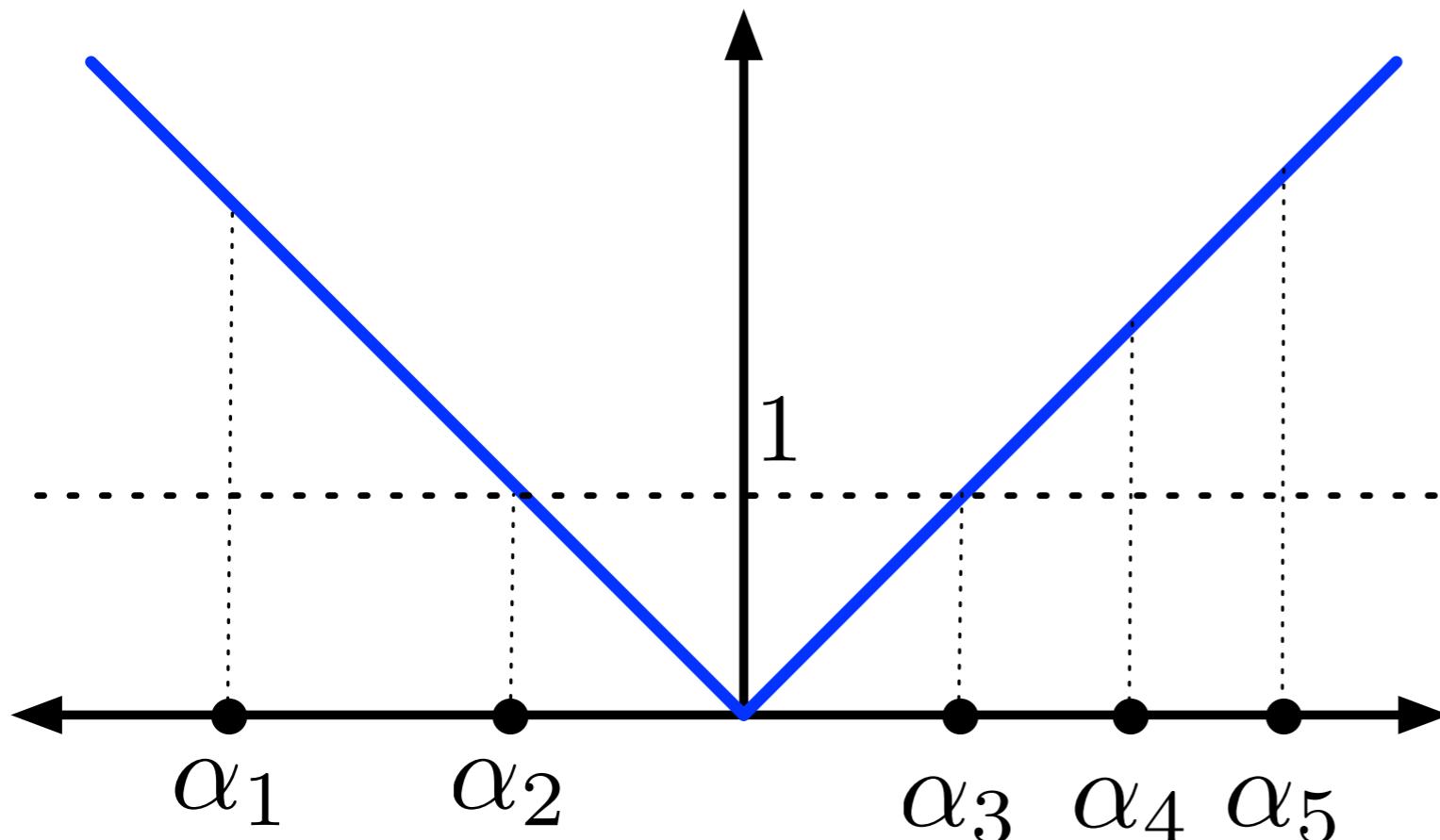
Solve by **binary** Branch-and-Bound: $\text{IP}_{\text{opt}} = 1, \text{LP}_{\text{opt}} = 0$

Branch on λ_2

- $\lambda_2 = 1 \rightarrow$ Feasible with $|x| = 1$
- $\lambda_2 = 0 \rightarrow$ Best Bound = 0

Branch on $\lambda_2, \lambda_4, \lambda_5 \rightarrow$ Best Bound = 0

Formulating Discrete Alternatives



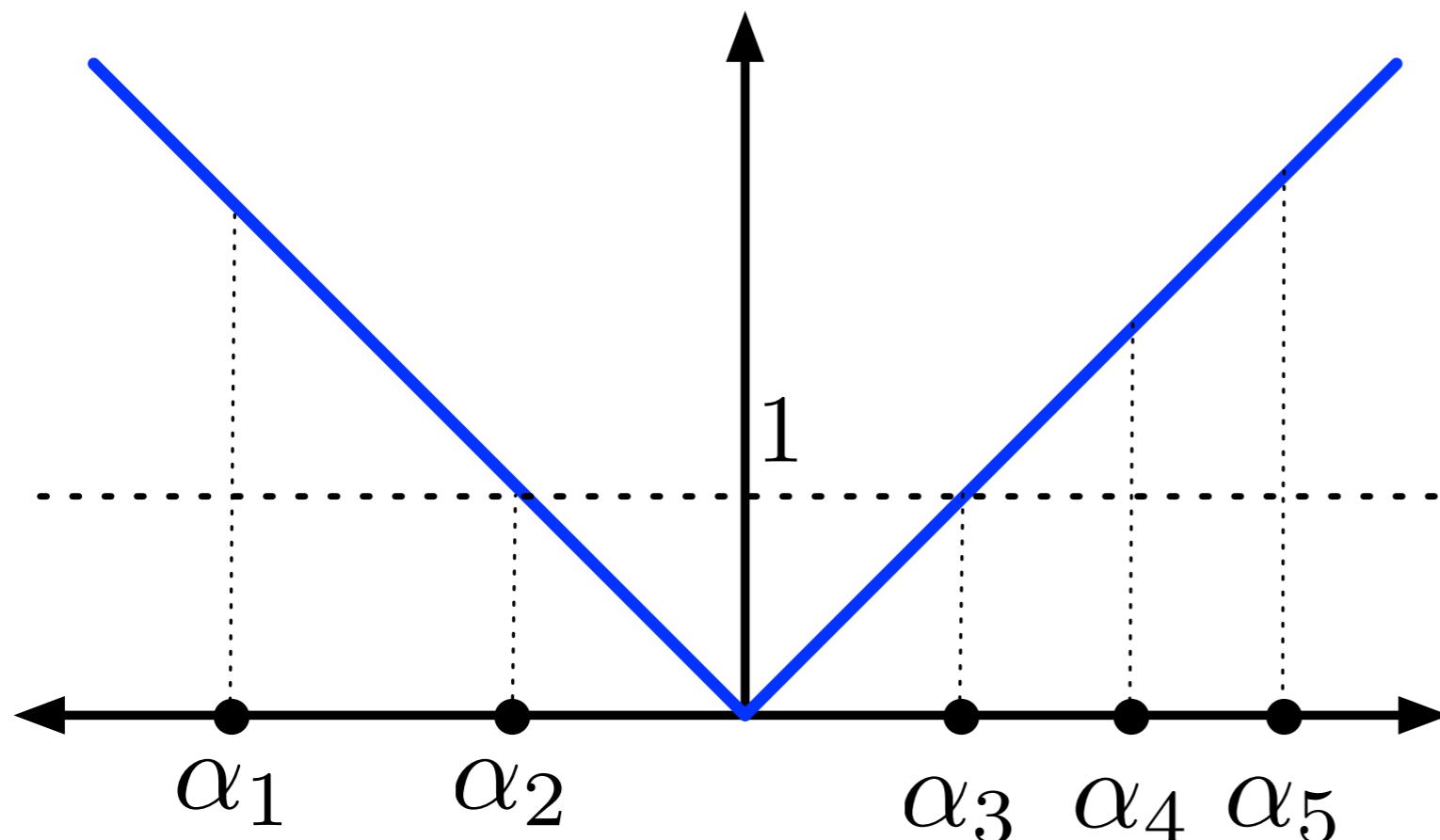
$$\begin{aligned}
 & \min && |x| \\
 & \text{s.t.} && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

Solve by **binary** Branch-and-Bound:

Worst case: $n/2$ branches to solve

$$\text{IP}_{\text{opt}} = 1, \text{LP}_{\text{opt}} = 0$$

Formulating Discrete Alternatives

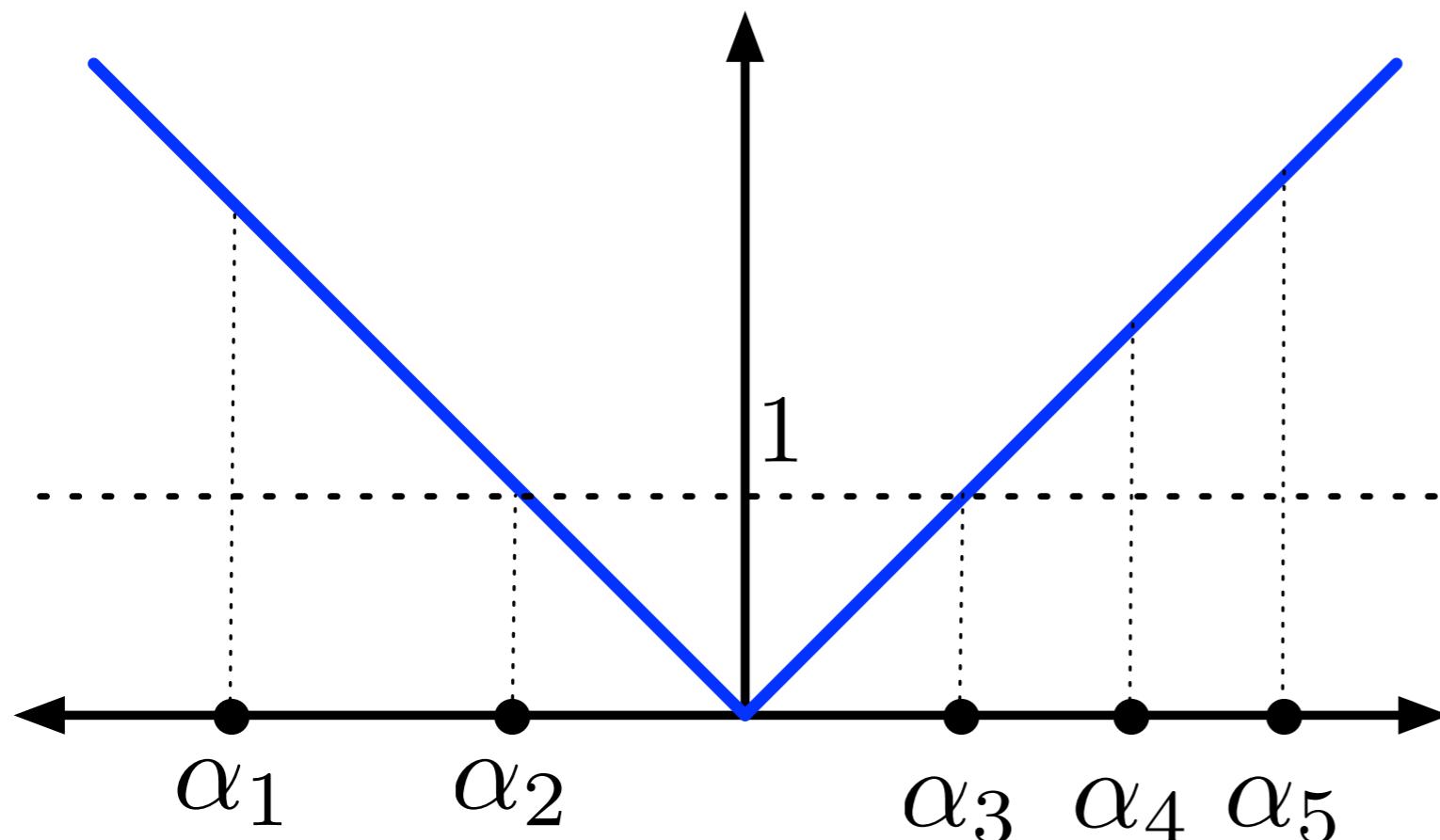


Solve by **constraint** B-and-B:

$$\begin{aligned} & \min && |x| \\ & s.t. && \sum_{i=1}^n \lambda_i \alpha_i = x \\ & && \sum_{i=1}^n \lambda_i = 1 \\ & && \lambda \in \{0, 1\}^n \end{aligned}$$

$$\text{IP}_{\text{opt}} = 1, \text{LP}_{\text{opt}} = 0$$

Formulating Discrete Alternatives



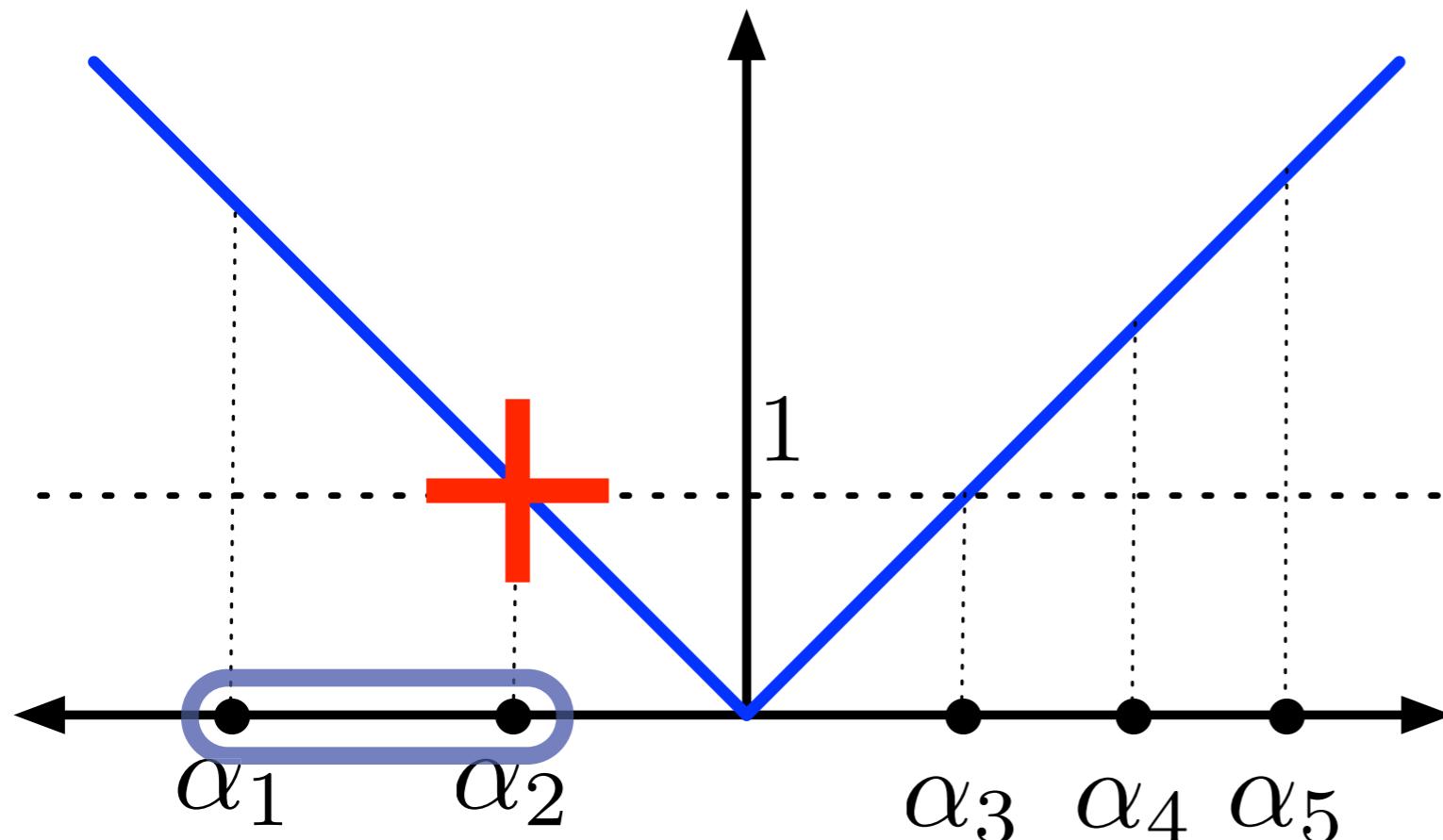
Solve by **constraint** B-and-B:

Branch on $\lambda_1 + \lambda_2$

$$\begin{aligned}
 & \min && |x| \\
 & \text{s.t.} && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

$$\text{IP}_{\text{opt}} = 1, \text{LP}_{\text{opt}} = 0$$

Formulating Discrete Alternatives



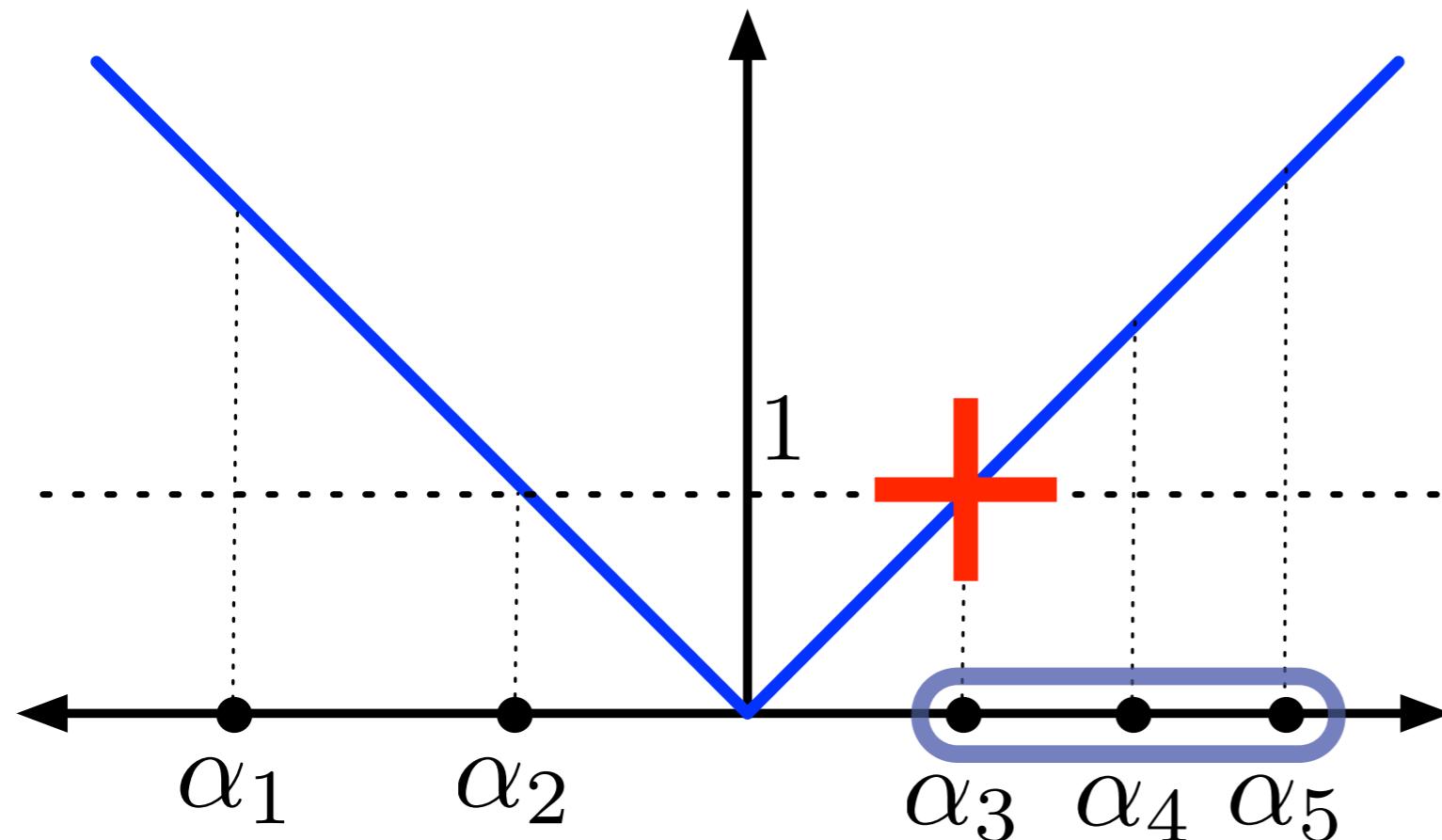
Solve by **constraint** B-and-B:

Branch on $\lambda_1 + \lambda_2$ $\bullet \quad \lambda_1 + \lambda_2 = 1 \rightarrow \text{Feasible with } |x| = 1$

$$\begin{aligned}
 & \min && |x| \\
 & \text{s.t.} && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

$$\text{IP}_{\text{opt}} = 1, \text{LP}_{\text{opt}} = 0$$

Formulating Discrete Alternatives



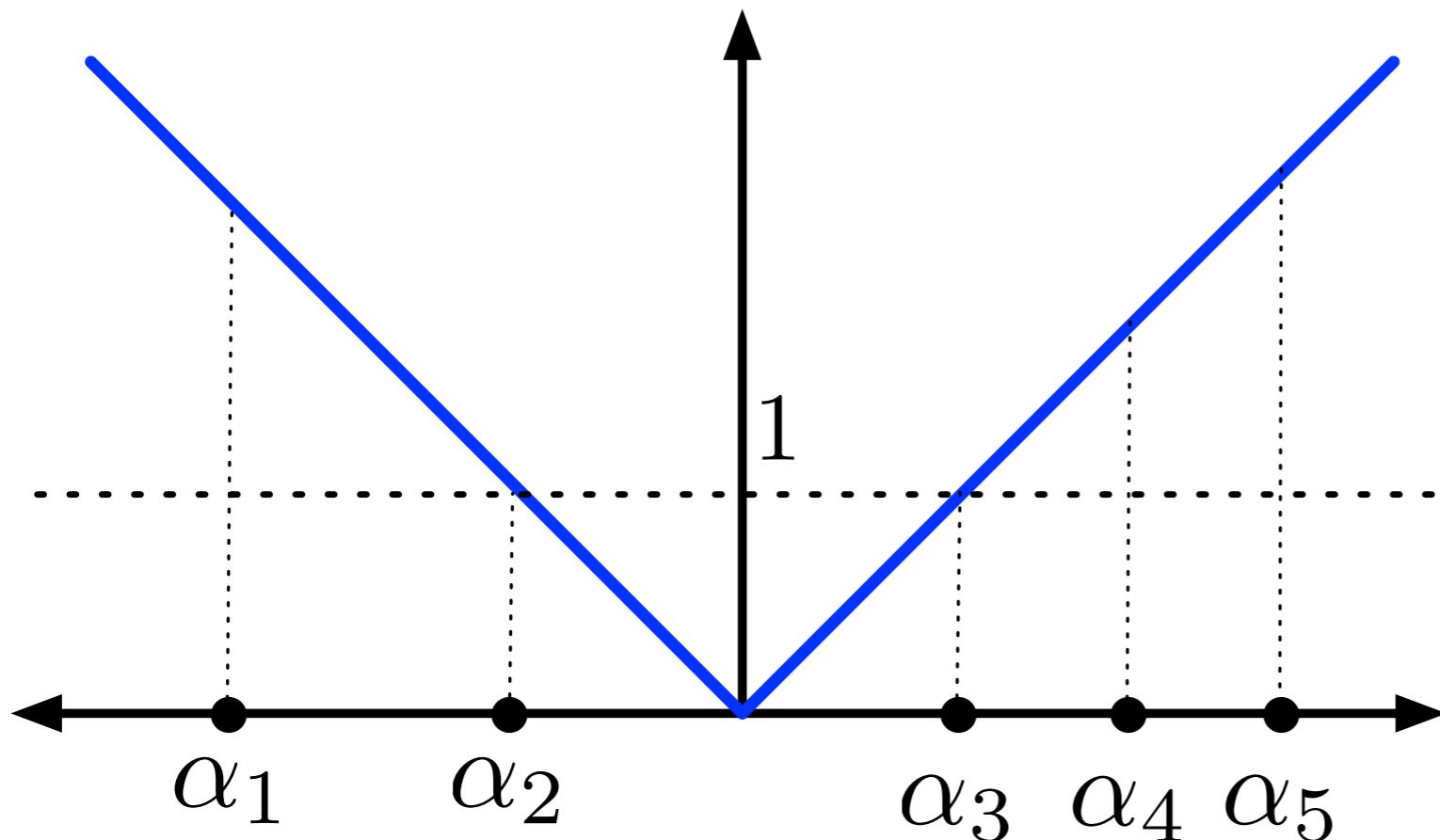
$$\begin{aligned}
 & \min && |x| \\
 & \text{s.t.} && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

Solve by **constraint** B-and-B:

Branch on $\lambda_1 + \lambda_2$

- $\lambda_1 + \lambda_2 = 1 \rightarrow$ Feasible with $|x| = 1$
- $\lambda_1 + \lambda_2 = 0 \rightarrow$ Feasible with $|x| = 1$

Formulating Discrete Alternatives



$$\begin{aligned}
 & \min && |x| \\
 & \text{s.t.} && \\
 & && \sum_{i=1}^n \lambda_i \alpha_i = x \\
 & && \sum_{i=1}^n \lambda_i = 1 \\
 & && \lambda \in \{0, 1\}^n
 \end{aligned}$$

Solve by **constraint** B-and-B:

Branch on $\lambda_1 + \lambda_2$

- $\lambda_1 + \lambda_2 = 1 \rightarrow$ Feasible with $|x| = 1$
- $\lambda_1 + \lambda_2 = 0 \rightarrow$ Feasible with $|x| = 1$

Never more than one branch (2 nodes).

Constraint Branching is the Solution?

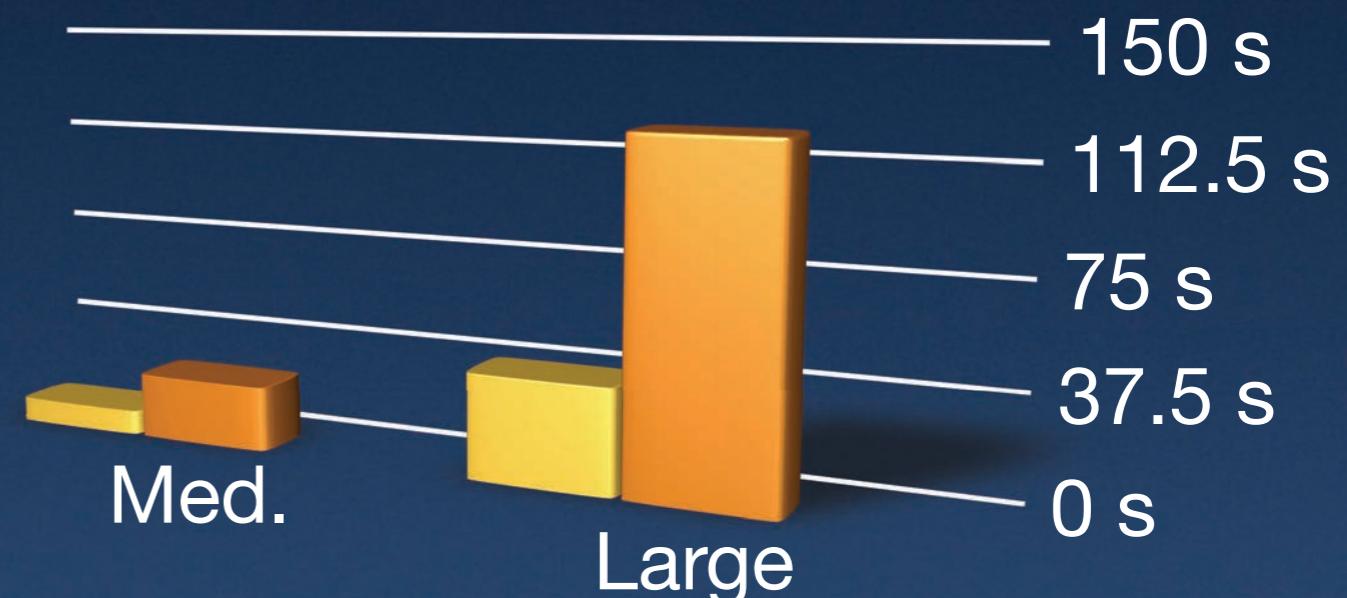
- Ryan and Foster, 1981.
- Discrete Alternatives: SOS1 branching of Beale and Tomlin 1970. Also SOS2 (B. and T, 70) and piecewise linear functions (Tomlin 1981).
- SOS1: $\sum_{i=1}^t \lambda_i = 1$ or $\sum_{i=1}^t \lambda_i = 0$
 $\lambda_i = 0 \quad \forall i > t$ or $\lambda_i = 0 \quad \forall i \leq t$
- Problem: Need to re-implement advanced branching selection (e.g. pseudocost).

Binary v/s Specialized Branching

Weak Integer SOS2 Branching Mystery Integer

Binary v/s Specialized Branching

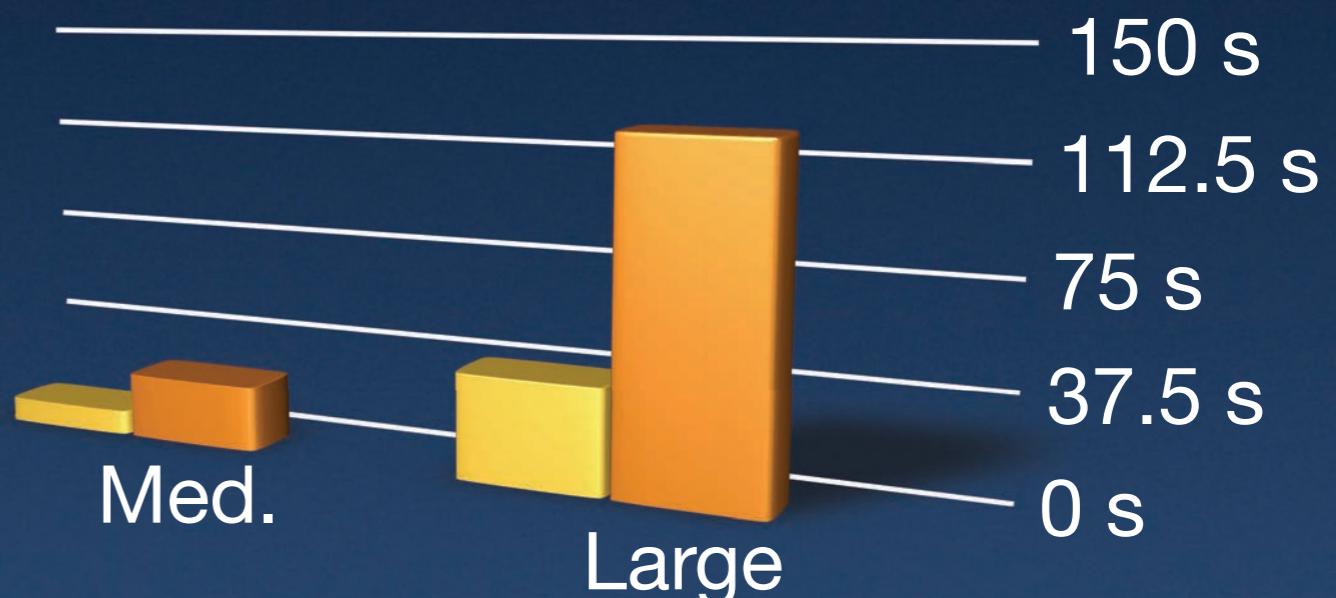
- CPLEX 9: Basic SOS2 branching implementation
(graph from Nemhauser, Keha and V. '08)



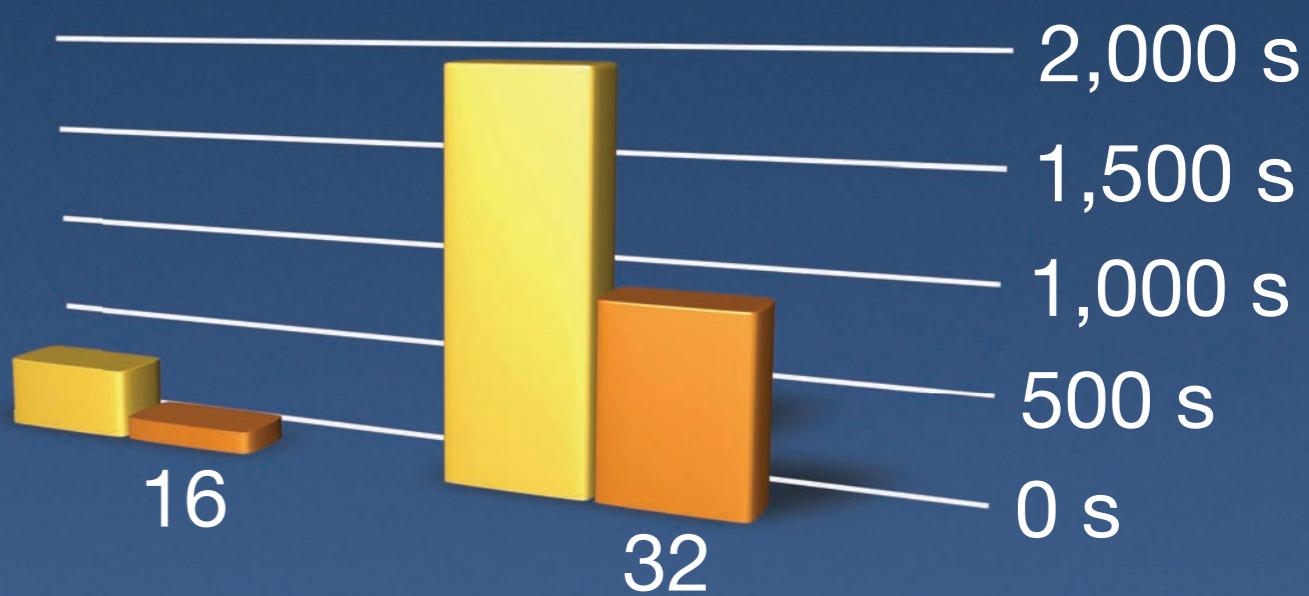
■ Weak Integer ■ SOS2 Branching ■ Mystery Integer

Binary v/s Specialized Branching

- CPLEX 9: Basic SOS2 branching implementation
(graph from Nemhauser, Keha and V. '08)



- CPLEX 11: Improved SOS2 branching implementation
(graph from Nemhauser, Ahmed and V. '10)



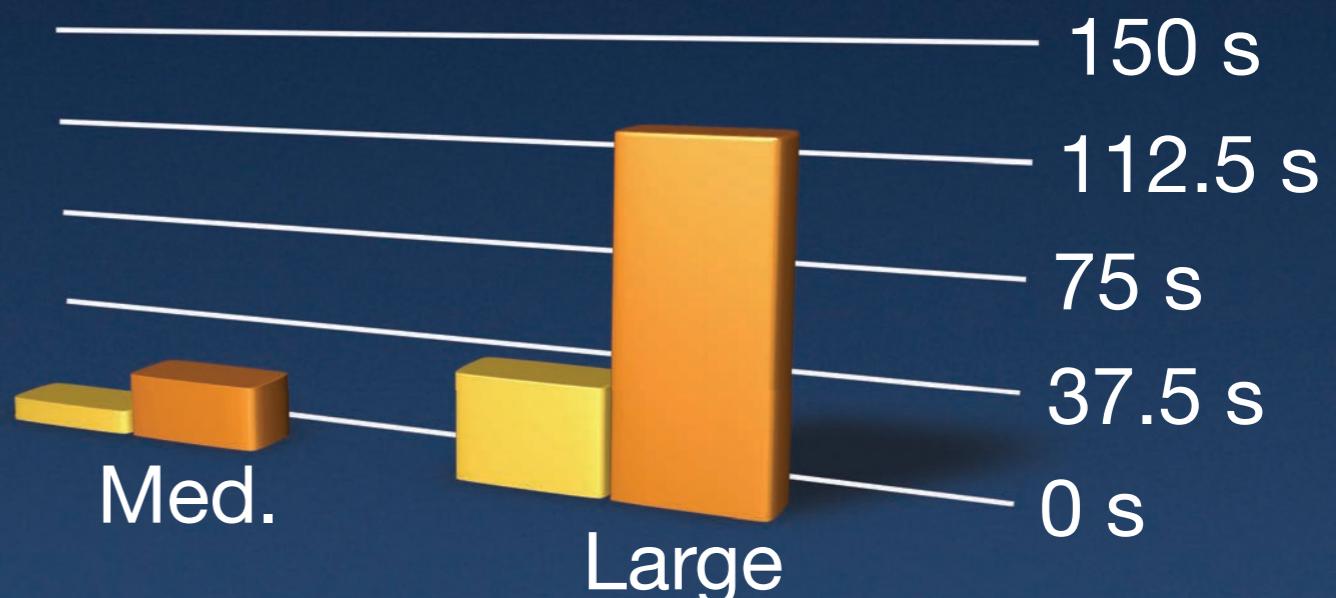
■ Weak Integer

■ SOS2 Branching

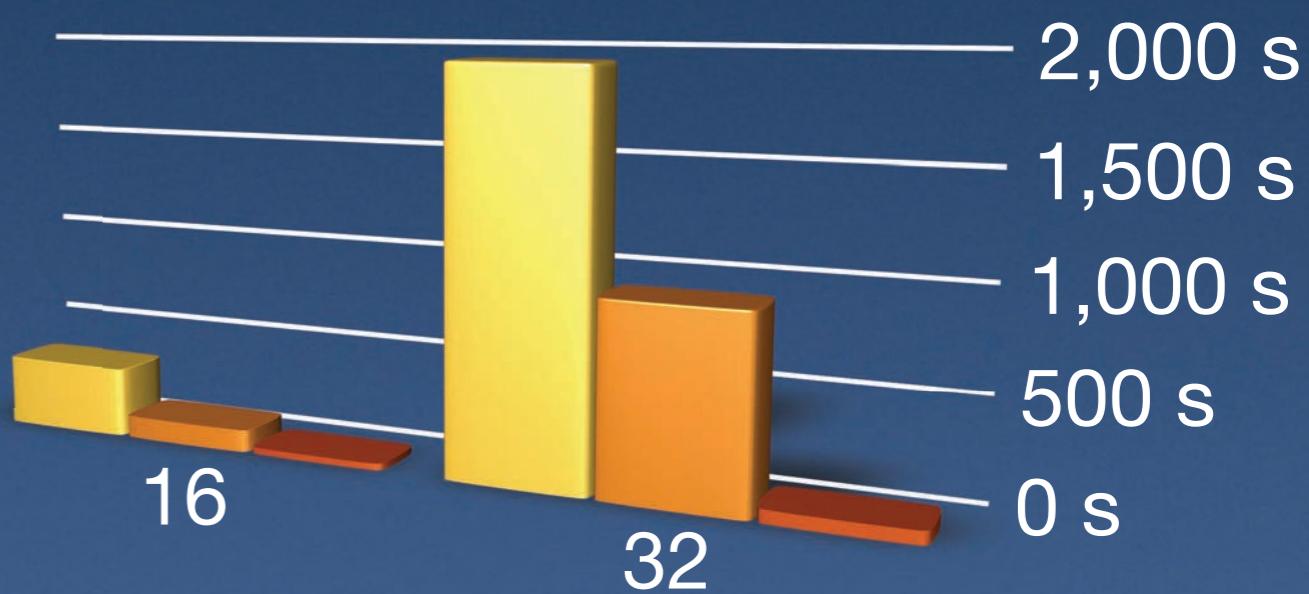
■ Mystery Integer

Binary v/s Specialized Branching

- CPLEX 9: Basic SOS2 branching implementation
(graph from Nemhauser, Keha and V. '08)



- CPLEX 11: Improved SOS2 branching implementation
(graph from Nemhauser, Ahmed and V. '10)



■ Weak Integer

■ SOS2 Branching

■ Mystery Integer

Formulation Step 1: Encoding Alternatives

Formulation for Discrete Alternatives

$$\sum_{i=1}^n \lambda_i = 1$$

$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^{\log_2 n}$$

$$\{b^i\}_{i=1}^n = \{0, 1\}^{\log_2 n}$$

- Li and Lu 2009, Adams and Henry 2011, V. and Nemhauser 2008.
- Sommer, TIMS 1972.
- Log = Binary Encoding
- Other choices of $\{b^i\}_{i=1}^n$ lead to standard and incremental formulations

Formulation for Discrete Alternatives

$$\sum_{i=1}^n \lambda_i = 1$$

$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^{\frac{\log_2 n}{m}}$$

$$\{b^i\}_{i=1}^n = \{0, 1\}^{\frac{\log_2 n}{m}}$$

- Li and Lu 2009, Adams and Henry 2011, V. and Nemhauser 2008.
- Sommer, TIMS 1972.
- Log = Binary Encoding
- Other choices of $\{b^i\}_{i=1}^n$ lead to standard and incremental formulations

Unary Encoding

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\lambda = y,$$

$$\sum_{i=1}^8 \lambda_i = 1, \\ \lambda \in \mathbb{R}^8, y \in \{0, 1\}^8$$



$$\lambda_i = y_i$$

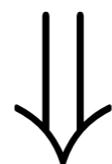
Binary Encoding

$$\left(\begin{array}{ccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array} \right) \lambda = y, \quad \sum_{i=1}^8 \lambda_i = 1, \\ \lambda \in \mathbb{R}^8, y \in \{0, 1\}^3$$

Incremental Encoding

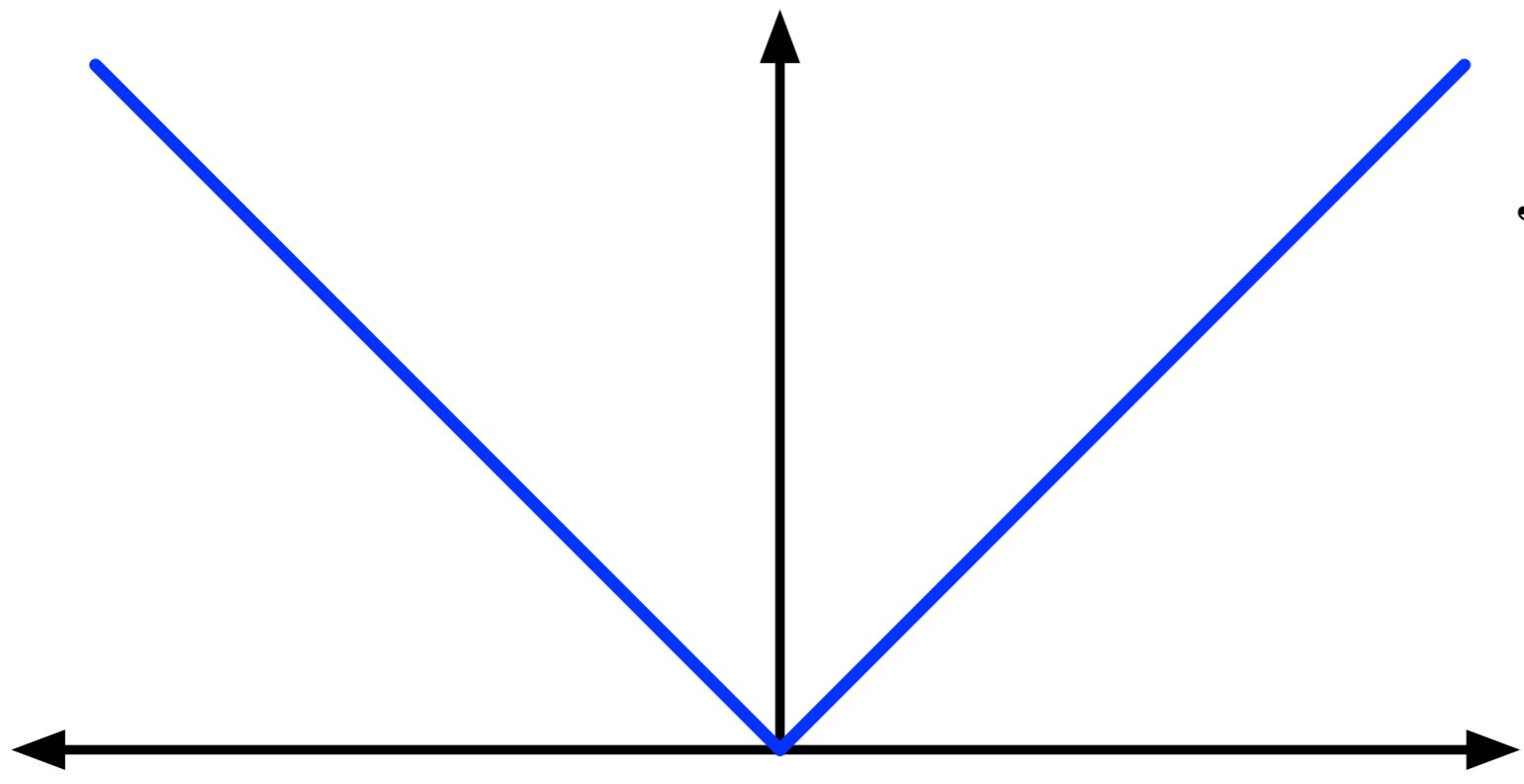
$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\lambda = y, \quad \sum_{i=1}^8 \lambda_i = 1, \\ \lambda \in \mathbb{R}^8, y \in \{0, 1\}^7$$



$$y_1 \geq y_2 \geq \dots \geq y_7$$

Example: Unary Encoding



min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

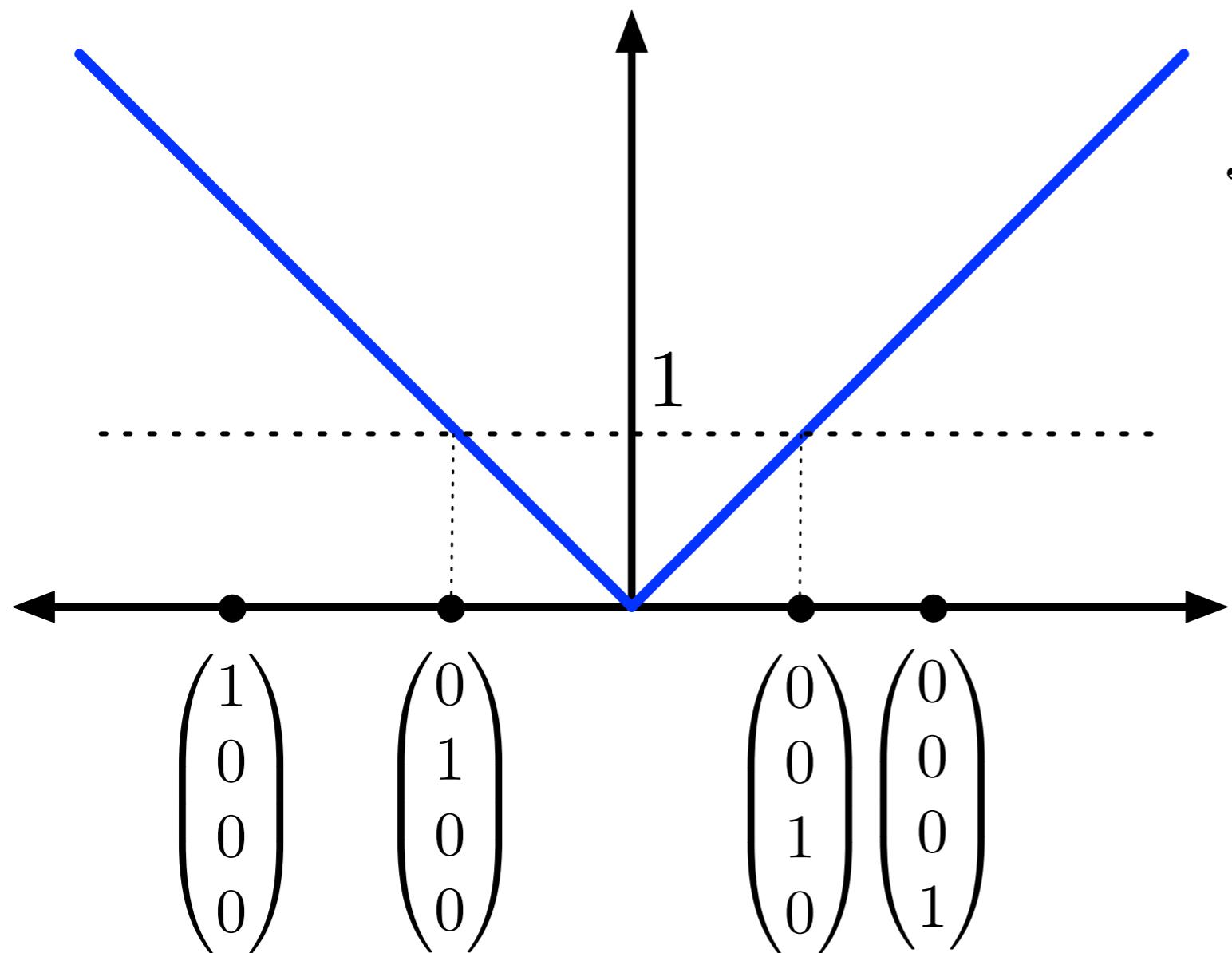
$$\sum_{i=1}^n \lambda_i = 1$$

$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^m$$

Example: Unary Encoding



min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

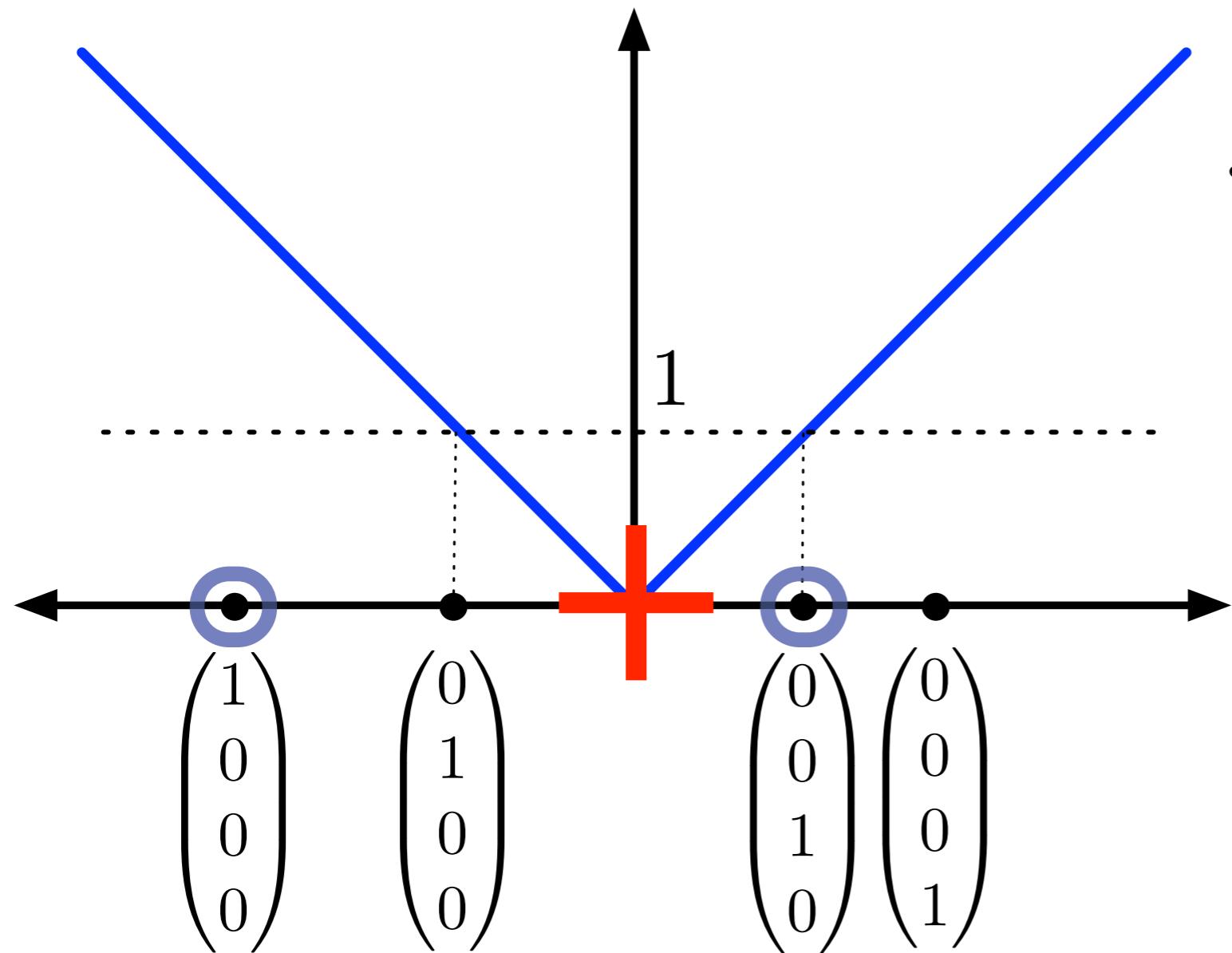
$$\sum_{i=1}^n \lambda_i = 1$$

$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^m$$

Example: Unary Encoding



\min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

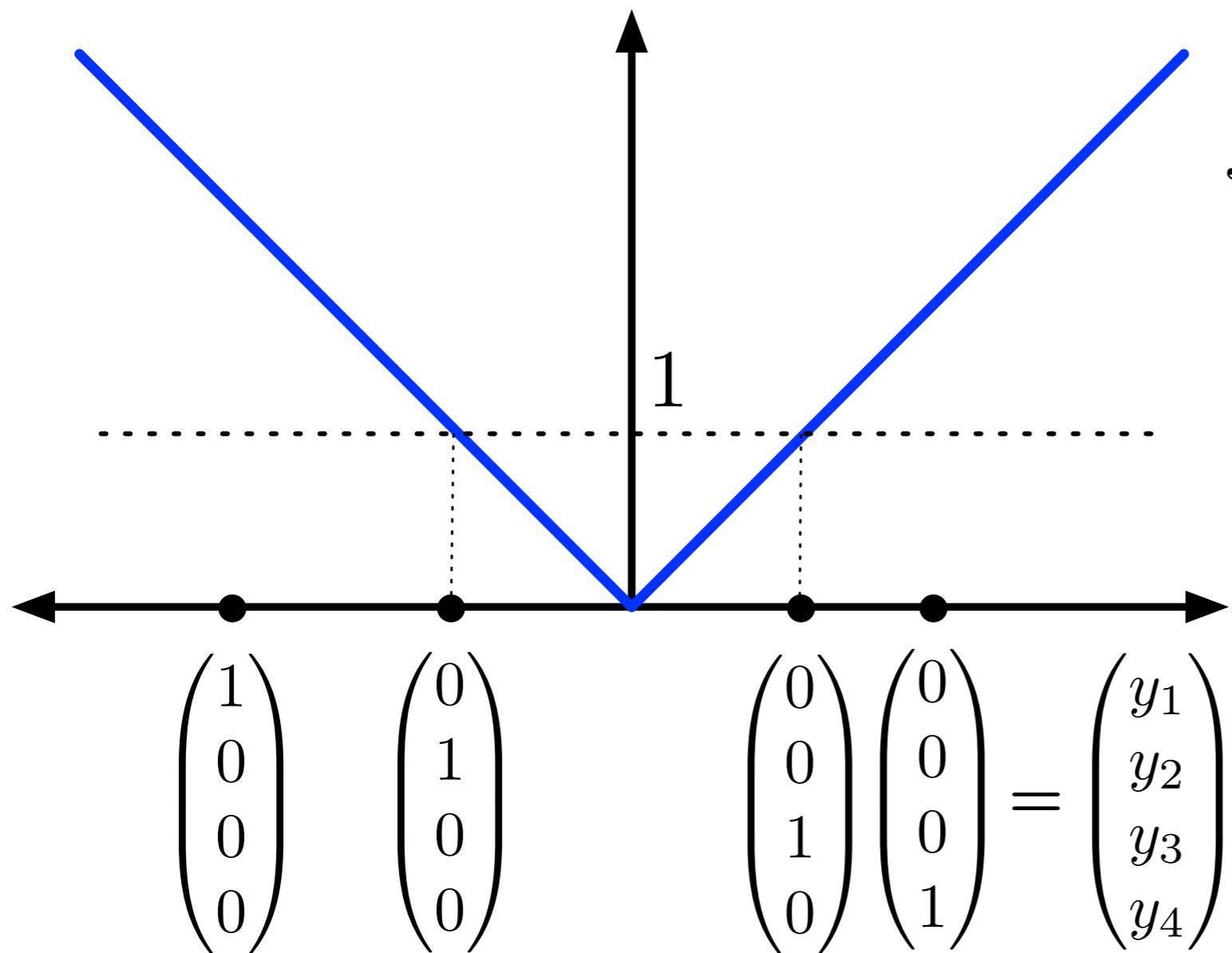
$$\sum_{i=1}^n \lambda_i = 1$$

$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^m$$

Example: Unary Encoding



min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

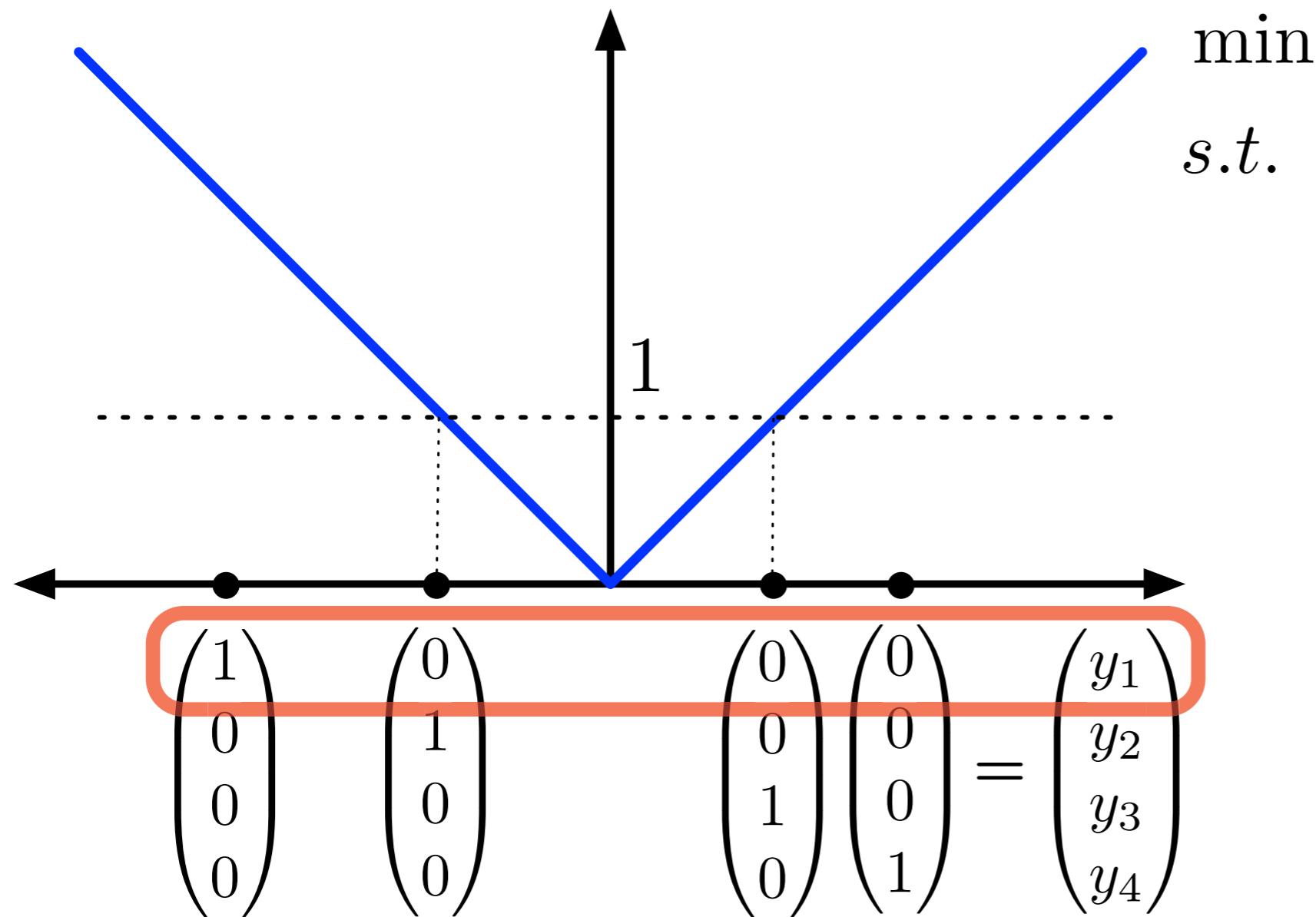
$$\sum_{i=1}^n \lambda_i = 1$$

$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^m$$

Example: Unary Encoding



min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

$$\sum_{i=1}^n \lambda_i = 1$$

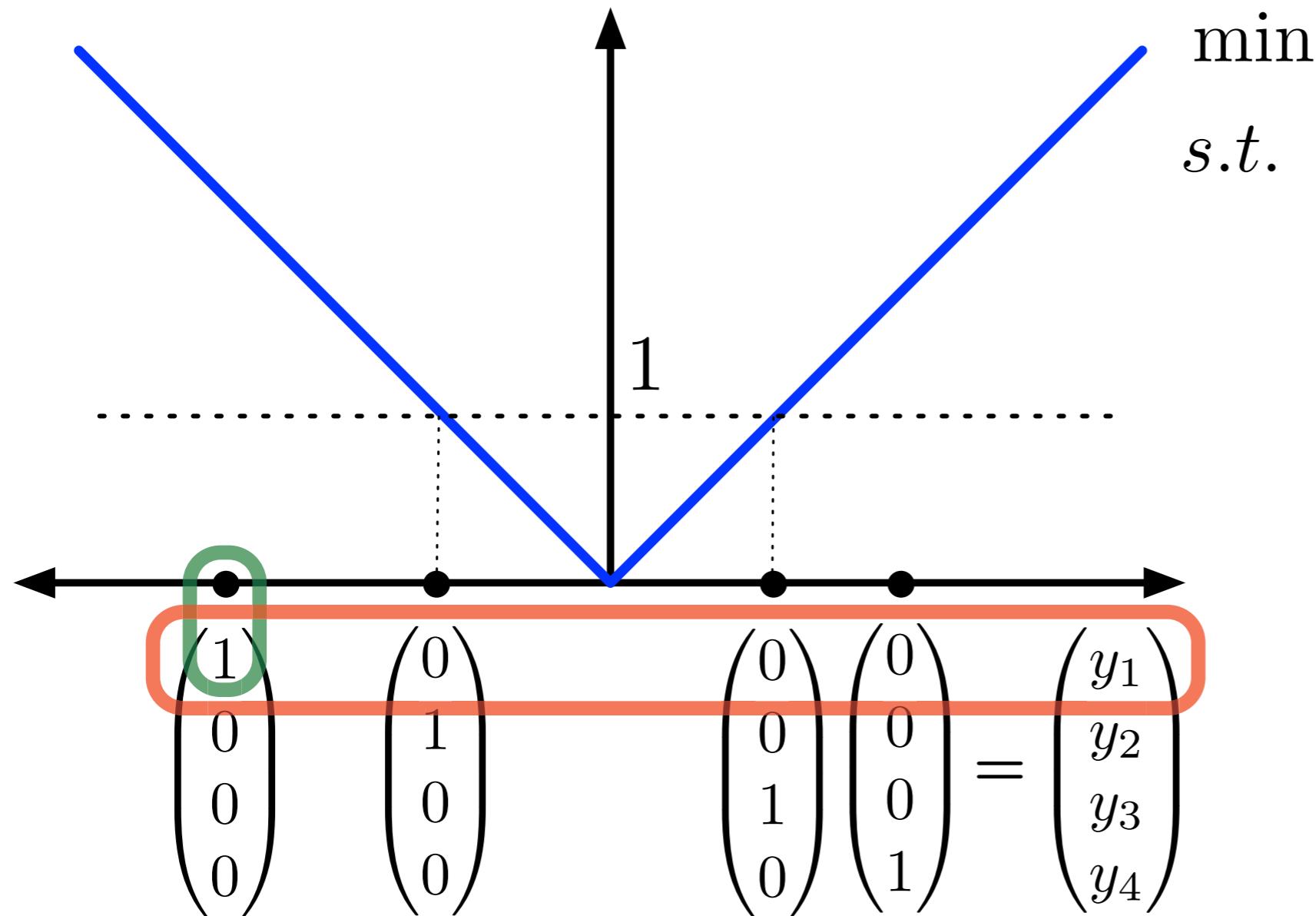
$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^m$$

$$y_1 =$$

Example: Unary Encoding



min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

$$\sum_{i=1}^n \lambda_i = 1$$

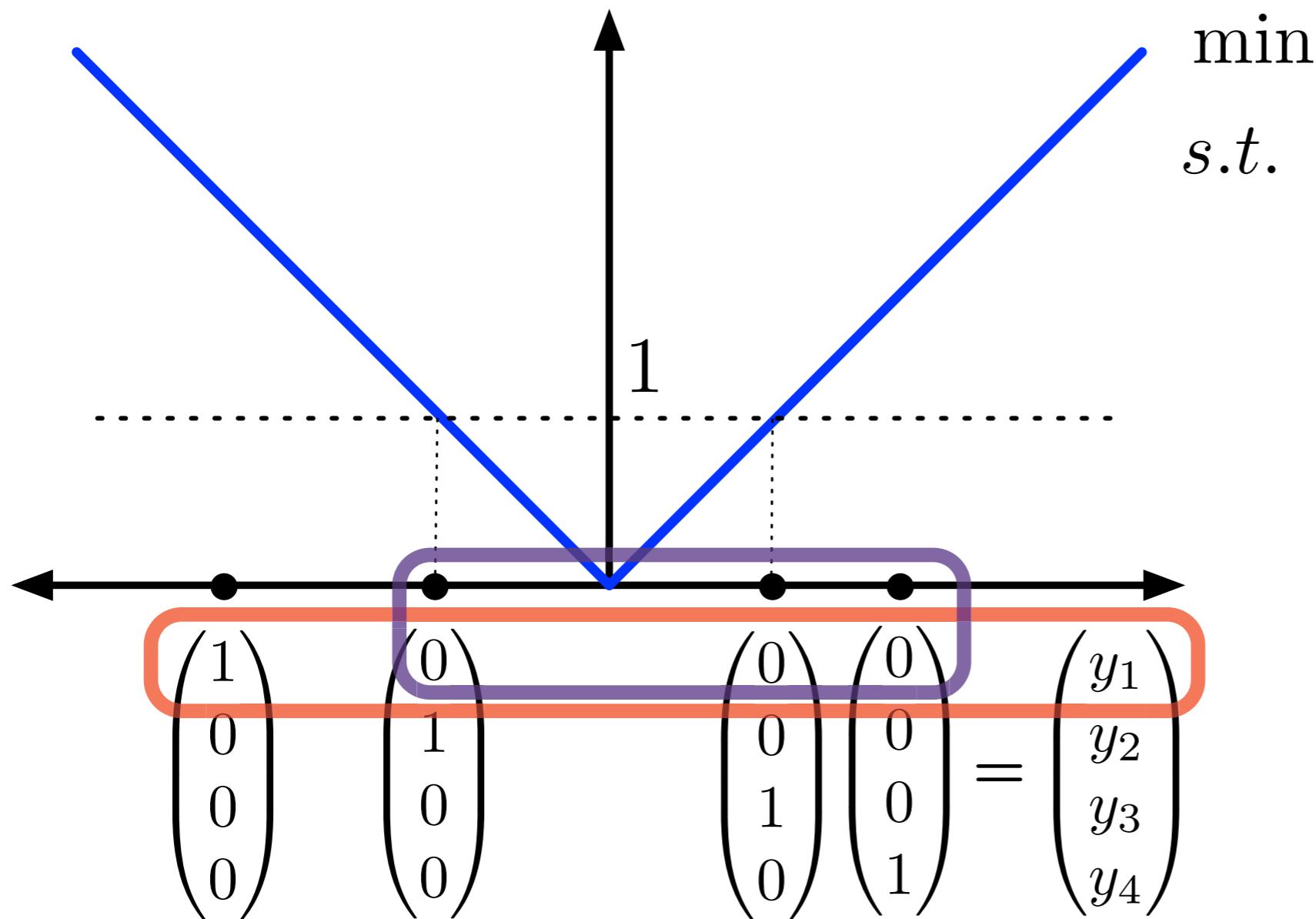
$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^m$$

$$y_1 = 1$$

Example: Unary Encoding



min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

$$\sum_{i=1}^n \lambda_i = 1$$

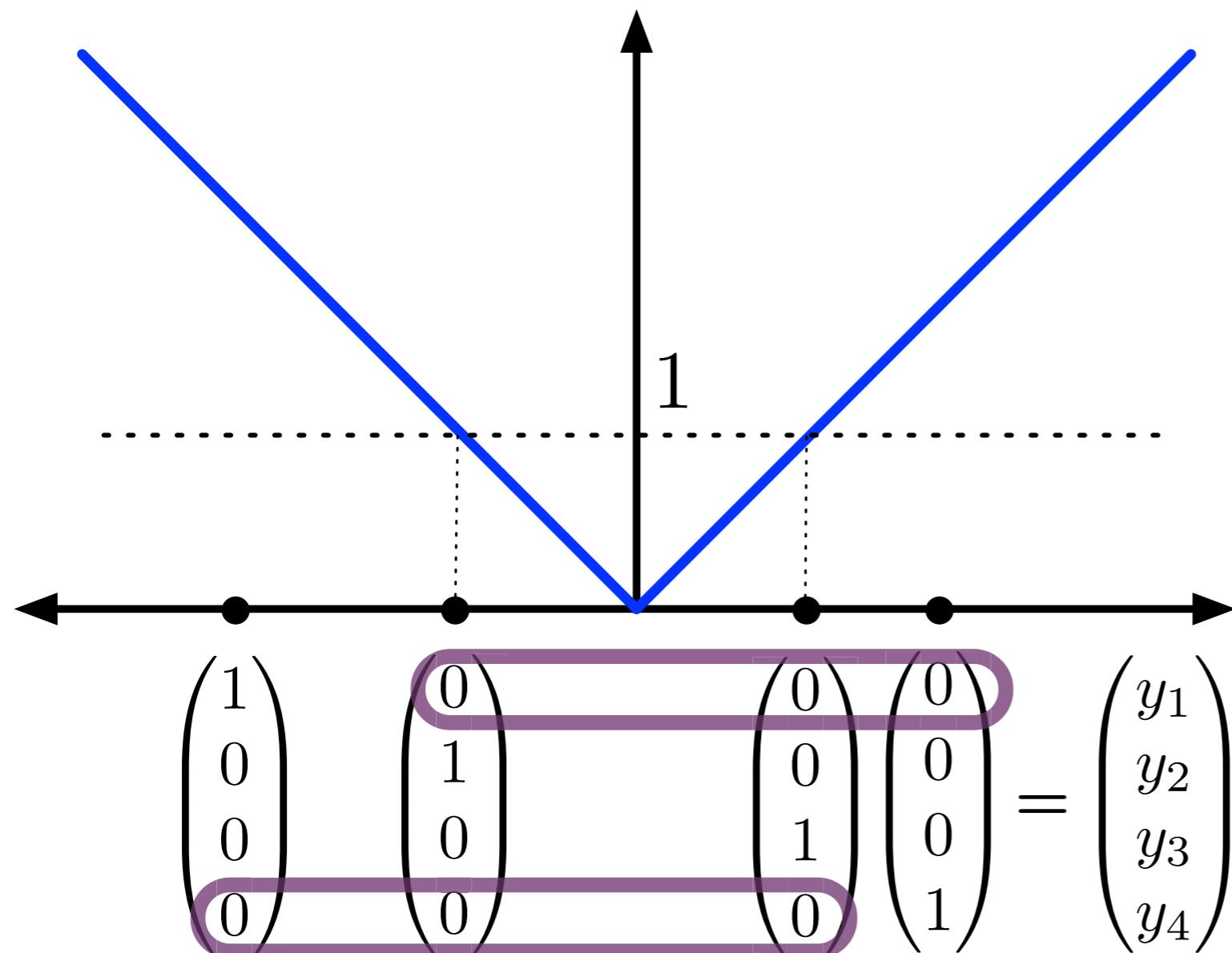
$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^m$$

$$y_1 = 0$$

Example: Unary Encoding



min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

$$\sum_{i=1}^n \lambda_i = 1$$

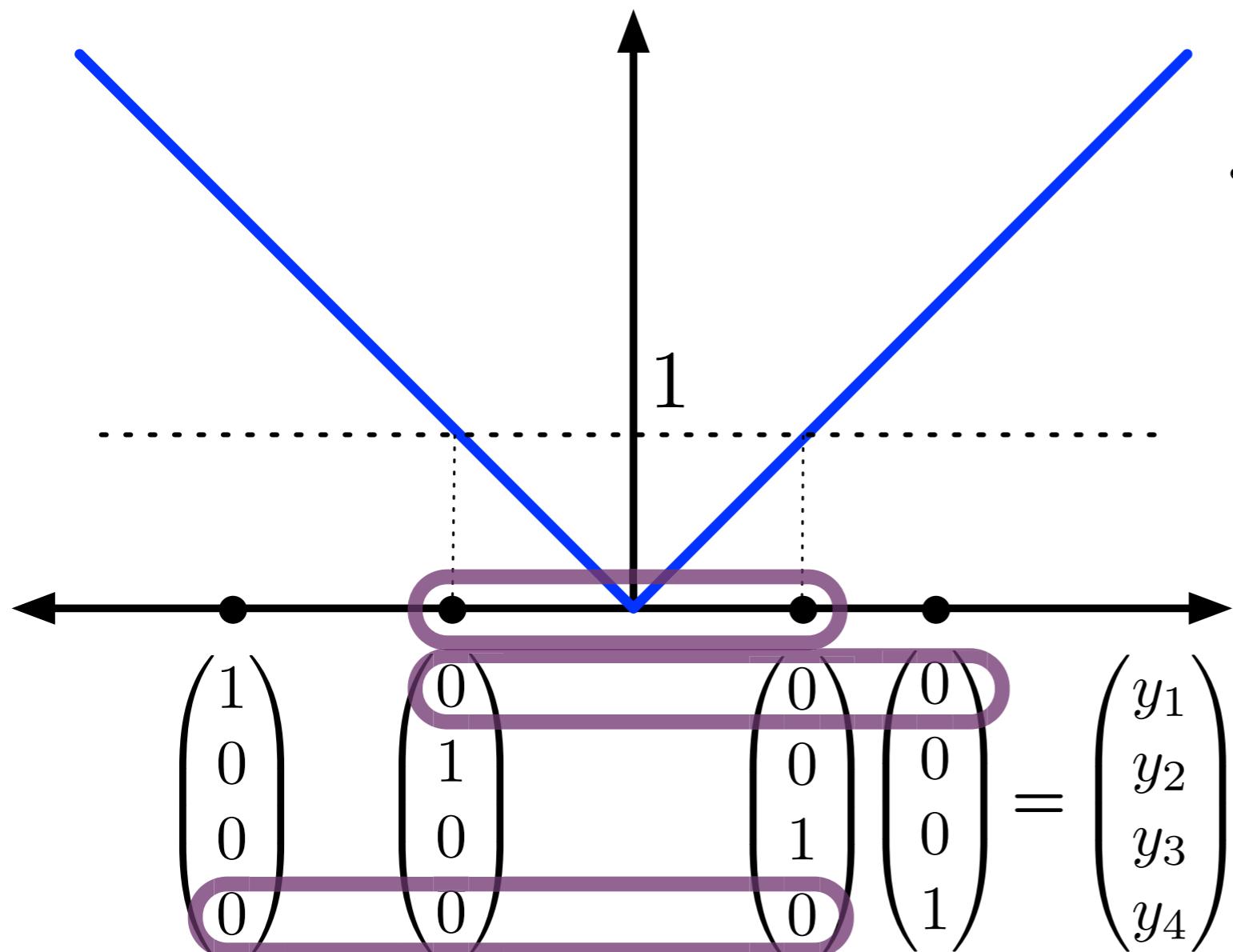
$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^m$$

$$y_1 = y_4 = 0$$

Example: Unary Encoding



min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

$$\sum_{i=1}^n \lambda_i = 1$$

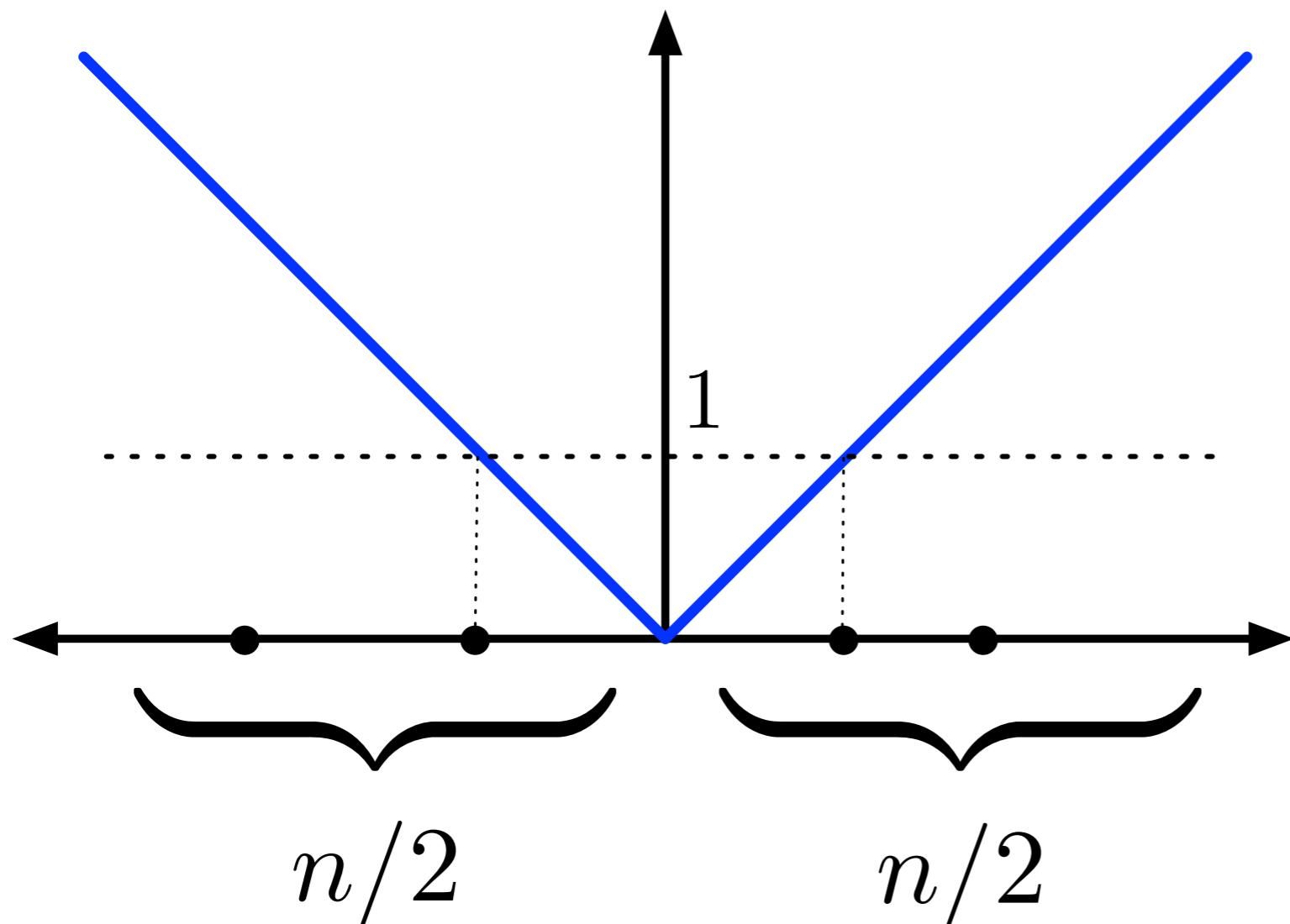
$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

$$y \in \{0, 1\}^m$$

$$y_1 = y_4 = 0$$

Example: Unary Encoding



\min
s.t.

$$\sum_{i=1}^n \lambda_i \alpha_i = x$$

$$\sum_{i=1}^n \lambda_i = 1$$

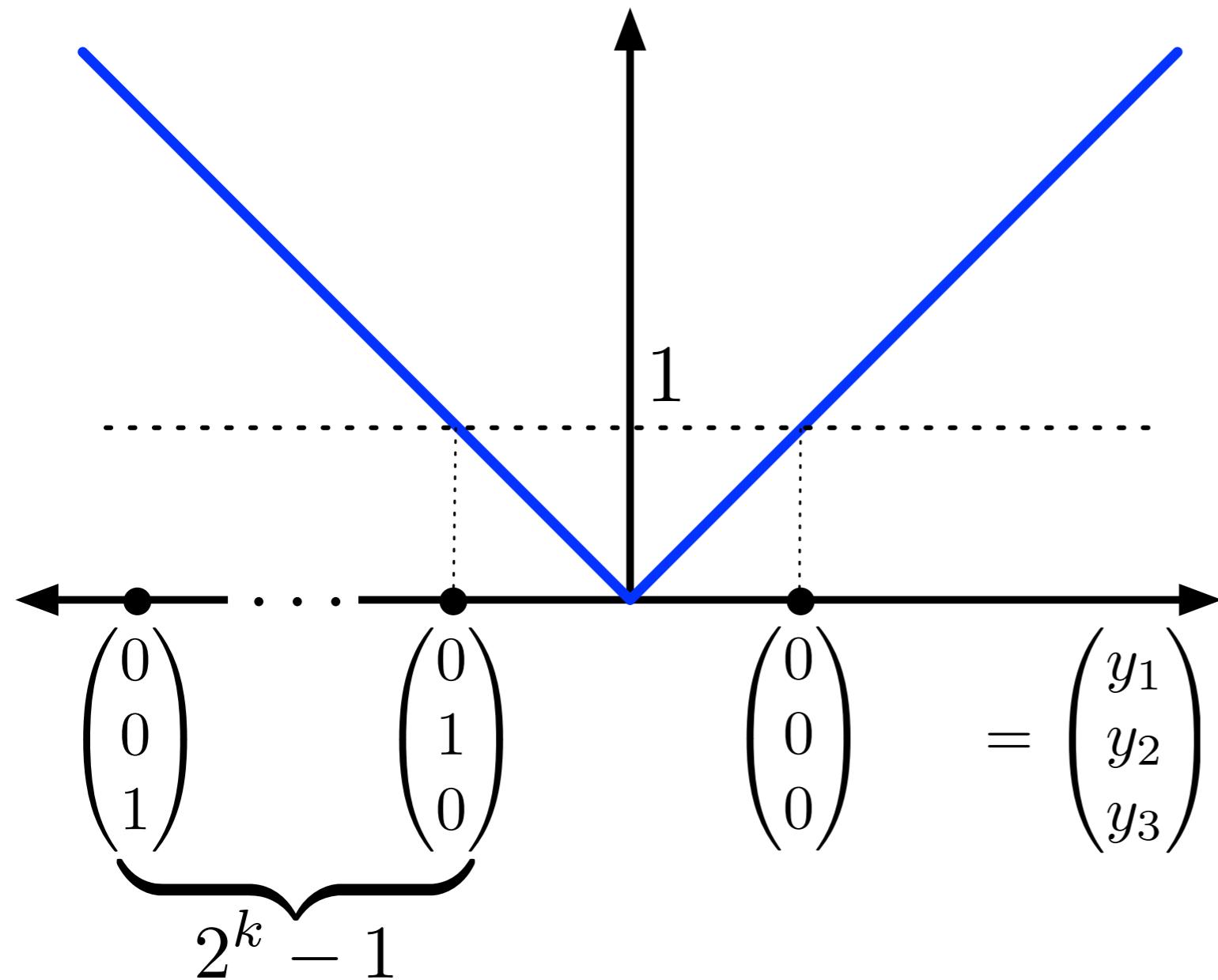
$$\sum_{i=1}^n b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}_+^n$$

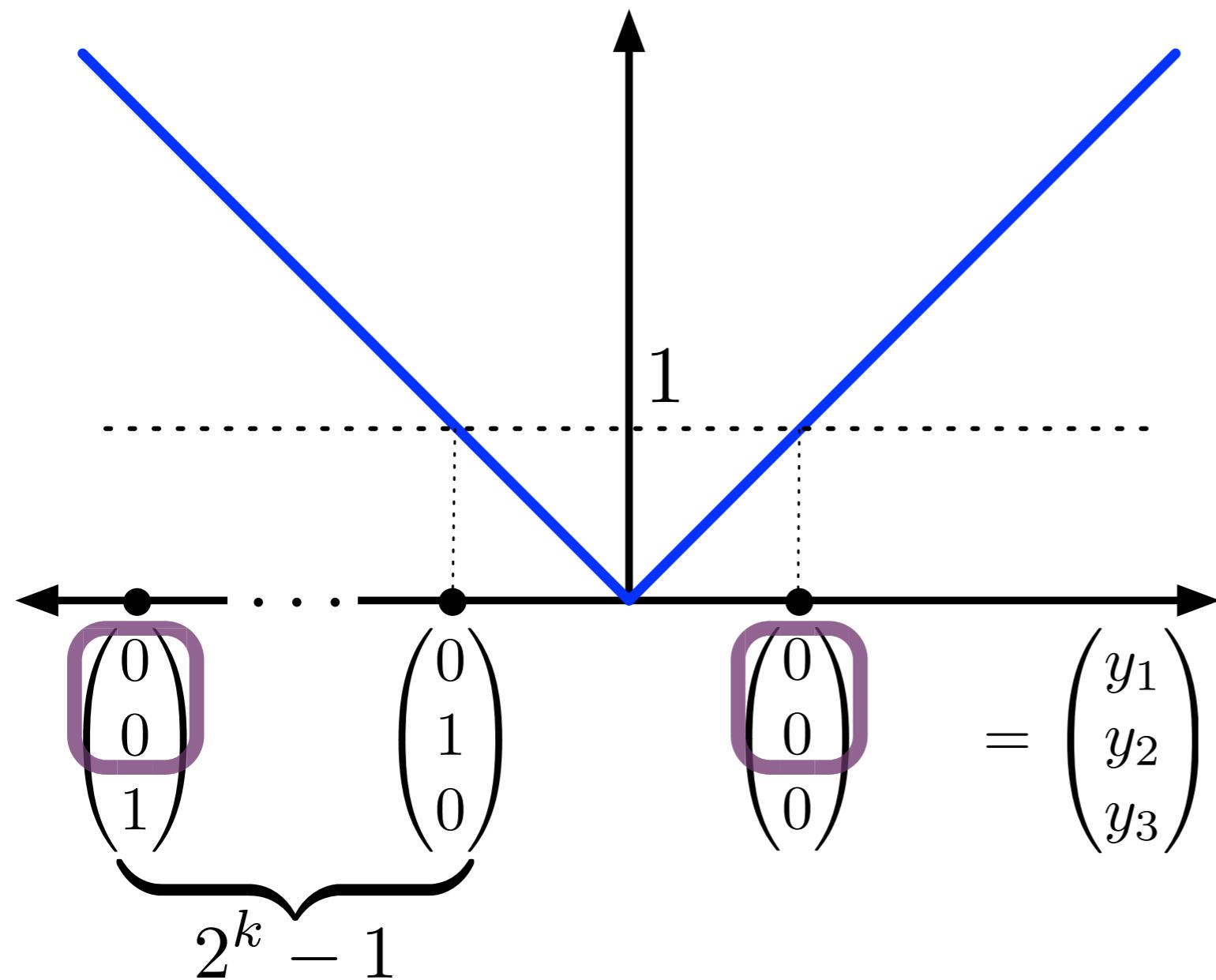
$$y \in \{0, 1\}^m$$

Need $n/2$
branches

Example: Binary Encoding

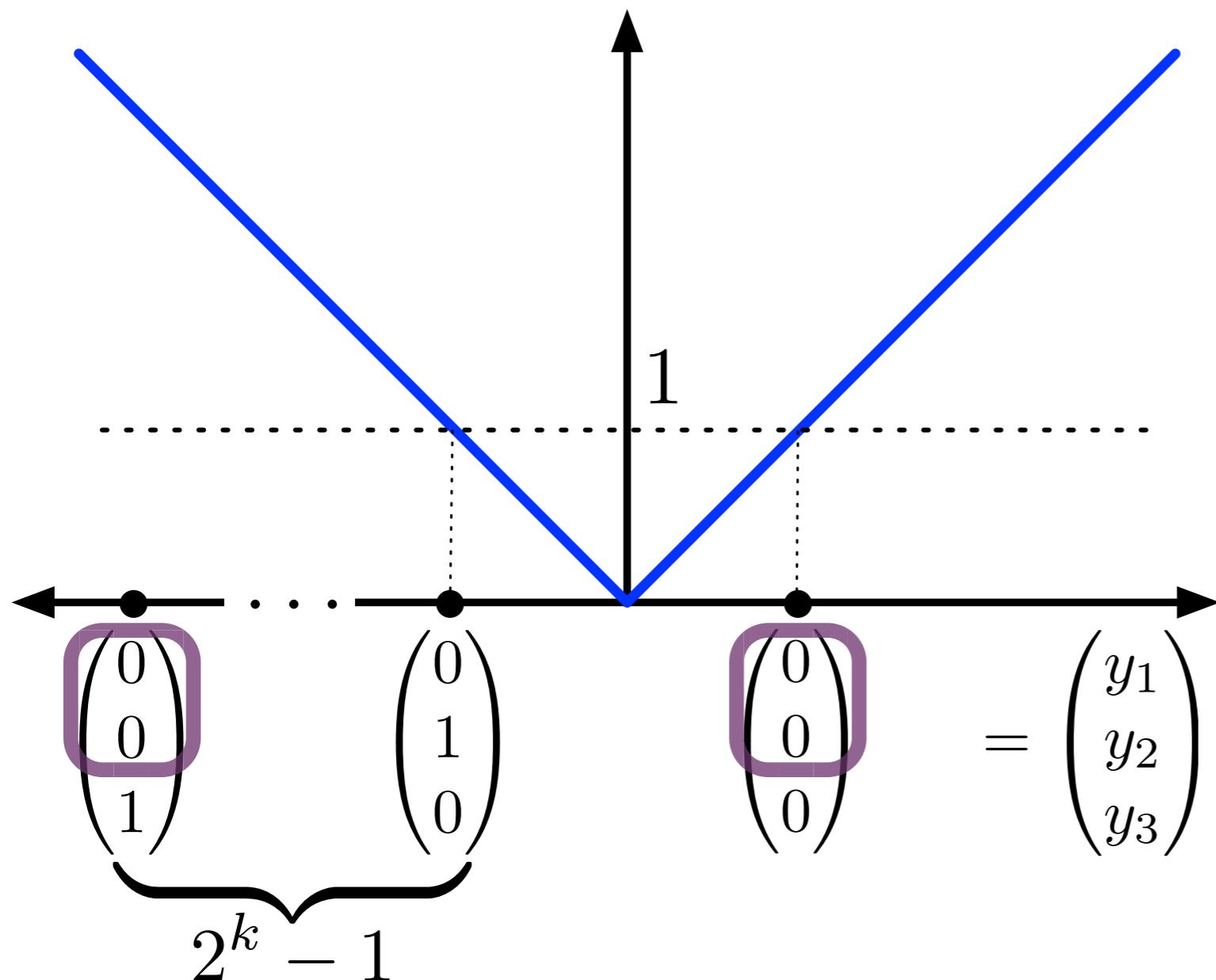


Example: Binary Encoding



$$y_1 = y_2 = 0$$

Example: Binary Encoding

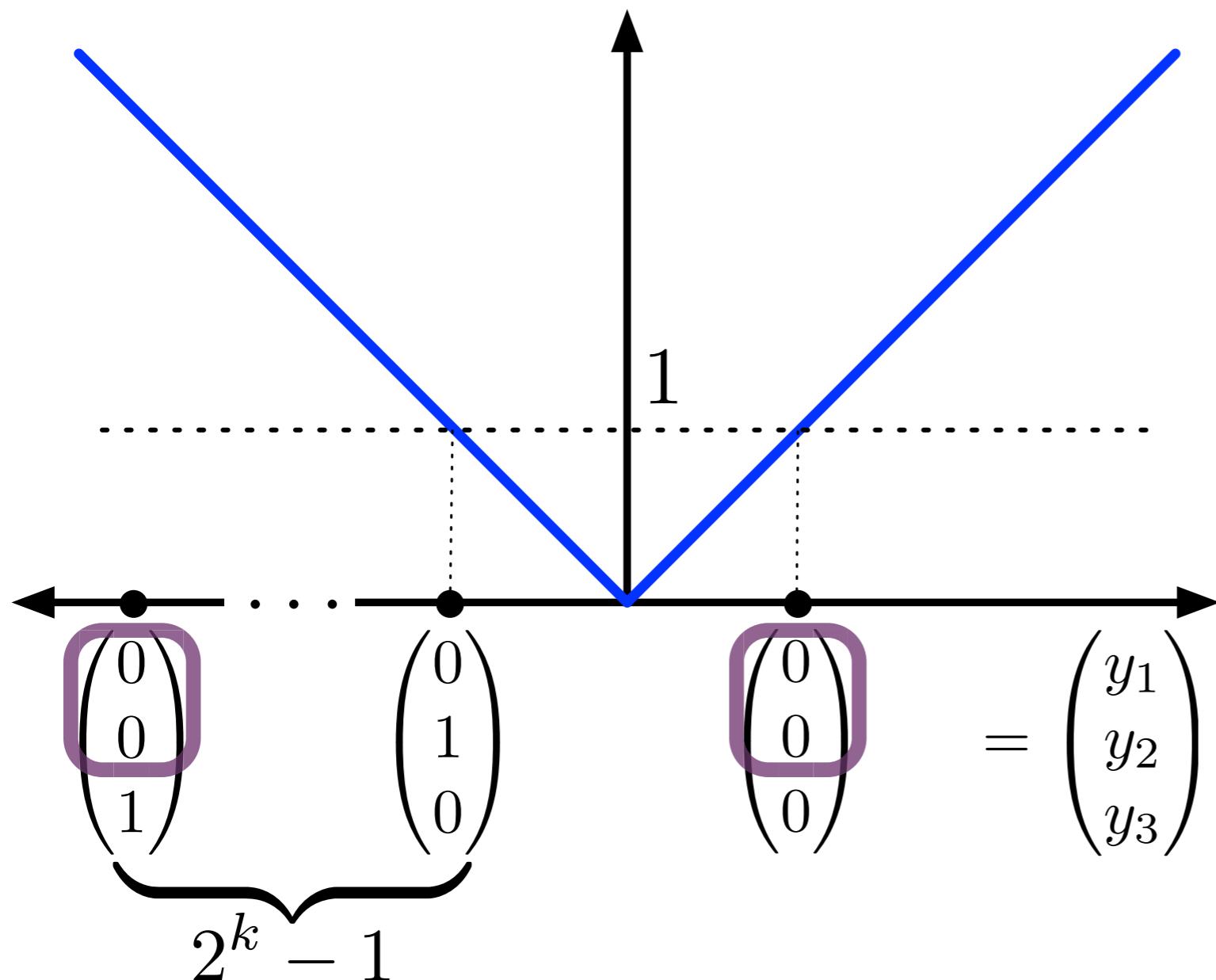


Best Bound = 0 unless:

$$y_i = 0 \quad \forall i$$

$$y_1 = y_2 = 0$$

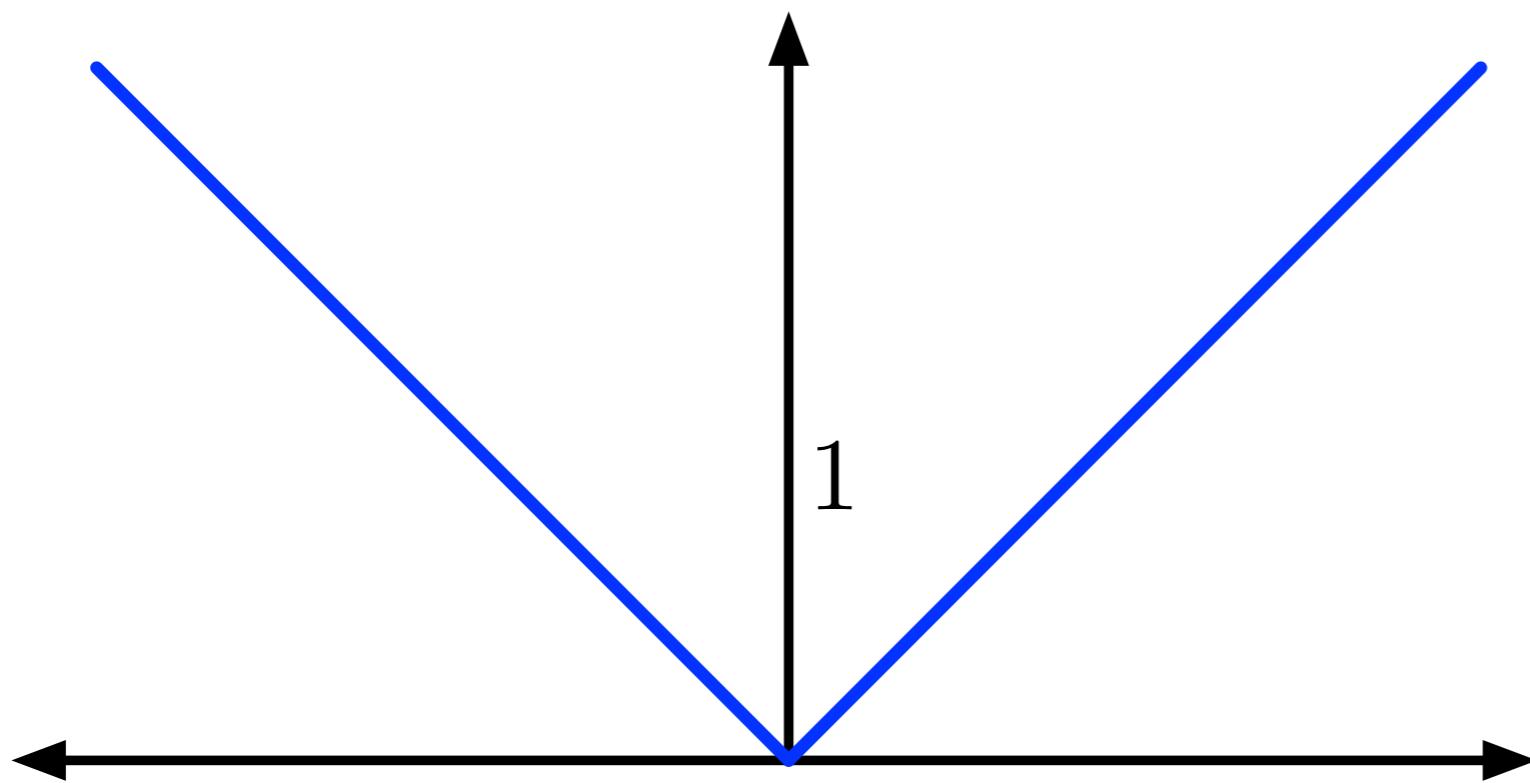
Example: Binary Encoding



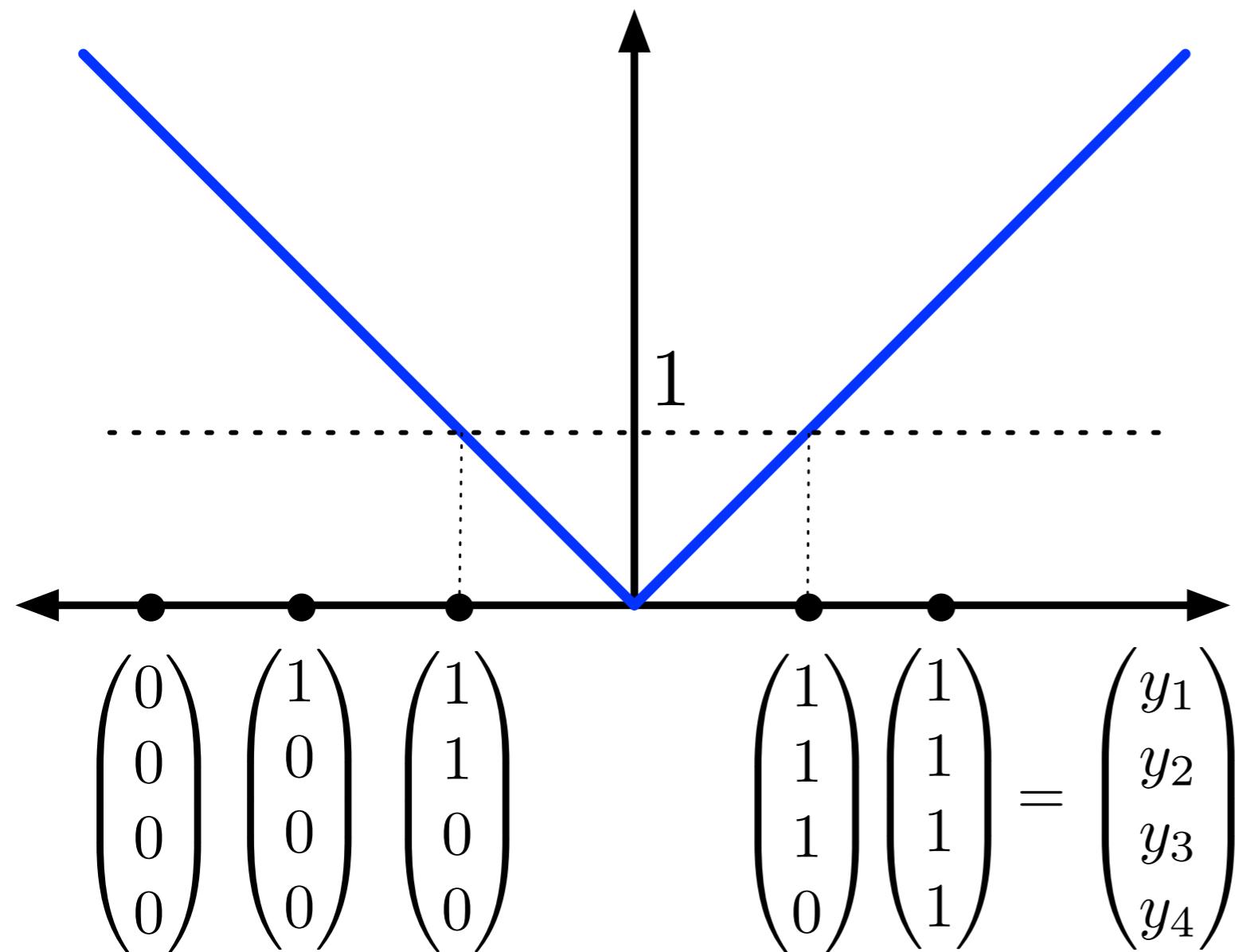
Best Bound = 0 unless:
 $y_i = 0 \quad \forall i$

Need $k = \log_2 n$ branches

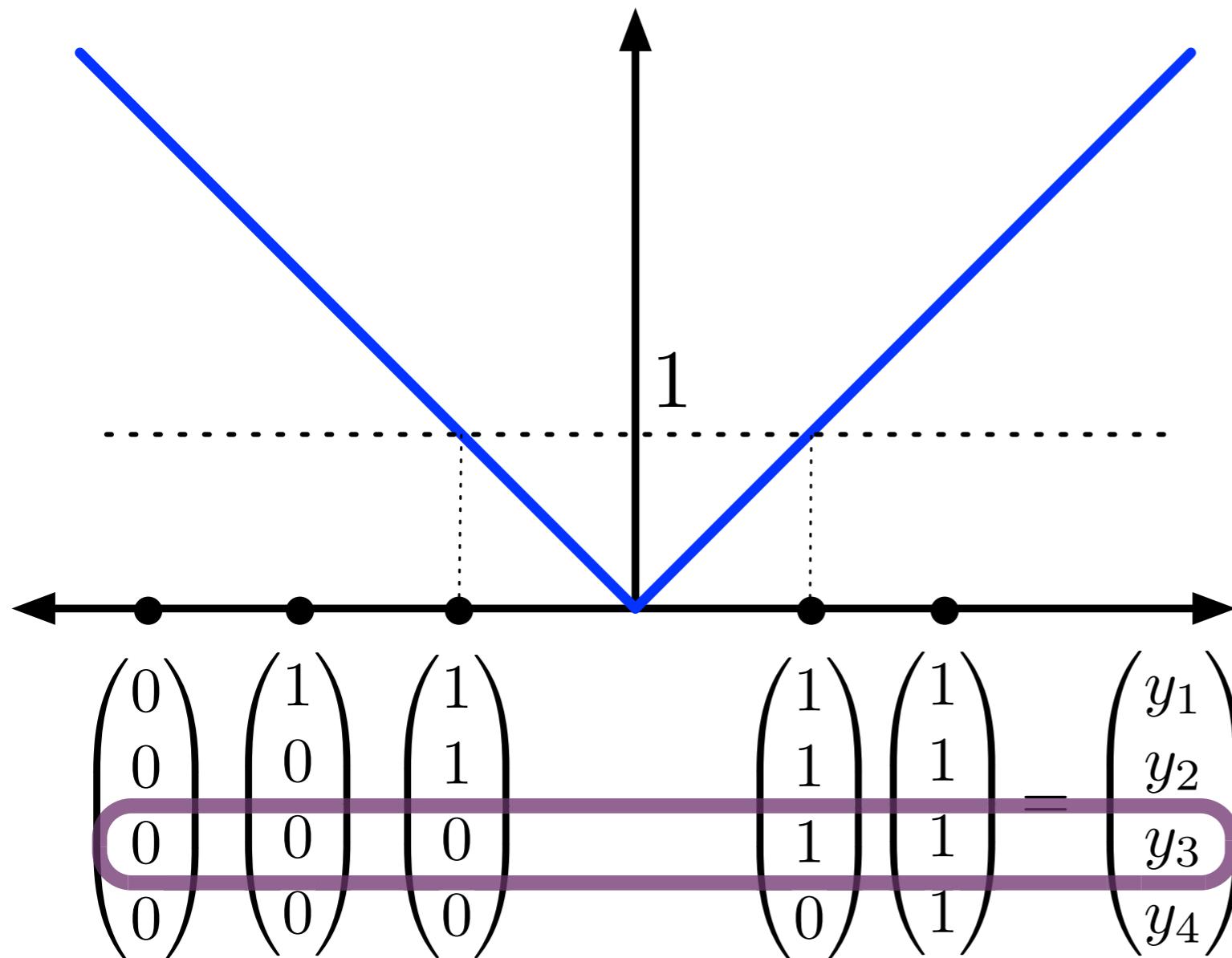
Example: Incremental Encoding



Example: Incremental Encoding

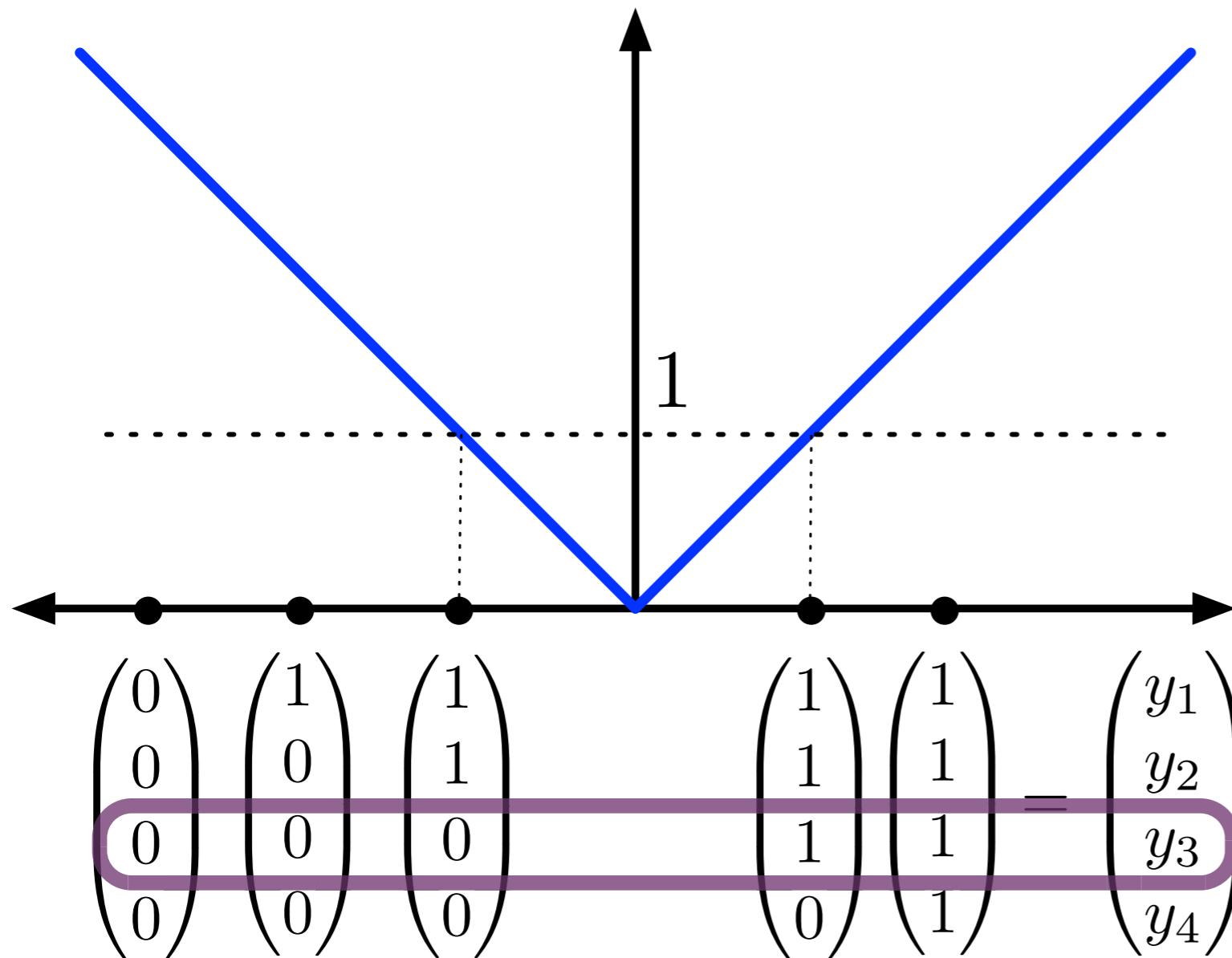


Example: Incremental Encoding



$$y_3 = 1 \vee y_3 = 0$$

Example: Incremental Encoding



Best Bound = 1 if:

$$y_{i^*} = 0 \vee y_{i^*} = 1$$

Only need
1 branch!

$$y_3 = 1 \vee y_3 = 0$$

Induced Constraint Branching

Incremental

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \lambda = y$$

Binary

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \lambda = y$$

Induced Constraint Branching

Incremental

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ \textcolor{purple}{\boxed{0}} & \textcolor{purple}{\boxed{0}} & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \lambda = y \quad \rightarrow$$

SOS1 Branching

$$\begin{aligned} \lambda_1 &= \lambda_2 = 0 \\ \textit{or} \\ \lambda_3 &= \lambda_4 = 0 \end{aligned}$$

Binary

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \lambda = y$$

Induced Constraint Branching

Incremental

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \lambda = y \quad \rightarrow$$

SOS1 Branching

$$\begin{aligned} \lambda_1 &= \lambda_2 = 0 \\ \text{or} \\ \lambda_3 &= \lambda_4 = 0 \end{aligned}$$

Binary

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \lambda = y \quad \rightarrow$$

SOS1 Branching

$$\begin{aligned} \lambda_1 &= \lambda_2 = 0 \\ \text{or} \\ \lambda_3 &= \lambda_4 = 0 \end{aligned}$$

Induced Constraint Branching

Incremental

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \lambda = y \quad \rightarrow$$

SOS1 Branching

$$\begin{aligned} \lambda_1 &= \lambda_2 = 0 \\ \text{or} \\ \lambda_3 &= \lambda_4 = 0 \end{aligned}$$

Binary

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \lambda = y \quad \rightarrow$$

Odd/Even Branching

$$\begin{aligned} \lambda_1 &= \lambda_3 = 0 \\ \text{or} \\ \lambda_2 &= \lambda_4 = 0 \end{aligned}$$

Formulation Step 2: Combining with Strong Formulation

Long Lost Integral Formulation

$\{P^i\}_{i=1}^n$ polytopes

$$x \in \bigcup_{i=1}^n P^i \Leftrightarrow$$

$$\sum_{i=1}^n \sum_{v \in \text{ext}(P^i)} v \lambda_v^i = x$$

$$\sum_{v \in \text{ext}(P^i)} \lambda_v^i = y_i$$

$$\sum_{i=1}^n y_i = 1$$

$$y \in \{0, 1\}^n, \lambda_v^i \geq 0$$

Also for general polyhedra
with common recession cones.

- Jeroslow and Lowe 1984.

Combining with Alternative Encoding

$\{P^i\}_{i=1}^n$ polytopes

$$x \in \bigcup_{i=1}^n P^i \Leftrightarrow$$

$$\sum_{i=1}^n \sum_{v \in \text{ext}(P^i)} v \lambda_v^i = x$$

$$\sum_{v \in \text{ext}(P^i)} \lambda_v^i = y_i$$

$$\sum_{i=1}^n y_i = 1$$

$$y \in \{0, 1\}^n, \lambda_v^i \geq 0$$

Also for general polyhedra
with common recession cones.

- Jeroslow and Lowe 1984.

Combining with Alternative Encoding

$\{P^i\}_{i=1}^n$ polytopes

$$x \in \bigcup_{i=1}^n P^i \iff$$

$$\sum_{i=1}^n \sum_{v \in \text{ext}(P^i)} v \lambda_v^i = x$$

$$\sum_{i=1}^n \sum_{v \in \text{ext}(P^i)} \lambda_v^i = 1$$

$$\sum_{i=1}^n \sum_{v \in \text{ext}(P^i)} b^i \lambda_v^i = y$$

$$y \in \{0, 1\}^m, \lambda_v^i \geq 0$$

Also for general polyhedra
with common recession cones.

- Jeroslow and Lowe 1984.

Combining with Alternative Encoding

$\{P^i\}_{i=1}^n$ polytopes

$$x \in \bigcup_{i=1}^n P^i \iff$$

$$\sum_{i=1}^n \sum_{v \in \text{ext}(P^i)} v \lambda_v^i = x$$

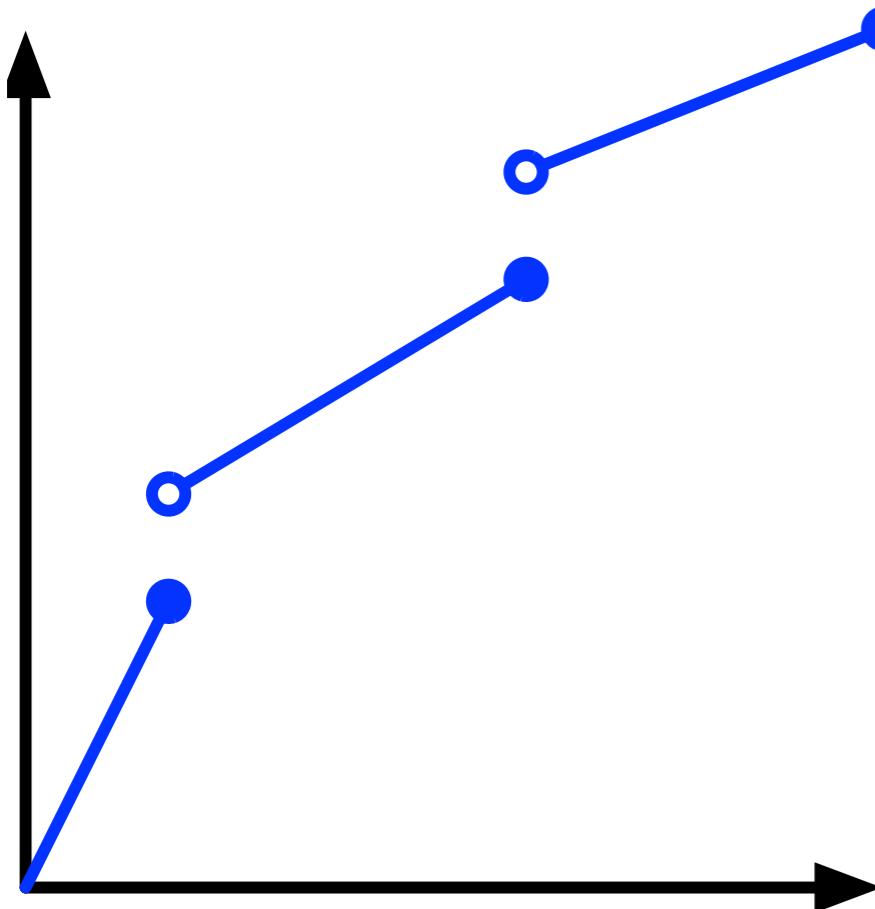
$$\sum_{i=1}^n \sum_{v \in \text{ext}(P^i)} \lambda_v^i = 1$$

$$\sum_{i=1}^n \sum_{v \in \text{ext}(P^i)} b^i \lambda_v^i = y$$

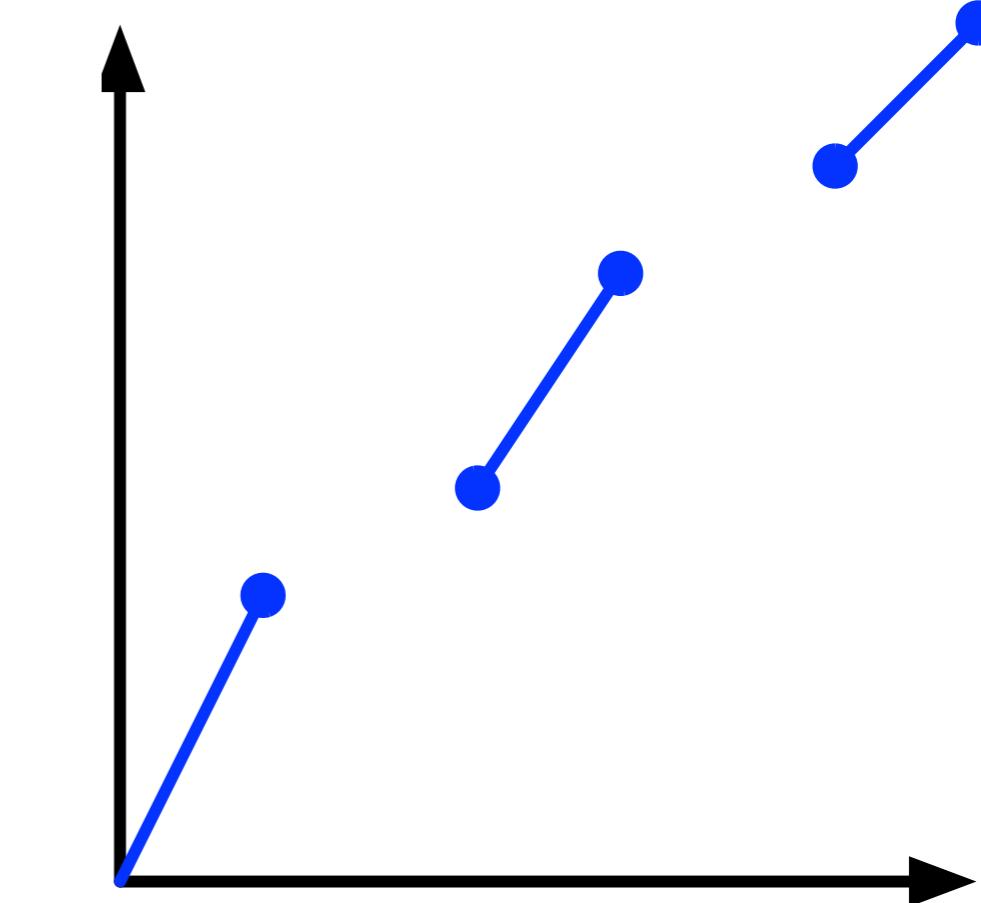
$$y \in \{0, 1\}^m, \lambda_v^i \geq 0$$

Also for general polyhedra
with common recession cones.

Univariate Transportation Problems



Discontinuous
Piecewise Linear



Disc. PWL
+
“Semicontinuous”

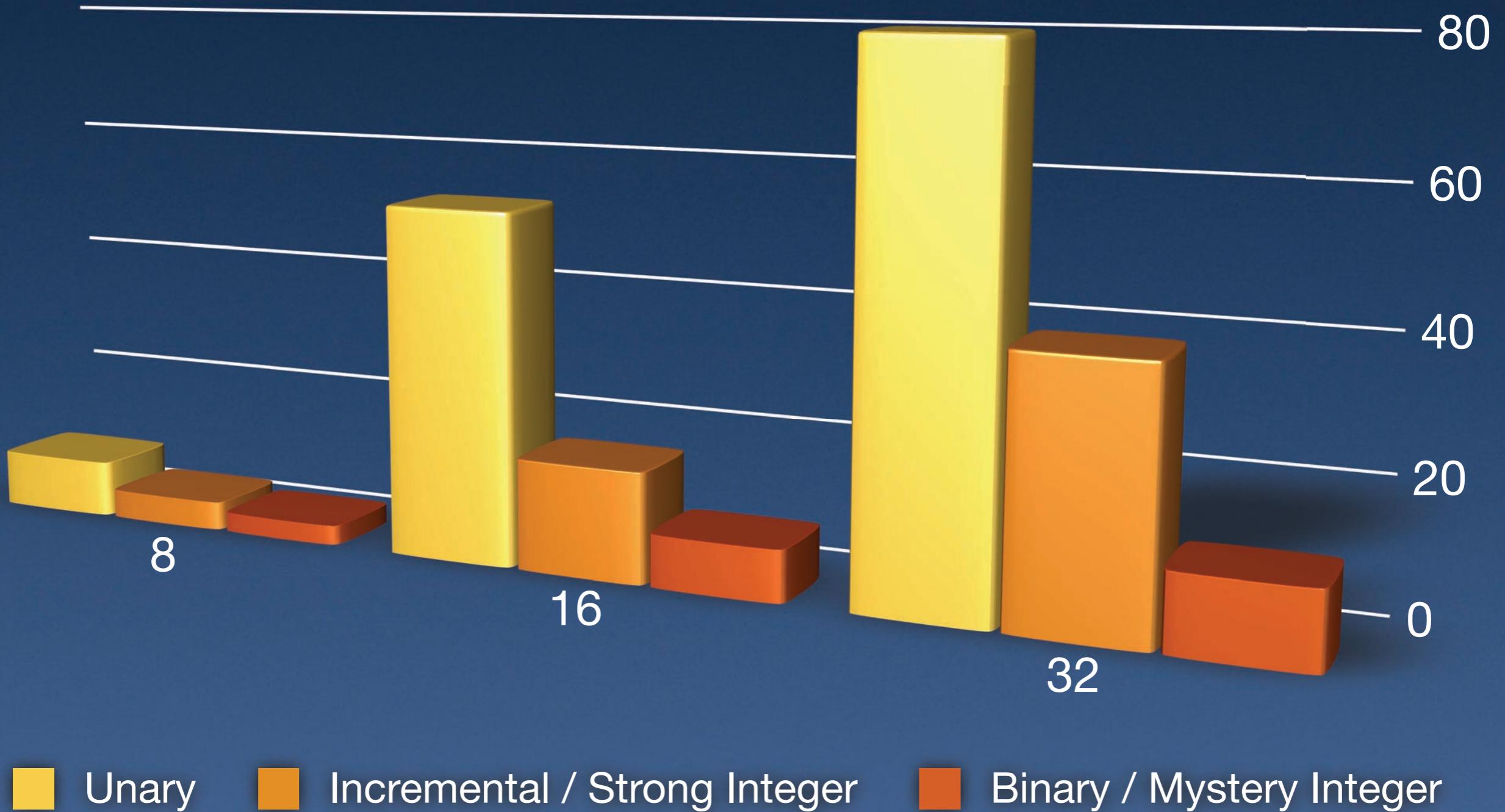
Piecewise Linear

■ Unary

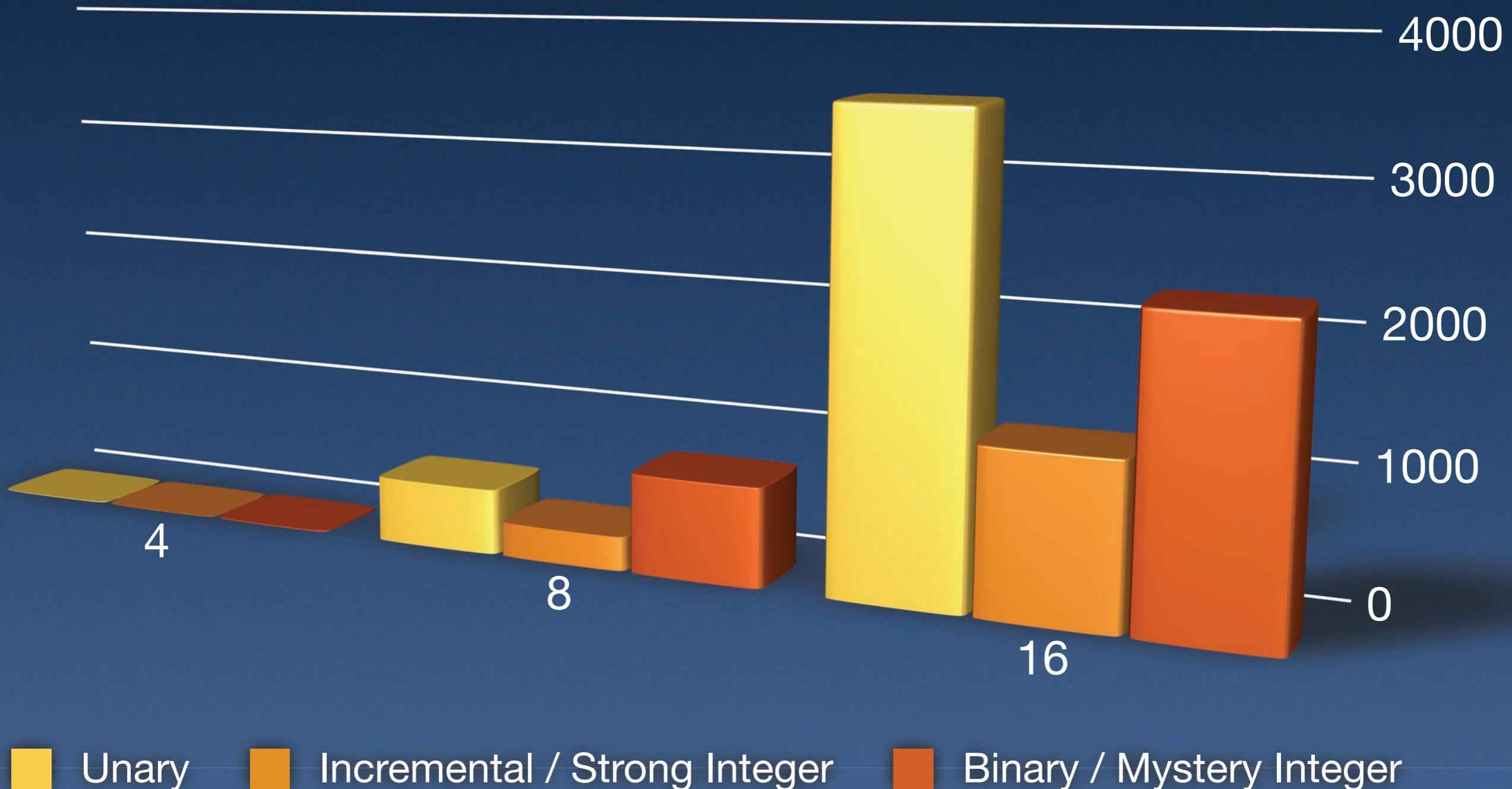
■ Incremental / Strong Integer

■ Binary / Mystery Integer

Piecewise Linear



Piecewise Linear + Semi Continuous



Summary, Extensions and More.

- Effective formulations: Encode and Formulate
 - Best encoding? Why not try a few.
 - Clever combination of encodings can be useful (e.g. V. and Nemhauser 2008 for multivariate piecewise linear functions)
- Smaller formulations for shared vertex case
 - Need encodings with special structure.

More Information

More Information

- Survey: V., “MIP Formulation Techniques”:
 - http://www.optimization-online.org/DB_HTML/2012/07/3539.html.

More Information

- Survey: V., “MIP Formulation Techniques”:
 - http://www.optimization-online.org/DB_HTML/2012/07/3539.html.
- Next year: automatic formulations for JUMP
 - Julia based modeling language:
 - As simple as AMPL + “faster” than C++
 - Solver independent call-backs and more!
 - JUMP/Julia tutorial in January