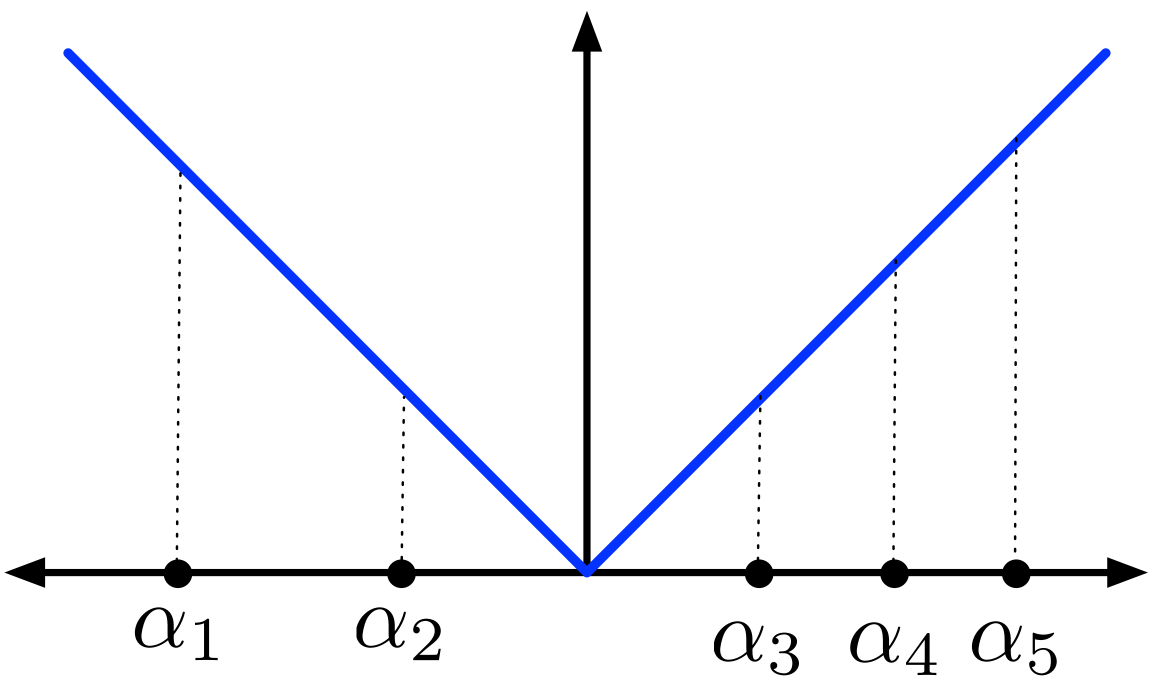
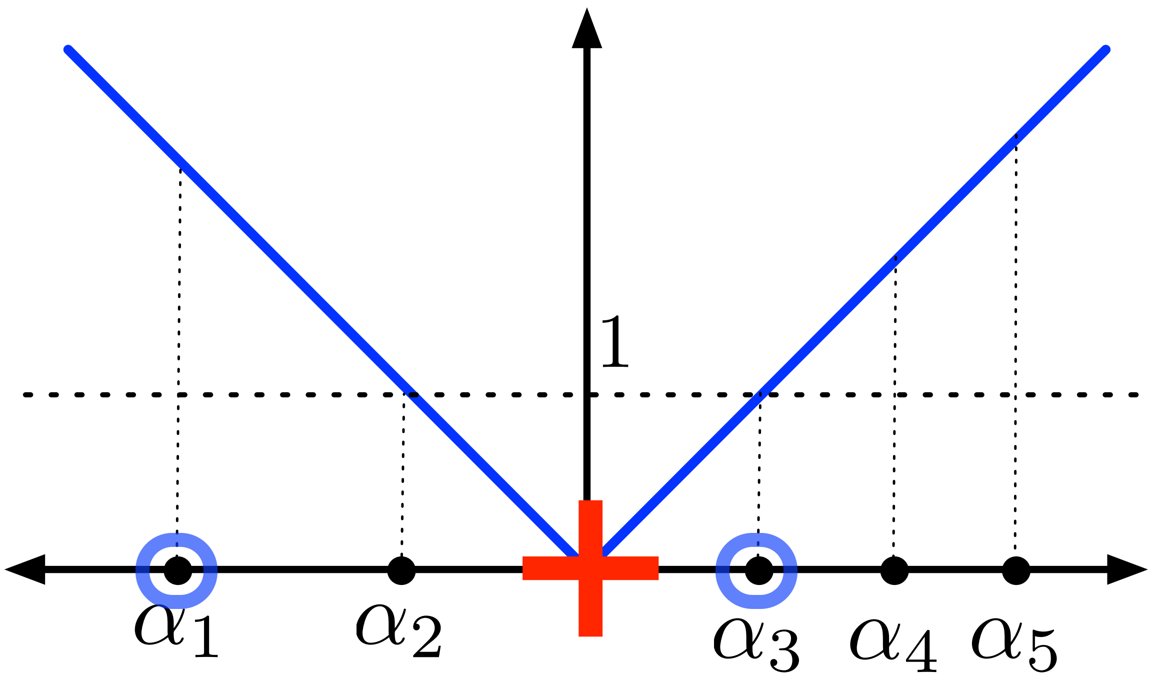
Formulating Discrete Alternatives

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



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

Formulating Discrete Alternatives



$$\min \quad |x|$$

$$s.t.$$

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

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

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

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

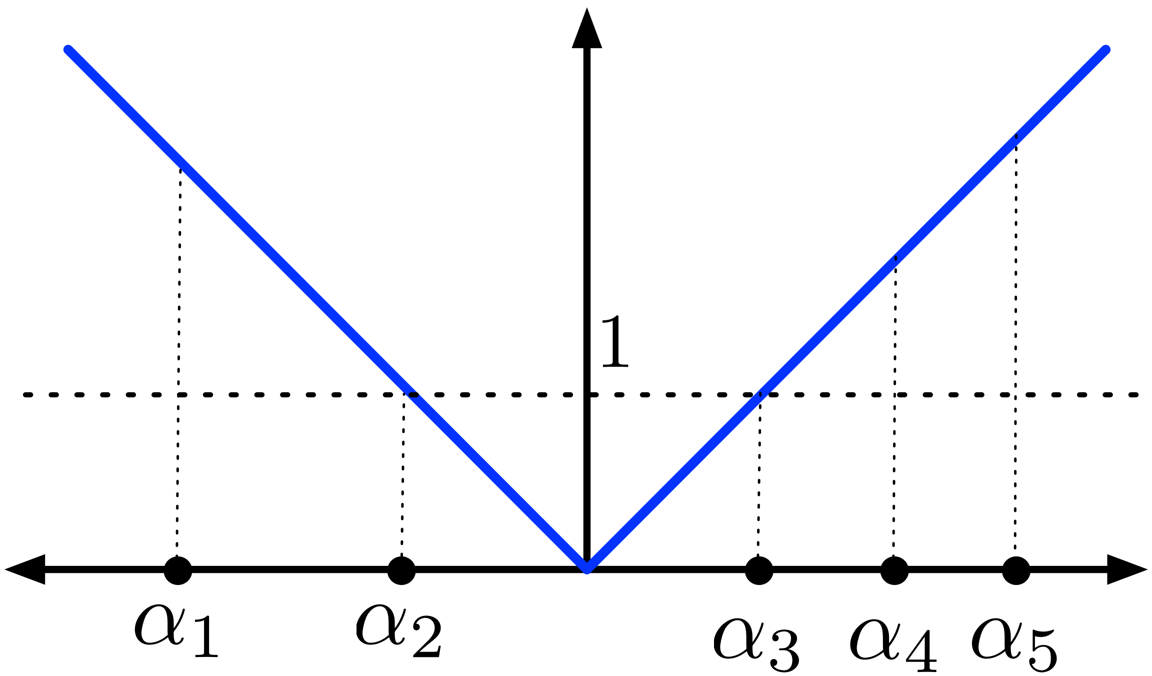
Solve by **binary** Branch-and-Bound:

Branch on λ_2

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

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

Formulating Discrete Alternatives



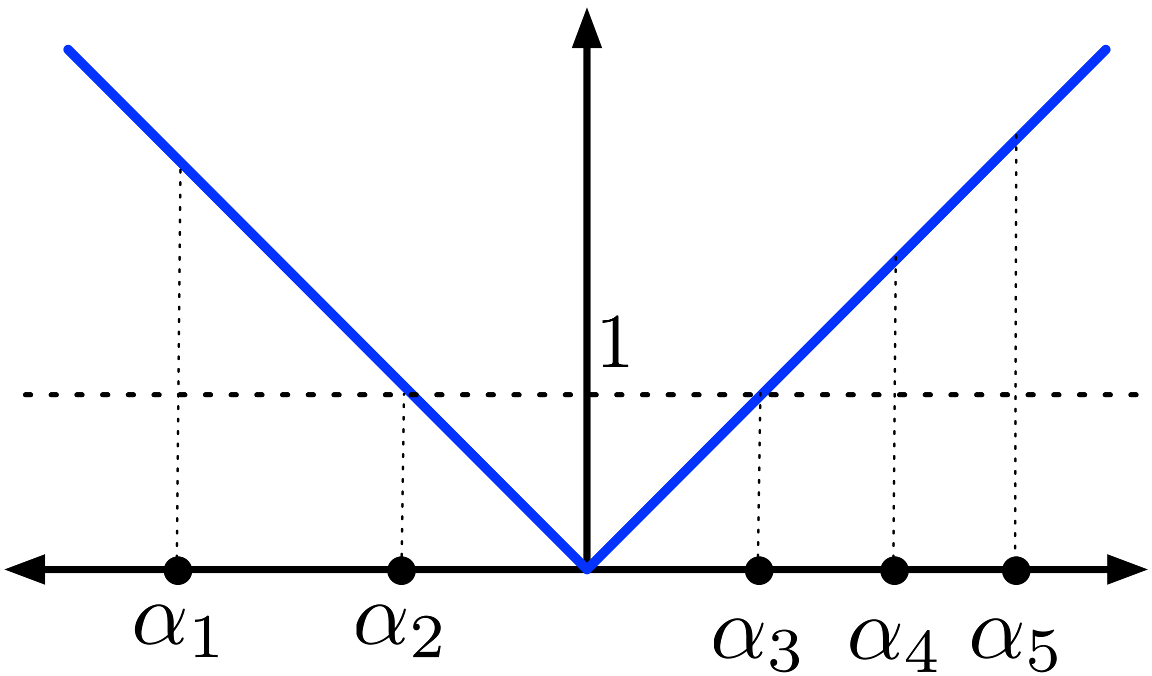
$$\begin{array}{ll} \min & \color{blue}{|x|} \\ s.t. & \\ & \sum_{i=1}^{n} \lambda_i \alpha_i = x \\ & \sum_{i=1}^{n} \lambda_i = 1 \\ & \lambda \in \{0,1\}^n \end{array}$$

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

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

Worst case: $n/2$ branches to solve

Formulating Discrete Alternatives



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

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

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

Branch on $\lambda_1 + \lambda_2$

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

Never more than one branch (2 nodes).

Constraint Branching is the Solution?

- For $\sum_{i=1}^{n} \lambda_i = 1, \quad \{0,1\}^n$

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

$$\Updownarrow$$

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

- Similar branching for Special Ordered Sets (SOS)
  - SOS1: at most 1 positive $\{\lambda_i\}_{i=1}^{n} \subset \mathbb{R}_+$
  - SOS2: at most 2 positive $\{\lambda_i\}_{i=1}^{n} \subset \mathbb{R}_+$

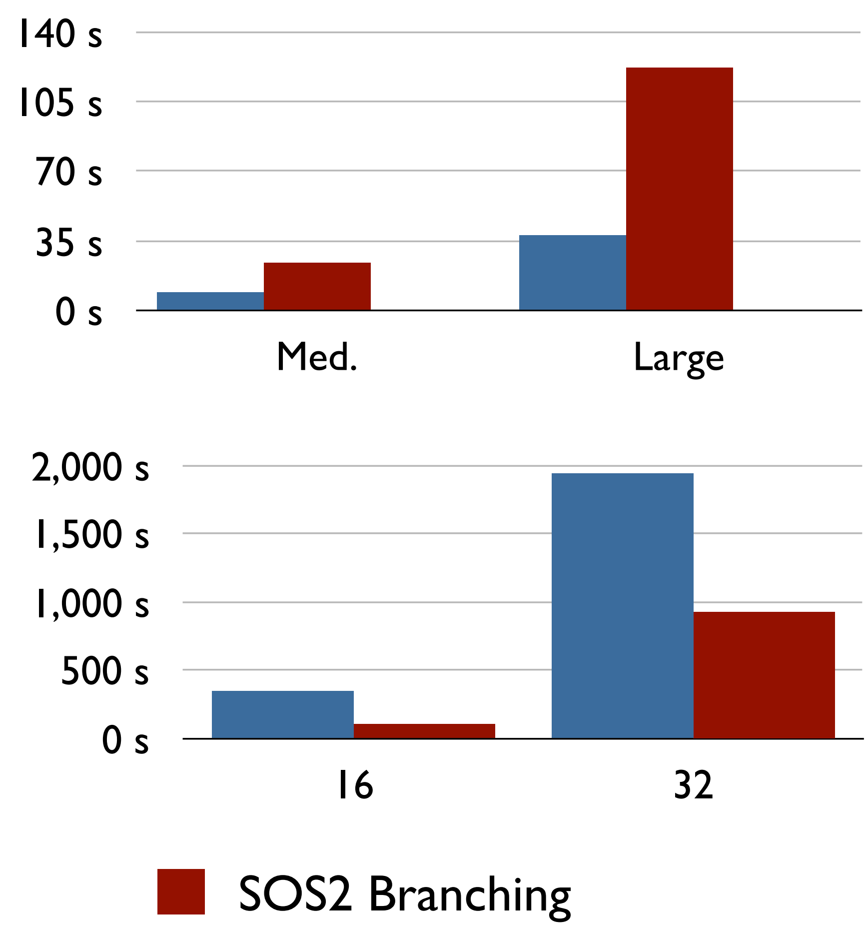
$$\text{and}$$

$$\text{if } \lambda_i > 0 \text{ and } \lambda_j > 0, \text{ then } |i - j| \leq 1$$

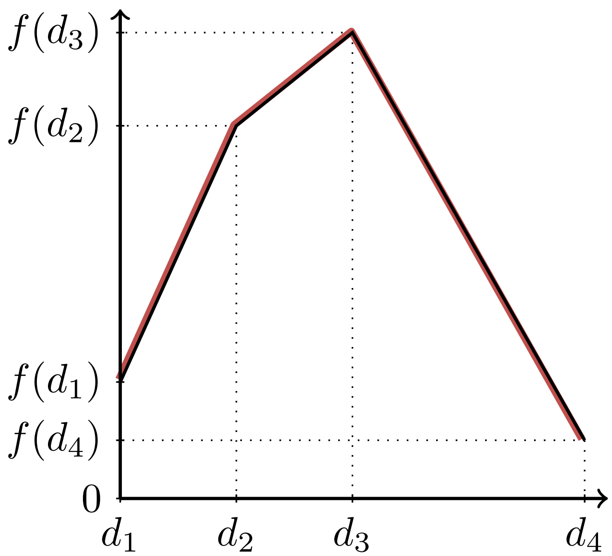
- Implemented by Gurobi and CPLEX

Effectiveness of SOS Branching in Practice

- Very sensitive to implementation of advanced branching techniques
  - CPLEX 9: Basic SOS2 branching implementation
  - CPLEX 11: Improved SOS2 branching implementation



Multiple Choice Plus SOS1 Constraints



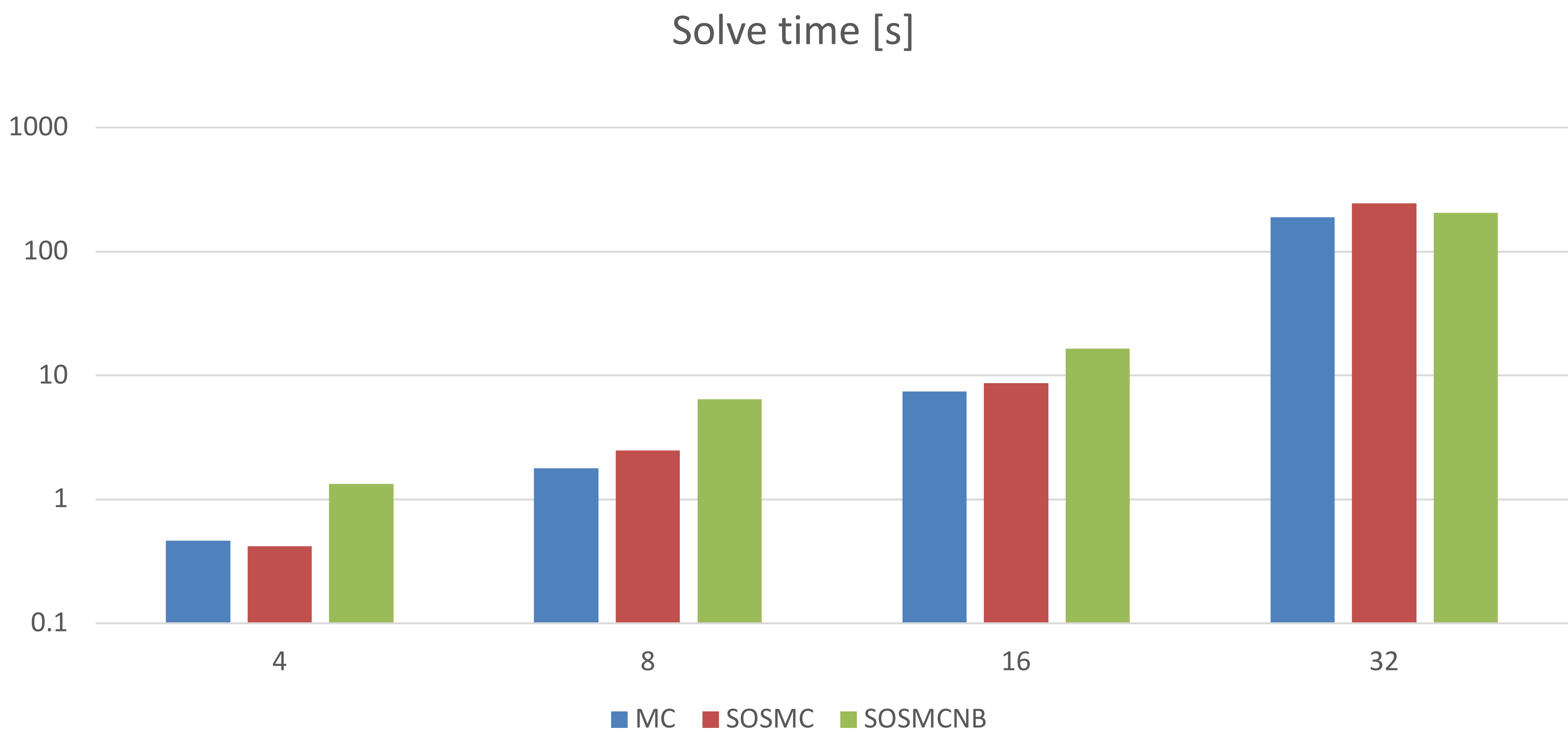
MC Formulation:

$$
\begin{aligned}
d_i y_i \le x^i \le d_{i+1} y_i & \quad \forall i \in [k] \\
m_i x^i + c_i y_i = z^i & \quad \forall i \in [k] \\
\sum_{i=1}^k x^i = x & \\
\sum_{i=1}^k z^i = z & \\
\sum_{i=1}^k y_i = 1 & \\
y \in \{0,1\}^k &
\end{aligned}
$$

- Plus SOS1 over $\{y_i\}_{i=1}^k$
  - SOSMC: keep integrality of y
  - SOSMCNB: relax integrality of y

Effect of SOS1 Constraints: Solve Time

- Gurobi 6.5 on Core i7-3770 (3.40GHz), 32 GB RAM.
  - Limit: 1000s (average over 5 instances)



Formulation for Discrete Alternatives

$$\sum_{i=1}^{n} \lambda_i = 1$$
$$\sum_{i=1}^{n} b^i \lambda_i = y$$
$$\lambda \in \mathbb{R}^n_+$$
$$y \in \{0,1\}^m$$

$$\Leftrightarrow$$

$$\sum_{i=1}^{n} \lambda_i = 1$$
$$\lambda \in \{0,1\}^n$$

$$\{b^i\}_{i=1}^{n} \subseteq \{0,1\}^m, \quad b^i \neq b^j \quad \forall i \neq j$$

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

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

Incremental Encoding

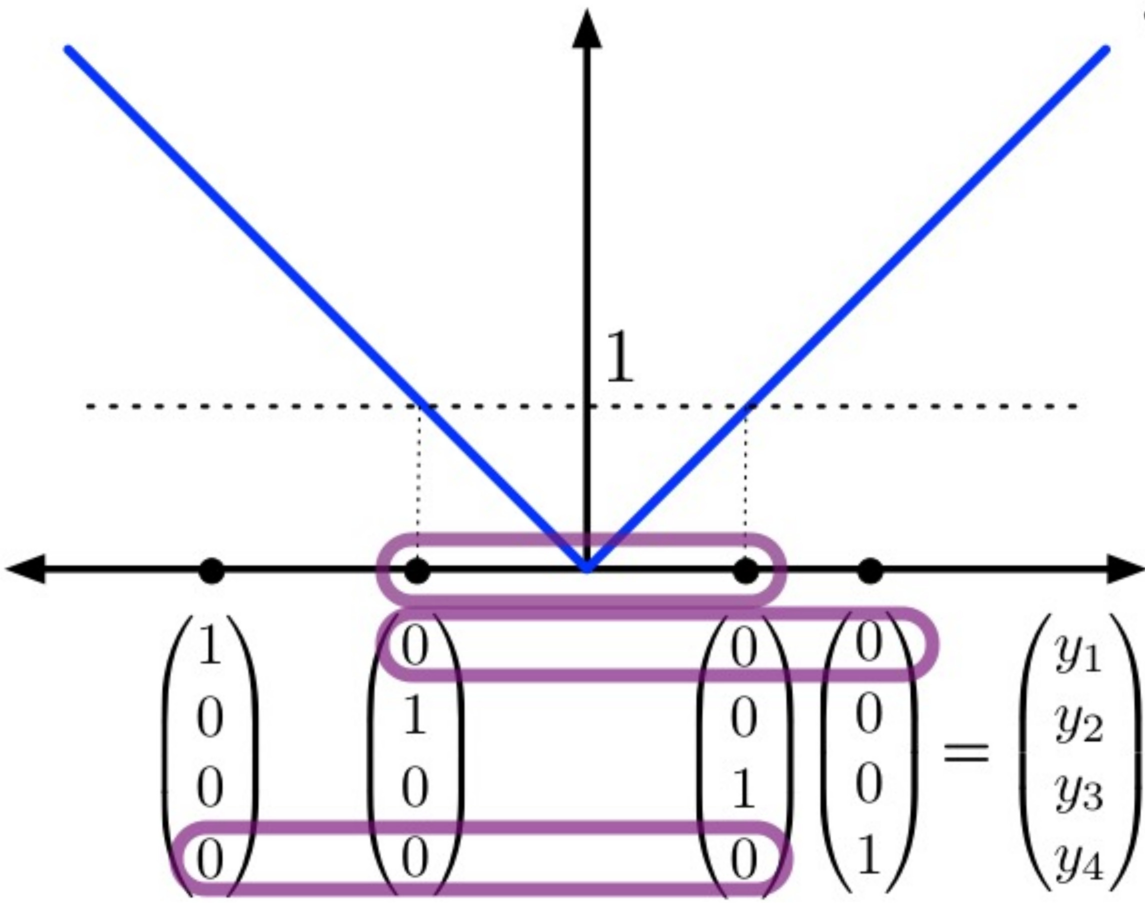
$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \lambda = y, \qquad \begin{array}{c} \sum_{i=1}^{8}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

$$\Downarrow$$

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

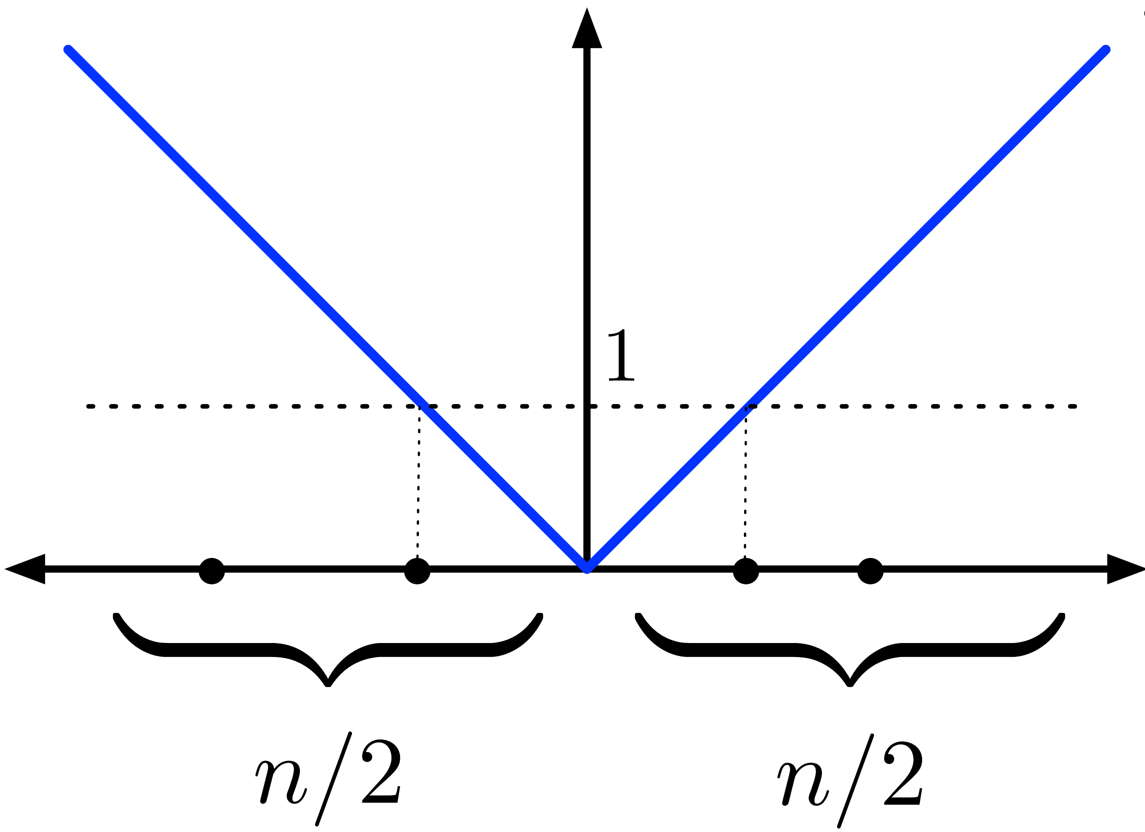
Example: Unary Encoding

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



$$y_1 = y_4 = 0$$

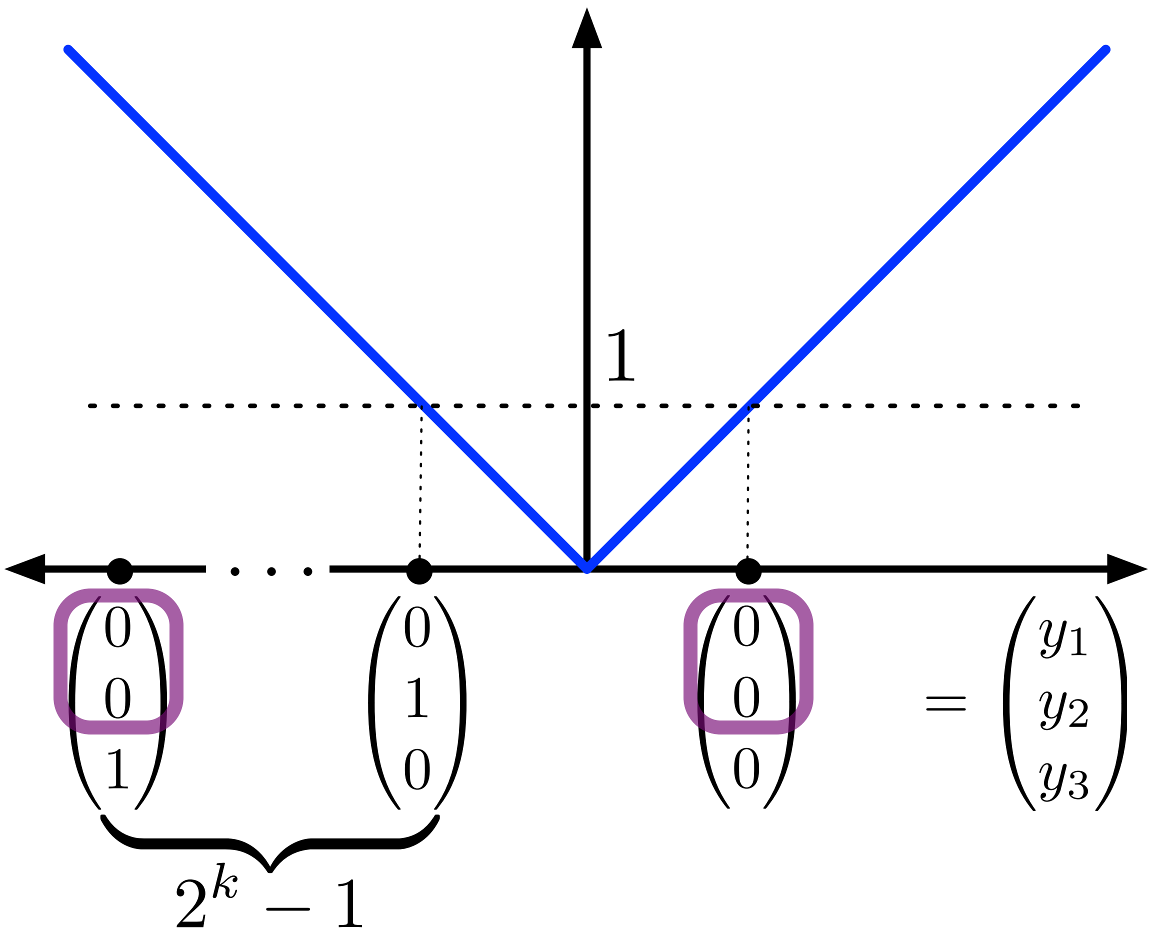
Example: Unary Encoding



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

Need $n/2$ branches to solve.

Example: Binary Encoding



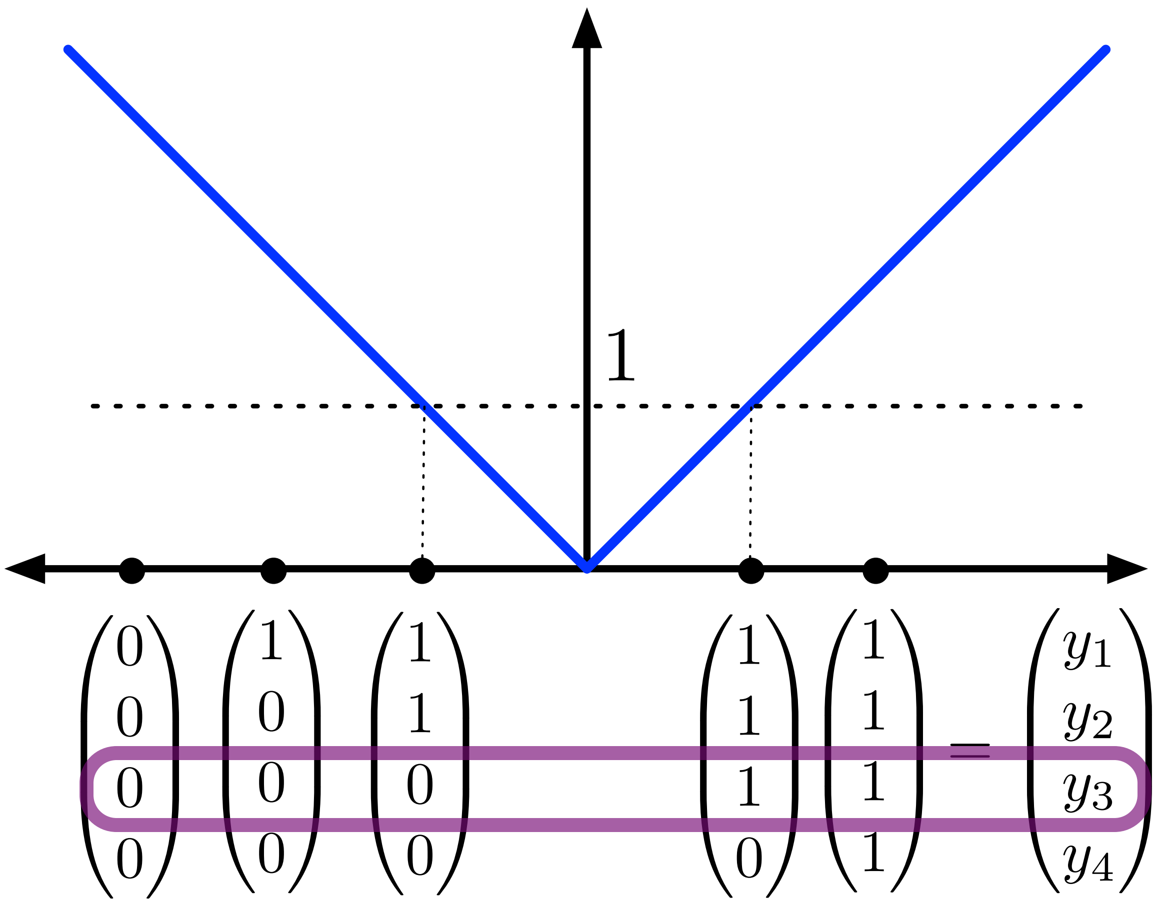
$\text{Best Bound} = 0$ unless:

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

Need $k = \log_2 n$ branches

$$y_1 = y_2 = 0$$

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

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

SOS1 Branching

$$\lambda_1 = \lambda_2 = 0$$

$$or$$

$$\lambda_3 = \lambda_4 = 0$$

Binary

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

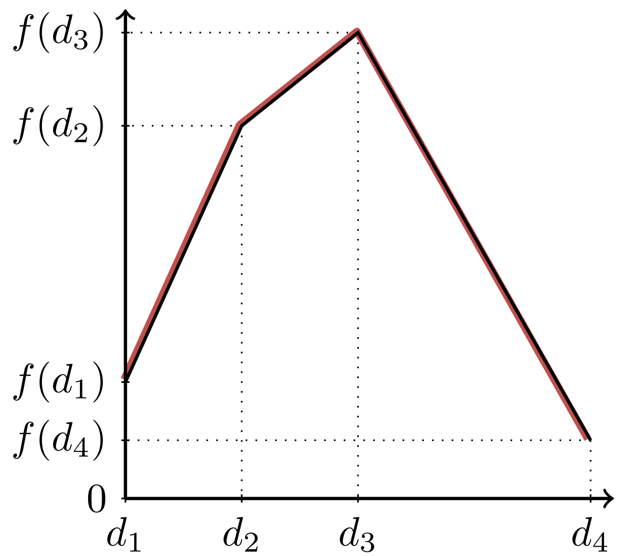
Odd/Even Branching

$$\lambda_1 = \lambda_3 = 0$$

$$or$$

$$\lambda_2 = \lambda_4 = 0$$

Multiple Choice Plus Encoding



MC Formulation:

$$d_i y_i \le x^i \le d_{i+1} y_i \qquad \forall i \in [k]$$

$$m_i x^i + c_i y_i = z^i \qquad \forall i \in [k]$$

$$\sum_{i=1}^k x^i = x$$

$$\sum_{i=1}^k z^i = z$$

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

$$\cancel{y \in \{0,1\}^k}$$

$$\sum_{i=1}^k b^i y_i = w$$

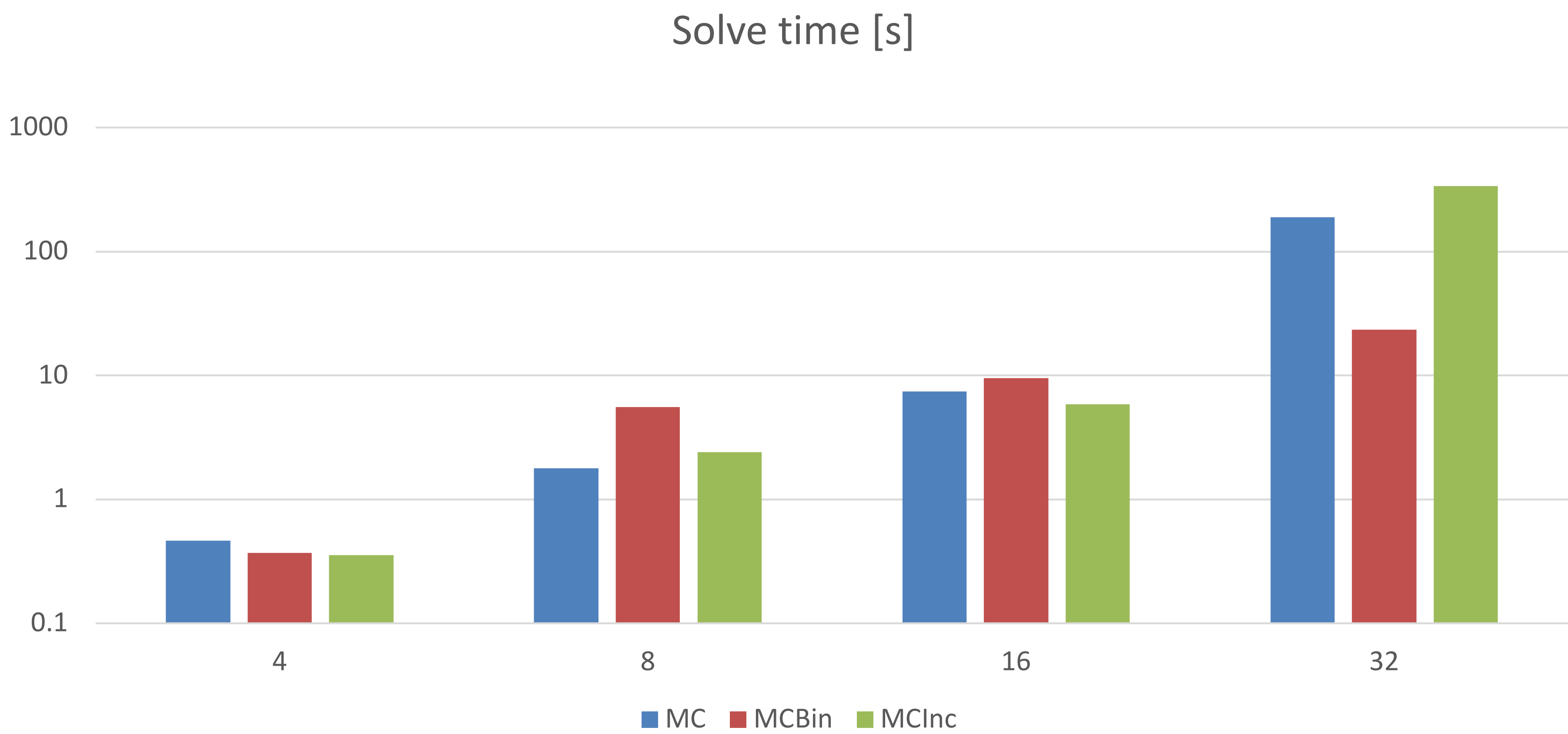
$$w \in \{0,1\}^m$$

$$y_i \ge 0 \quad \forall i \in [k]$$

- $\{b^i\}_{i=1}^k \subseteq \{0,1\}^m$
  - Binary encoding (MCBin)
  - Incremental encoding (MCInc)

Effect of Encoding Constraints: Solve Time

- Gurobi 6.5 on Core i7-3770 (3.40GHz), 32 GB RAM.
  - Limit: 1000s (average over 5 instances)



General Extreme Point Formulation

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

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

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

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

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

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

Also for general polyhedra
with common recession cones.

Combining with Alternative Encoding

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

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

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

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

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

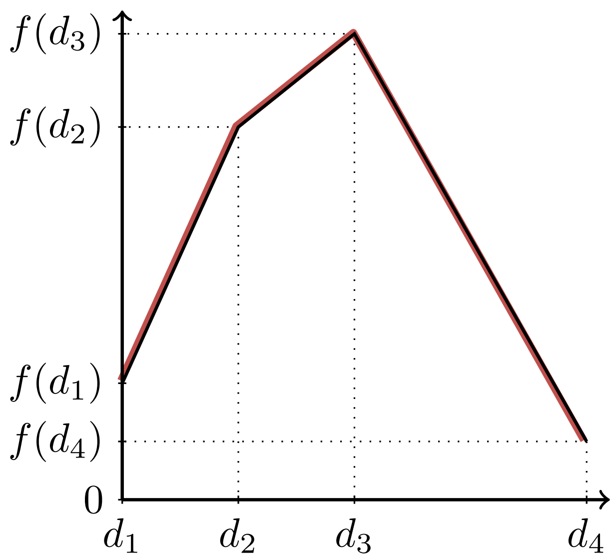
$$y \in \{0,1\}^m, \lambda_v^i \geq 0$$

Also for general polyhedra
with common recession cones.

Extended $\mathcal{V}$-formulation for PWL Functions

$$S = \mathrm{gr}\,(f) = \bigcup_{i=1}^{k} \left\{ (x,z) \in \mathbb{R}^2 \;:\; \begin{array}{r} d_i \leq x \leq d_{i+1} \\ m_i x + c_i = z \end{array} \right\}$$

DCC Formulation:



$$\sum_{i=1}^{k} \lambda_i^i d_i + \lambda_{i+1}^i d_{i+1} = x$$

$$\sum_{i=1}^{k} \lambda_i^i f(d_i) + \lambda_{i+1}^i f(d_{i+1}) = z$$

$$\lambda_i^i + \lambda_{i+1}^i = y_i \qquad \forall i \in [k]$$

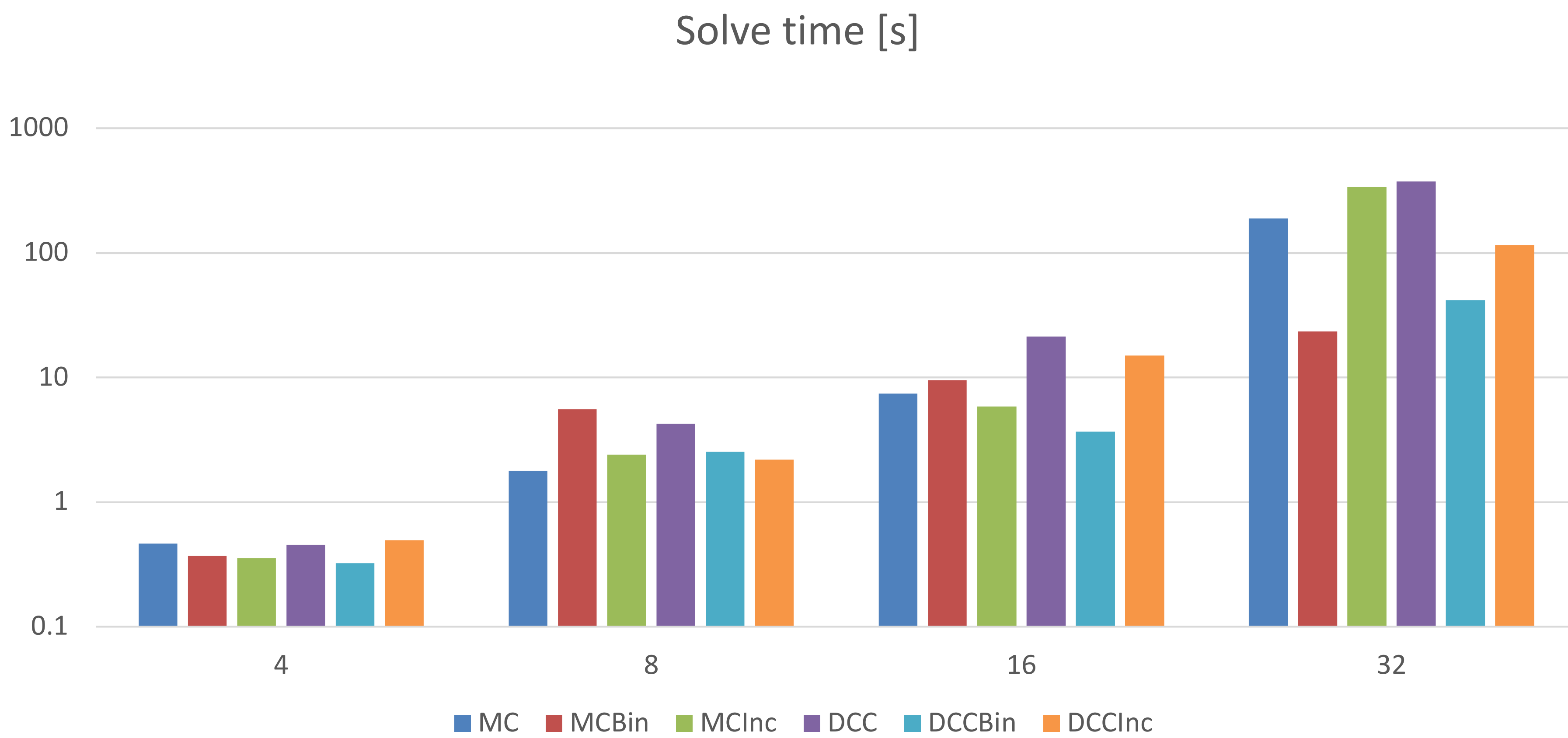
$$\sum_{i=1}^{k} y_i = 1$$

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

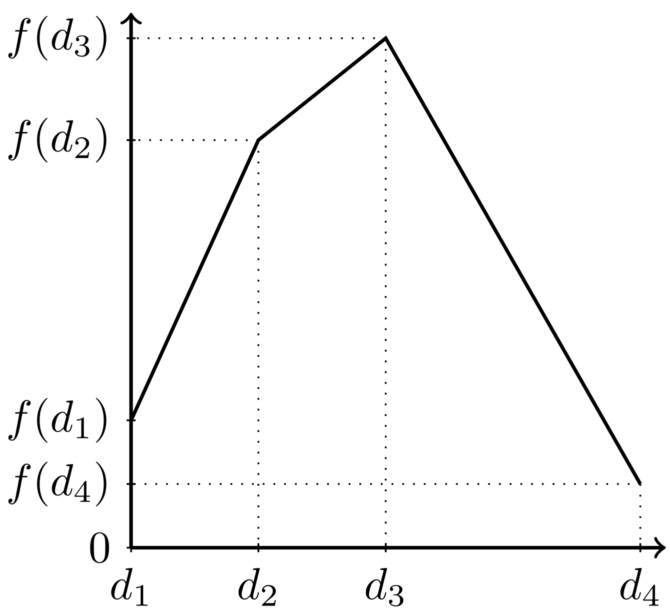
- DCC + Encoding
  - Binary encoding (DCCBin)
  - Incremental encoding (DCCInc)

Effect of Encoding Constraints: Solve Time

- Gurobi 6.5 on Core i7-3770 (3.40GHz), 32 GB RAM.
  - Limit: 1000s (average over 5 instances)



Integral Formulation



Formulation for f(x)=z

$$d_1 + \sum_{i=1}^{3} (d_{i+1} - d_i)\delta_i = x,$$

$$f(d_1) + \sum_{i=1}^{3} (f(d_{i+1}) - f(d_i))\delta_i = z$$

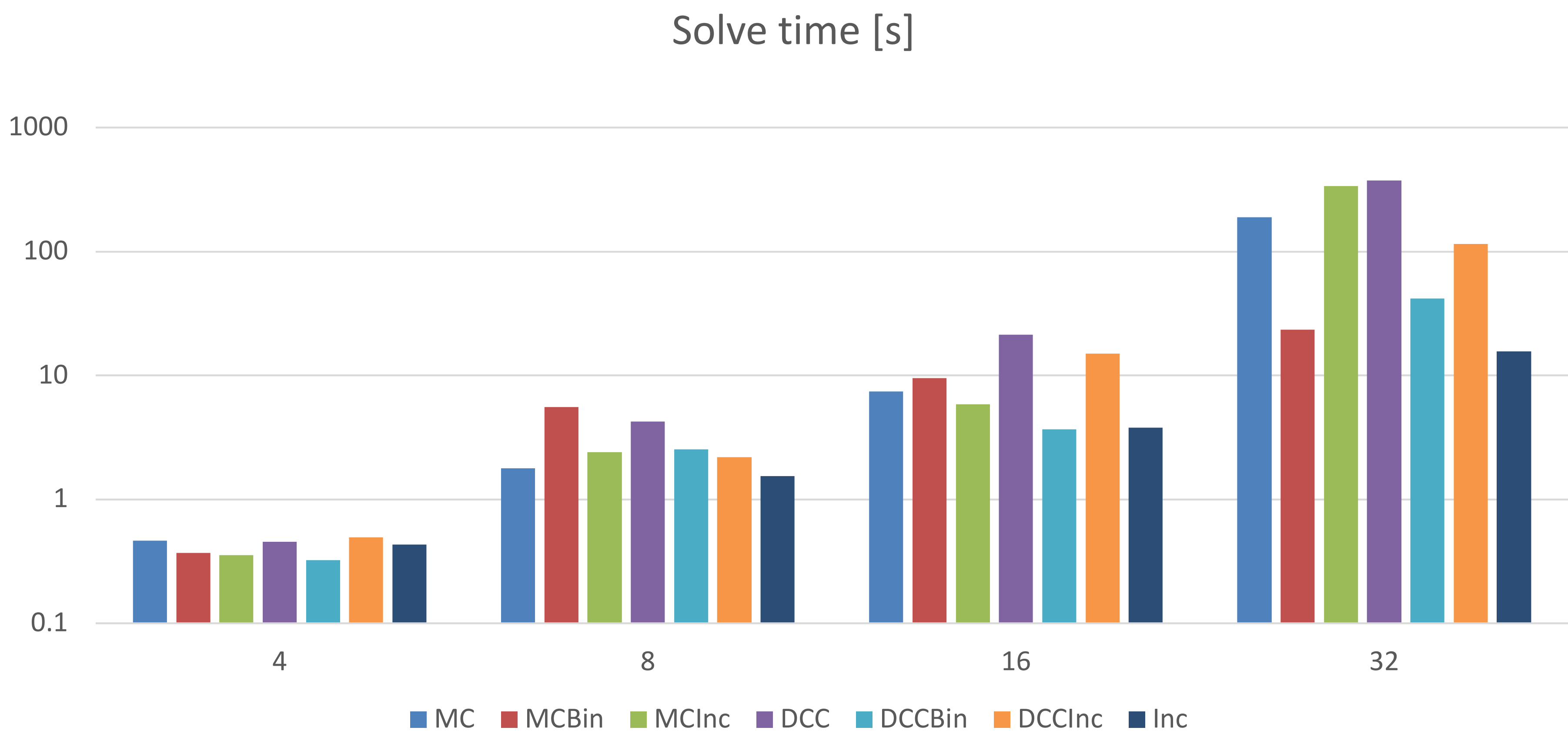
$$0 \le \delta_3 \le y_2 \le \delta_2 \le y_1 \le \delta_1 \le 1$$

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

Incremental Formulation: Inc
Affine transformation of DCCInc

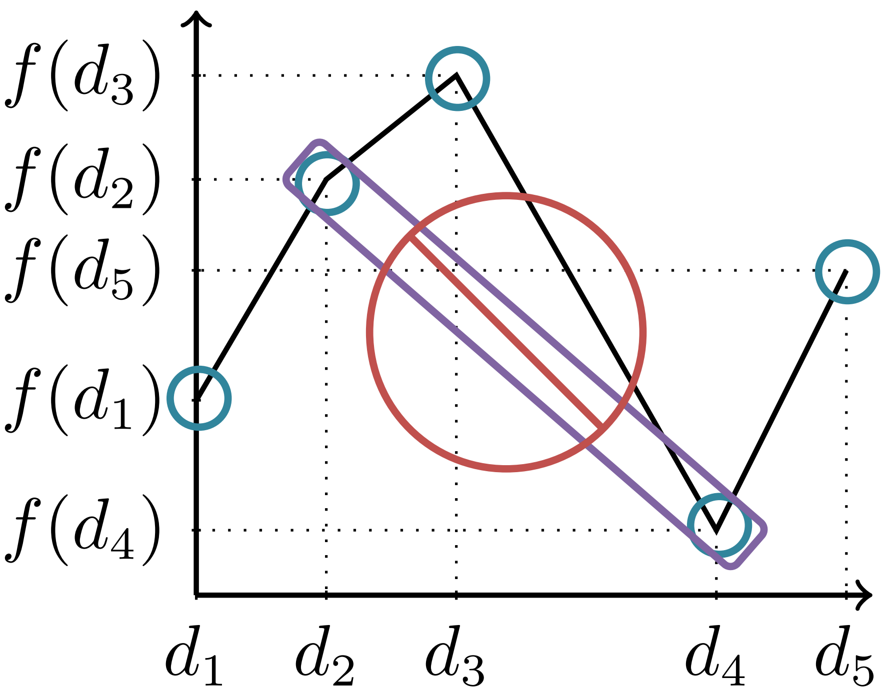
Effect of Encoding Constraints: Solve Time

- Gurobi 6.5 on Core i7-3770 (3.40GHz), 32 GB RAM.
  - Limit: 1000s (average over 5 instances)



Simple Formulation for Univariate Functions

$$z = f(x)$$



$$\text{Size} = O\left(\#\text{ of segments}\right)$$

Non-Ideal: Fractional Extreme Points

$$\begin{pmatrix} x \\ z \end{pmatrix} = \sum_{j=1}^{5} \begin{pmatrix} d_j \\ f(d_j) \end{pmatrix} \lambda_j$$

$$1 = \sum_{j=1}^{5} \lambda_j, \quad \lambda_j \geq 0$$

$$y \in \{0,1\}^4, \quad \sum_{i=1}^{4} y_i = 1$$

$$0 \leq \lambda_1 \leq y_1$$

$$0 \leq \lambda_2 \leq y_1 + y_2$$

$$0 \leq \lambda_3 \leq y_2 + y_3$$

$$0 \leq \lambda_4 \leq y_3 + y_4$$

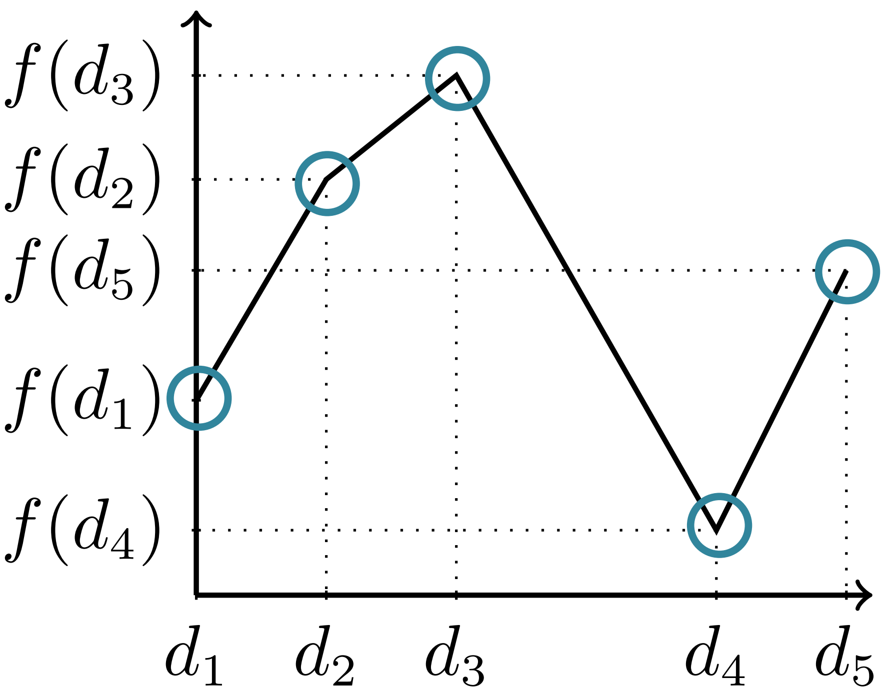
$$0 \leq \lambda_5 \leq y_4$$

Advanced Formulation for Univariate Functions

$$z = f(x)$$

$$\begin{pmatrix} x \\ z \end{pmatrix} = \sum_{j=1}^{5} \begin{pmatrix} d_j \\ f(d_j) \end{pmatrix} \lambda_j$$

$$1 = \sum_{j=1}^{5} \lambda_j, \quad \lambda_j \geq 0$$



$$y \in \{0,1\}^2$$

$$0 \leq \lambda_1 + \lambda_5 \leq 1 - y_1$$

$$0 \leq \lambda_3 \leq y_1$$

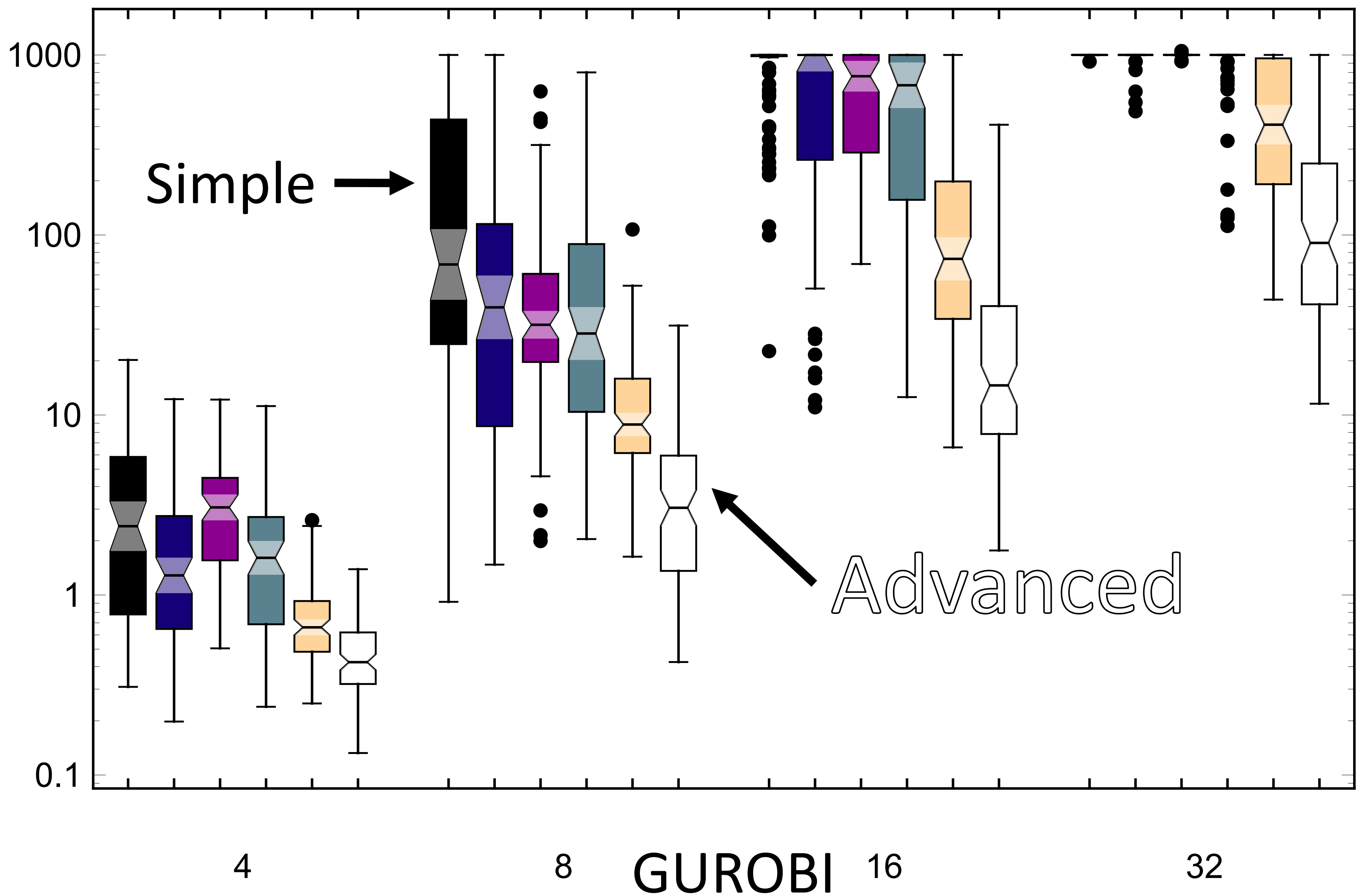
$$0 \leq \lambda_4 + \lambda_5 \leq 1 - y_2$$

$$0 \leq \lambda_1 + \lambda_2 \leq y_2$$

$$\text{Size} = O(\log_2 \# \text{ of segments})$$

Ideal: Integral Extreme Points

Formulation Improvements can be Significant



Can we do even better?

Yes, by focusing on branching

More elaborate: SOS3(26)