Encodings in Mixed Integer Linear Programming

Juan Pablo Vielma

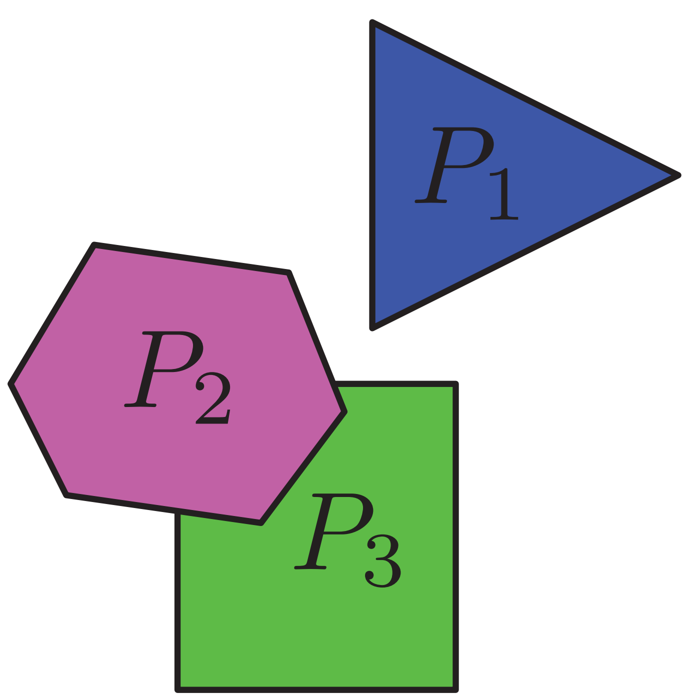
Massachusetts Institute of Technology

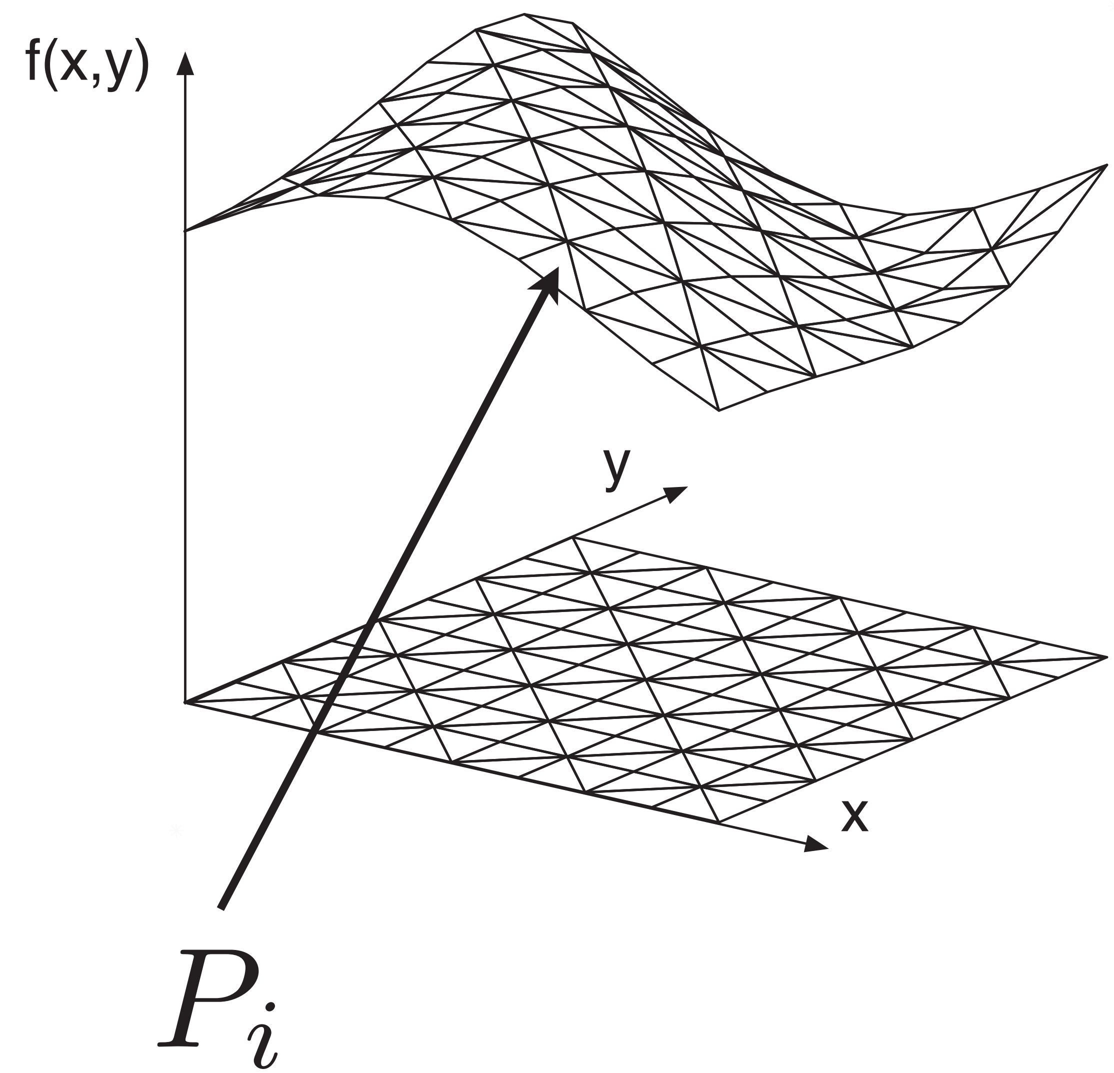
ORC Seminar,
December 2012 – Cambridge, Massachusetts

Mixed Integer Binary Formulations

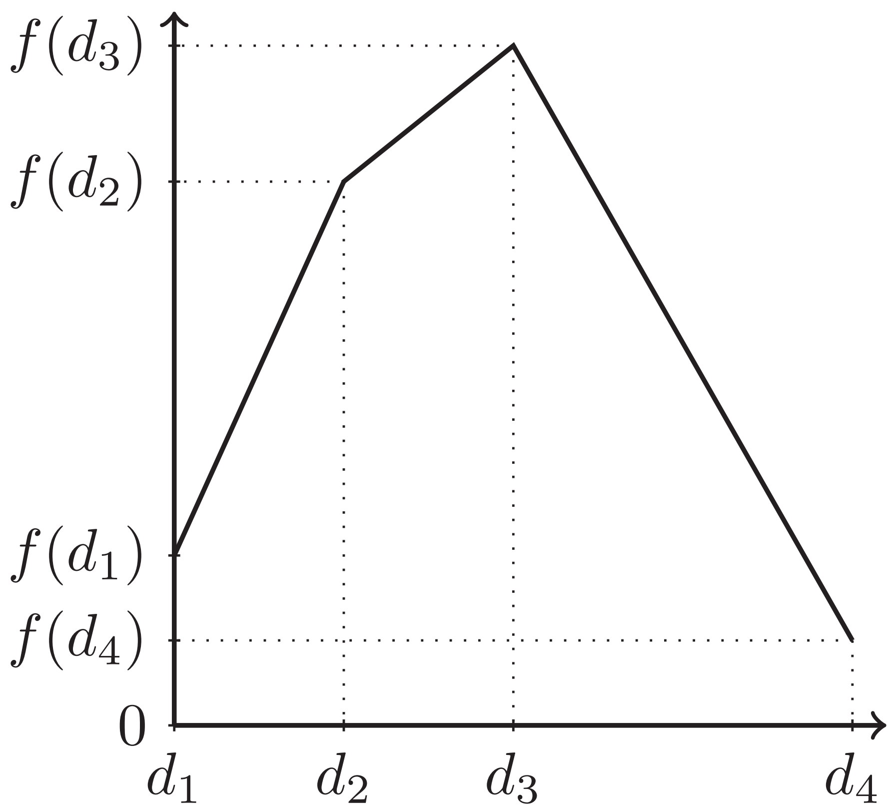
- MIP Formulations = Model Finite Alternatives

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





Textbook Formulation



Formulation for f(x)=z

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

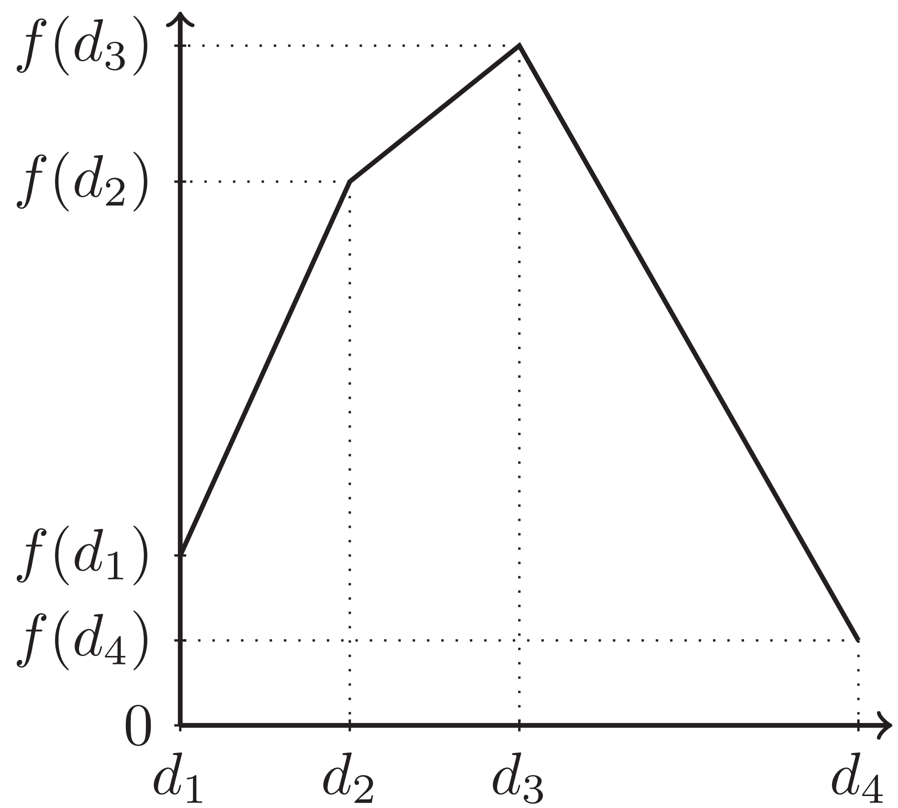
$$\sum_{i=1}^{4} \lambda_i = 1, \qquad \lambda_i \geq 0$$

$$\sum_{i=1}^{3} y_i = 1, \qquad y_i \in \{0, 1\}$$

$$\lambda_1 \leq y_1, \qquad \lambda_2 \leq y_1 + y_2$$

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

Textbook Formulation



Formulation for f(x)=z

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

$$\sum_{i=1}^{4} \lambda_i = 1, \qquad \lambda_i \geq 0$$

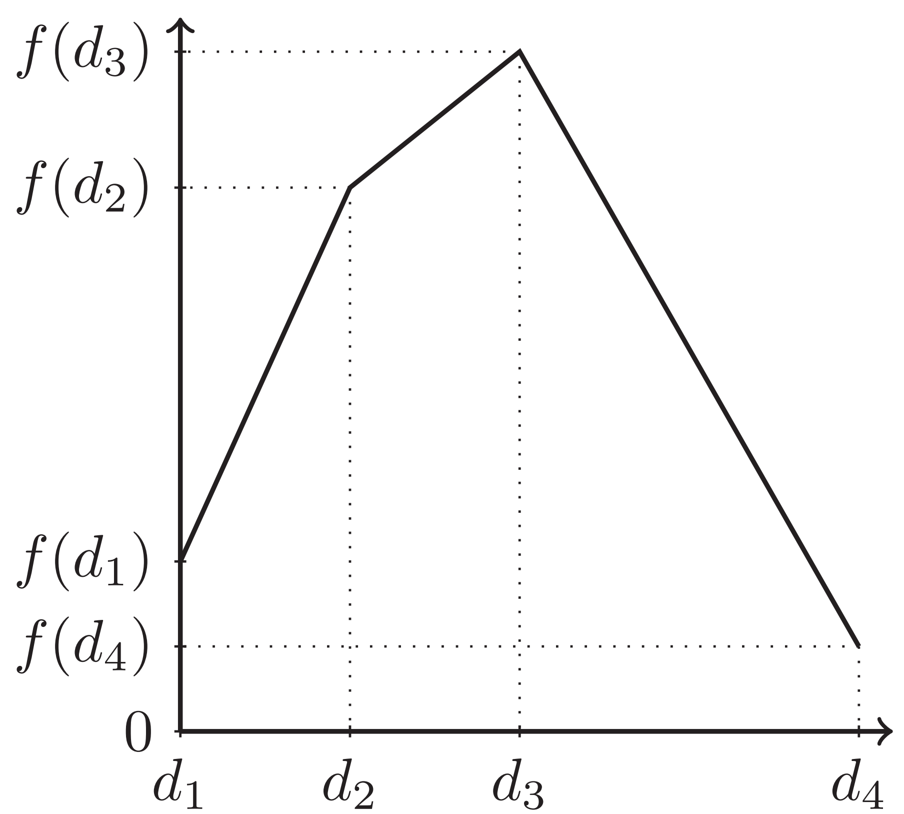
$$\sum_{i=1}^{3} y_i = 1, \qquad y_i \in \{0, 1\}$$

$$\lambda_1 \leq y_1, \qquad \lambda_2 \leq y_1 + y_2$$

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

“Weak”

Better Formulation



Formulation for f(x)=z

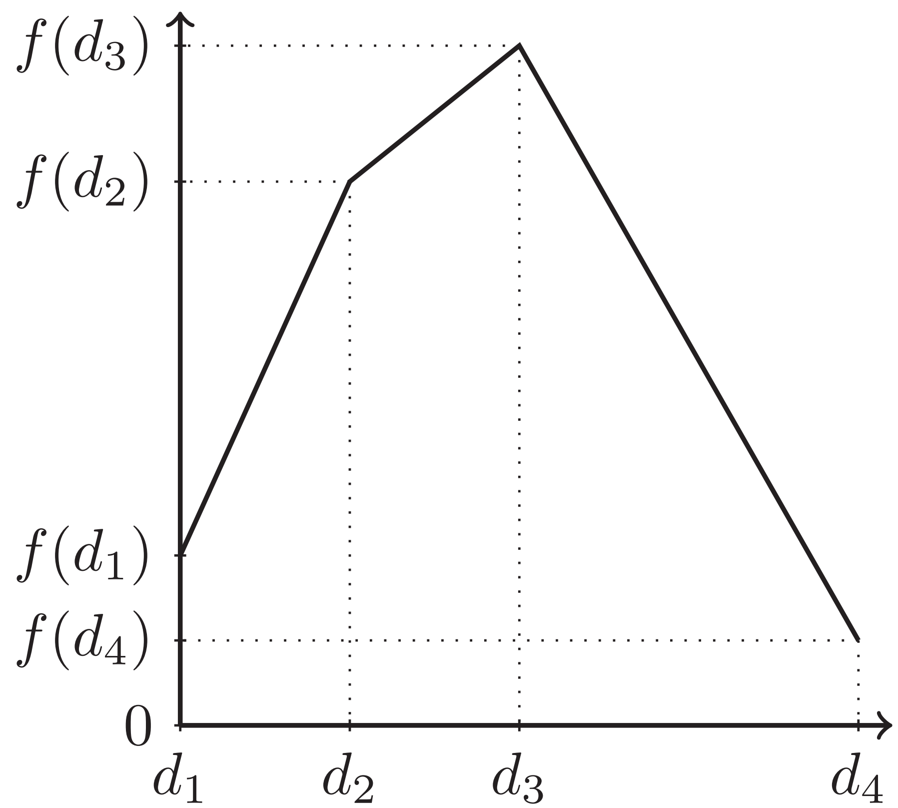
$$d_0 + \sum_{i=1}^{3} (d_{i+1} - d_i)\delta_i = x,$$

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

$$\delta_3 \leq y_2 \leq \delta_2 \leq y_1 \leq \delta_1$$

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

Better Formulation



Formulation for f(x)=z

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

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

$$\delta_3 \le y_2 \le \delta_2 \le y_1 \le \delta_1$$

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

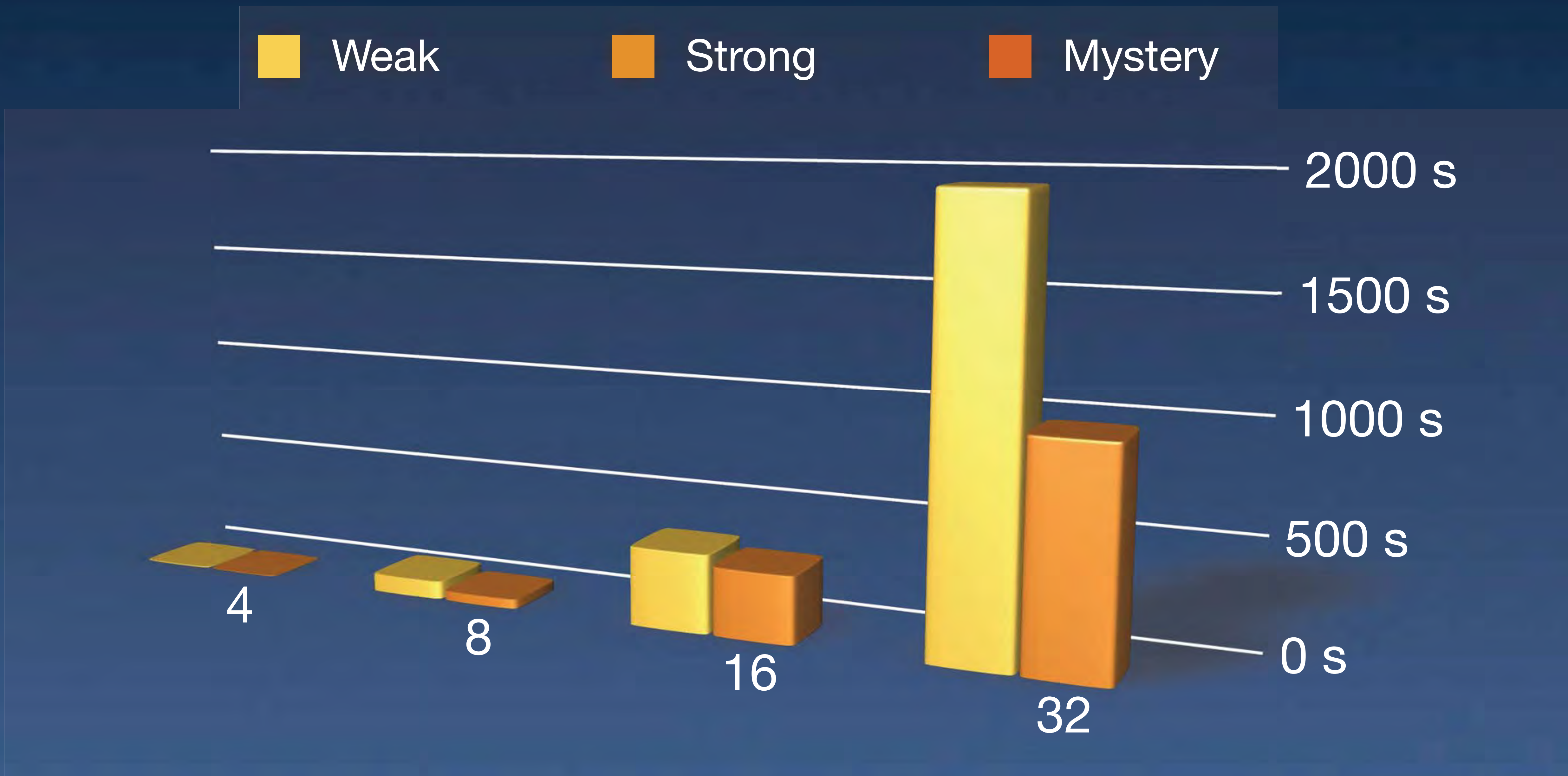
Integral

Solve Times in CPLEX 11

Weak   Strong   Mystery

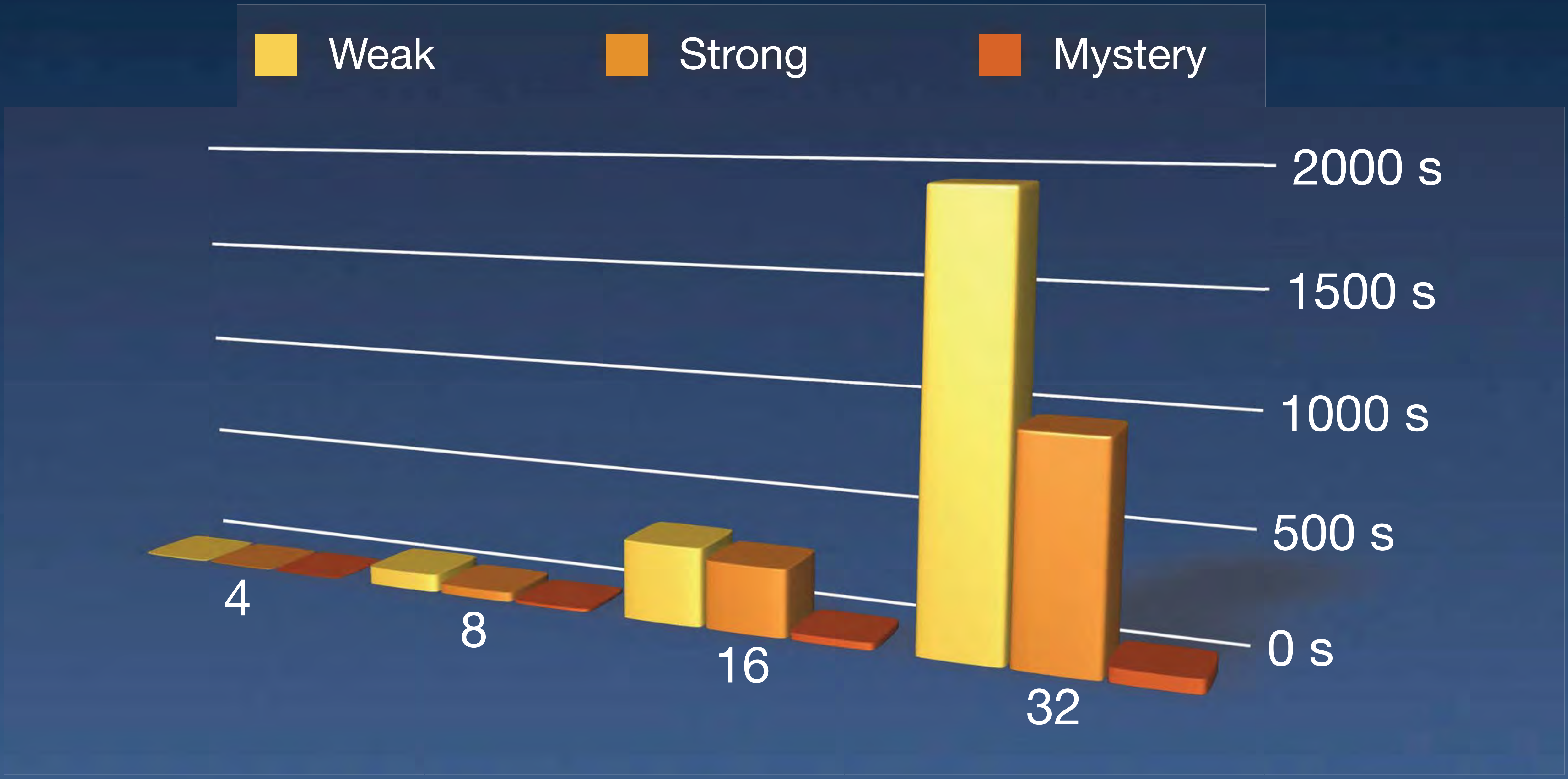
- Transportation Problems in V., Ahmed and Nemhauser '10.

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

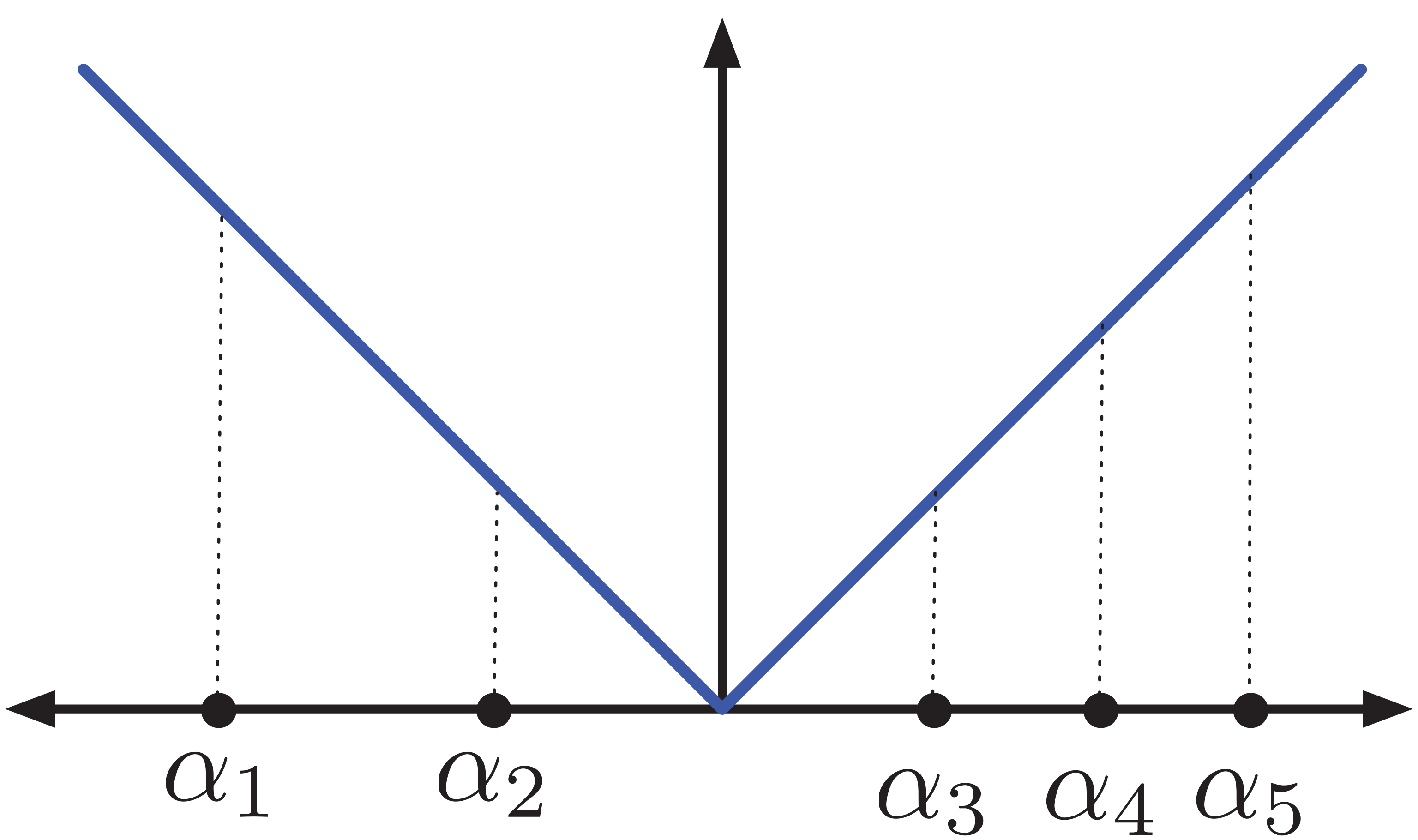
- MIP v/s constraint branching.
- “Have your cake and eat it too” formulation
  - Step 1: Encoding alternatives.
  - Step 2: Combine with strong “standard” formulation.
- Summary, Extensions and More.

Formulating Discrete Alternatives

$$\min \quad |x|$$

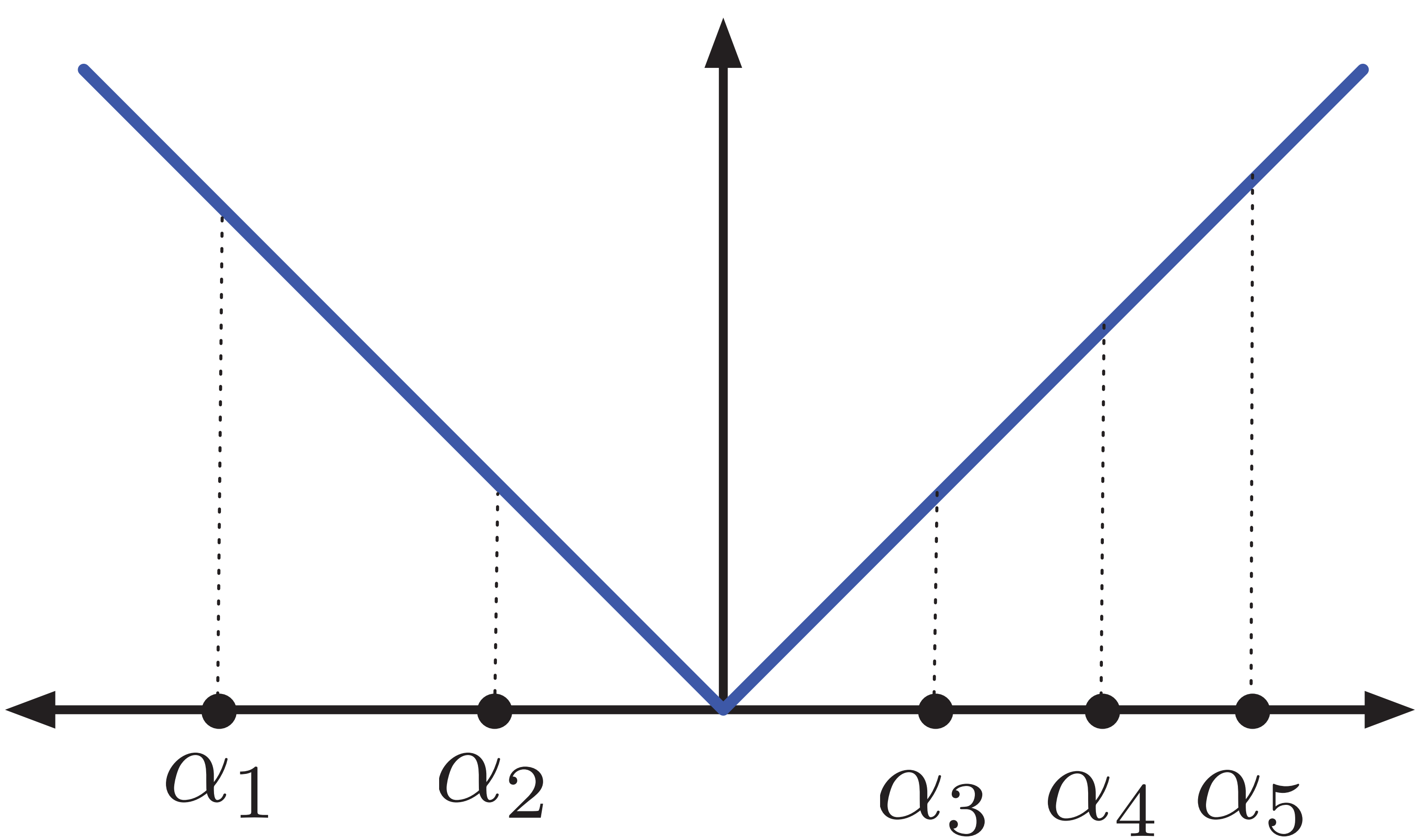
$$s.t.$$

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



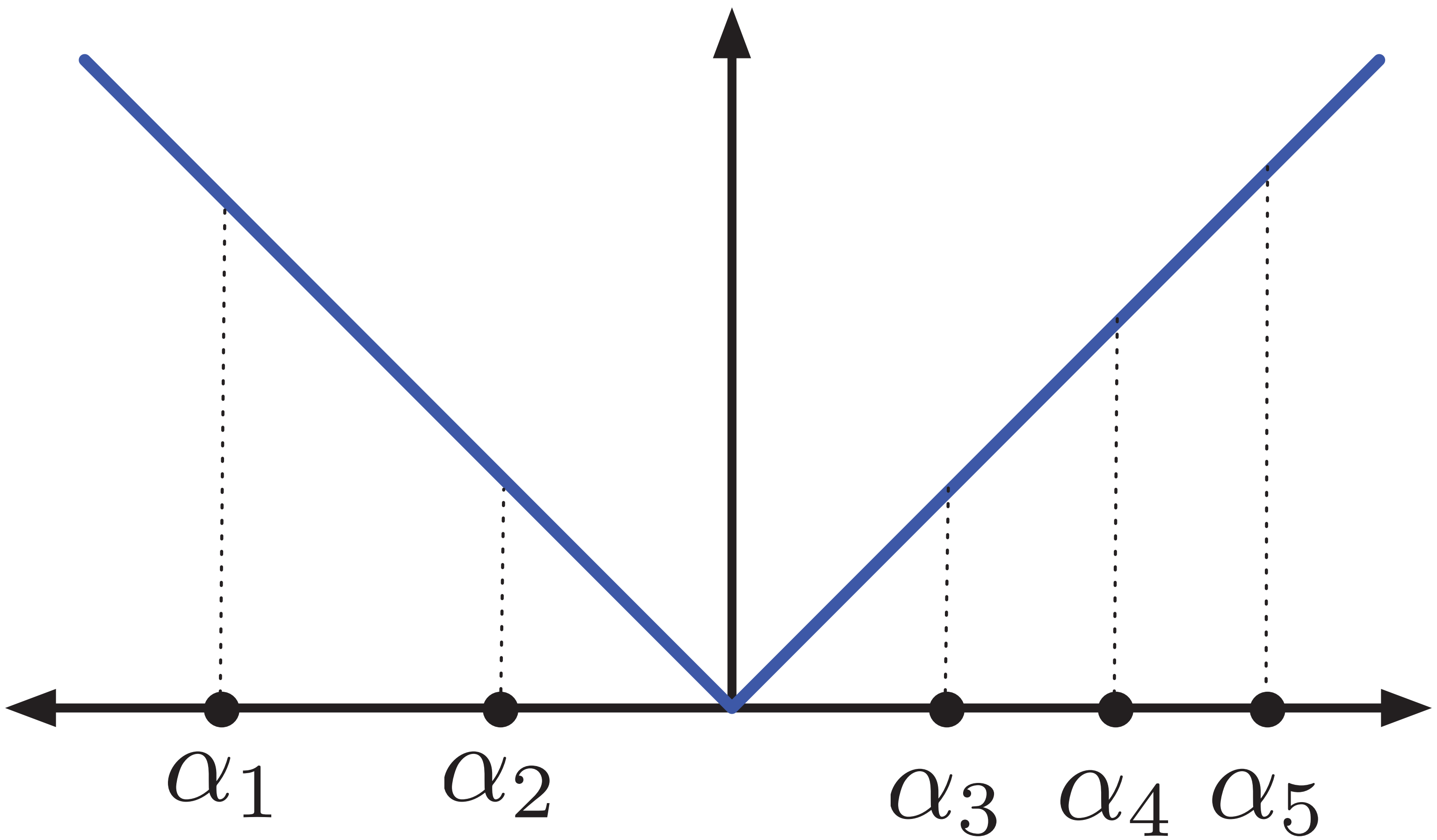
Formulating Discrete Alternatives

$$\begin{aligned} \min \quad & |x| \\ s.t. \quad & \\ & x \in \{\alpha_i\}_{i=1}^n \end{aligned}$$



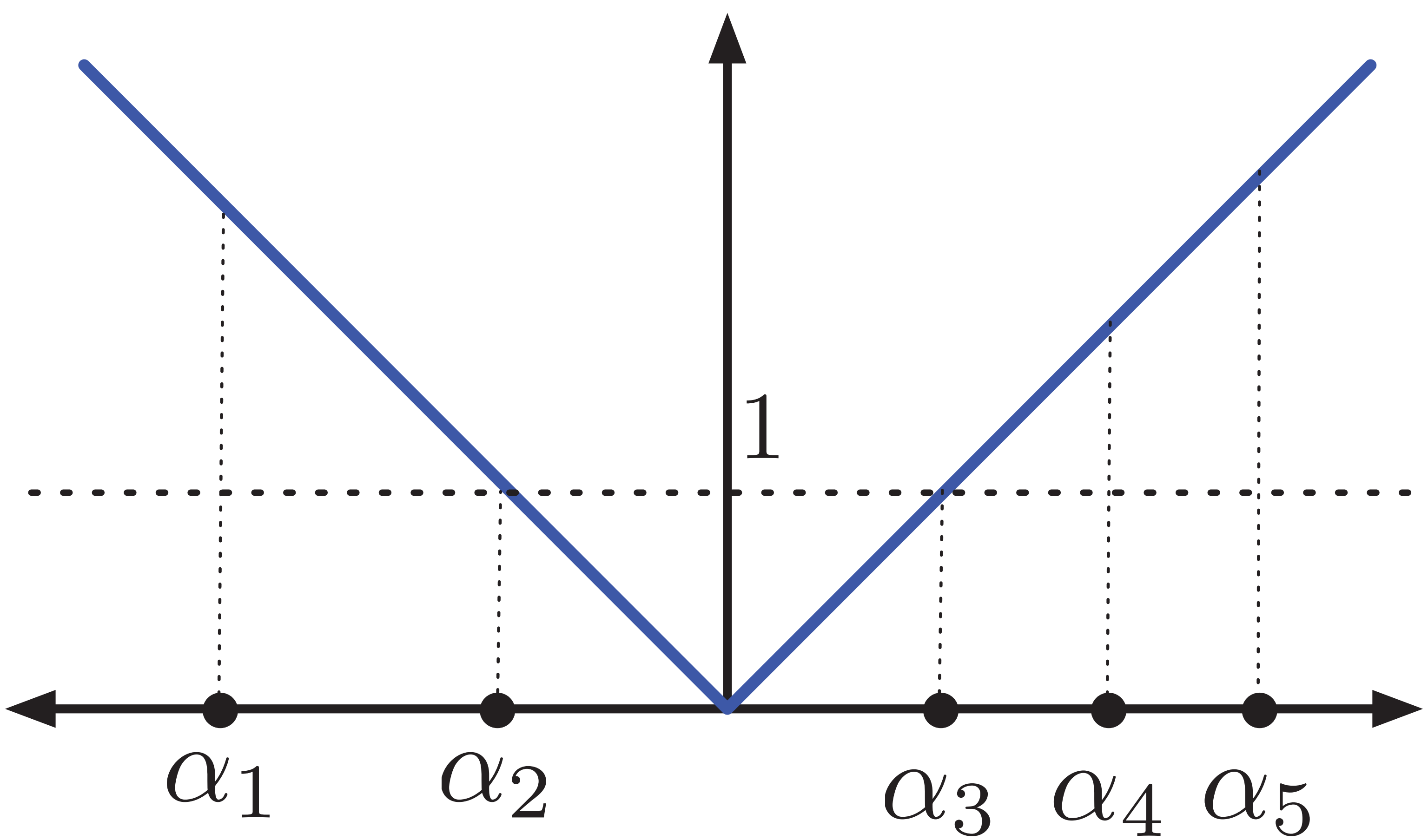
$$\begin{aligned} \min \quad & |x| \\ s.t. \quad & \\ & \sum_{i=1}^n \lambda_i \alpha_i = x \\ & \sum_{i=1}^n \lambda_i = 1 \\ & \lambda \in \{0,1\}^n \end{aligned}$$

Formulating Discrete Alternatives



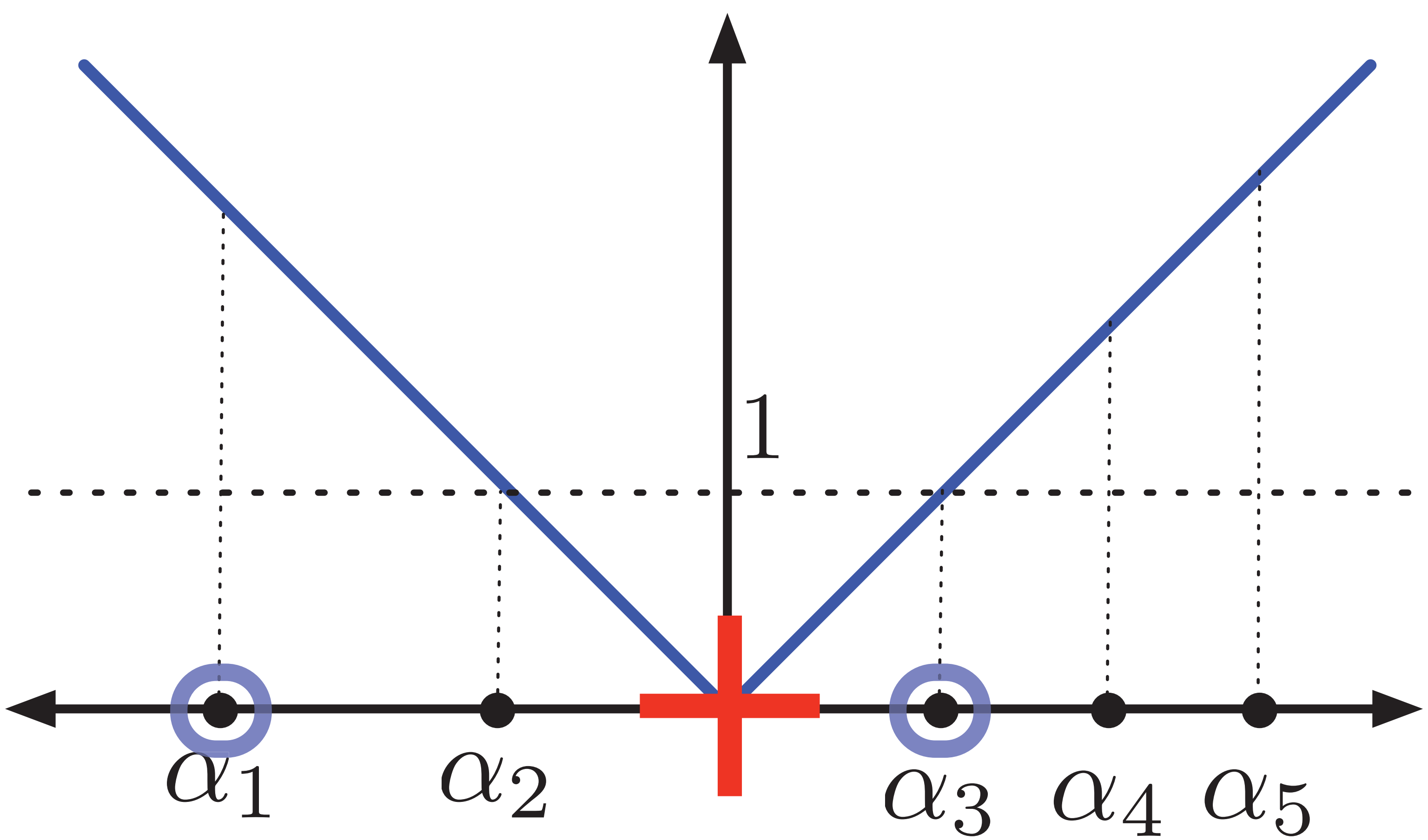
$$\begin{aligned} \min \quad & |x| \\ s.t. \quad & \\ & \sum_{i=1}^{n} \lambda_i \alpha_i = x \\ & \sum_{i=1}^{n} \lambda_i = 1 \\ & \lambda \in \{0,1\}^n \end{aligned}$$

Formulating Discrete Alternatives



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

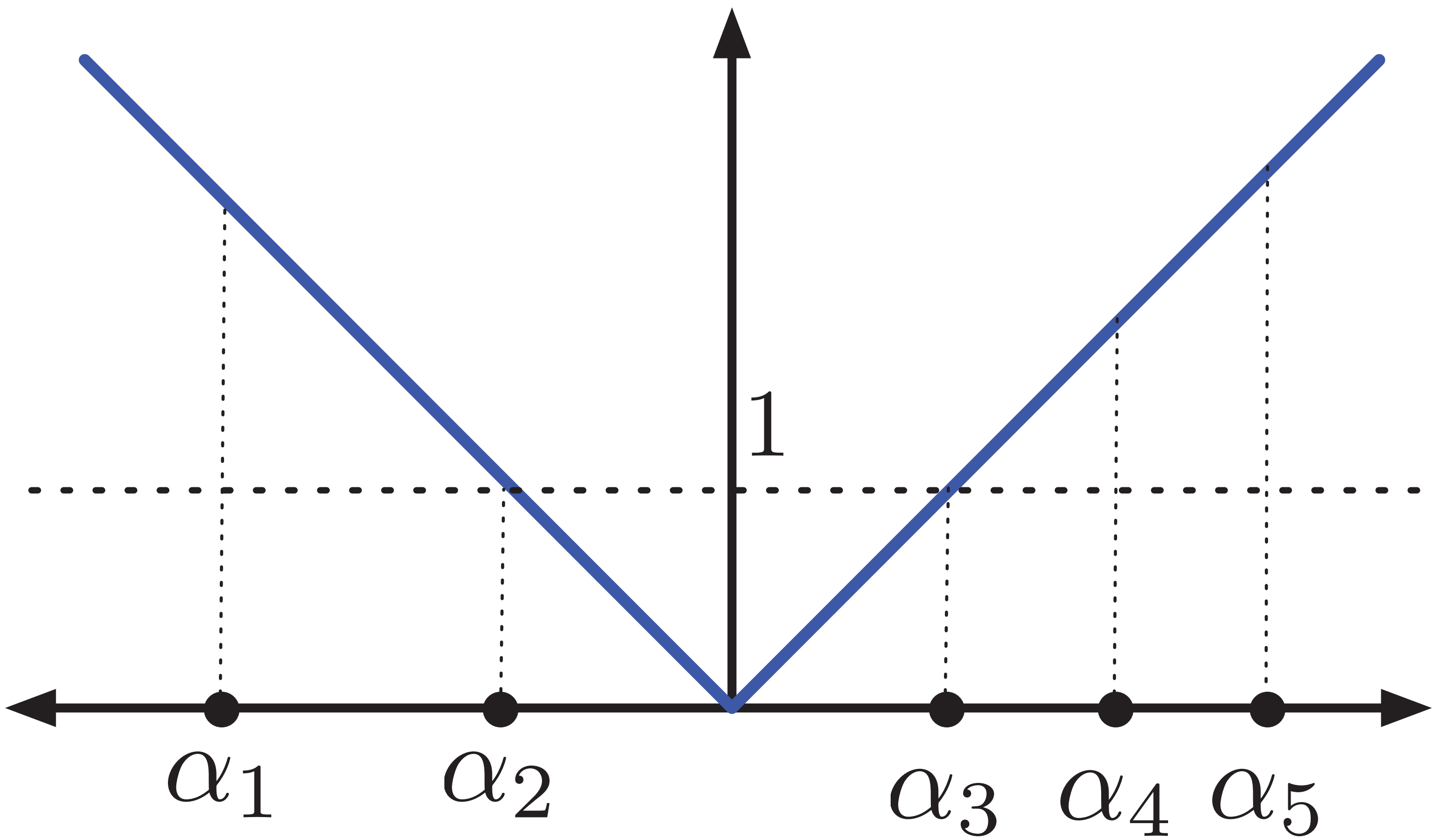
Formulating Discrete Alternatives



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

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

Formulating Discrete Alternatives



$$\min \quad |x|$$

$s.t.$

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

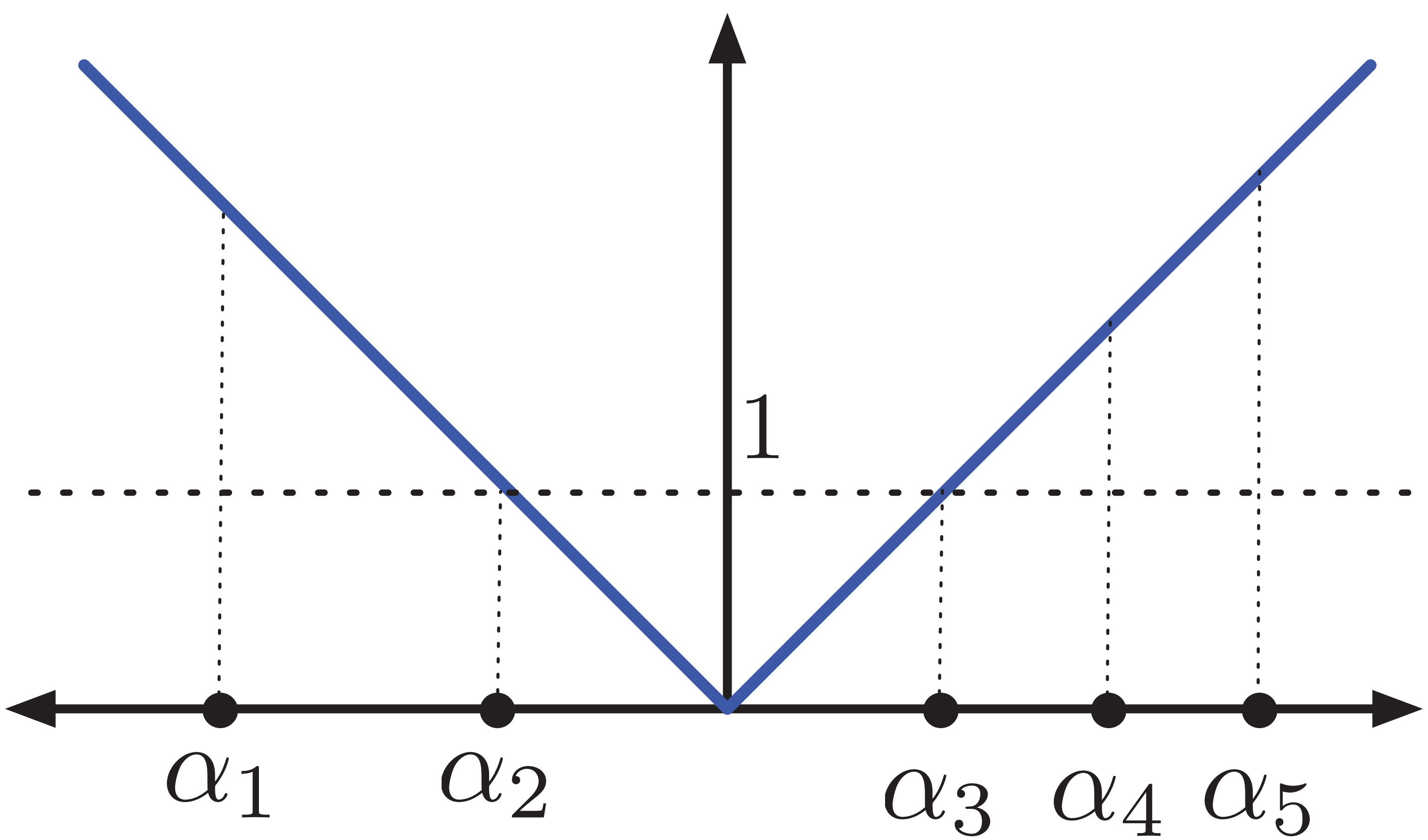
$$\sum_{i=1}^{n} \lambda_i = 1$$

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

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

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

Formulating Discrete Alternatives



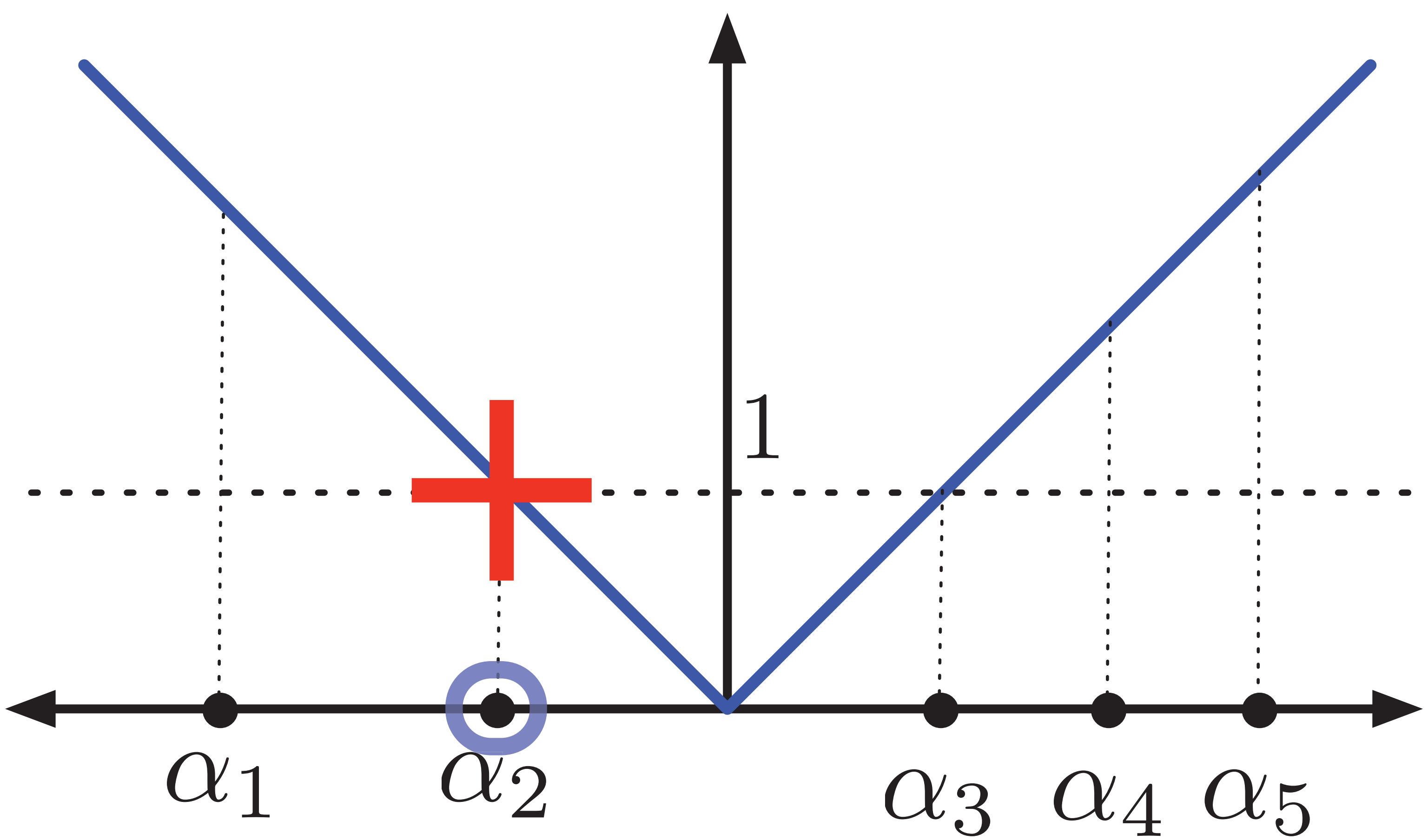
$$
\begin{aligned}
\min \quad & |x| \\
s.t. \quad & \\
& \textstyle\sum_{i=1}^{n} \lambda_i \alpha_i = x \\
& \textstyle\sum_{i=1}^{n} \lambda_i = 1 \\
& \lambda \in \{0,1\}^n
\end{aligned}
$$

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

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

Branch on λ_2

Formulating Discrete Alternatives



$$\min \quad |x|$$

$$s.t.$$

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

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

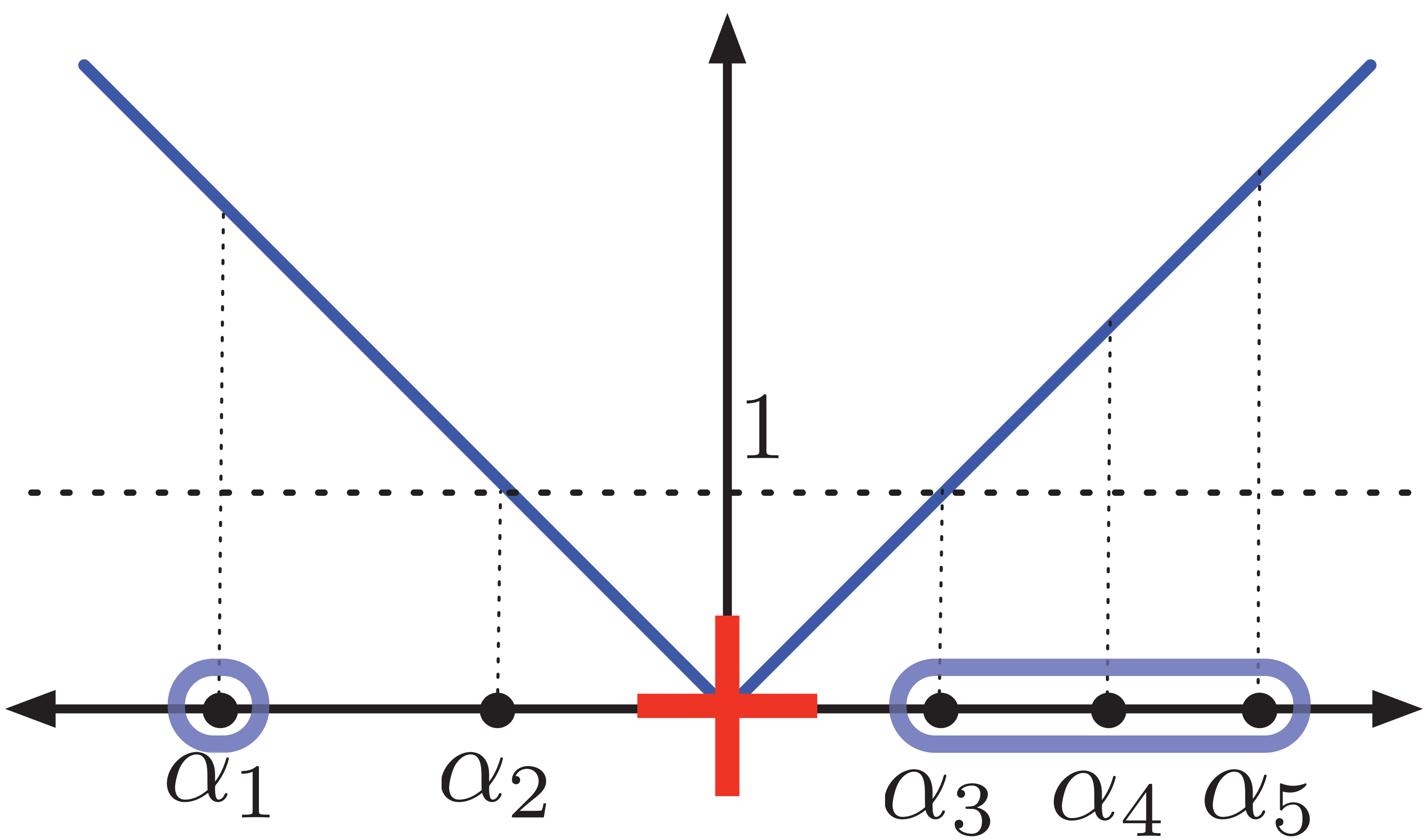
$$\lambda \in \{0,1\}^n$$

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

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

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

Formulating Discrete Alternatives



$$\min \quad |x|$$

$$s.t.$$

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

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

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

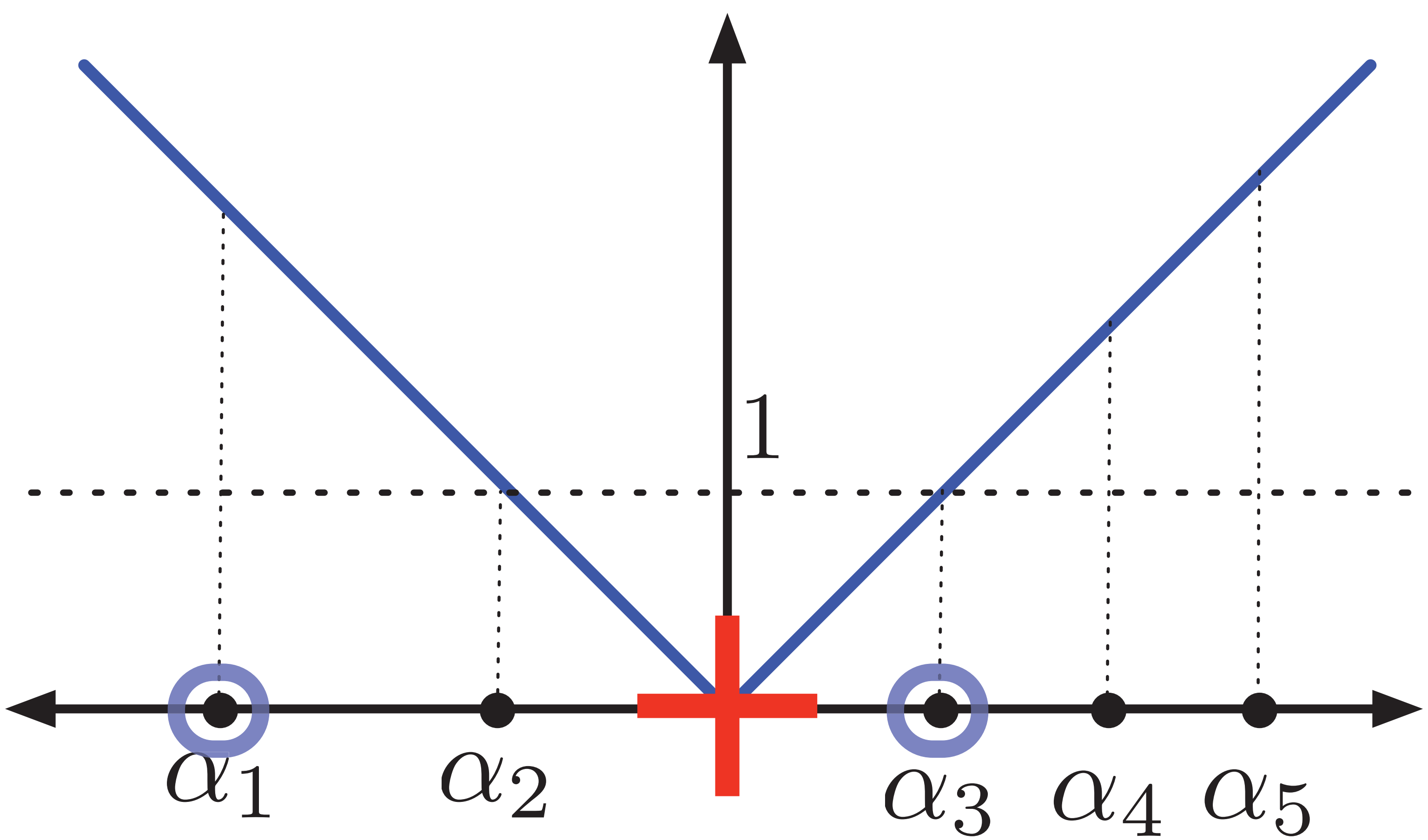
$$\mathrm{IP}_{\mathrm{opt}} = 1, \ \mathrm{LP}_{\mathrm{opt}} = 0$$

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



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

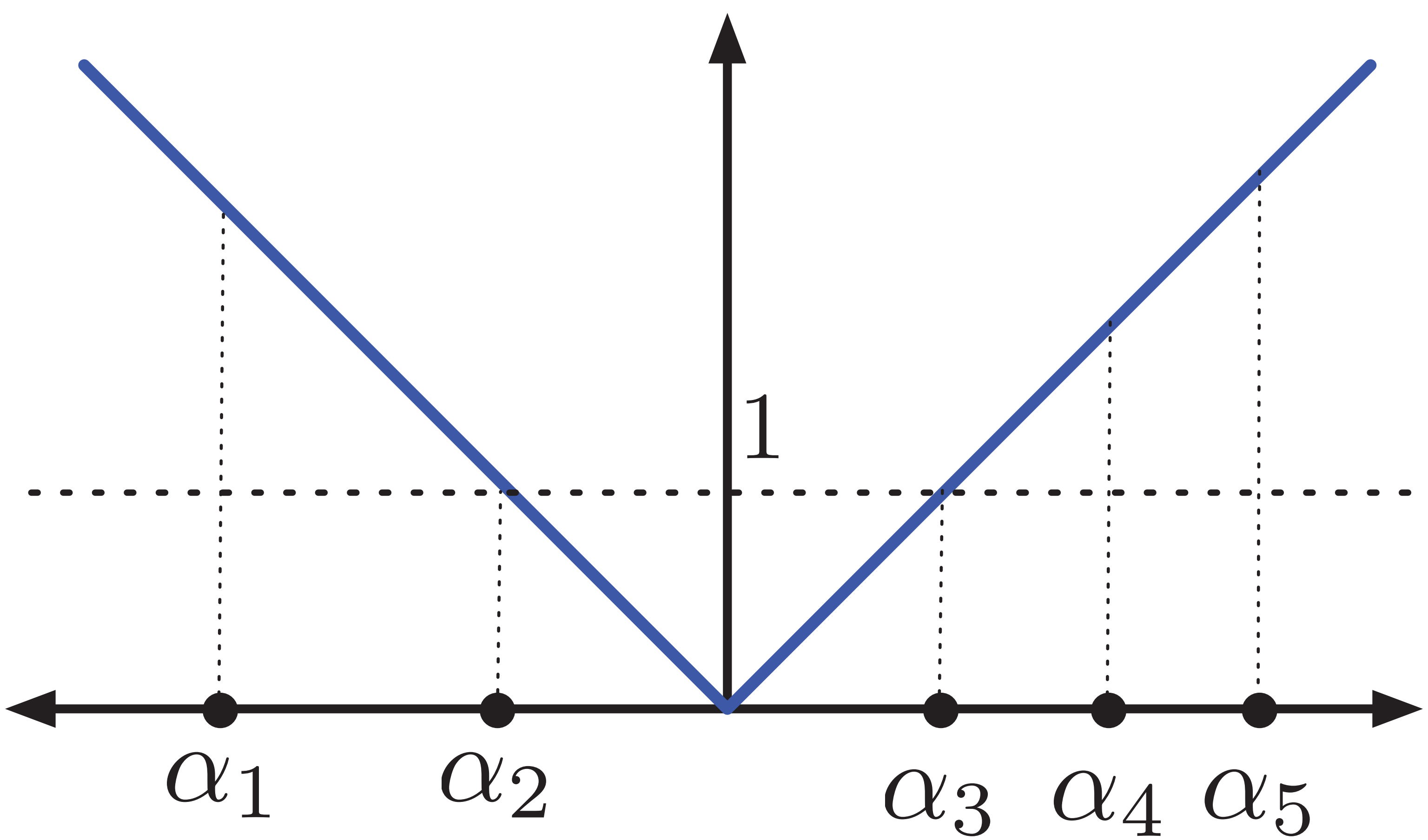
$\mathrm{IP}_{\mathrm{opt}} = 1,\ \mathrm{LP}_{\mathrm{opt}} = 0$

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

Branch on λ_2
- $\lambda_2 = 1 \rightarrow$ Feasible with $|x| = 1$
- $\lambda_2 = 0 \rightarrow$ Best Bound $= 0$

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

Formulating Discrete Alternatives



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

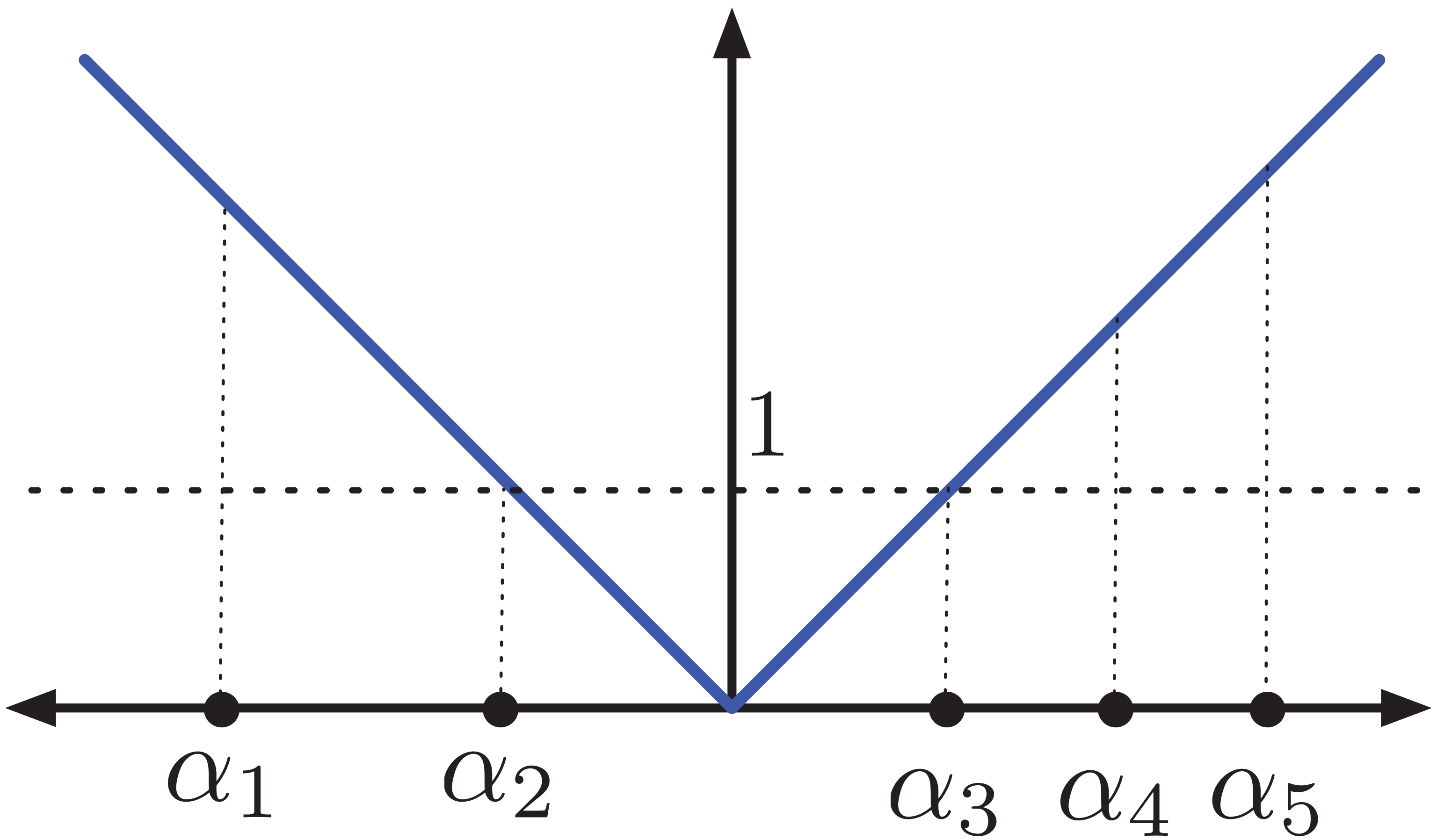
$$\mathrm{IP}_{\mathrm{opt}} = 1, \ \mathrm{LP}_{\mathrm{opt}} = 0$$

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

Worst case: $n/2$ branches to solve

(i.e. $2^{n/2}$ B-and-B nodes!).

Formulating Discrete Alternatives

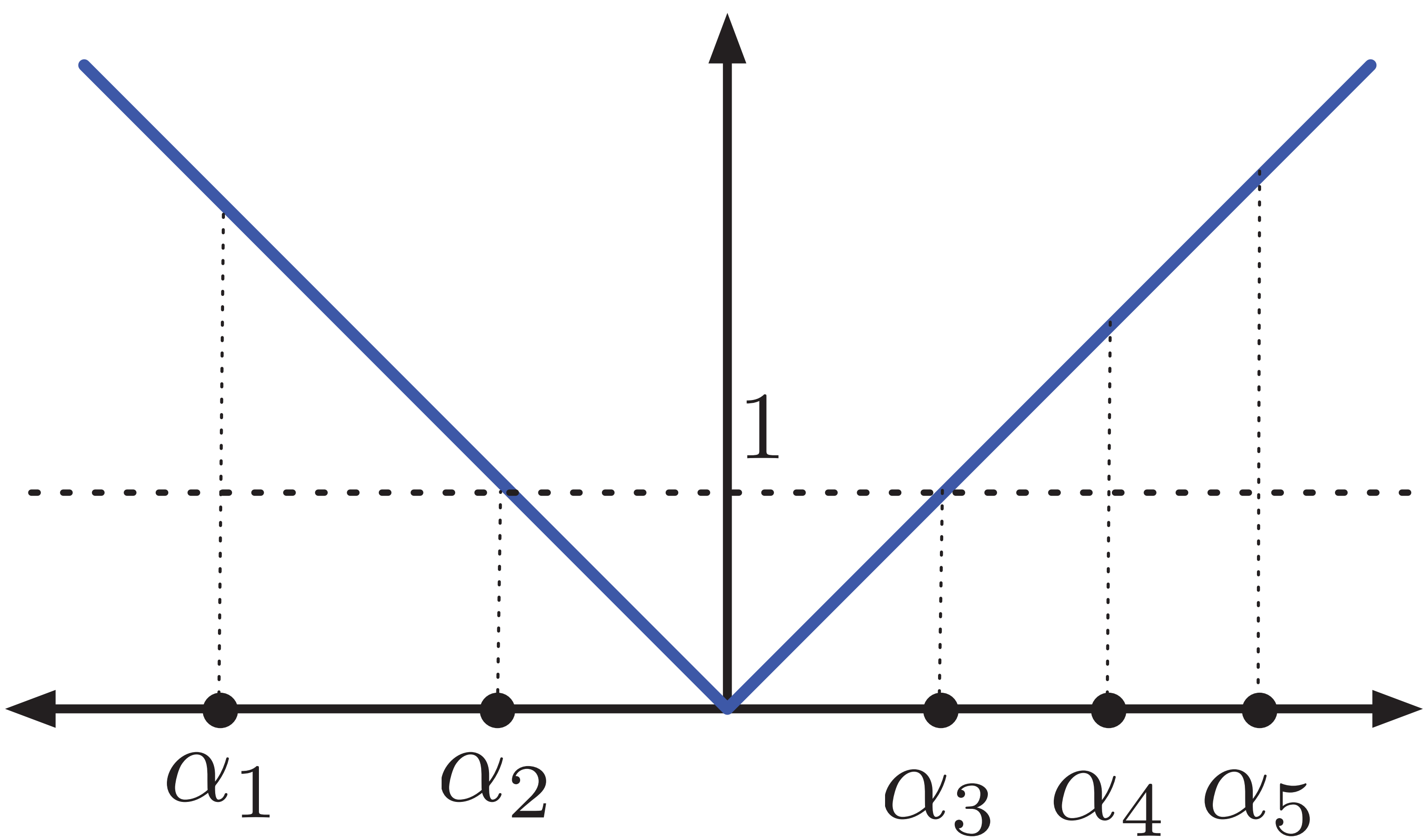


$$
\begin{aligned}
\min \quad & |x| \\
s.t. \quad & \\
& \sum_{i=1}^{n} \lambda_i \alpha_i = x \\
& \sum_{i=1}^{n} \lambda_i = 1 \\
& \lambda \in \{0,1\}^n
\end{aligned}
$$

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

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

Formulating Discrete Alternatives



$$\min \quad |x|$$

$$s.t.$$

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

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

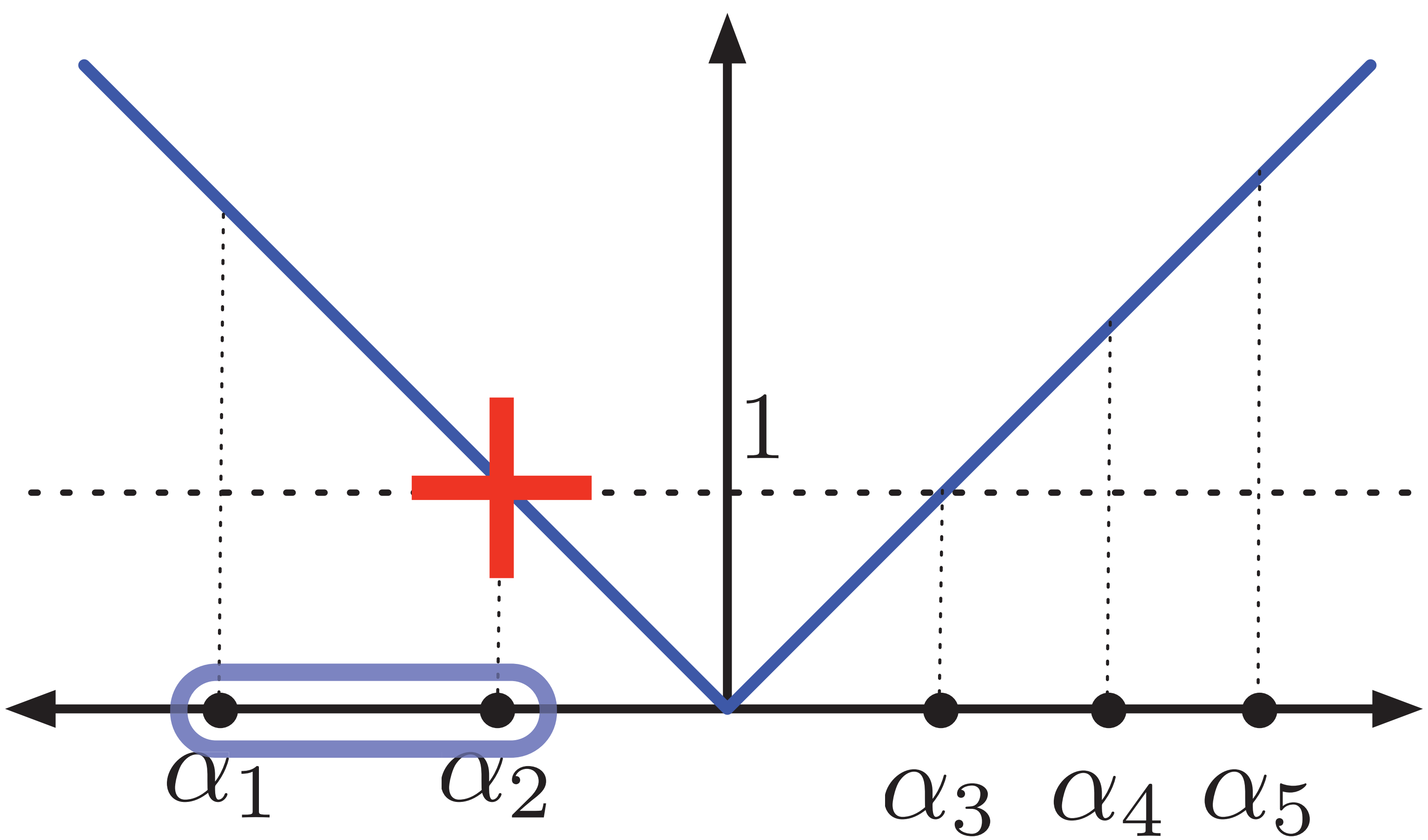
$$\lambda \in \{0,1\}^n$$

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

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

Branch on $\lambda_1 + \lambda_2$

Formulating Discrete Alternatives



$$\min \quad |x|$$

$$s.t.$$

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

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

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

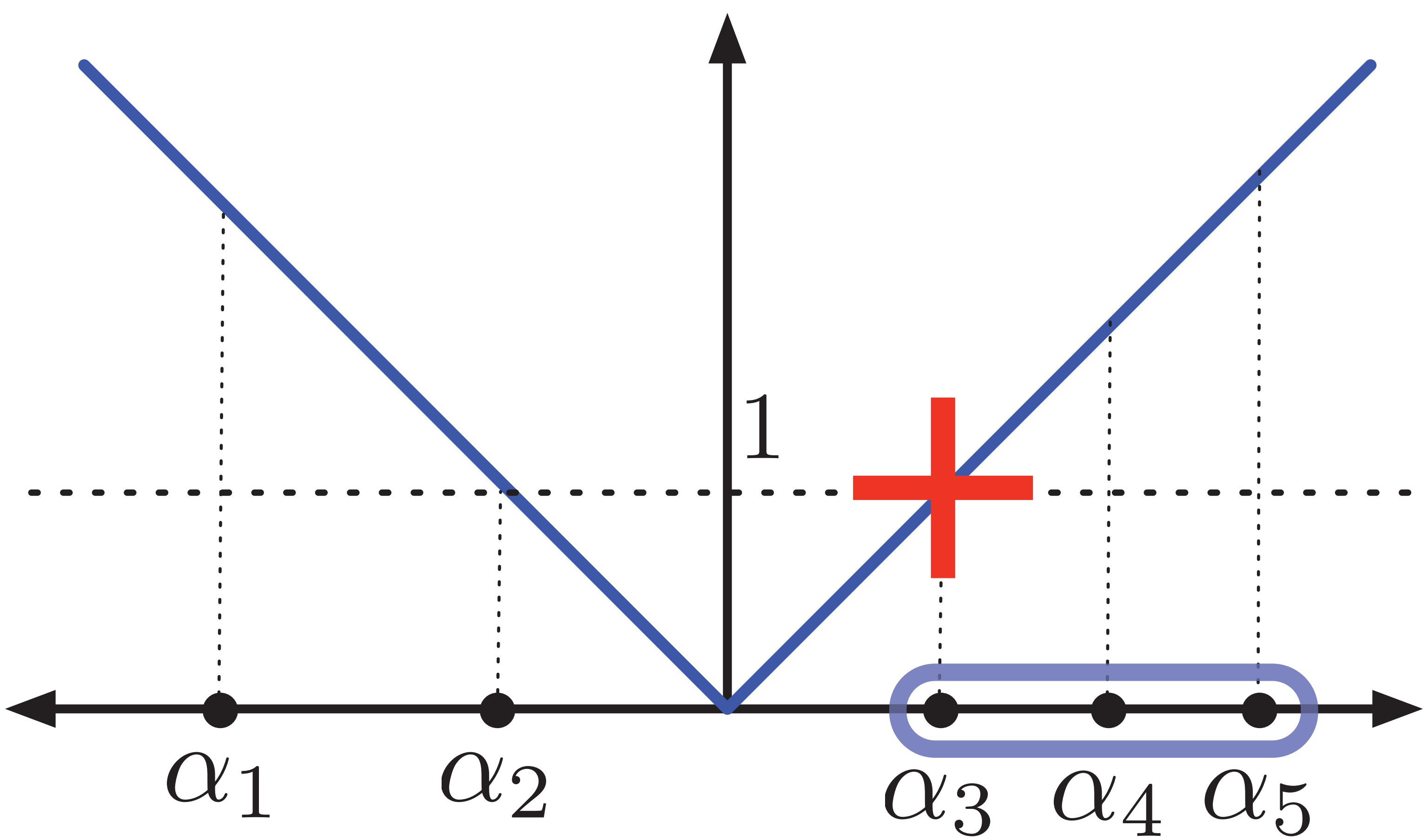
$$\mathrm{IP_{opt}} = 1,\ \mathrm{LP_{opt}} = 0$$

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

Branch on $\lambda_1 + \lambda_2$ →

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

Formulating Discrete Alternatives



$$\min \quad |x|$$

$$s.t.$$

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

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

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

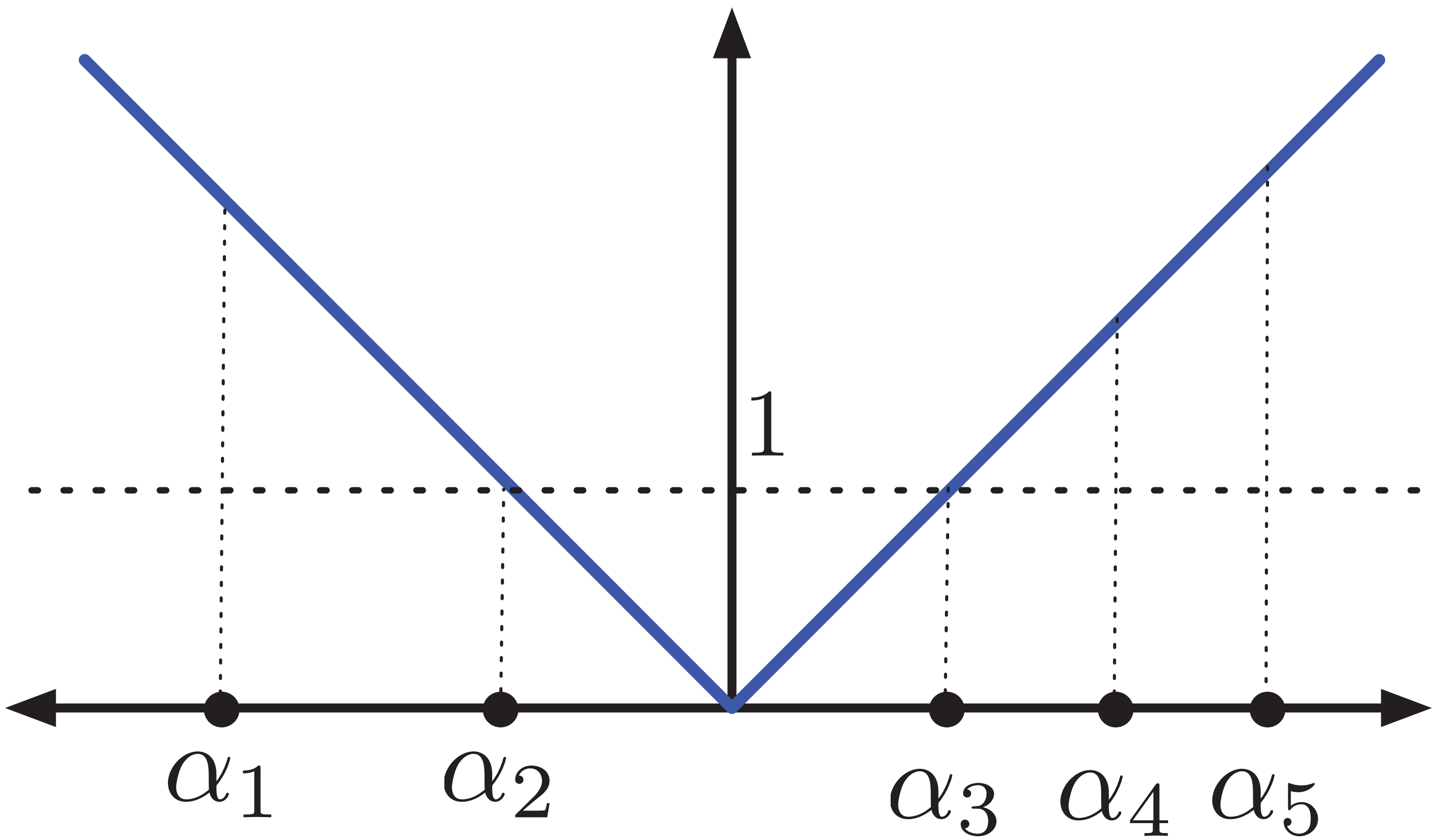
$$\mathrm{IP}_{\mathrm{opt}} = 1,\ \mathrm{LP}_{\mathrm{opt}} = 0$$

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



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

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

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: $\sum_{i=1}^{t} \lambda_i = 1 \quad \text{or} \quad \sum_{i=1}^{t} \lambda_i = 0$

$$\Updownarrow$$

$$\lambda_i = 0 \quad \forall i > t \quad \text{or} \quad \lambda_i = 0 \quad \forall i \leq t$$

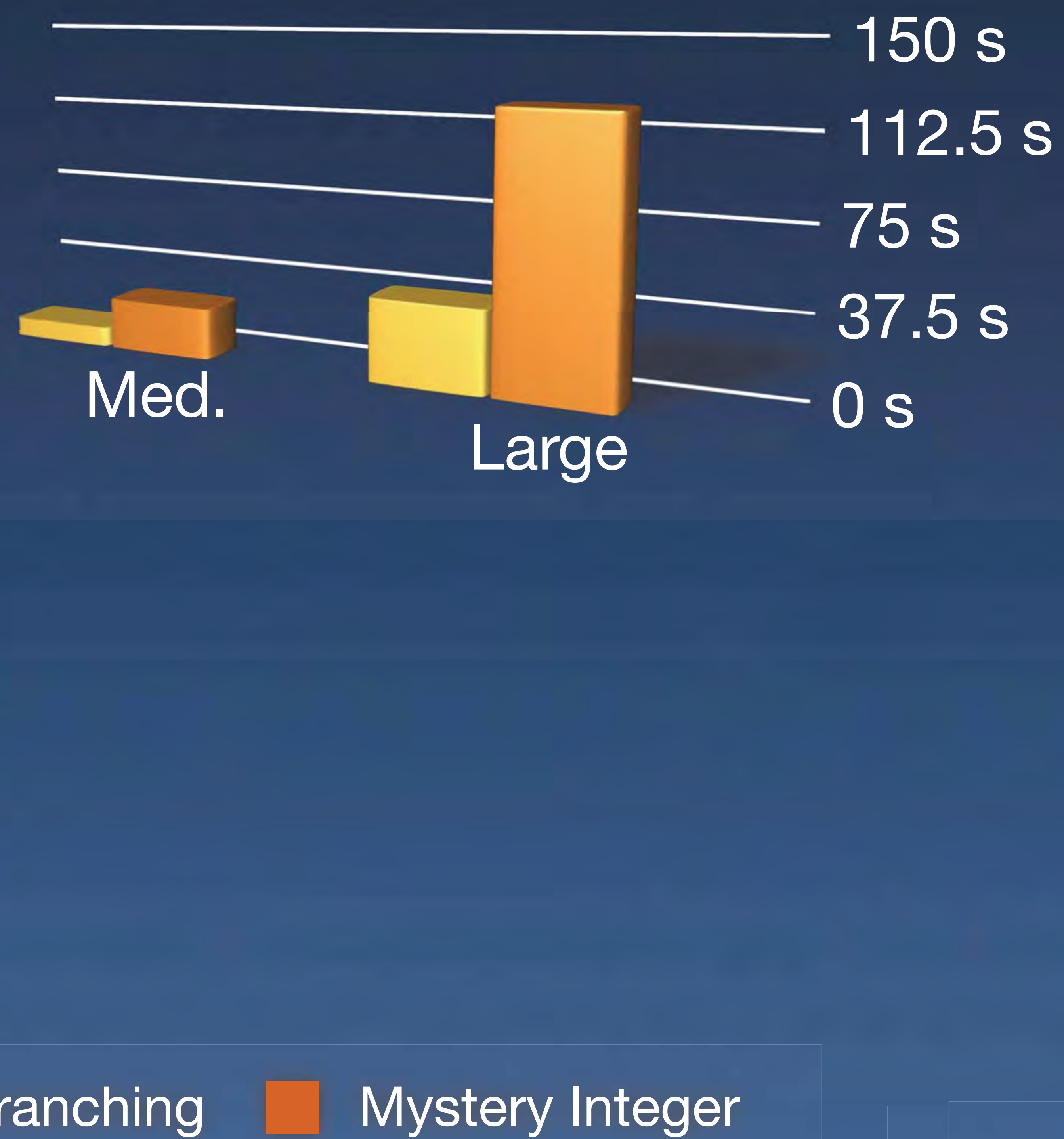
- Problem: Need to re-implement advanced branching selection (e.g. pseudocost).

Binary v/s Specialized Branching

Weak Integer SOS2 Branching Mystery Integer

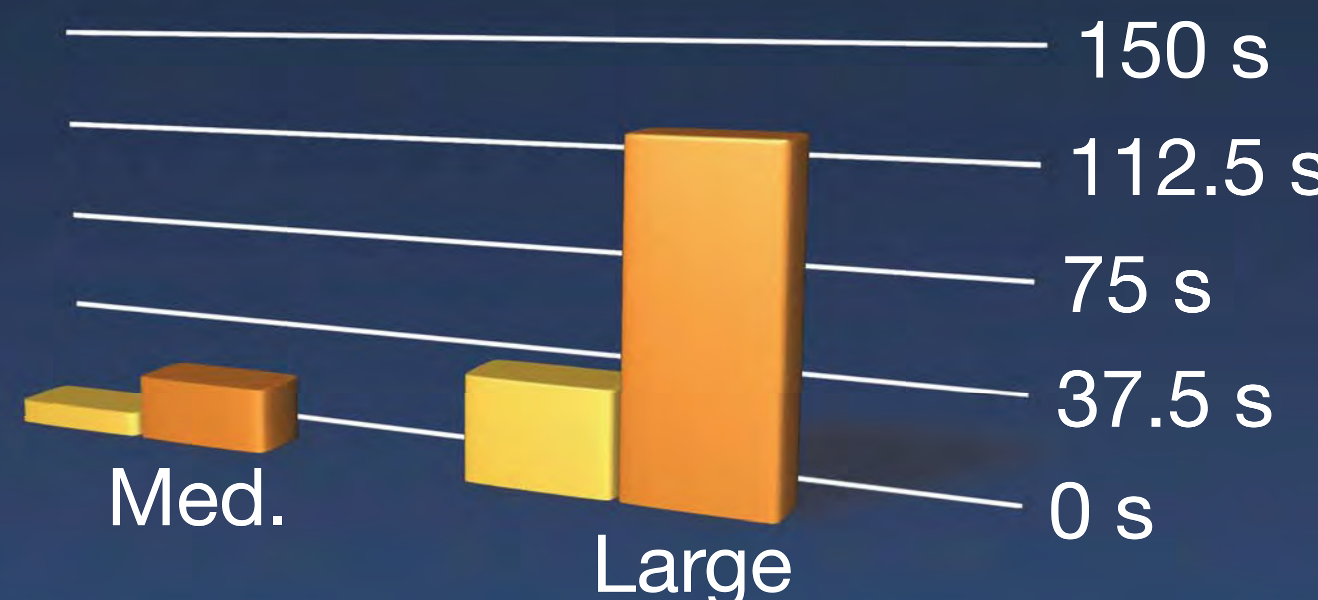
Binary v/s Specialized Branching

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

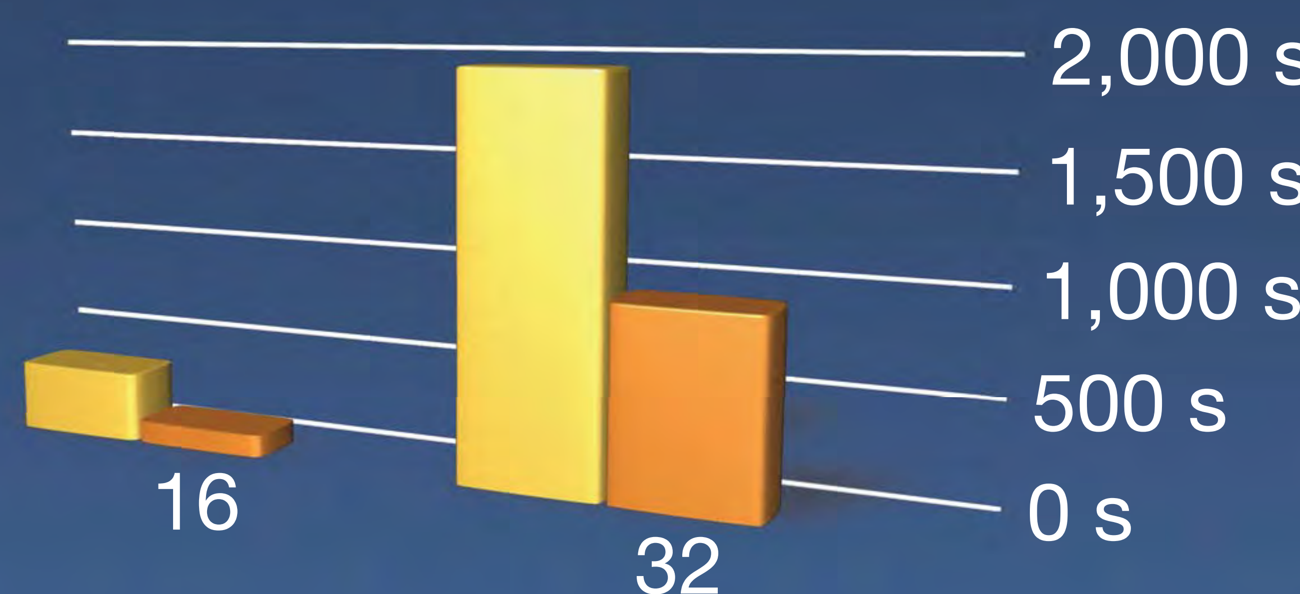


Binary v/s Specialized Branching

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



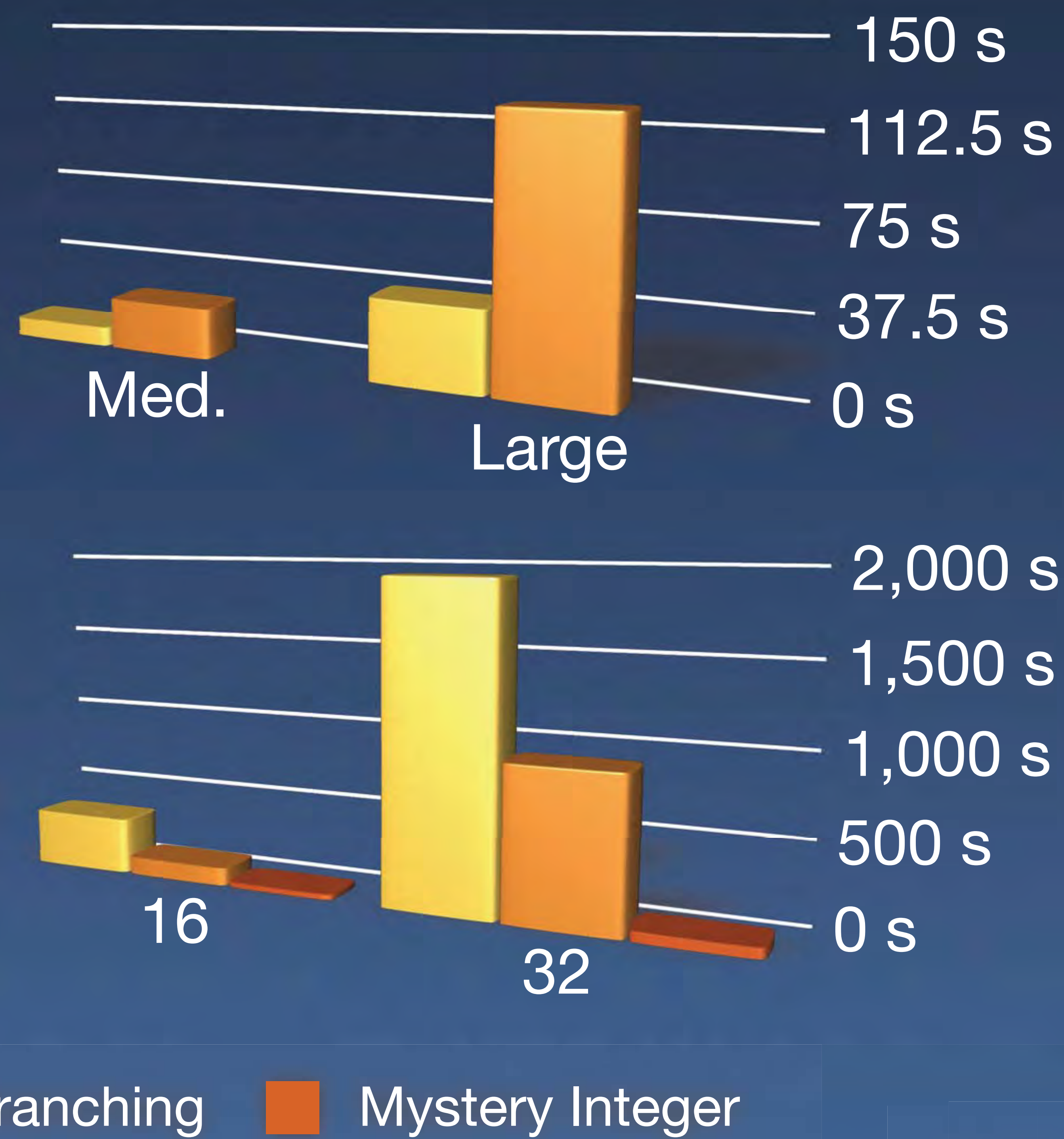
- CPLEX 11: Improved SOS2 branching implementation
  (graph from Nemhauser, Ahmed and V. '10)



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)



Formulation Step 1: Encoding Alternatives

Formulation for Discrete Alternatives

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

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

$$\lambda \in \mathbb{R}^n_+$$

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

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

- 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

Unary Encoding

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \lambda = y, \qquad \begin{array}{c} \sum_{i=1}^{8} \lambda_i = 1, \\ \lambda \in \mathbb{R}^8,\ y \in \{0,1\}^8 \end{array}$$

$$\Updownarrow$$

$$\lambda_i = y_i$$

Binary Encoding

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \lambda = y, \qquad \sum_{i=1}^{8} \lambda_i = 1, \quad \lambda \in \mathbb{R}^8,\ y \in \{0,1\}^3$$

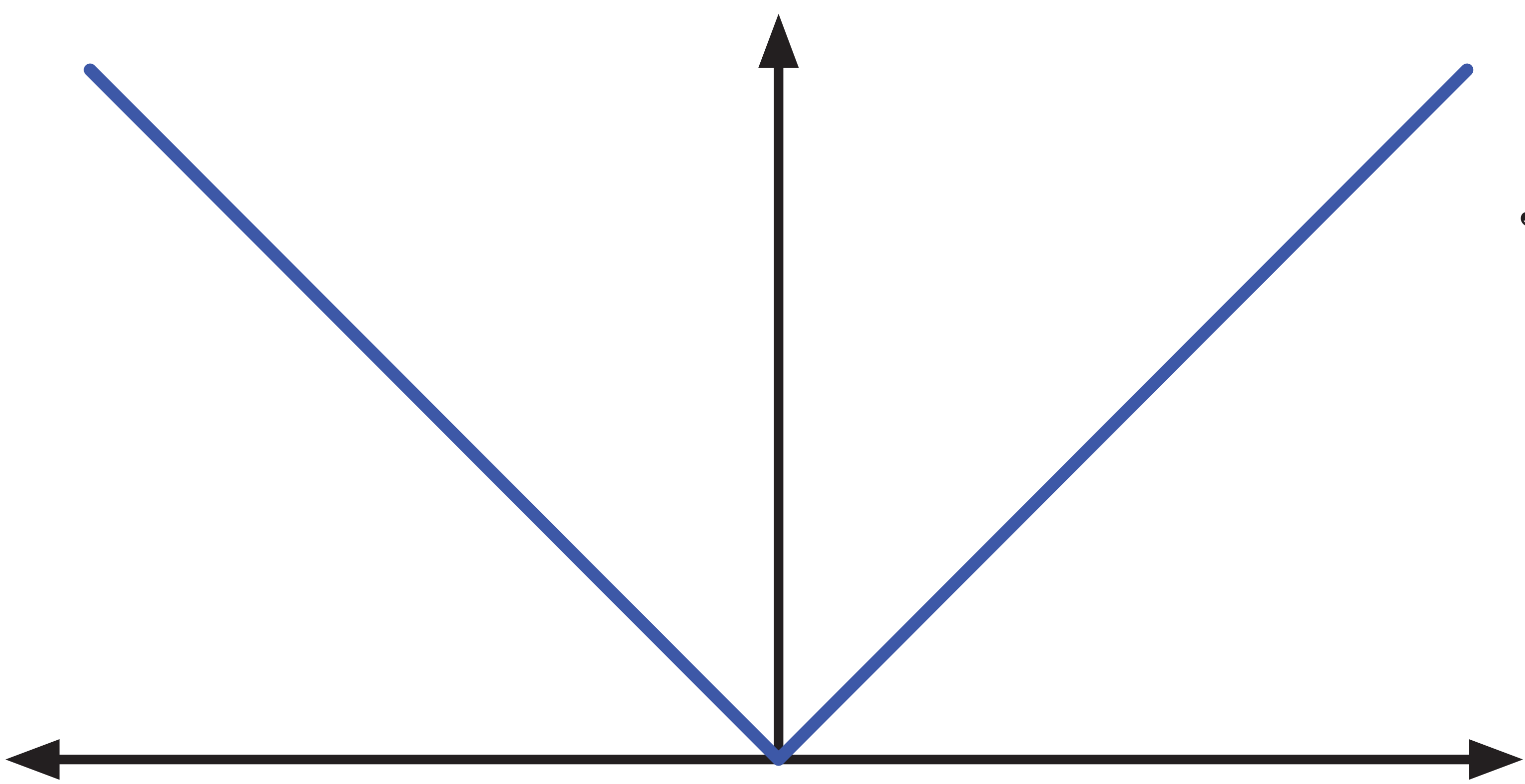
Incremental Encoding

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \lambda = y, \qquad \begin{array}{c} \sum_{i=1}^{8} \lambda_i = 1, \\ \lambda \in \mathbb{R}^8,\ y \in \{0,1\}^7 \end{array}$$

$$\Downarrow$$

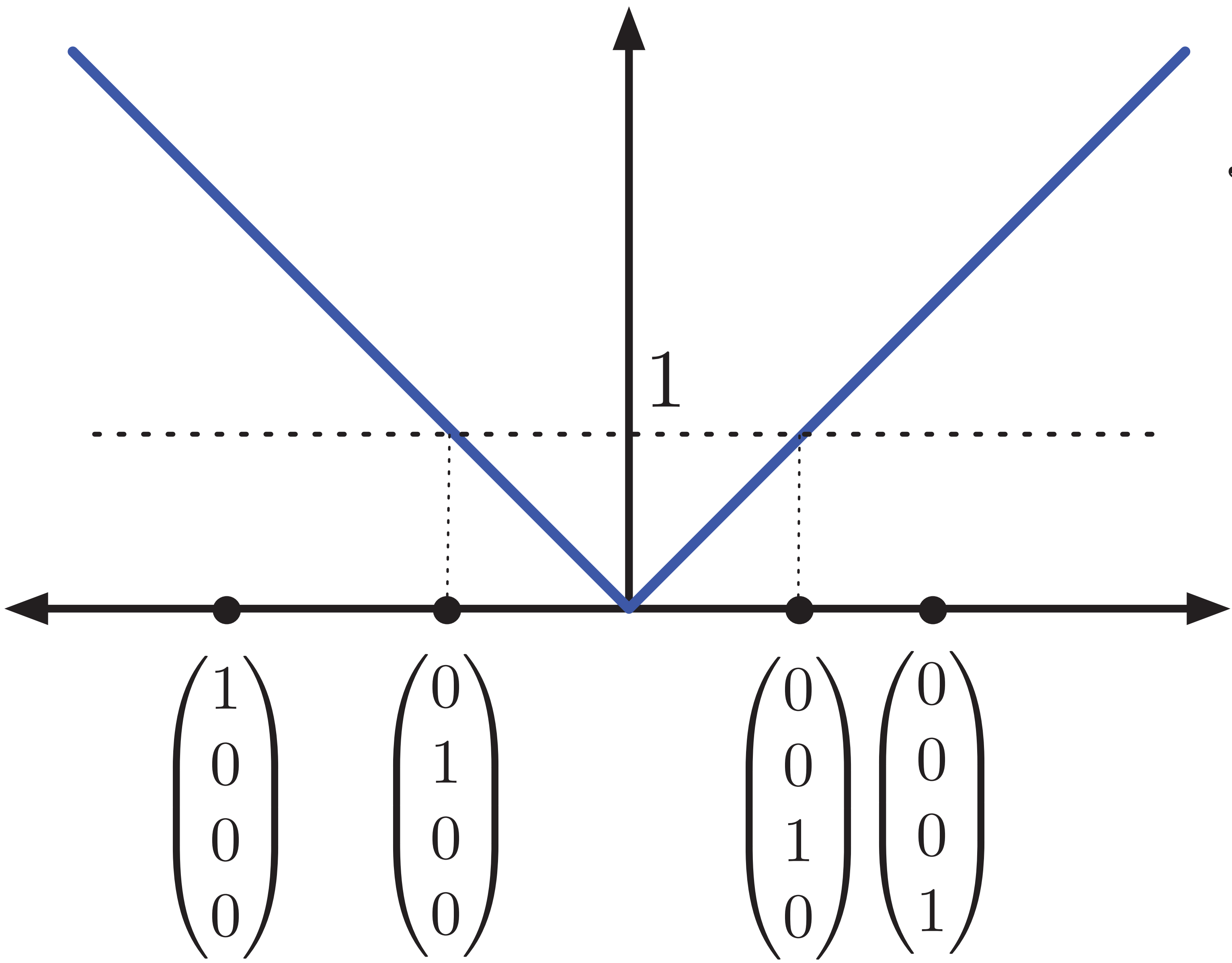
$$y_1 \geq y_2 \geq \cdots \geq y_7$$

Example: Unary Encoding



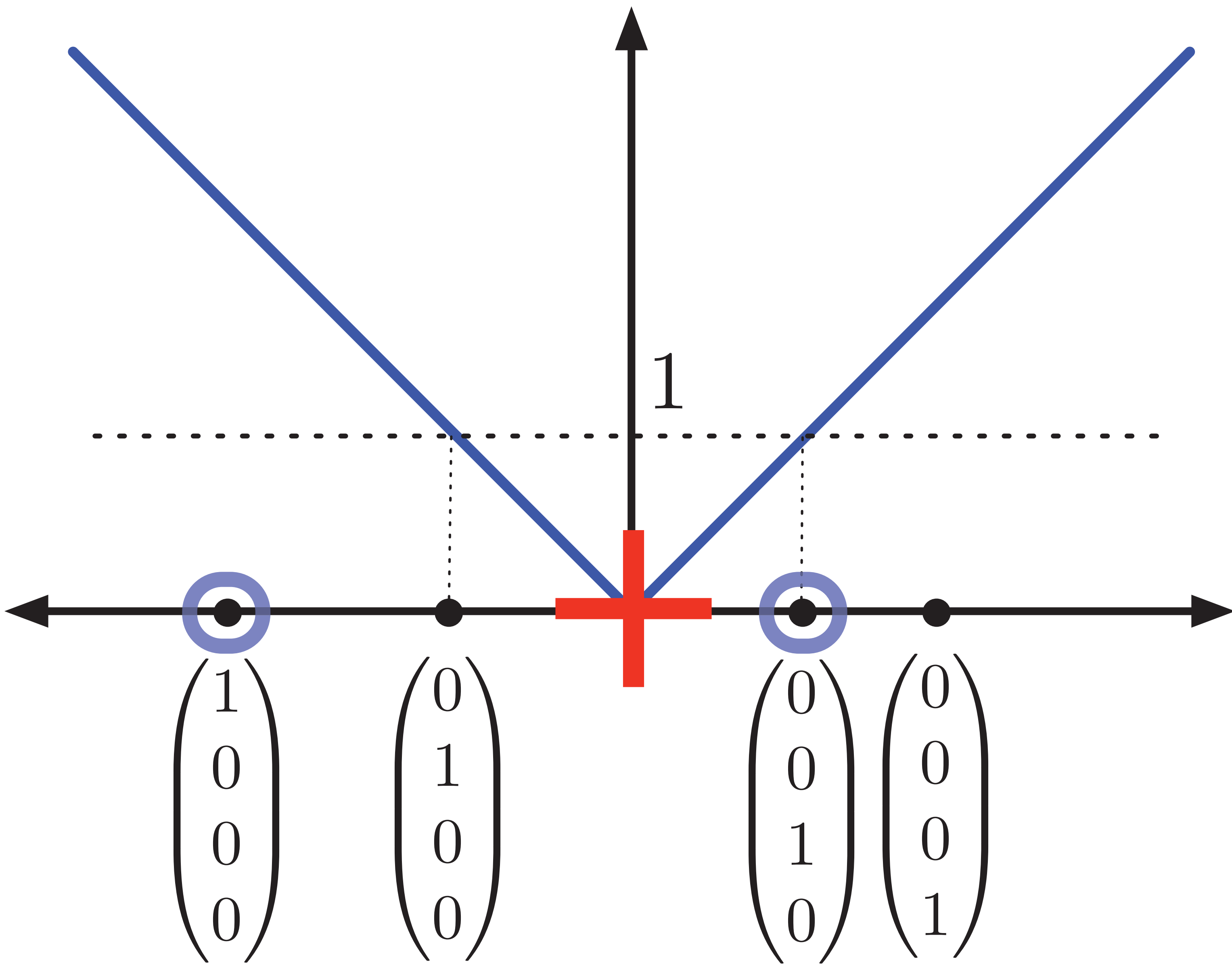
$$\begin{aligned}
\min \quad & |x| \\
s.t. \quad & \\
& \sum_{i=1}^{n} \lambda_i \alpha_i = x \\
& \sum_{i=1}^{n} \lambda_i = 1 \\
& \sum_{i=1}^{n} b^i \lambda_i = y \\
& \lambda \in \mathbb{R}_+^n \\
& y \in \{0,1\}^m
\end{aligned}$$

Example: Unary Encoding



$$
\begin{aligned}
\min \quad & |x| \\
s.t. \quad & \textstyle\sum_{i=1}^{n} \lambda_i \alpha_i = x \\
& \textstyle\sum_{i=1}^{n} \lambda_i = 1 \\
& \textstyle\sum_{i=1}^{n} b^i \lambda_i = y \\
& \lambda \in \mathbb{R}^n_+ \\
& y \in \{0,1\}^m
\end{aligned}
$$

Example: Unary Encoding



$$\min \quad |x|$$

$s.t.$

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

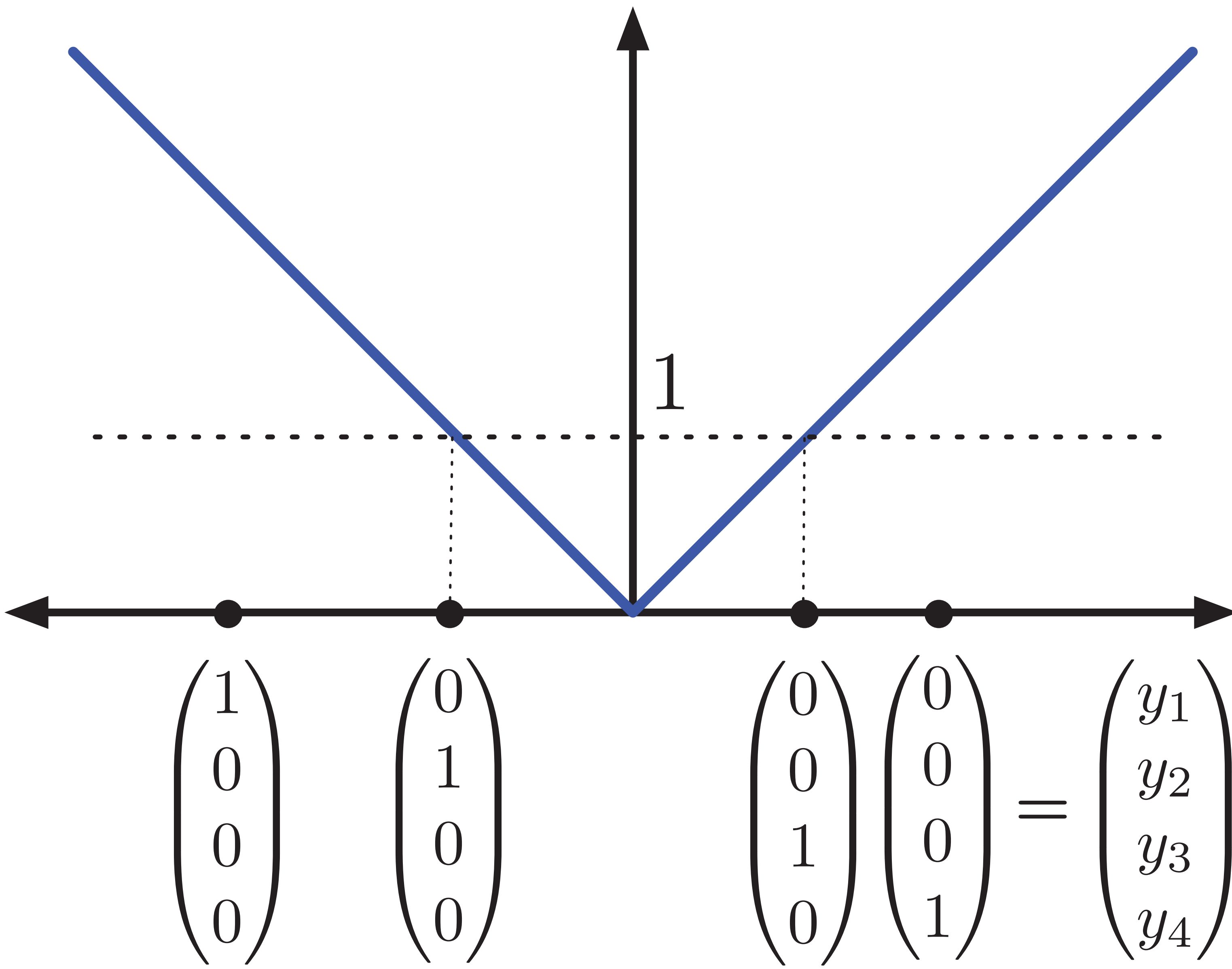
$$\sum_{i=1}^{n} \lambda_i = 1$$

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

$$\lambda \in \mathbb{R}^n_+$$

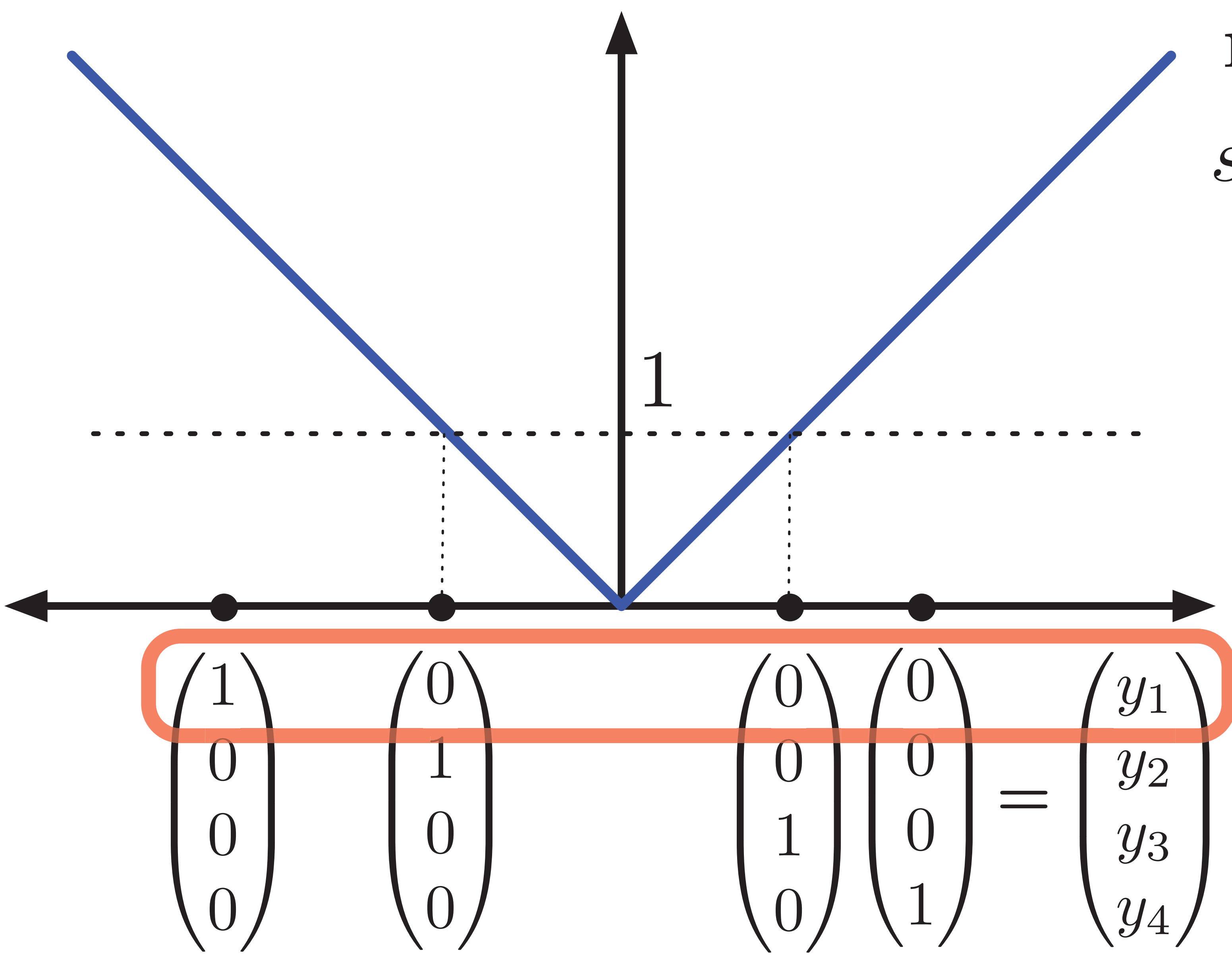
$$y \in \{0,1\}^m$$

Example: Unary Encoding



$$
\begin{aligned}
\min \quad & |x| \\
s.t. \quad & \\
& \sum_{i=1}^{n} \lambda_i \alpha_i = x \\
& \sum_{i=1}^{n} \lambda_i = 1 \\
& \sum_{i=1}^{n} b^i \lambda_i = y \\
& \lambda \in \mathbb{R}^n_+ \\
& y \in \{0,1\}^m
\end{aligned}
$$

Example: Unary Encoding



$$\min \quad |x|$$

$$s.t.$$

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

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

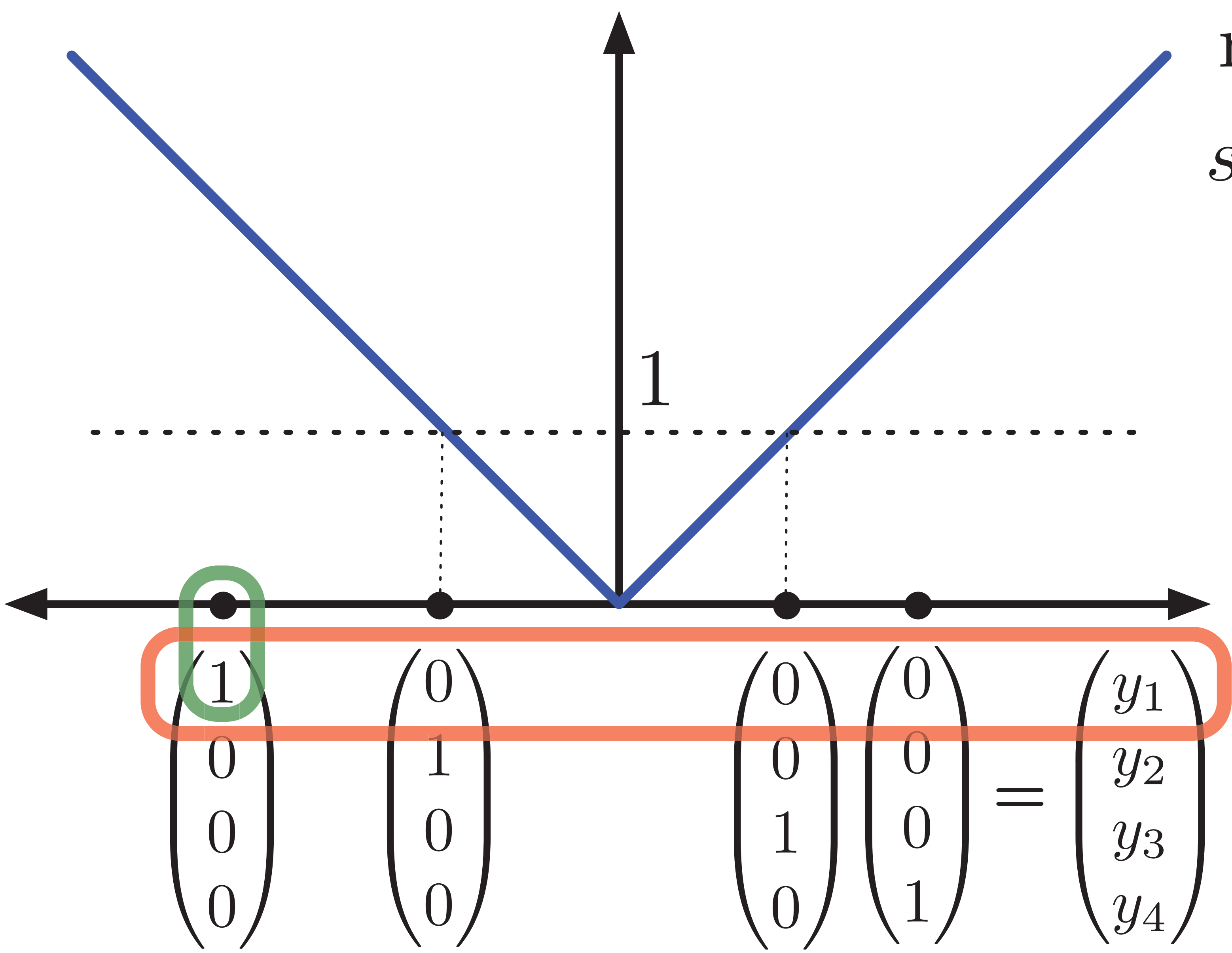
$$\sum_{i=1}^{n} b^i \lambda_i = y$$

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

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

$$y_1 =$$

Example: Unary Encoding



$$\min \quad |x|$$

$$s.t.$$

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

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

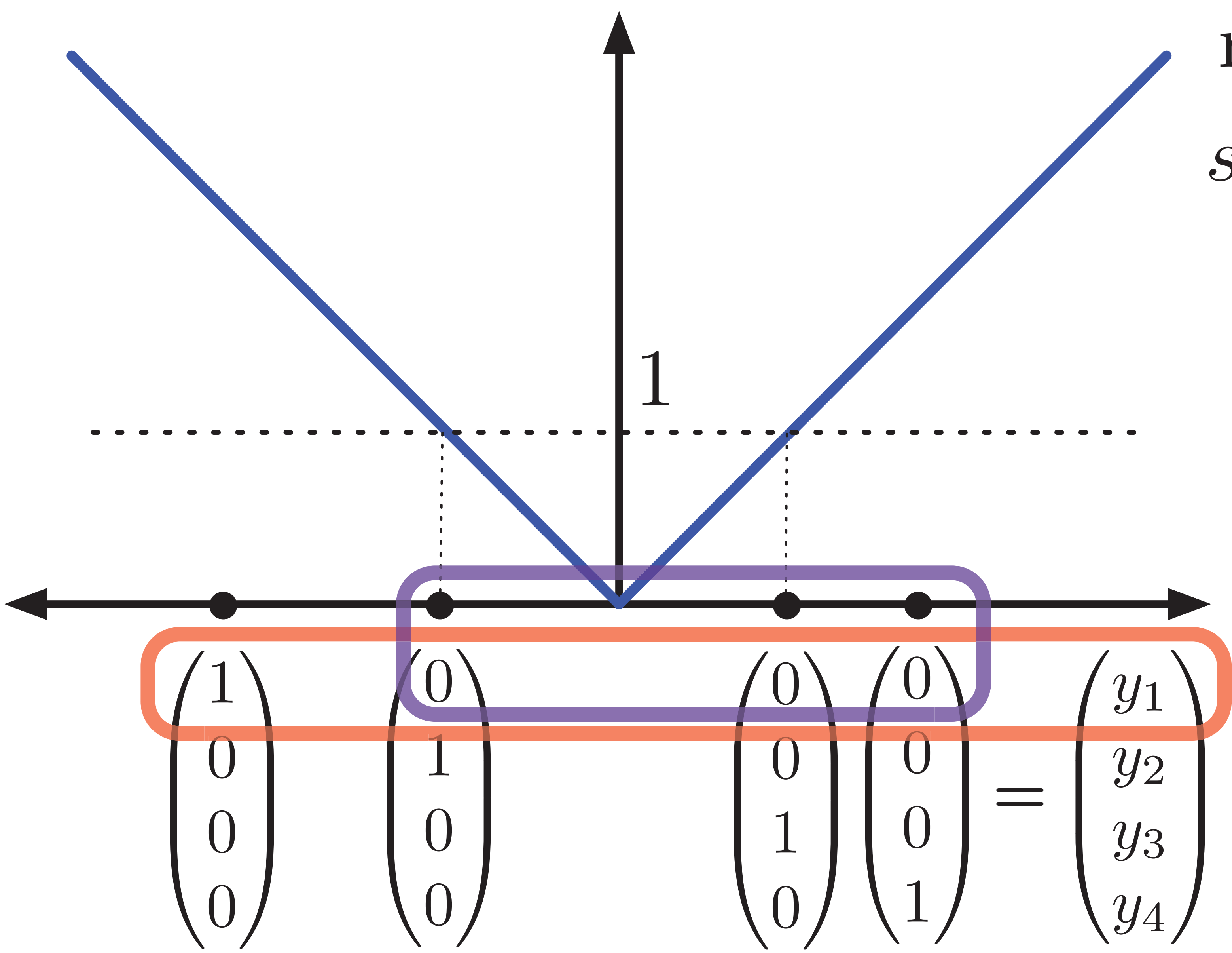
$$\sum_{i=1}^{n} b^i \lambda_i = y$$

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

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

$$y_1 = 1$$

Example: Unary Encoding



$$\min \quad |x|$$

$$s.t.$$

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

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

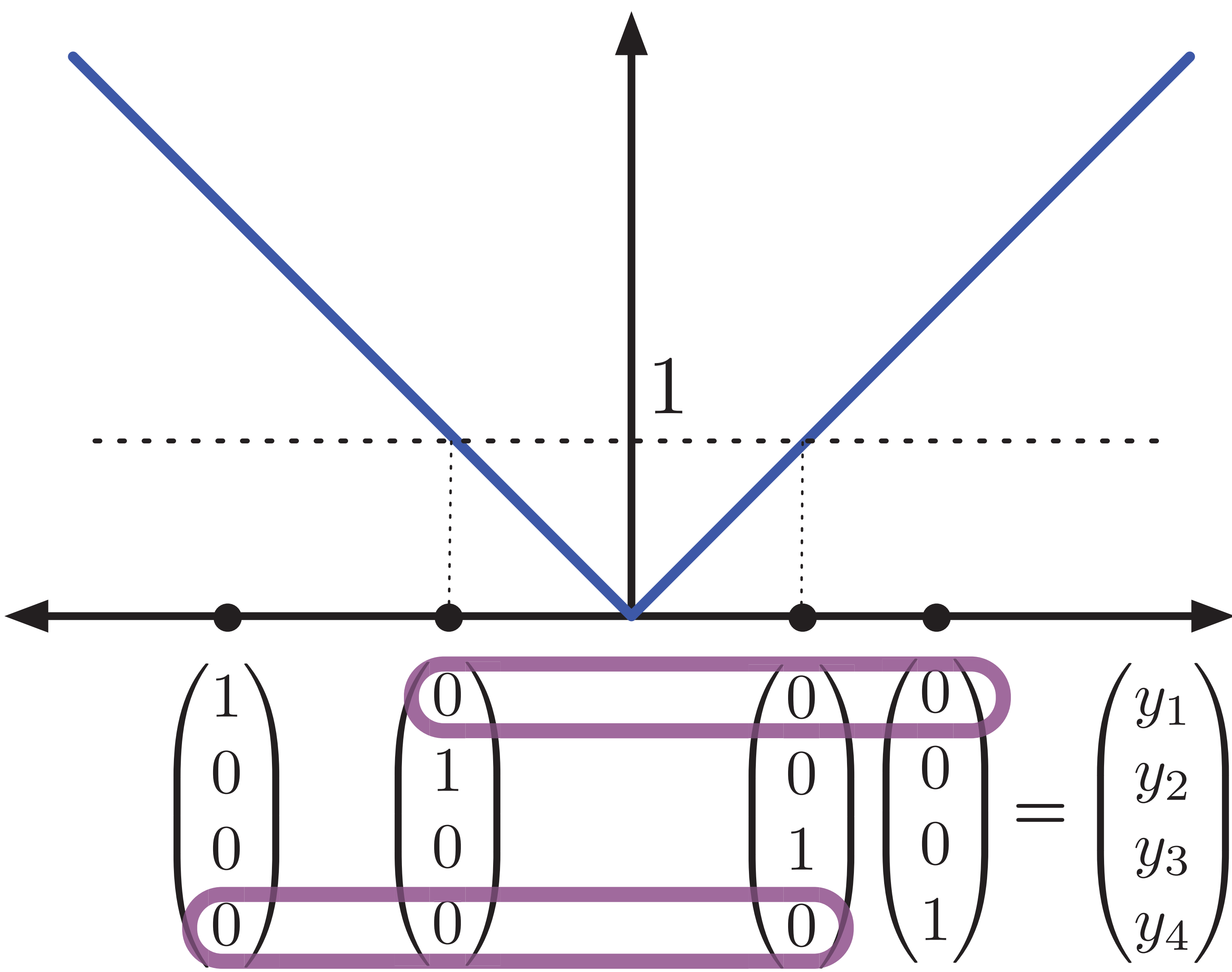
$$\sum_{i=1}^{n} b^i \lambda_i = y$$

$$\lambda \in \mathbb{R}^n_+$$

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

$$y_1 = 0$$

Example: Unary Encoding



$$\min \quad |x|$$

$$s.t.$$

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

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

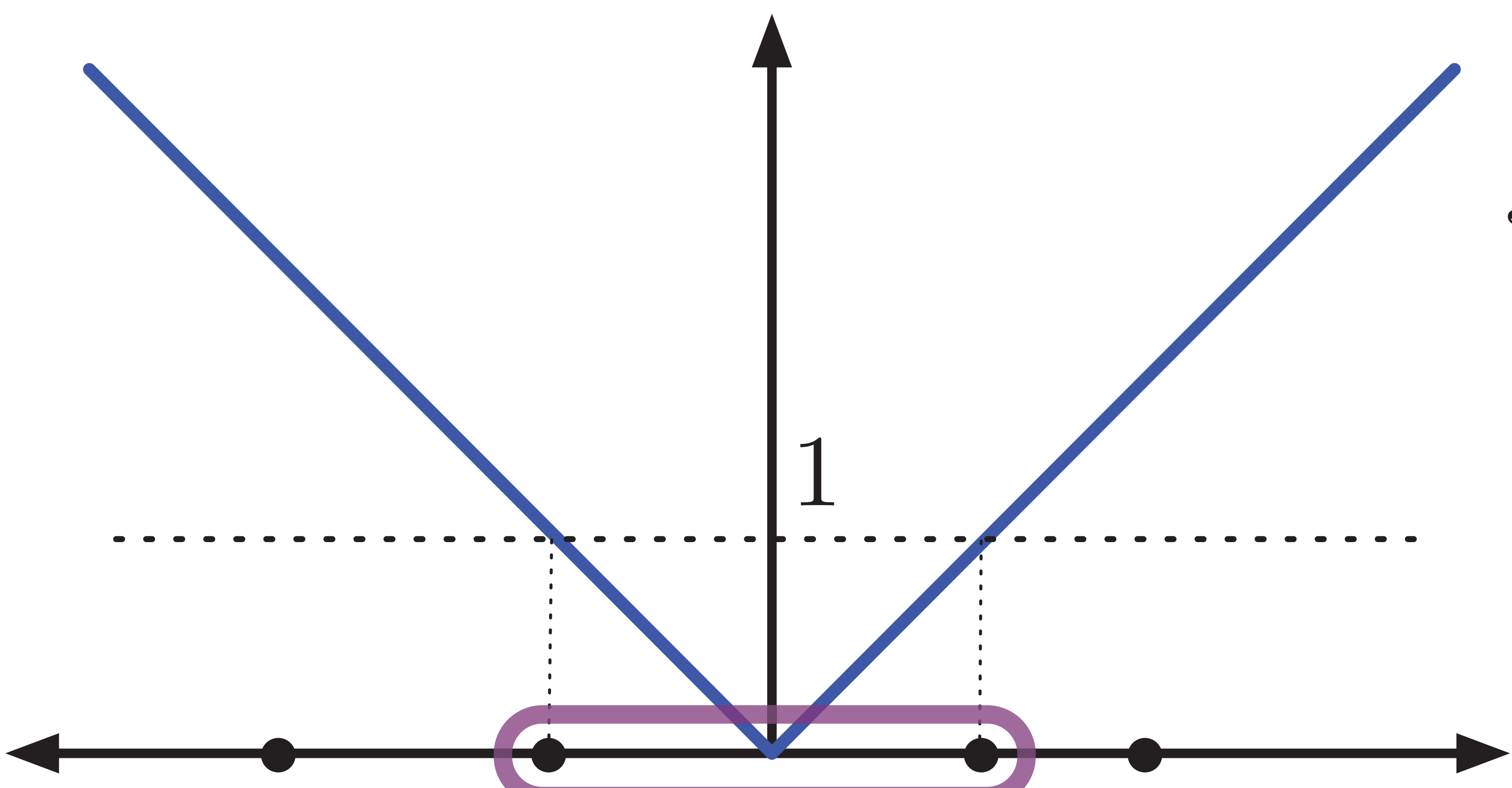
$$\sum_{i=1}^{n} b^i \lambda_i = y$$

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

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

$$y_1 = y_4 = 0$$

Example: Unary Encoding



$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \begin{pmatrix}y_1\\y_2\\y_3\\y_4\end{pmatrix}$$

$$\min \quad |x|$$

$$s.t.$$

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

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

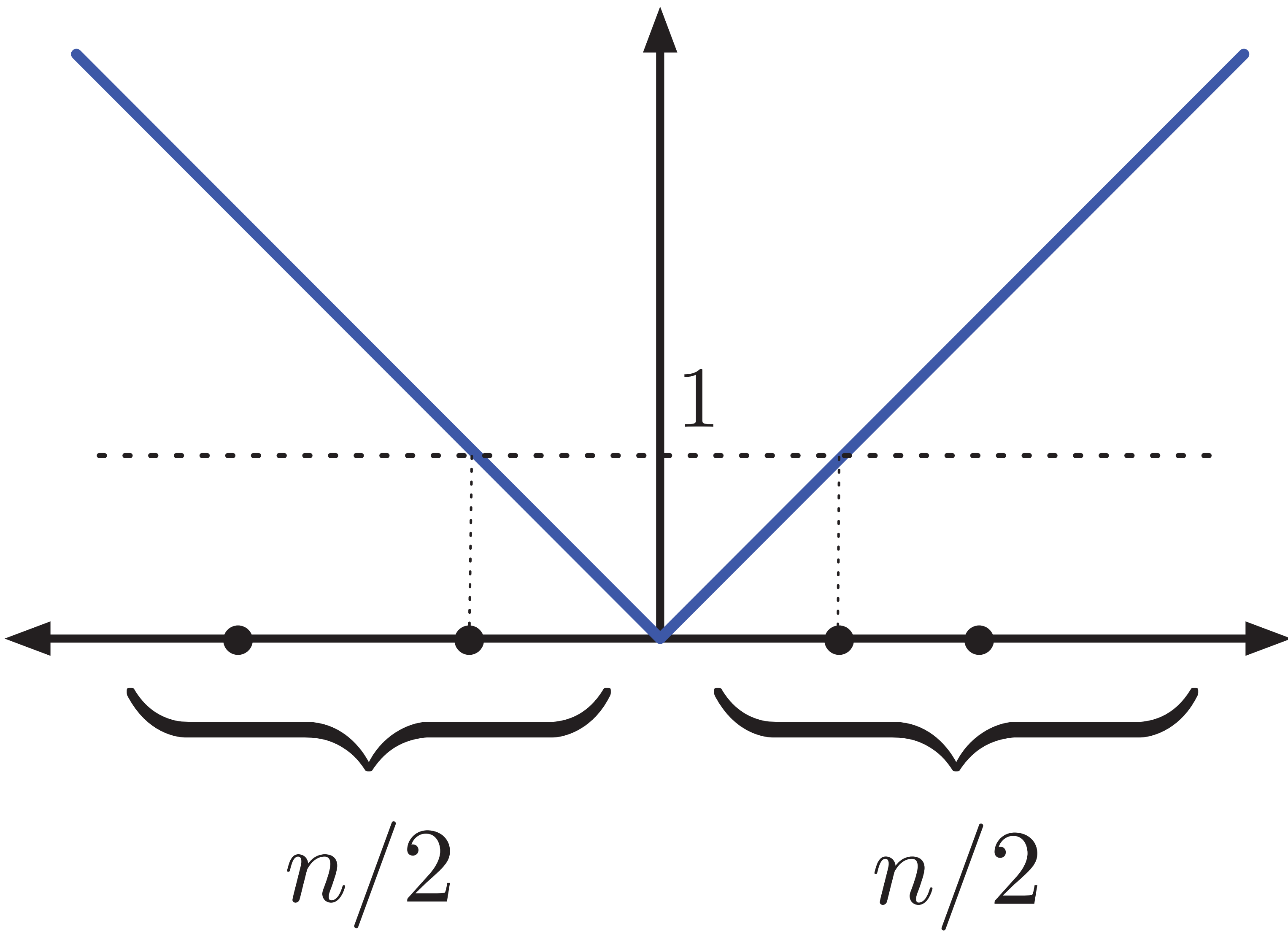
$$\sum_{i=1}^{n} b^i \lambda_i = y$$

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

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

$$y_1 = y_4 = 0$$

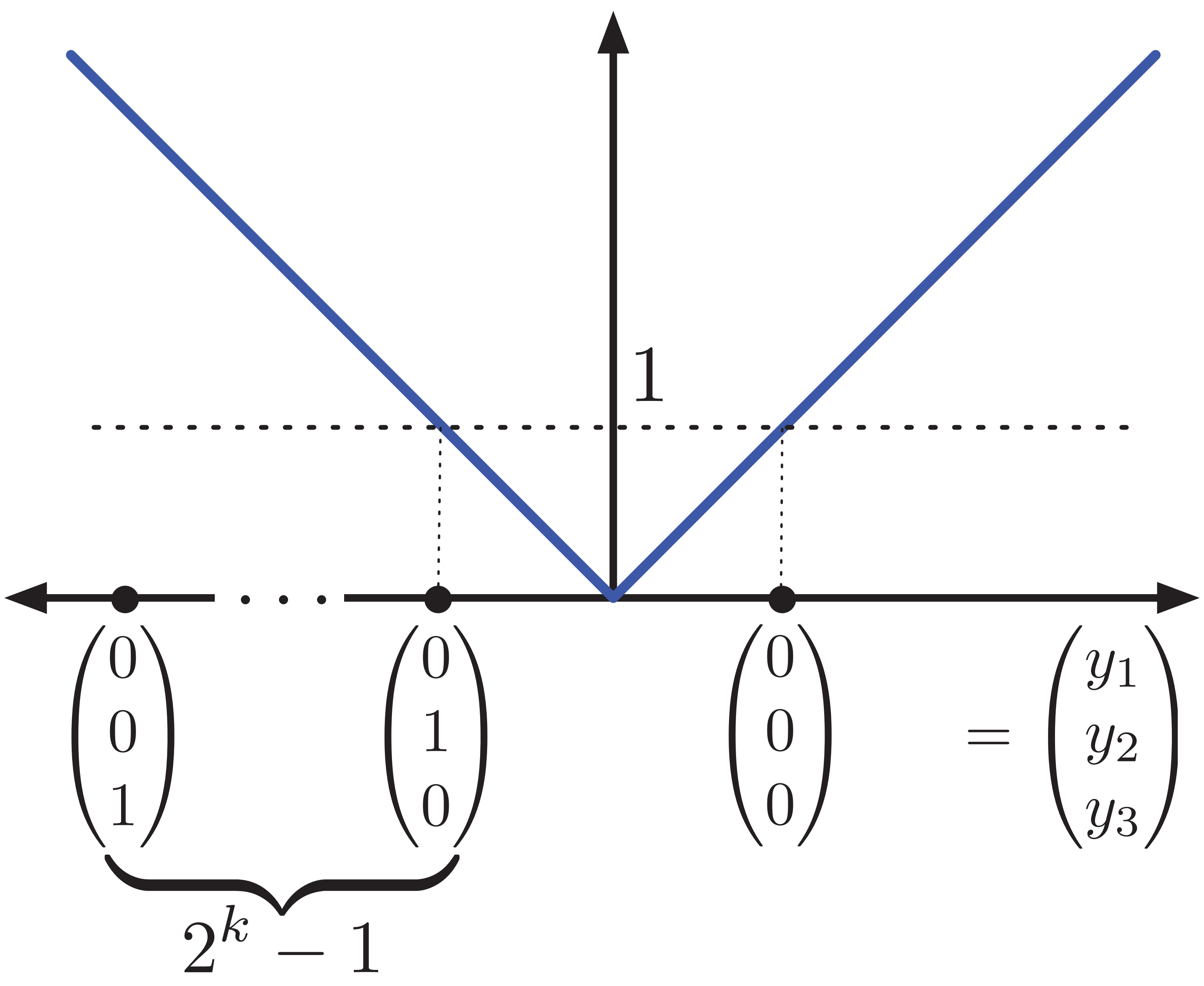
Example: Unary Encoding



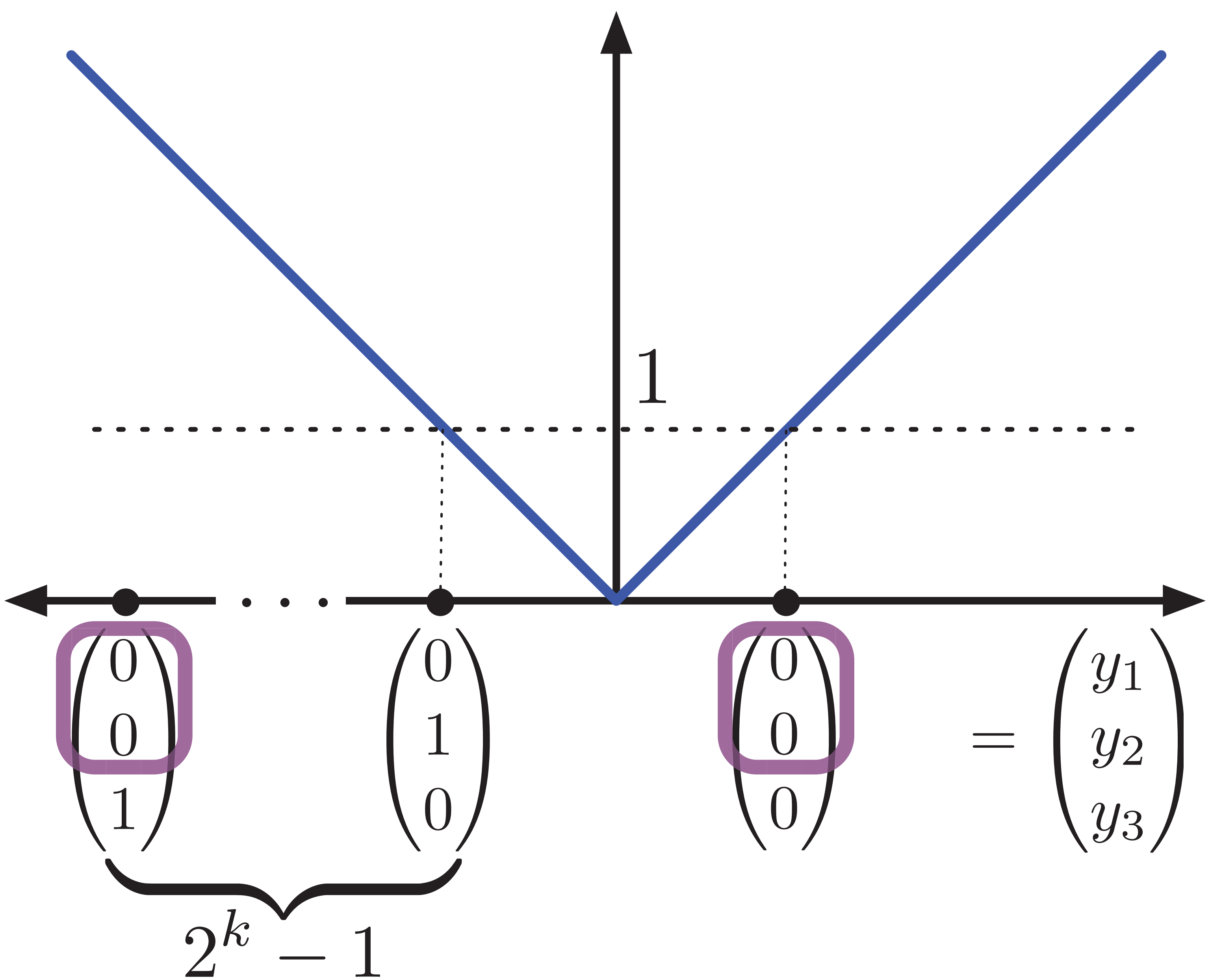
$$
\begin{aligned}
\min \quad & |x| \\
s.t. \quad & \\
& \sum_{i=1}^{n} \lambda_i \alpha_i = x \\
& \sum_{i=1}^{n} \lambda_i = 1 \\
& \sum_{i=1}^{n} b^i \lambda_i = y \\
& \lambda \in \mathbb{R}^n_+ \\
& y \in \{0,1\}^m
\end{aligned}
$$

Need $n/2$ branches

Example: Binary Encoding

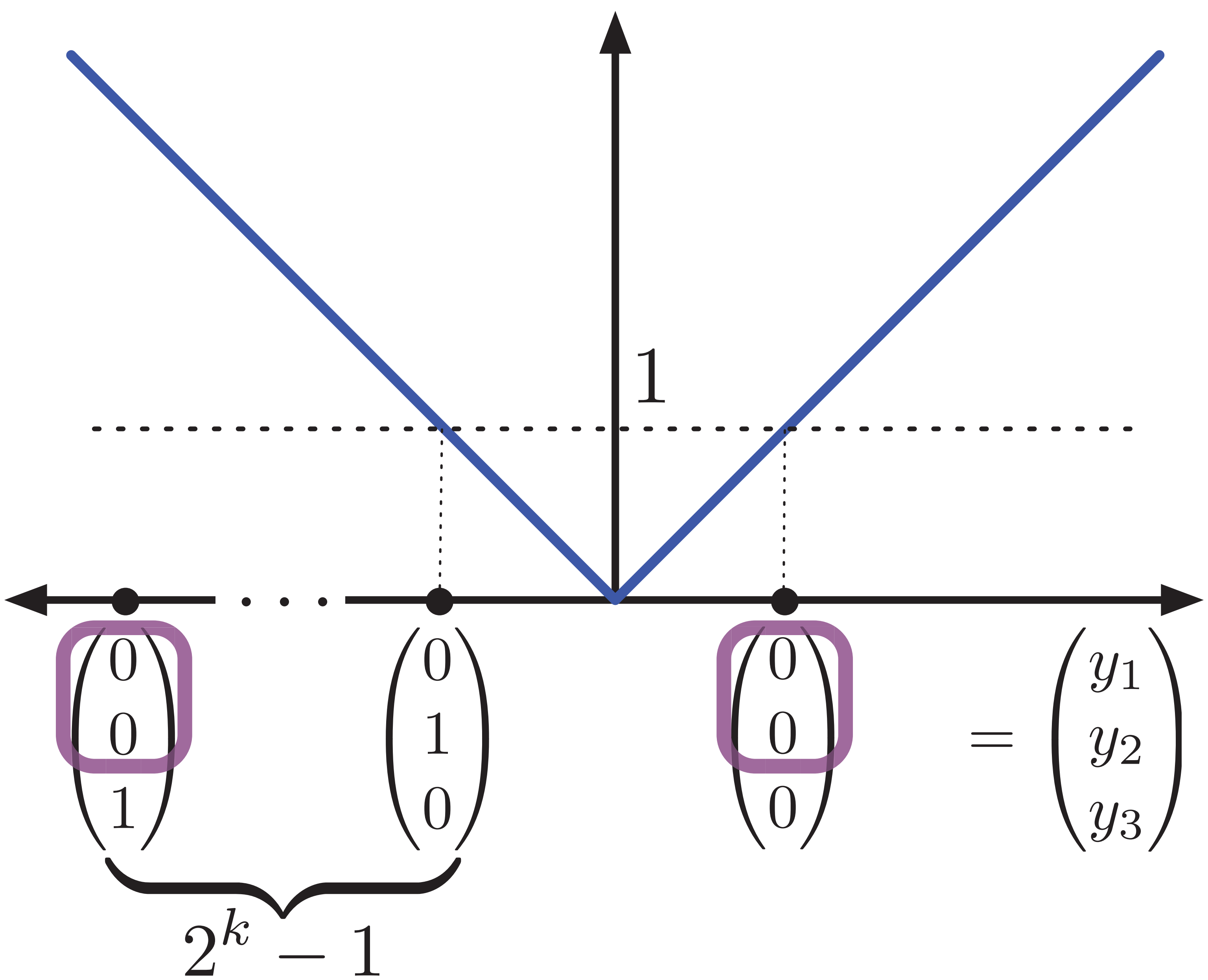


Example: Binary Encoding



$$y_1 = y_2 = 0$$

Example: Binary Encoding

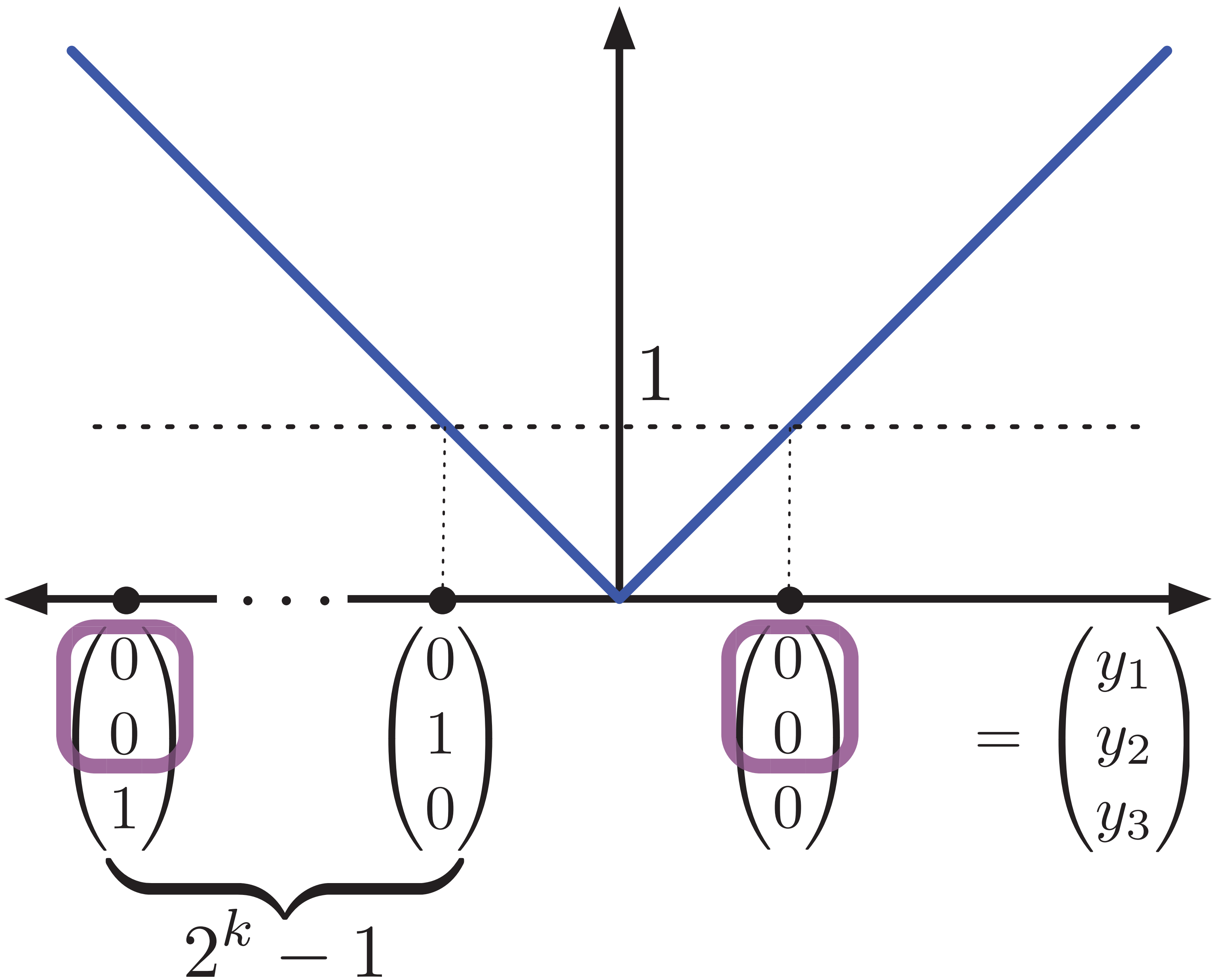


Best Bound $= 0$ unless:

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

$$y_1 = y_2 = 0$$

Example: Binary Encoding



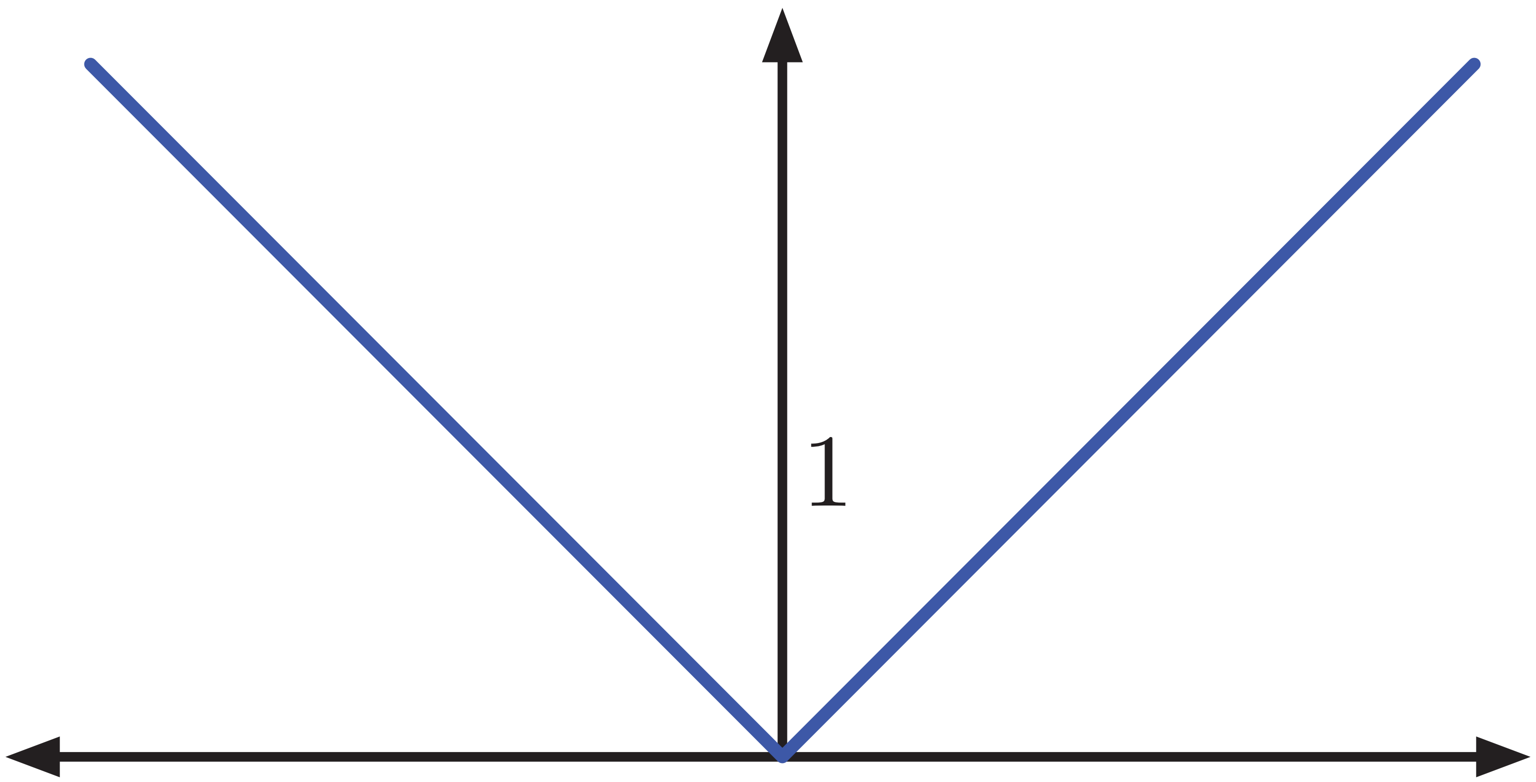
Best Bound $= 0$ unless:

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

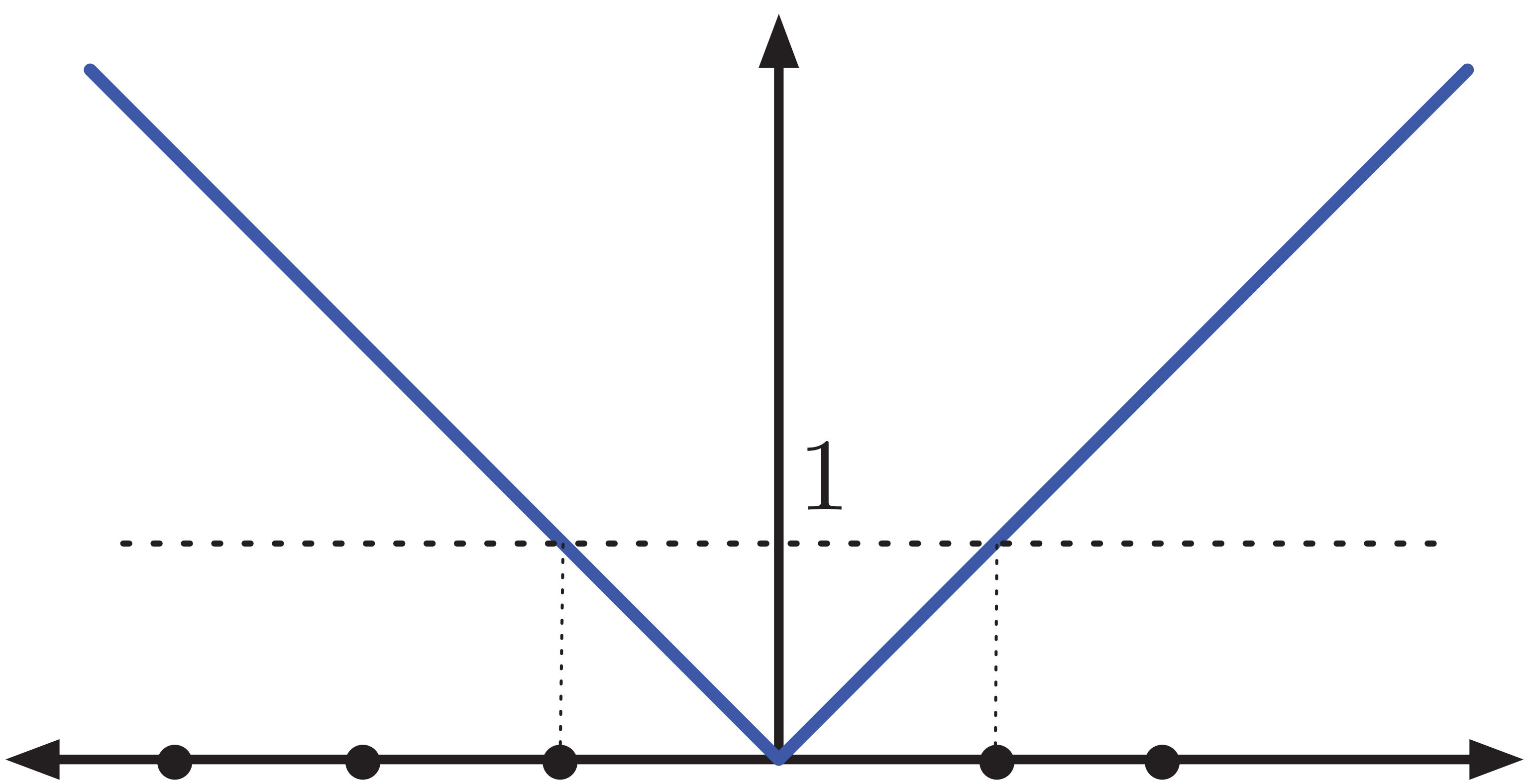
Need $k = \log_2 n$ branches

$$y_1 = y_2 = 0$$

Example: Incremental Encoding

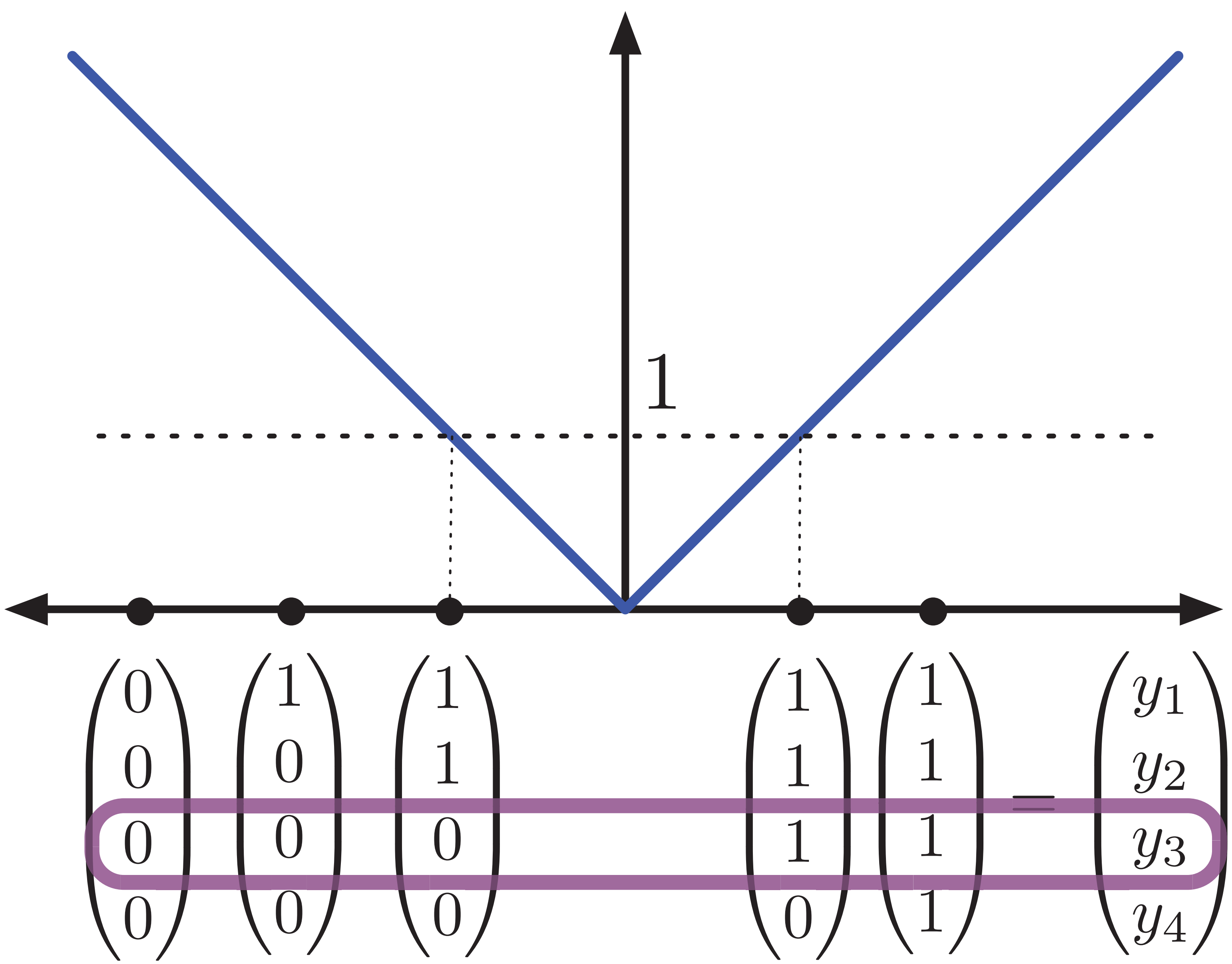


Example: Incremental Encoding



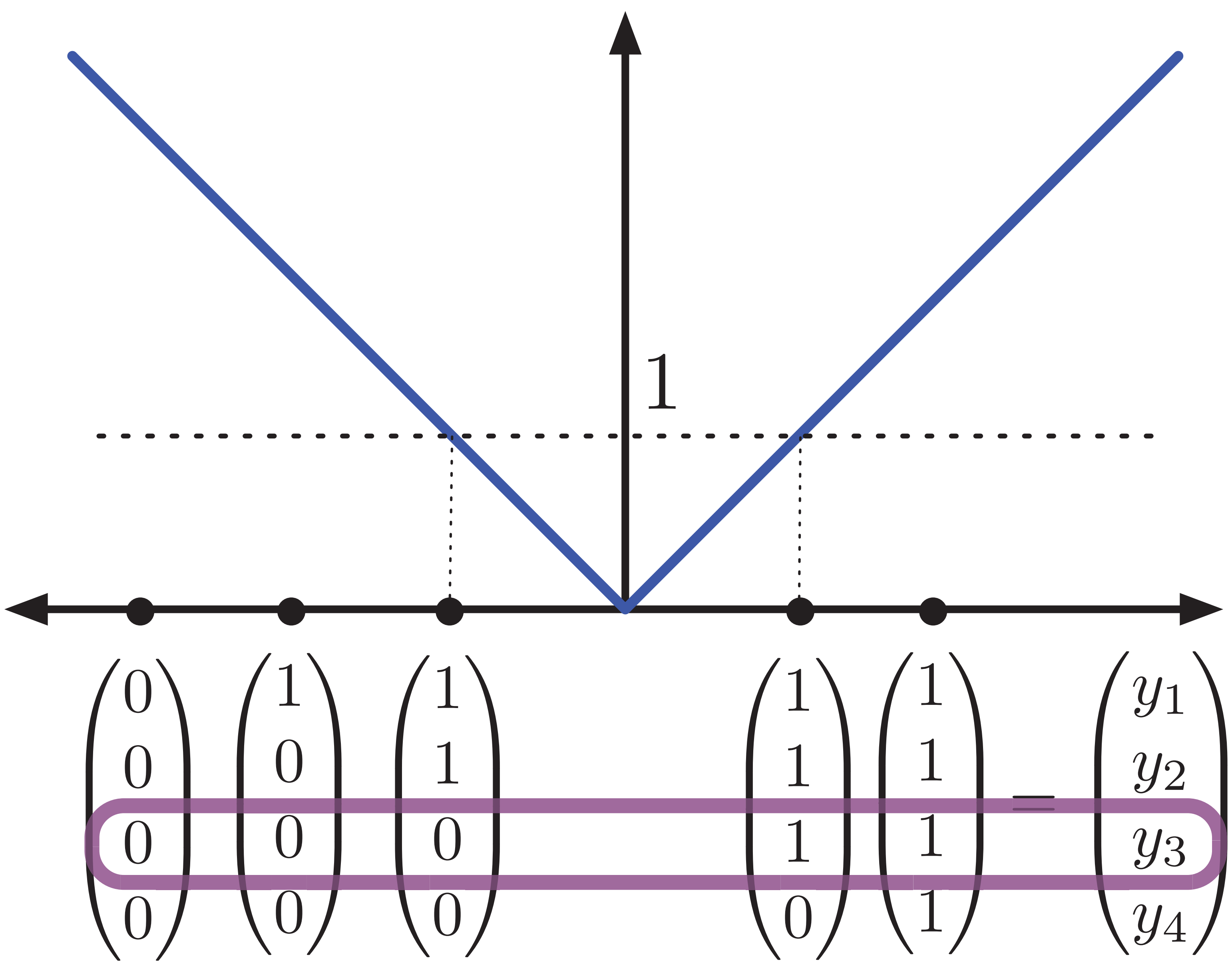
$$\begin{pmatrix}0\\0\\0\\0\end{pmatrix} \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \begin{pmatrix}1\\1\\0\\0\end{pmatrix} \quad \begin{pmatrix}1\\1\\1\\0\end{pmatrix} \begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \begin{pmatrix}y_1\\y_2\\y_3\\y_4\end{pmatrix}$$

Example: Incremental Encoding



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

Example: Incremental Encoding



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

Best Bound = 1 if:

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

Only need 1 branch!

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

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

SOS1 Branching

$$\lambda_1 = \lambda_2 = 0$$

$$or$$

$$\lambda_3 = \lambda_4 = 0$$

Binary

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

Induced Constraint Branching

Incremental

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

SOS1 Branching

$$\lambda_1 = \lambda_2 = 0$$

$$or$$

$$\lambda_3 = \lambda_4 = 0$$

Binary

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

SOS1 Branching

$$\lambda_1 = \lambda_2 = 0$$

$$or$$

$$\lambda_3 = \lambda_4 = 0$$

Induced Constraint Branching

Incremental → **SOS1 Branching**

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \lambda = y \quad \longrightarrow \quad \begin{array}{c} \lambda_1 = \lambda_2 = 0 \\ or \\ \lambda_3 = \lambda_4 = 0 \end{array}$$

Binary → **Odd/Even Branching**

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \lambda = y \quad \longrightarrow \quad \begin{array}{c} \lambda_1 = \lambda_3 = 0 \\ or \\ \lambda_2 = \lambda_4 = 0 \end{array}$$

Formulation Step 2:
Combining with Strong Formulation

Long Lost Integral Formulation

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

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

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

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

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

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

Also for general polyhedra with common recession cones.

- Jeroslow and Lowe 1984.

Combining with Alternative Encoding

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

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

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

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

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

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

Also for general polyhedra with common recession cones.

- Jeroslow and Lowe 1984.

Combining with Alternative Encoding

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

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

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

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

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

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

Also for general polyhedra with common recession cones.

- Jeroslow and Lowe 1984.

Combining with Alternative Encoding

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

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

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

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

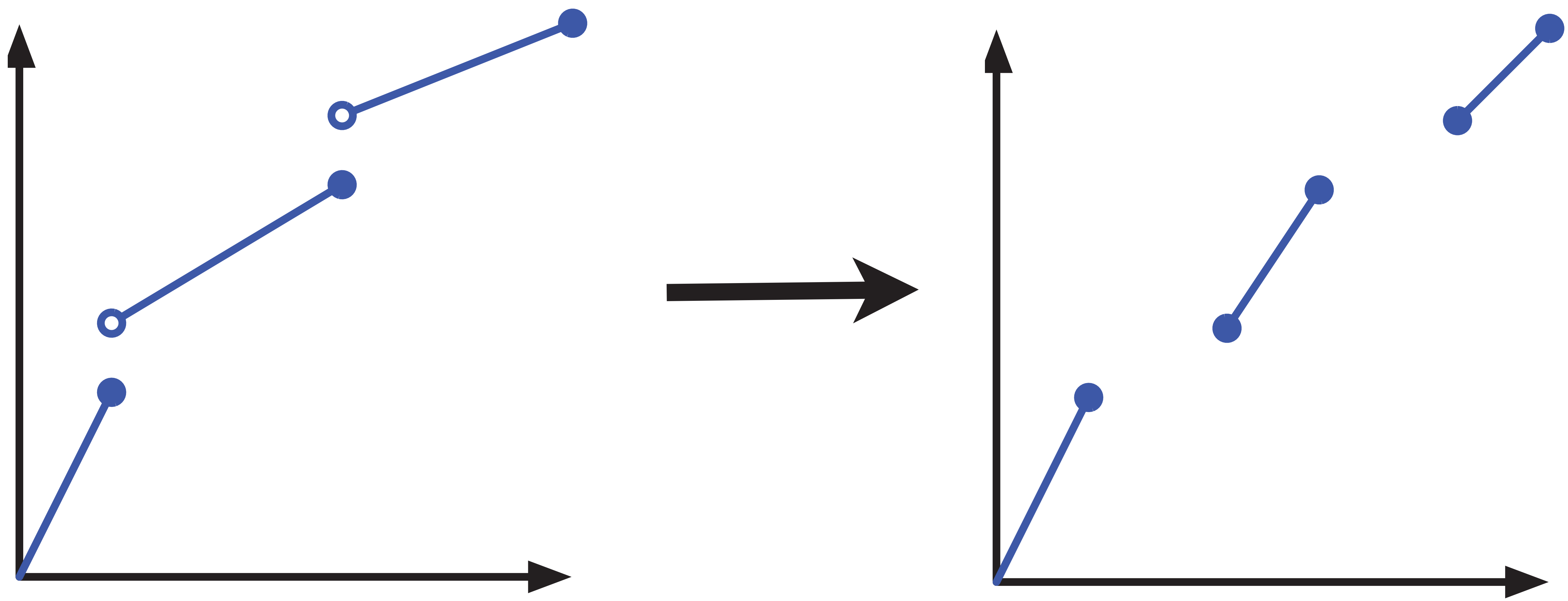
$$\sum_{i=1}^n \sum_{v \in \text{ext}(P^i)} b^i \lambda_v^i = y$$

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

Also for general polyhedra with common recession cones.

- V., Ahmed and Nemhauser 2010; V. 2012.

Univariate Transportation Problems



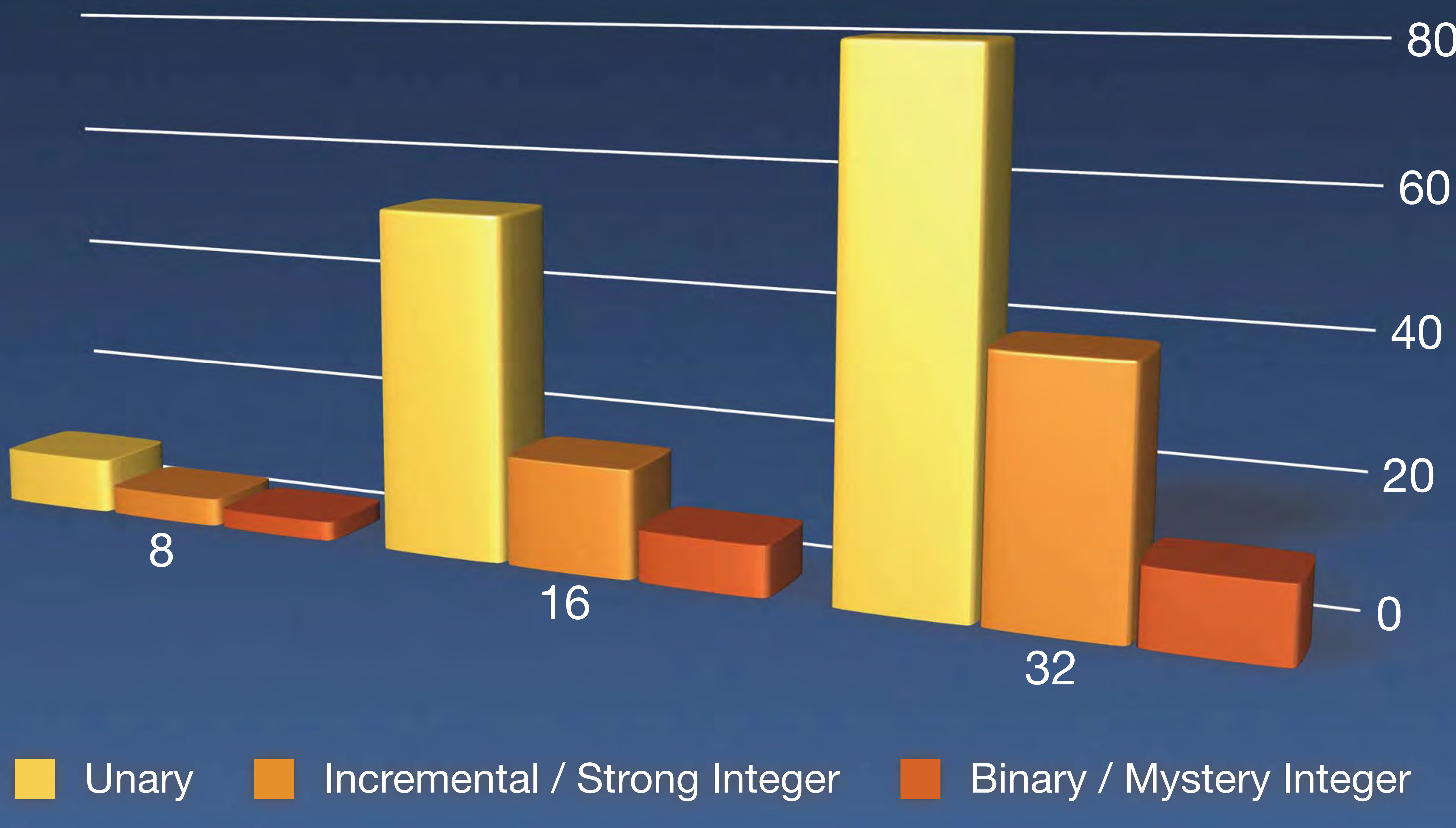
Discontinuous Piecewise Linear

Disc. PWL + "Semicontinuous"

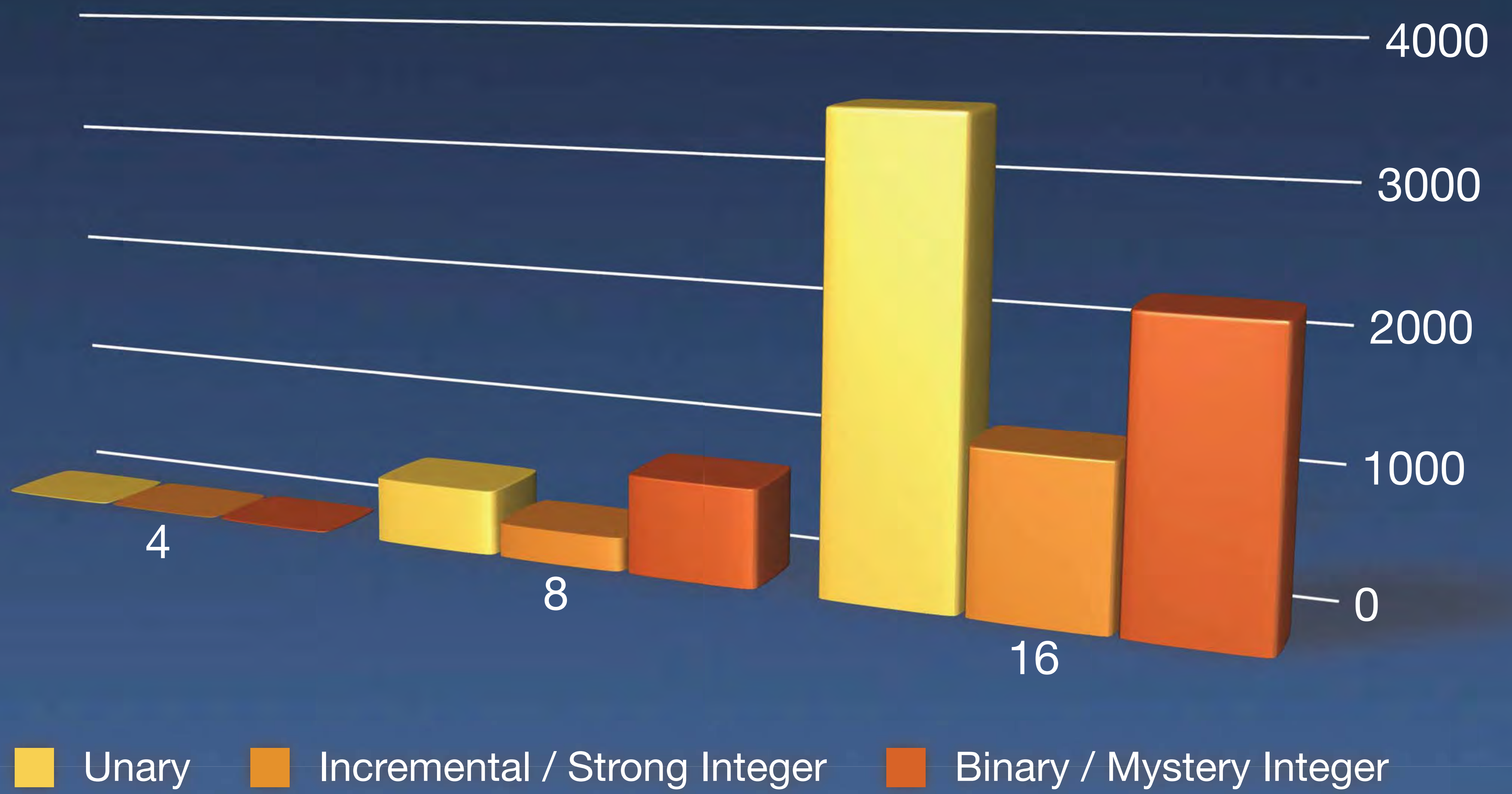
Piecewise Linear

Unary   Incremental / Strong Integer   Binary / Mystery Integer

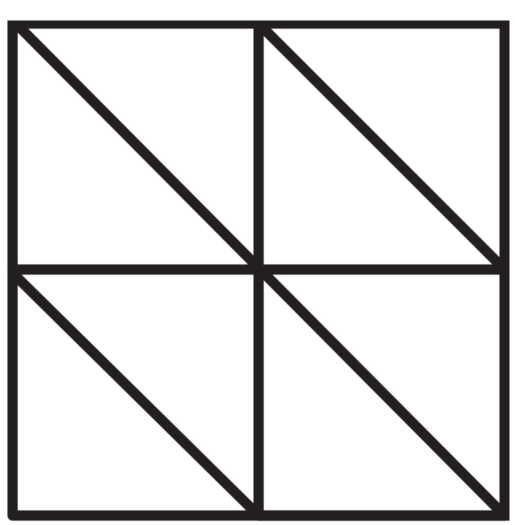
Piecewise Linear



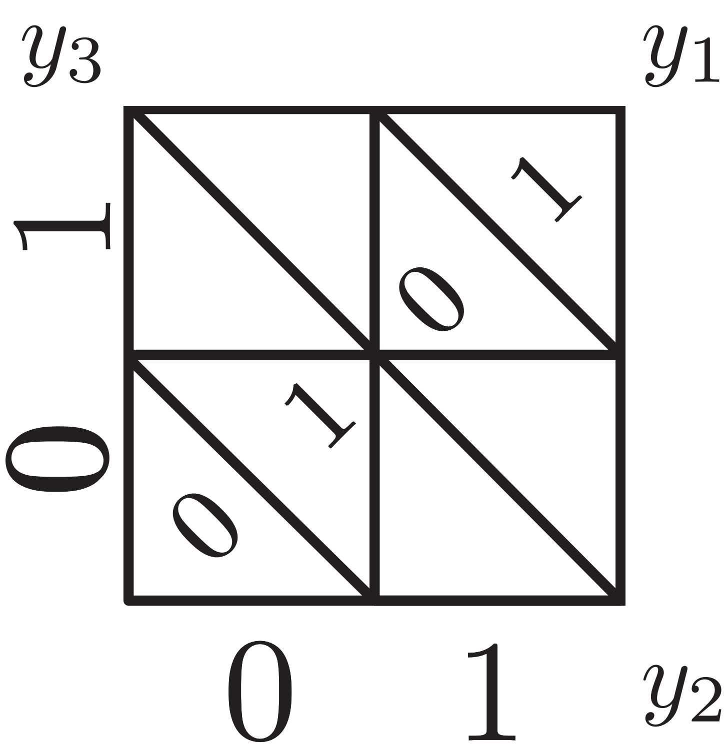
Piecewise Linear



Combining Codes for Multivariate

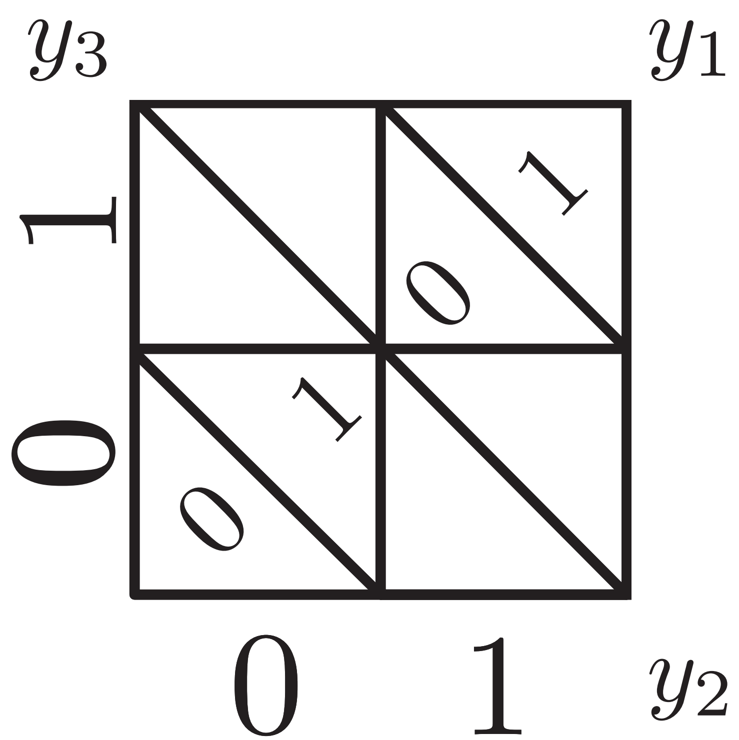


Combining Codes for Multivariate



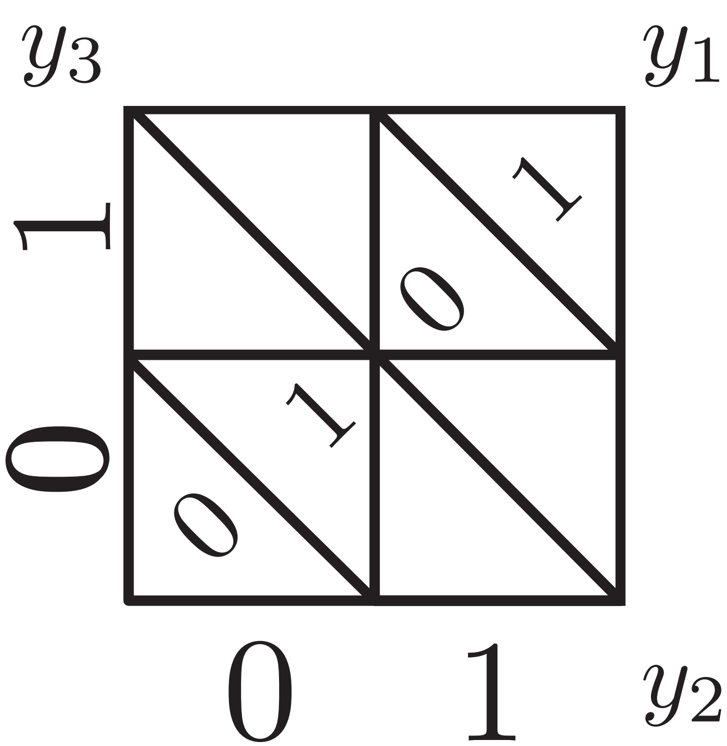
Combining Codes for Multivariate

$0 \quad 1 \quad 0 \quad 1$



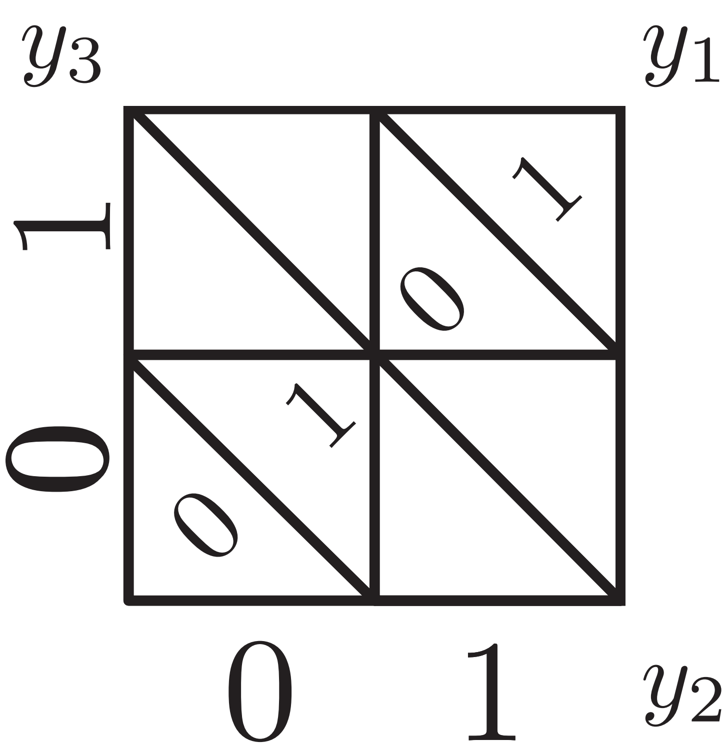
Combining Codes for Multivariate

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



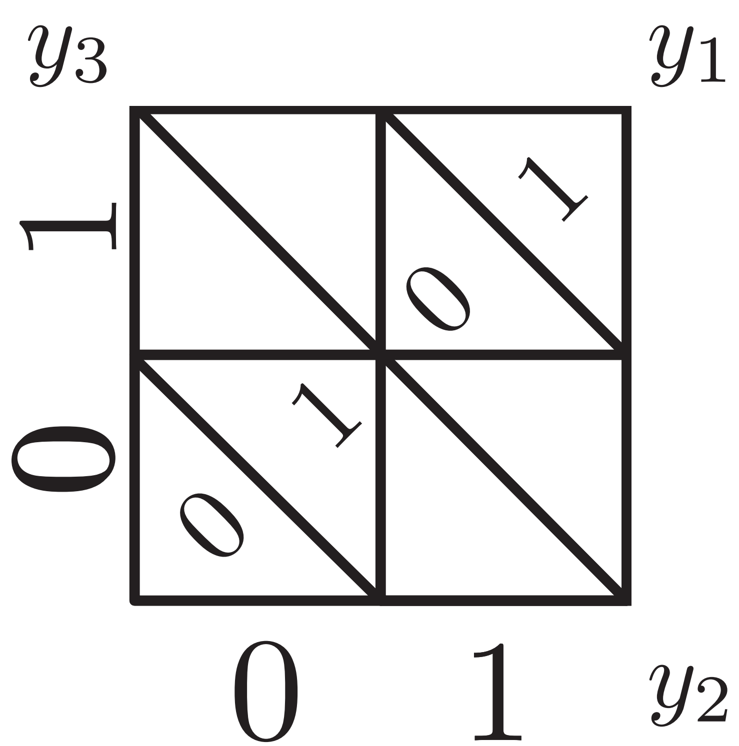
Combining Codes for Multivariate

0 1 0 1
0



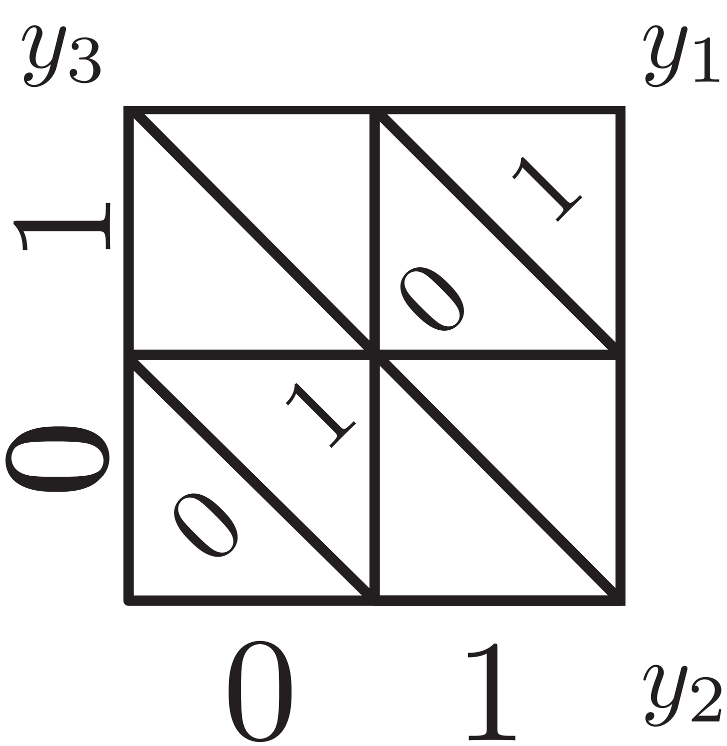
Combining Codes for Multivariate

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



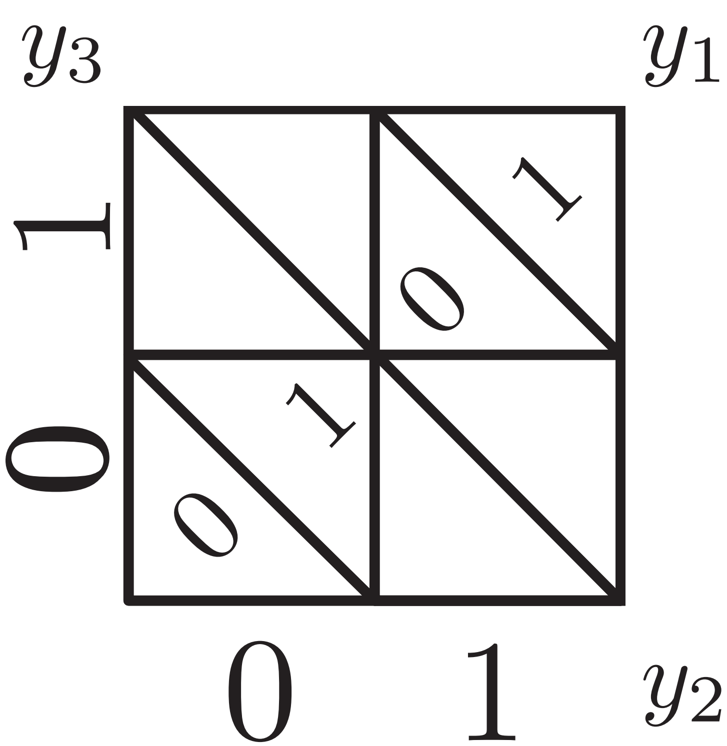
Combining Codes for Multivariate

$$
\begin{array}{cccc}
0 & 1 & 0 & 1 \\
0 & 0 & 1 &
\end{array}
$$



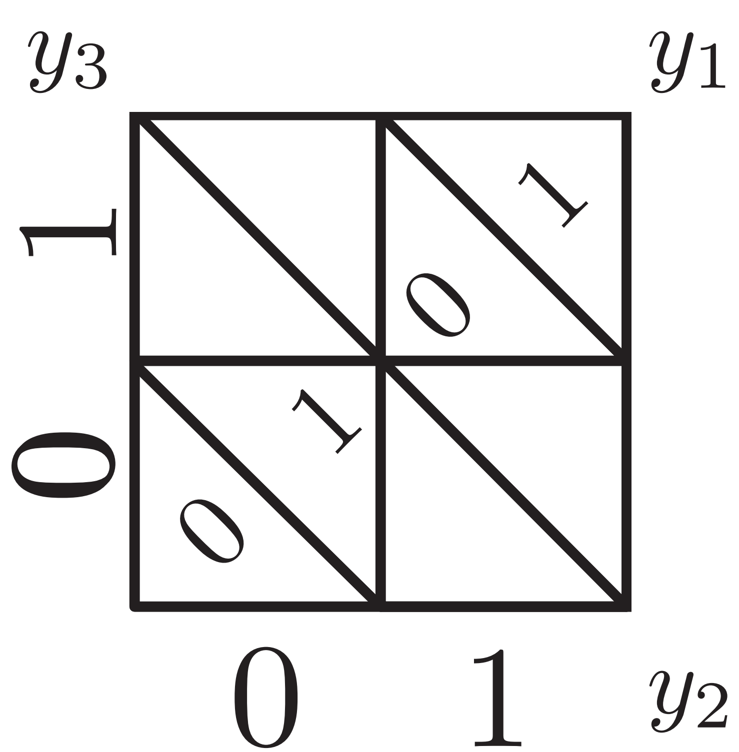
Combining Codes for Multivariate

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



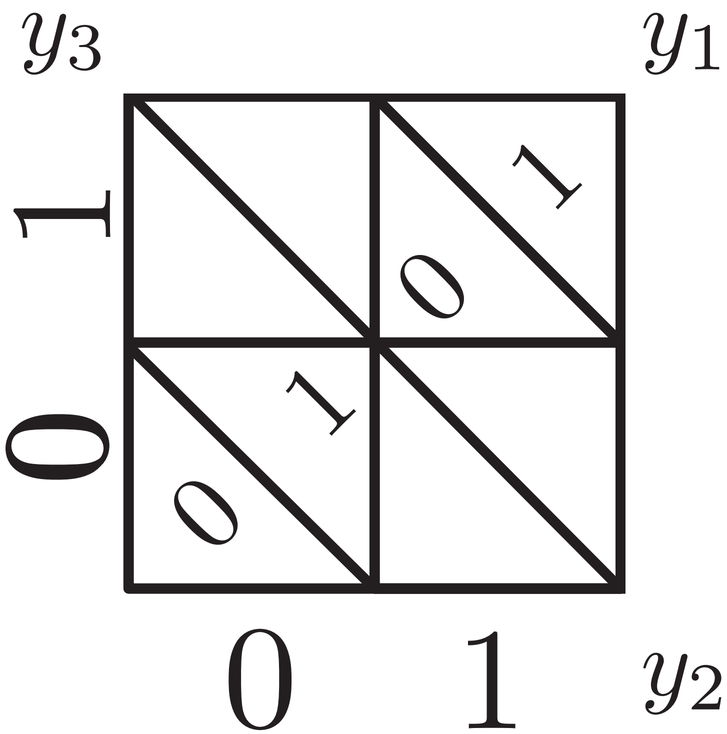
Combining Codes for Multivariate

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



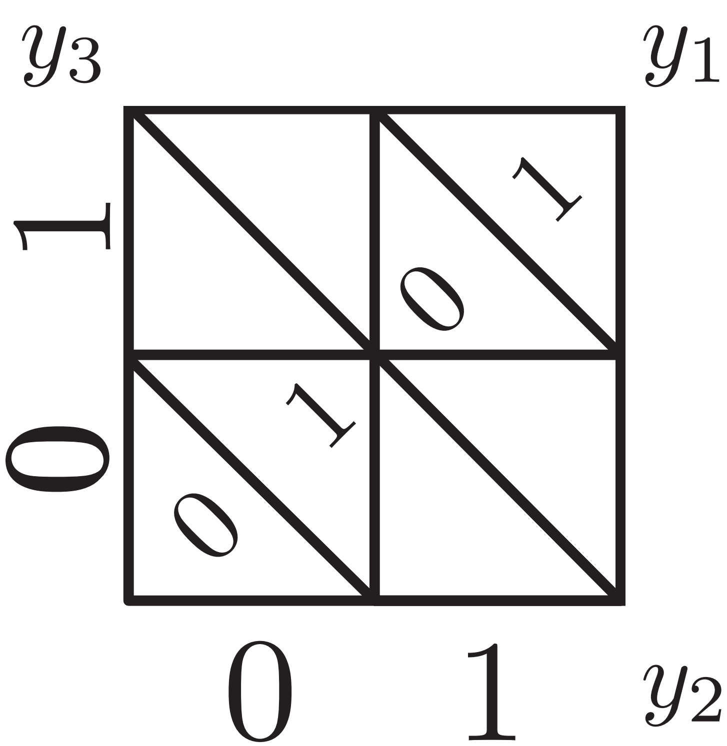
Combining Codes for Multivariate

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



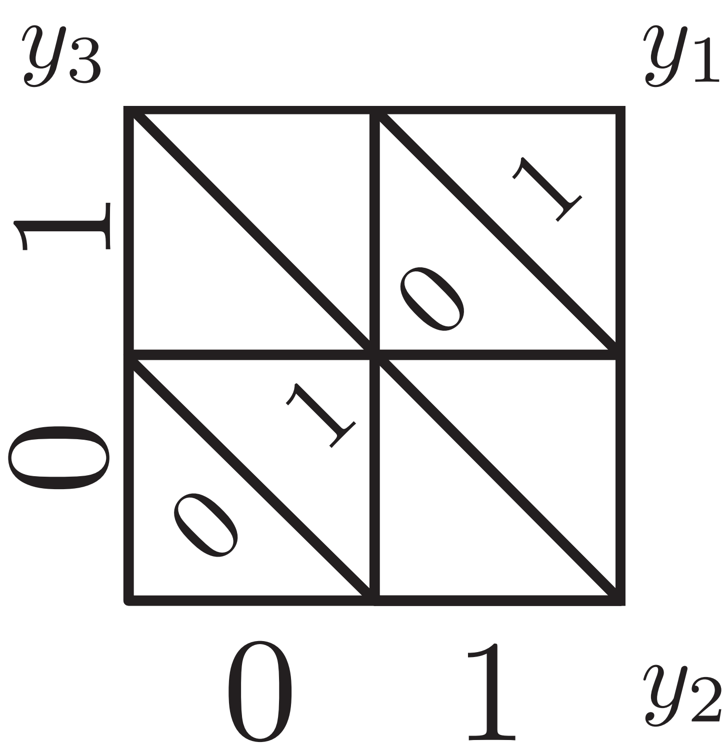
Combining Codes for Multivariate

0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1



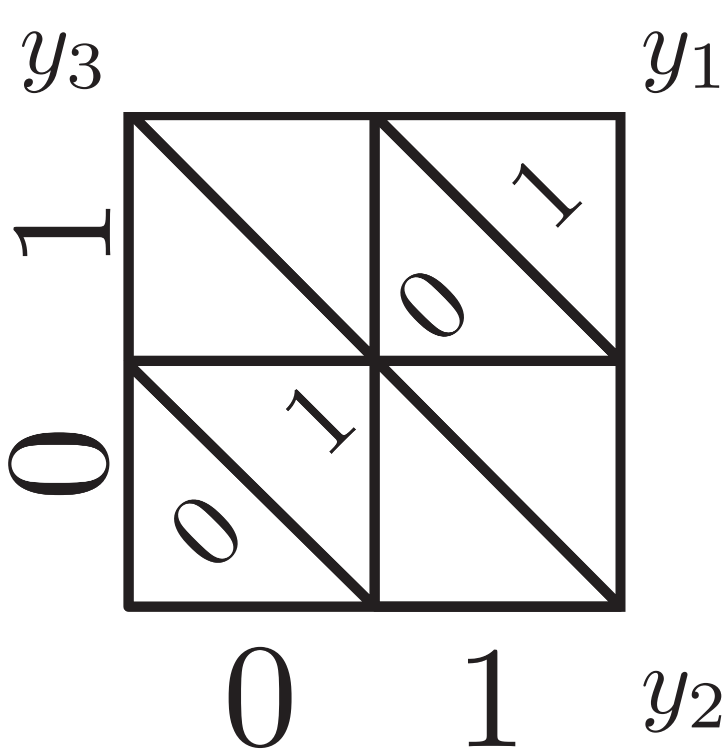
Combining Codes for Multivariate

0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0							



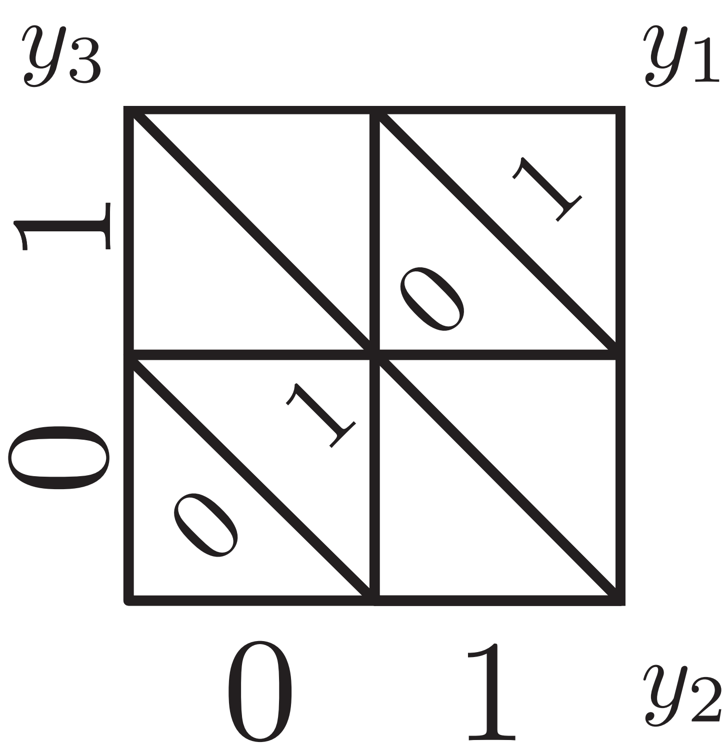
Combining Codes for Multivariate

0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0							



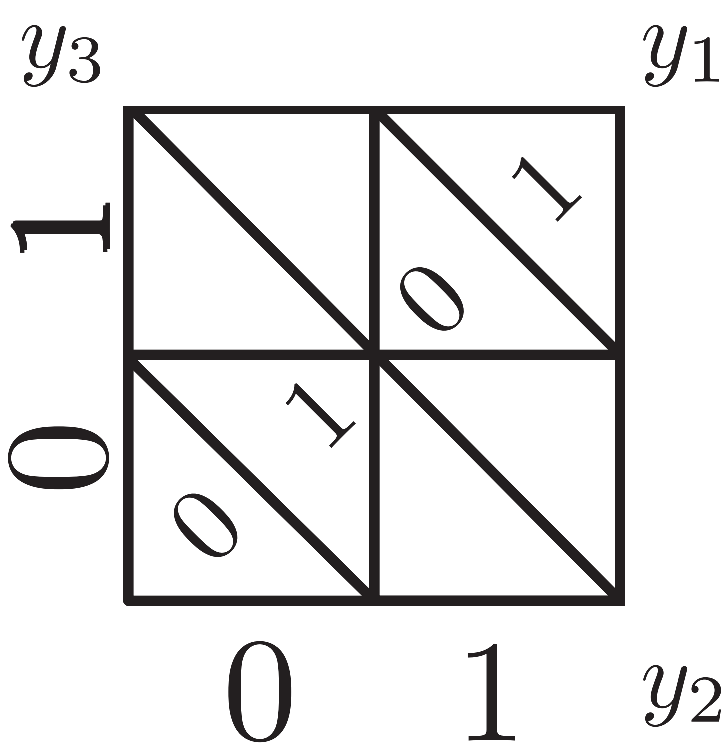
Combining Codes for Multivariate

0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0				



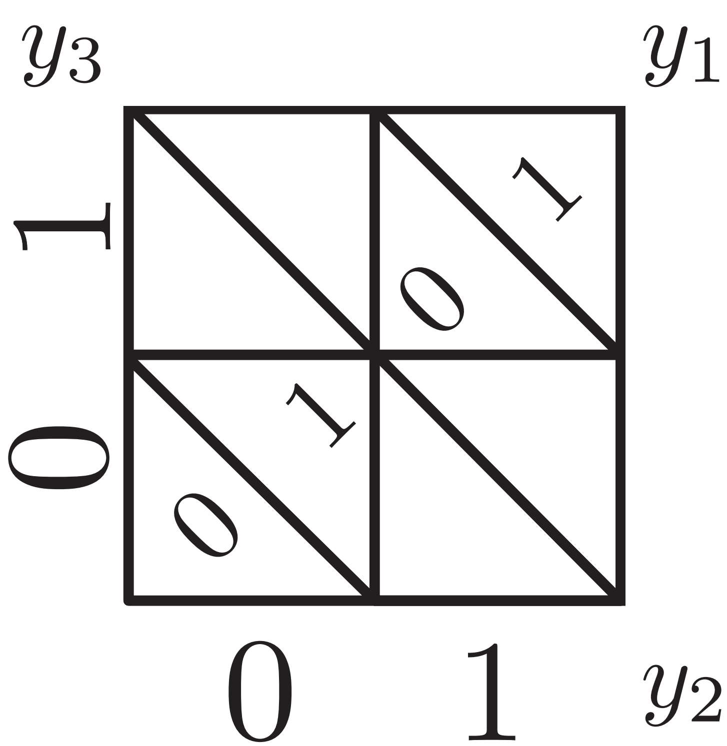
Combining Codes for Multivariate

0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0				



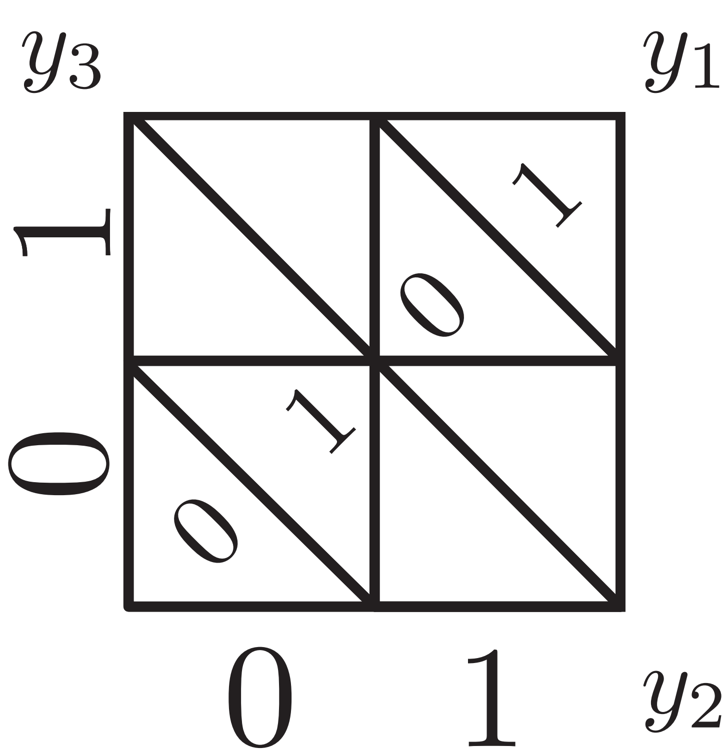
Combining Codes for Multivariate

0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1			



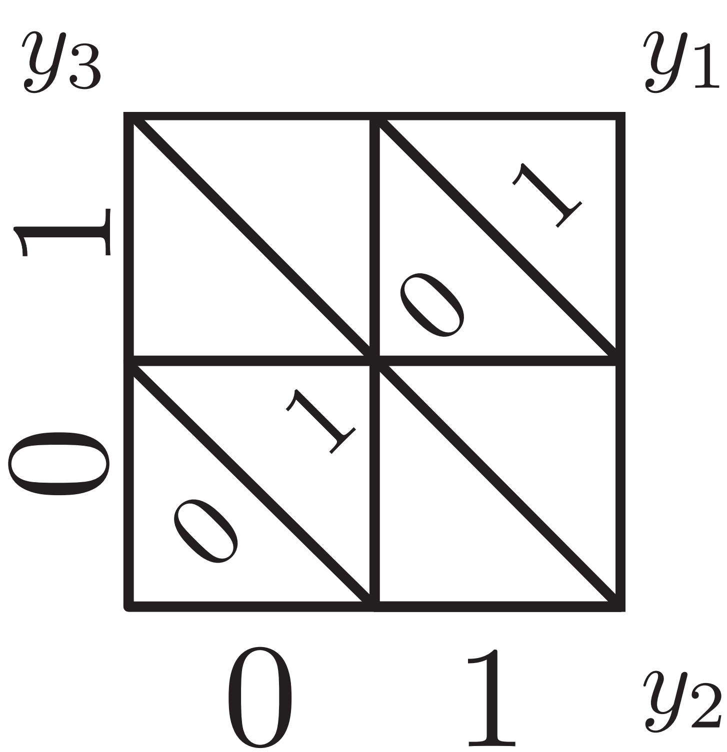
Combining Codes for Multivariate

0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1			



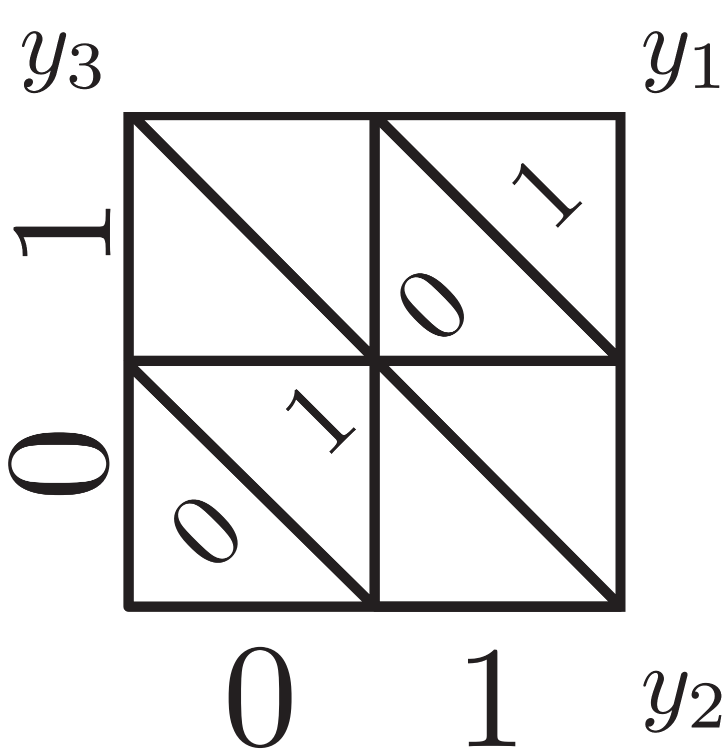
Combining Codes for Multivariate

0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1



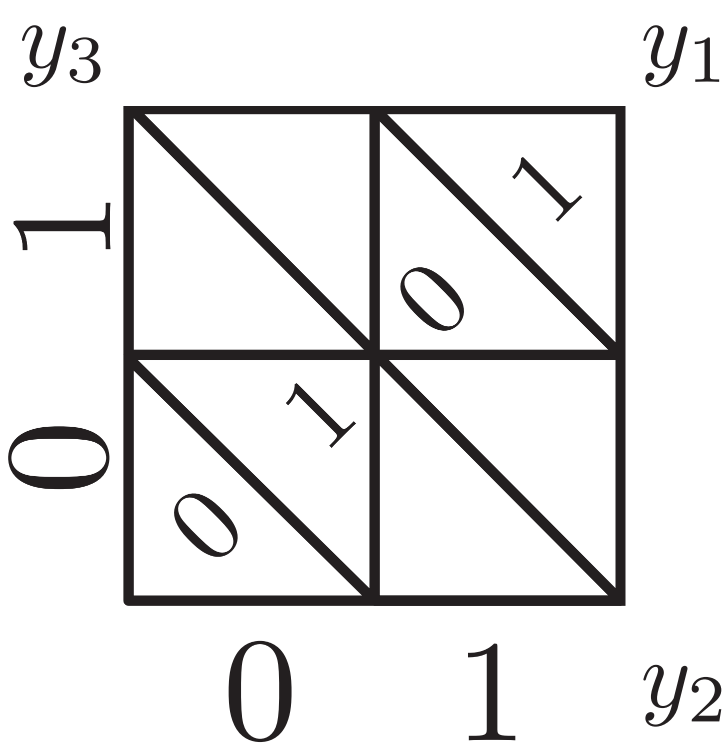
Combining Codes for Multivariate

0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1



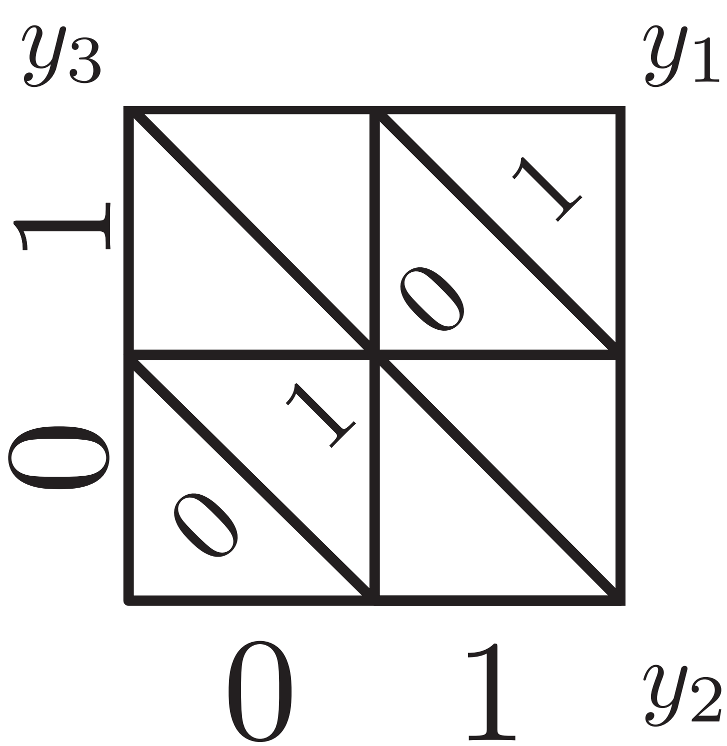
Combining Codes for Multivariate

y_1	0	1	0	1	0	1	0	1
y_2	0	0	1	1	0	0	1	1
y_3	0	0	0	0	1	1	1	1



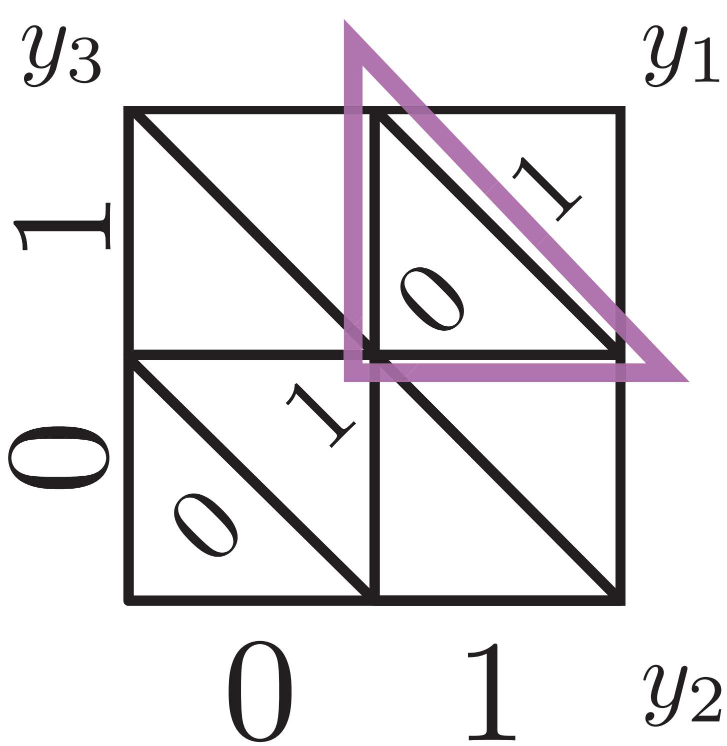
Combining Codes for Multivariate

Binary	y_1	0	1	0	1	0	1	0	1
	y_2	0	0	1	1	0	0	1	1
	y_3	0	0	0	0	1	1	1	1



Combining Codes for Multivariate

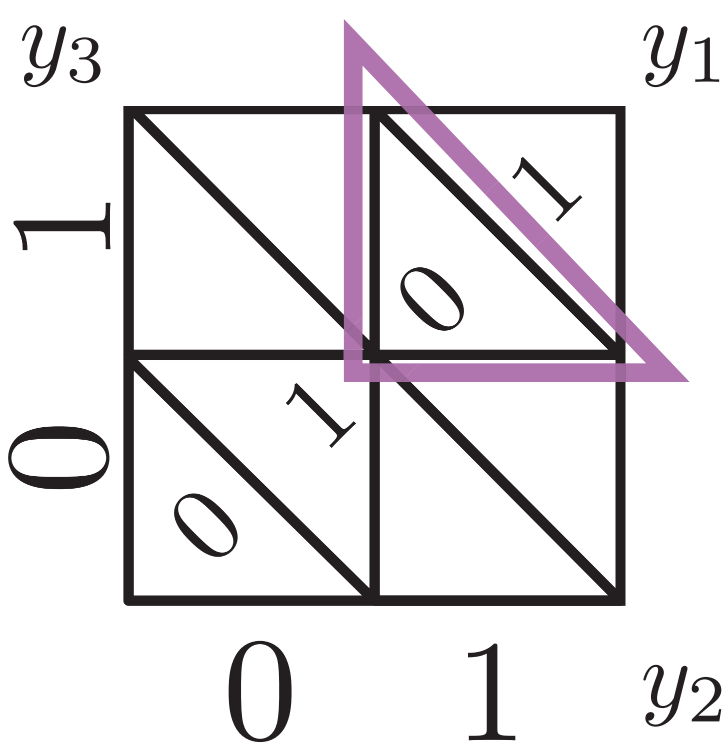
Binary	y_1	0	1	0	1	0	1	0	1
	y_2	0	0	1	1	0	0	1	1
	y_3	0	0	0	0	1	1	1	1

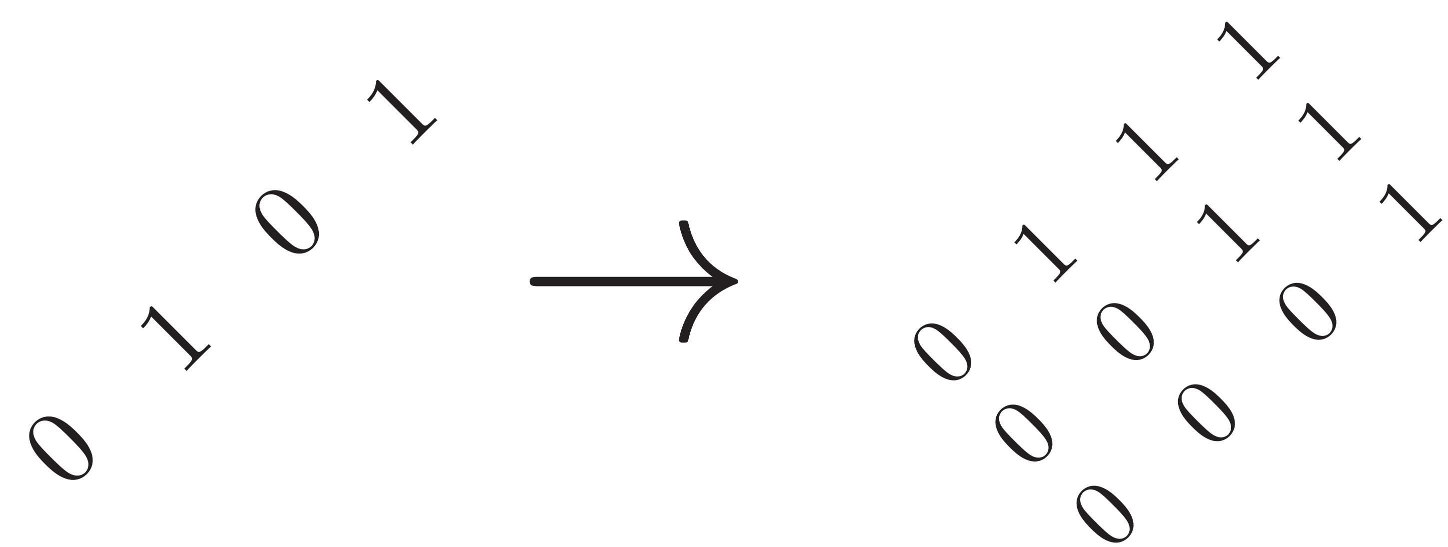


Combining Codes for Multivariate

Binary

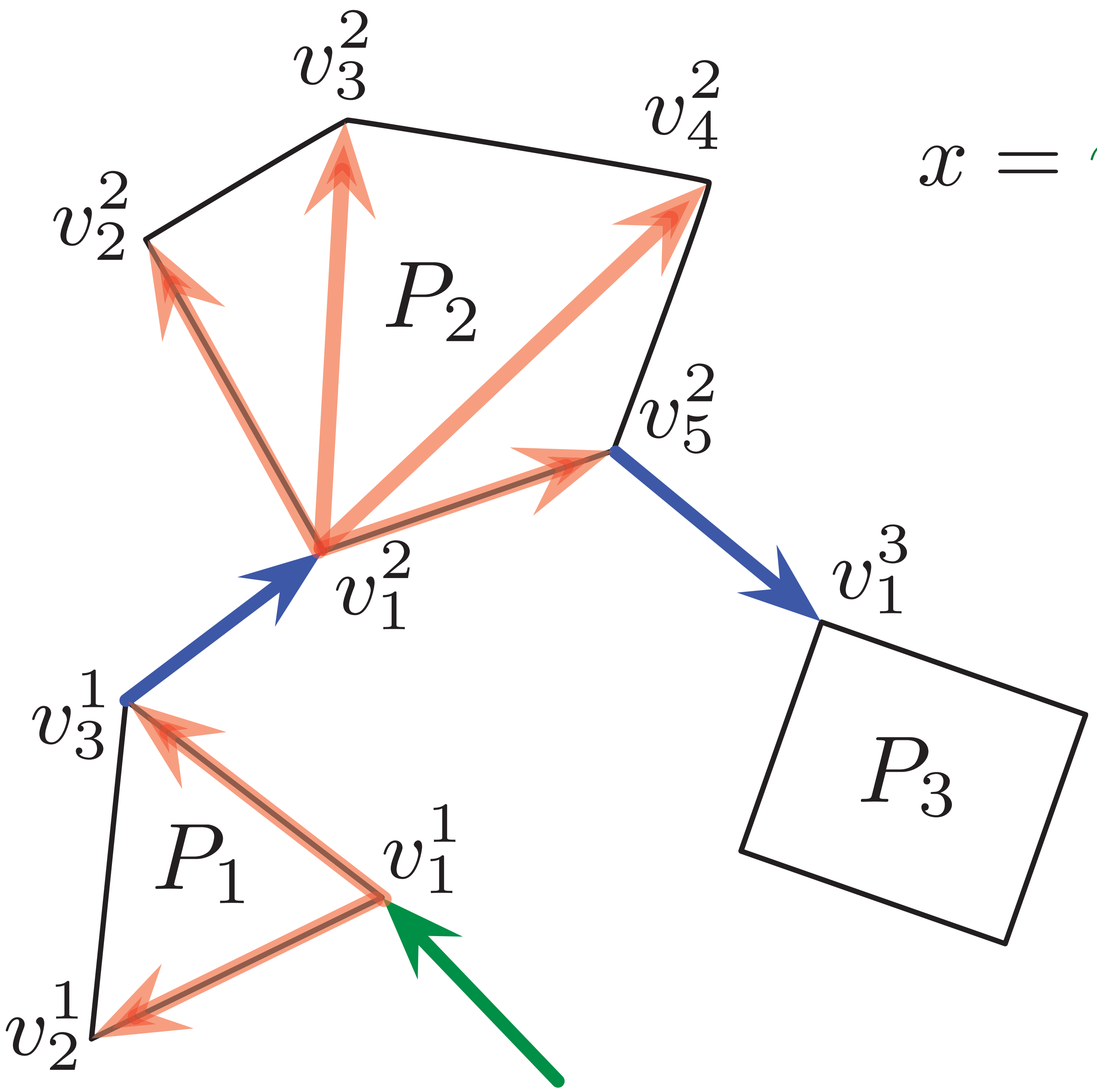
y_1	0	1	0	1	0	1	0	1
y_2	0	0	1	1	0	0	1	1
y_3	0	0	0	0	1	1	1	1





Incremental Diagonal

Incremental “Delta” Formulation

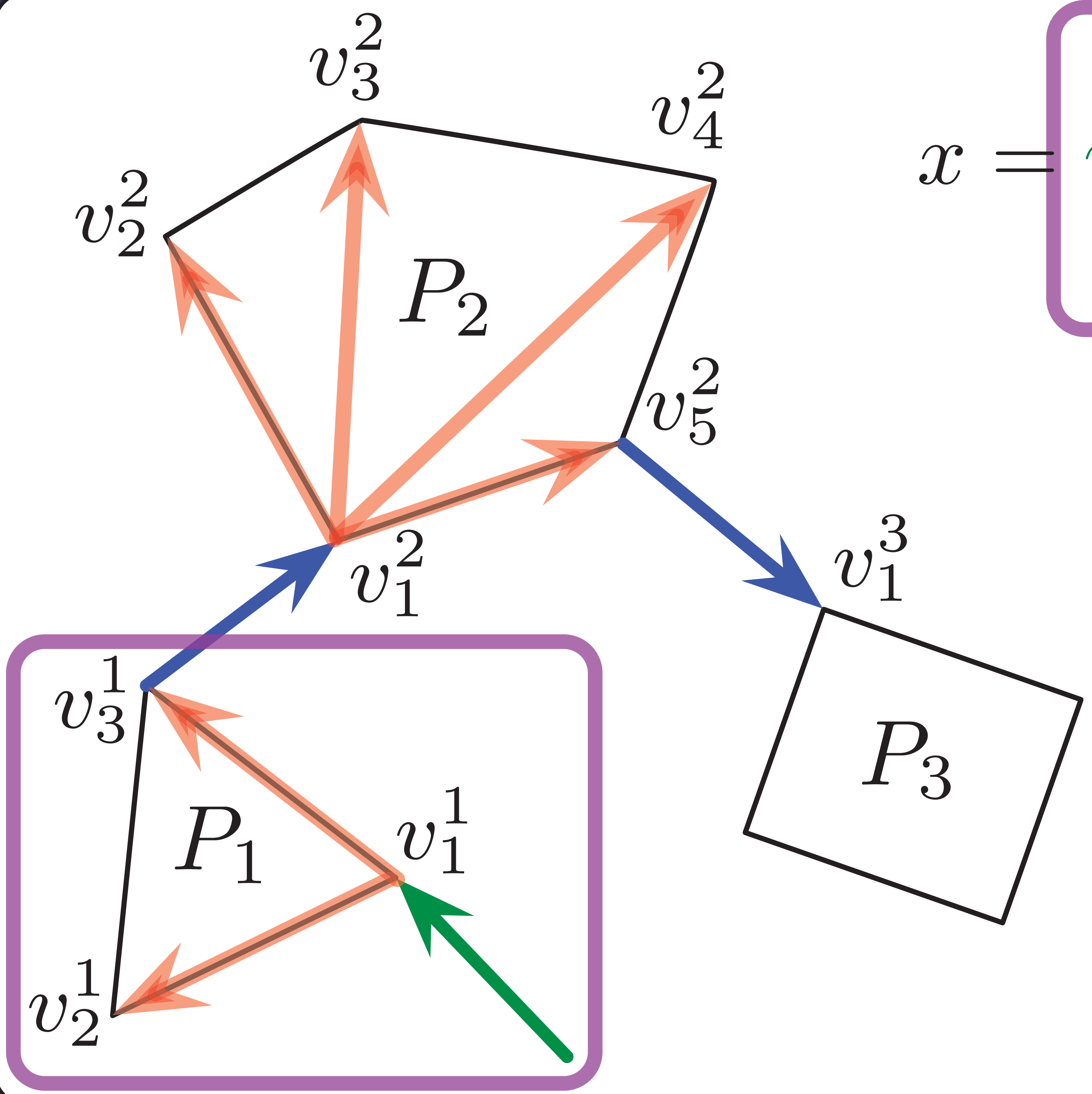


$$x = v_1^1 + \sum_{i=1}^{k} \sum_{j=2}^{r_i} \left(v_j^i - v_1^i\right) \delta_j^i + \sum_{i=2}^{k} \left(v_1^i - v_{r_i}^{i-1}\right) y_i$$

$$\delta_j^i \in [0,1], \; y_i \in \{0,1\}$$

- Yıldız and V. ‘12 (Generalization of Lee and Wilson 1999)

Incremental "Delta" Formulation

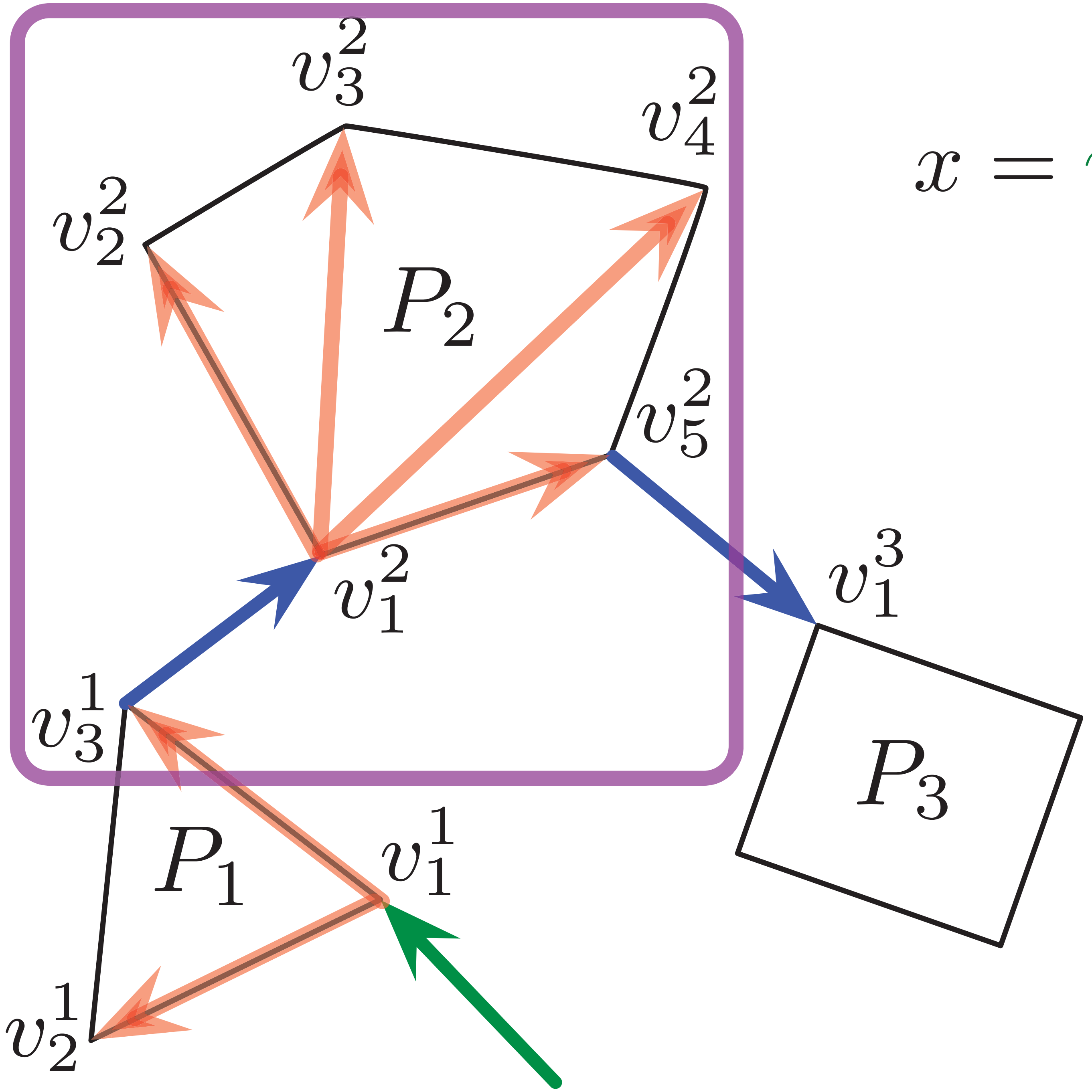


$$x = v_1^1 + \sum_{i=1}^{k} \sum_{j=2}^{r_i} \left(v_j^i - v_1^i\right) \delta_j^i + \sum_{i=2}^{k} \left(v_1^i - v_{r_i}^{i-1}\right) y_i$$

$$\sum_{j=2}^{r_i} \delta_j^i \leq 1, \quad \delta_j^i \geq 0$$

- Yıldız and V. '12 (Generalization of Lee and Wilson 1999)

Incremental "Delta" Formulation

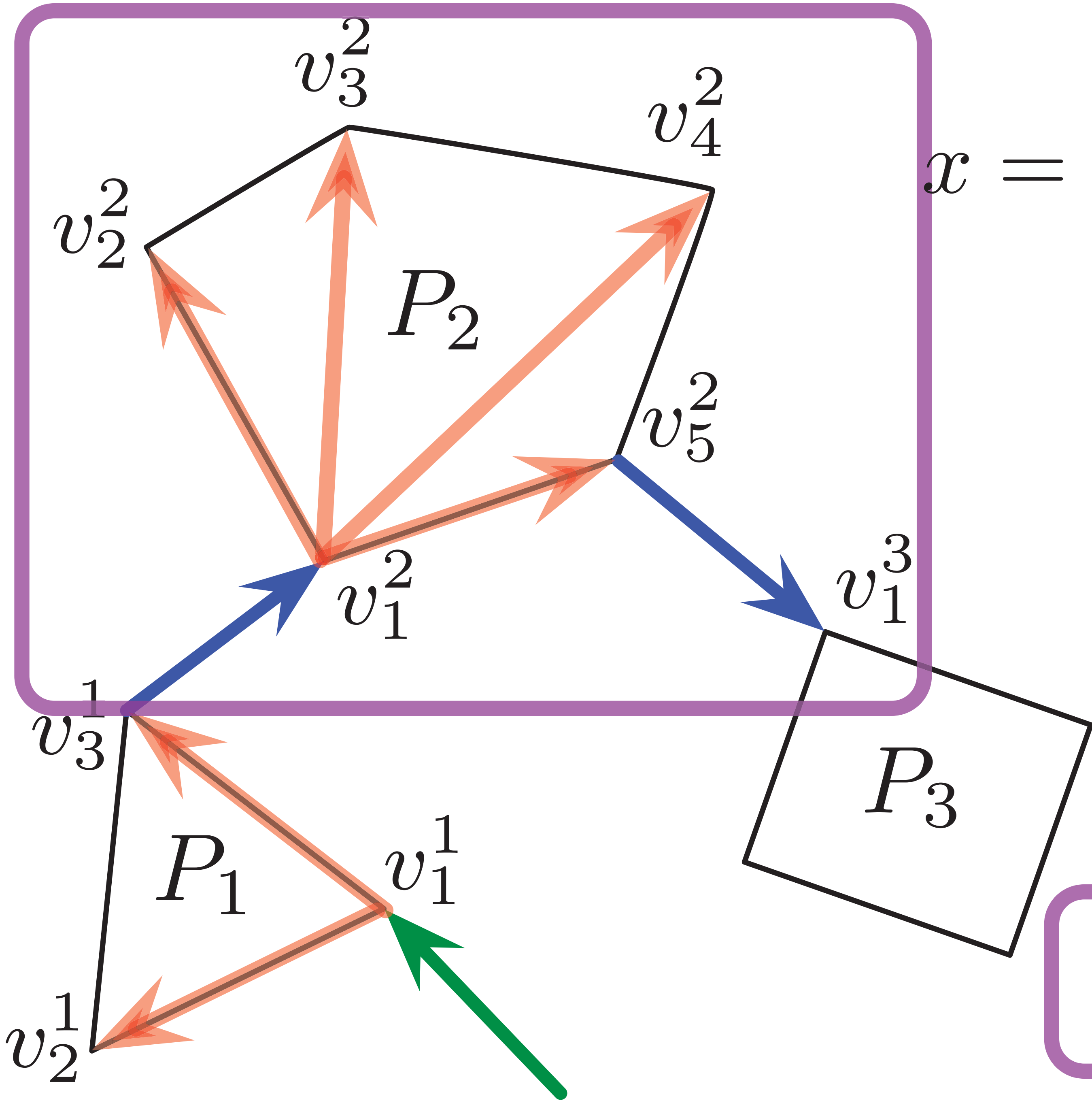


$$x = v_1^1 + \sum_{i=1}^{k} \sum_{j=2}^{r_i} \left(v_j^i - v_1^i\right) \delta_j^i + \sum_{i=2}^{k} \left(v_1^i - v_{r_i}^{i-1}\right) y_i$$

$$\sum_{j=2}^{r_i} \delta_j^i \leq y_i$$

- Yıldız and V. '12 (Generalization of Lee and Wilson 1999)

Incremental "Delta" Formulation



$$x = v_1^1 + \sum_{i=1}^{k} \sum_{j=2}^{r_i} \left(v_j^i - v_1^i\right) \delta_j^i + \sum_{i=2}^{k} \left(v_1^i - v_{r_i}^{i-1}\right) y_i$$

$$y_{i+1} \leq \delta_{r_i}^i, \quad y_{i+1} \leq y_i$$

- Yıldız and V. '12 (Generalization of Lee and Wilson 1999)

Summary, Extensions and More.

- Effective formulations: Encode and Formulate
  - Best encoding? Why not try a few.
- Smaller formulations for shared vertex case
  - Need encodings with special structure.
- Where to find more:
  - 15.099 Spring ’13: Theory / Practice.
  - Survey: V., “MIP Formulation Techniques”.