Embedding Formulations and Complexity for Unions of Polyhedra

Juan Pablo Vielma

Massachusetts Institute of Technology

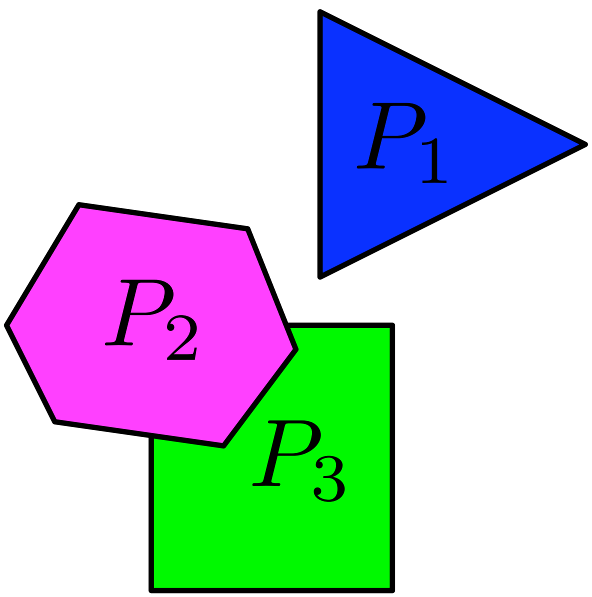
Workshop on Combinatorial Optimization,
Mathematisches Forschungsinstitut Oberwolfach,
Oberwolfach, Germany. November, 2014.

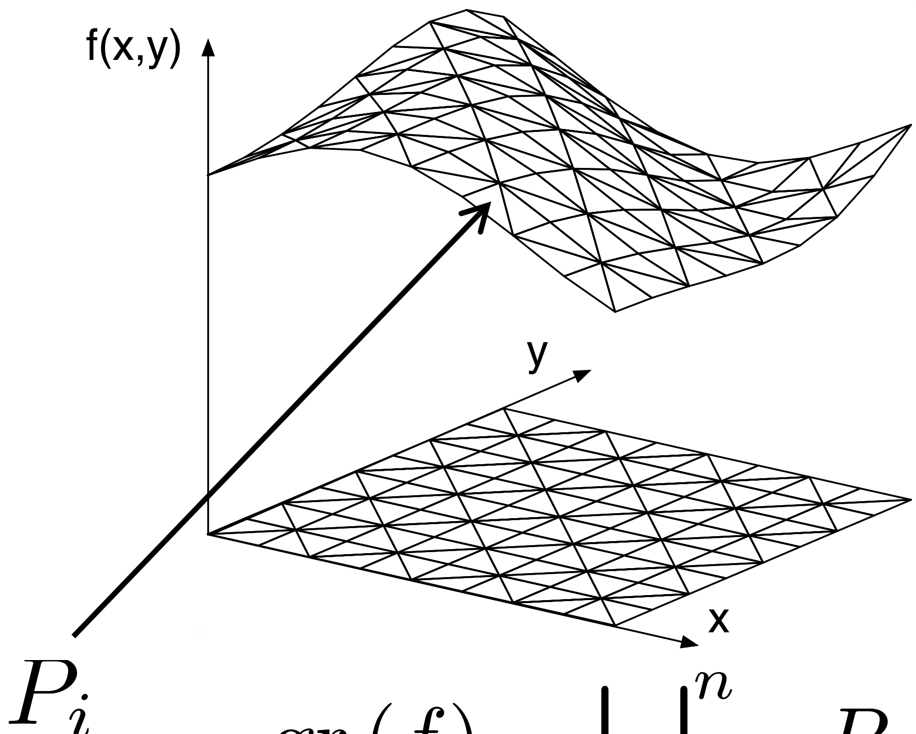
Supported by NSF grant CMMI-1351619

(Linear) Mixed <u>0-1</u> Integer Formulations

- Modeling Finite Alternatives = Unions of Polyhedra

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





$$\mathrm{gr}\,(f) = \bigcup_{i=1}^{n} P_i$$

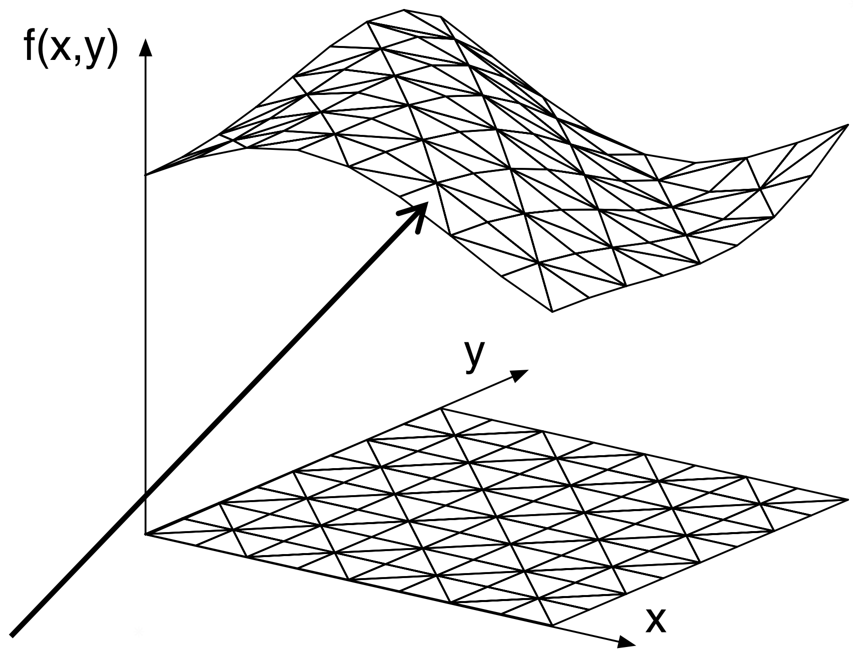
(Linear) Mixed 0-1 Integer Formulations

- Modeling Finite Alternatives = Unions of Polyhedra

$$\min \quad \sum_{j=1}^{m} f_j(x_j, y_j)$$

$$s.t.$$

$$(x, y) \in X$$



$$P_i$$

$$\operatorname{gr}(f) = \bigcup_{i=1}^{n} P_i$$

Outline

- Introduction
  - Classical Formulations v/s Specialized Branching
- Encoding Formulations
  - Role of Binary Variables and Specialized Branching
- Embedding Formulations
  - Smallest Strong Formulations

Strong Extended Formulations for $x \in \bigcup_{i=1}^{n} P_i$

- Balas, Jeroslow and Lowe ’70s early ’80s

$$P_i = \left\{ x \in \mathbb{R}^d \ : \ A^i x \le b^i \right\}$$

$$\begin{aligned} A^i x^i &\le b^i y_i \quad \forall i \\ \sum_{i=1}^{n} x^i &= x \\ \sum_{i=1}^{n} y_i &= 1 \\ y &\in \{0,1\}^n \end{aligned}$$

$\mathcal{H}$-formulation

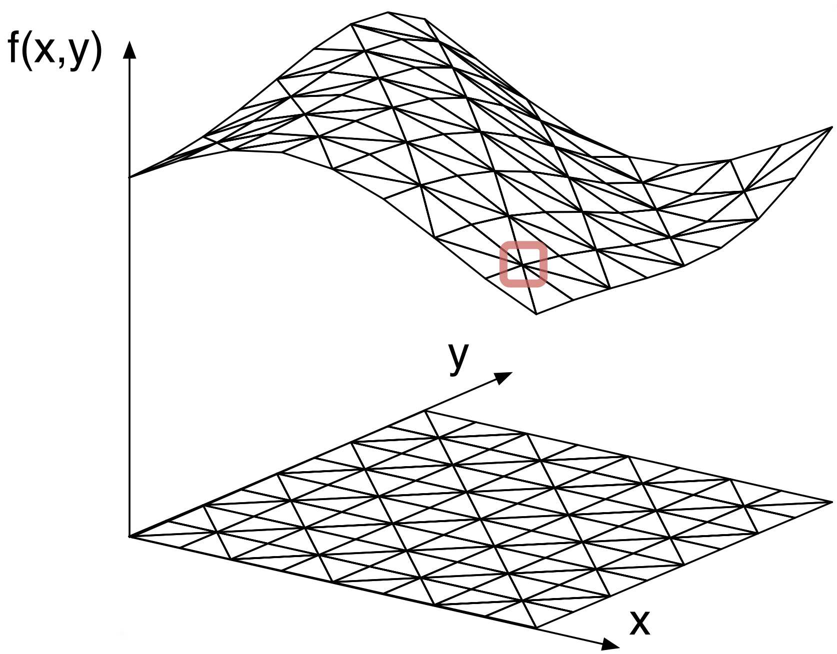
$$\begin{aligned} \sum_{i=1}^{n} \sum_{v \in \operatorname{ext}(P_i)} v \lambda_v^i &= x \\ \sum_{v \in \operatorname{ext}(P_i)} \lambda_v^i &= y_i \quad \forall i \\ \sum_{i=1}^{n} y_i &= 1 \\ \lambda^i &\in \mathbb{R}_+^{\operatorname{ext}(P_i)} \\ y &\in \{0,1\}^n \end{aligned}$$

$\mathcal{V}$-formulation

- **Convex Hull** (Sharp) = LP relaxation projects to $\operatorname{conv}\left(\bigcup_{i=1}^{n} P_i \right)$
- **Integral** (Locally Ideal) = LP relaxation has integral extreme points (y)

Strong Extended Formulations for $x \in \bigcup_{i=1}^n P_i$

- Balas, Jeroslow and Lowe ’70s early ’80s



$$
\begin{aligned}
\sum_{i=1}^n \sum_{v \in \operatorname{ext}(P_i)} v\boxed{\lambda_v^i} &= x \\
\sum_{v \in \operatorname{ext}(P_i)} \lambda_v^i &= y_i \quad \forall i \\
\sum_{i=1}^n y_i &= 1 \\
\lambda^i &\in \mathbb{R}_+^{\operatorname{ext}(P_i)} \\
y &\in \{0,1\}^n
\end{aligned}
$$

$\mathcal{V}$-formulation

- **Convex Hull** (Sharp) = LP relaxation projects to $\operatorname{conv}\left(\bigcup_{i=1}^n P_i\right)$
- **Integral** (Locally Ideal) = LP relaxation has integral extreme points (y)

“Strong” ~~Extended~~ Formulations for $x \in \bigcup_{i=1}^{n} P_i$

- Balas, Blair and Jeroslow late ‘80s

$$P_i = \left\{ x \in \mathbb{R}^d \;:\; A^{\cancel{i}} x \le b^i \right\}$$

$$\boxed{\begin{aligned} Ax &\le \sum\nolimits_{i=1}^{n} b^i y_i \\ \sum\nolimits_{i=1}^{n} y_i &= 1 \\ y &\in \{0,1\}^n \end{aligned}}$$

$\mathcal{H}$-formulation

- Lee and Wilson late ‘90s

$$V := \bigcup\nolimits_{i=1}^{n} \operatorname{ext}(P_i)$$

$$\boxed{\begin{aligned} \sum\nolimits_{v \in V} v \lambda_v &= x \\ \sum\nolimits_{v \in V} \lambda_v &= 1 \\ \lambda_v &\le \sum\nolimits_{i : v \in \operatorname{ext}(P_i)} y_i \\ \sum\nolimits_{i=1}^{n} y_i &= 1 \\ y \in \{0,1\}^n, \quad \lambda &\in \mathbb{R}_+^V \end{aligned}}$$

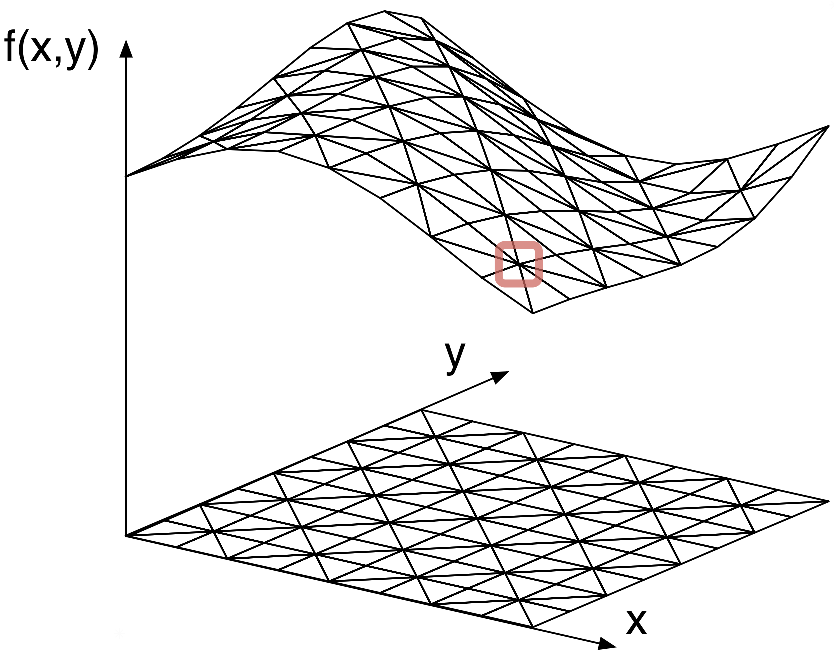
$\mathcal{V}$-formulation

Sometimes

- **Convex Hull** (Sharp) = LP relaxation projects to $\operatorname{conv}\left(\bigcup_{i=1}^{n} P_i\right)$
- ~~**Integral** (Locally Ideal) = LP relaxation has integral extreme points (y)~~

“Strong” ~~Extended~~ Formulations for $x \in \bigcup_{i=1}^{n} P_i$

- Balas, Blair and Jeroslow late ‘80s



- Lee and Wilson late ‘90s

$$V := \bigcup_{i=1}^{n} \operatorname{ext}(P_i)$$

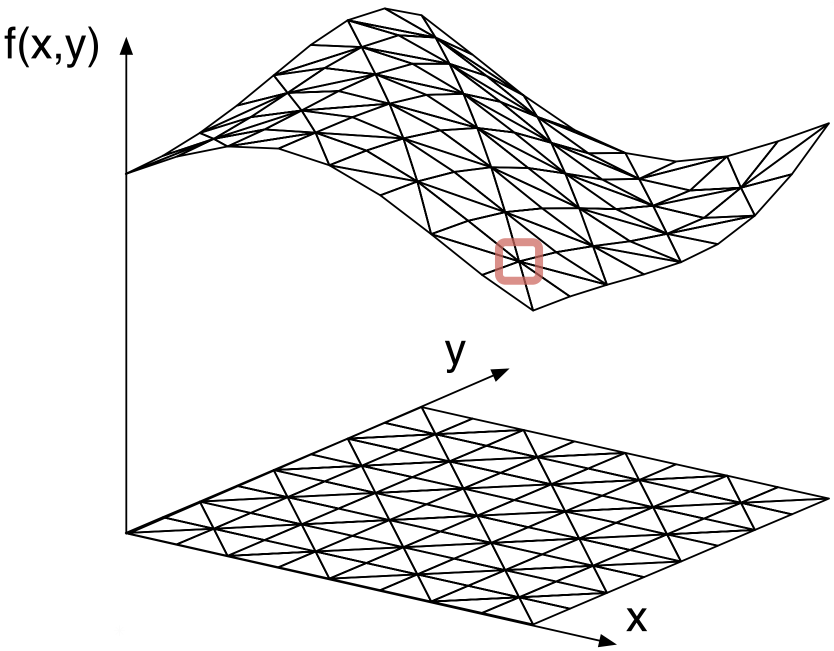
$$\begin{aligned}
\sum_{v \in V} \boxed{v \lambda_v} &= x \\
\sum_{v \in V} \lambda_v &= 1 \\
\lambda_v &\leq \sum_{i : v \in \operatorname{ext}(P_i)} y_i \\
\sum_{i=1}^{n} y_i &= 1 \\
y \in \{0,1\}^n, \quad & \lambda \in \mathbb{R}_+^V
\end{aligned}$$

$\mathcal{V}$-formulation

- **Convex Hull** (Sharp) = LP relaxation projects to $\operatorname{conv}\left(\bigcup_{i=1}^{n} P_i\right)$
- ~~**Integral** (Locally Ideal) = LP relaxation has integral extreme points (y)~~

"Strong" Projected Formulations for $x \in \bigcup_{i=1}^{n} P_i$

- Balas, Blair and Jeroslow late '80s



- Lee and Wilson late '90s

$$V := \bigcup_{i=1}^{n} \operatorname{ext}(P_i)$$

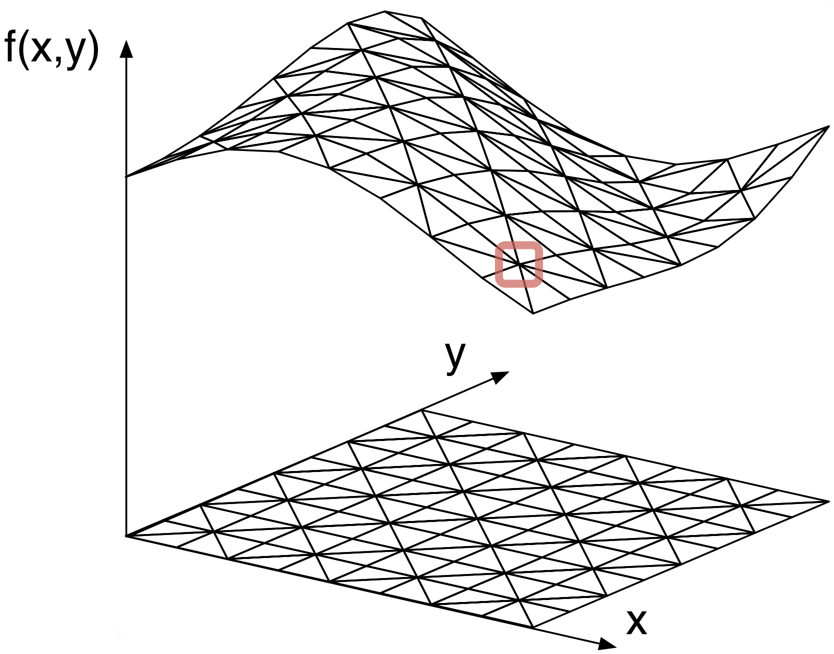
$$\boxed{\begin{aligned} \sum_{v\in V} v\boxed{\lambda_v} &= x \\ \sum_{v\in V} \lambda_v &= 1 \\ \lambda_v &\le \sum_{i:v\in \operatorname{ext}(P_i)} y_i \\ \sum_{i=1}^{n} y_i &= 1 \\ y \in \{0,1\}^n, \quad \lambda &\in \mathbb{R}_+^V \end{aligned}}$$

$\mathcal{V}$-formulation

- **Convex Hull** (Sharp) = LP relaxation projects to $\operatorname{conv}\left(\bigcup_{i=1}^{n} P_i\right)$
- ~~**Integral** (Locally Ideal) = LP relaxation has integral extreme points (y)~~

“Strong” Projected Formulations for $x \in \bigcup_{i=1}^{n} P_i$

- Balas, Blair and Jeroslow late ‘80s



- Lee and Wilson late ‘90s

$$V := \bigcup_{i=1}^{n} \operatorname{ext}(P_i)$$

$$\sum_{v \in V} v \boxed{\lambda_v} = x$$

$$\sum_{v \in V} \lambda_v = 1$$

$$\lambda_v \leq \sum_{i : v \in \operatorname{ext}(P_i)} y_i$$

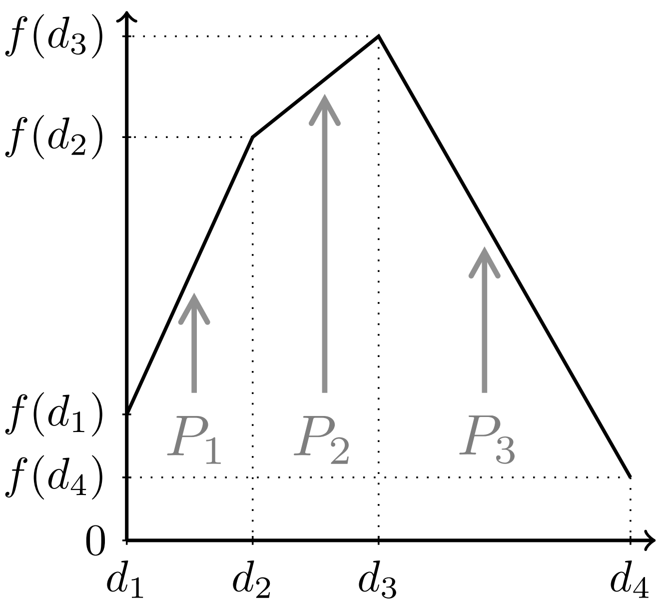
$$\sum_{i=1}^{n} y_i = 1$$

$$y \in \{0,1\}^n, \quad \lambda \in \mathbb{R}_+^V$$

$\mathcal{V}$-formulation

- **Convex Hull** (Sharp) = LP relaxation projects to $\operatorname{conv}\left(\bigcup_{i=1}^{n} P_i\right)$
- ~~**Integral** (Locally Ideal) = LP relaxation has integral extreme points (y)~~

Projected Formulation for Univariate Functions



$$\sum_{j=1}^{4} d_j \lambda_{d_j} = x, \qquad \sum_{j=1}^{4} f(d_j)\lambda_{d_j} = z$$

$$\sum_{j=1}^{4} \lambda_{d_j} = 1, \qquad \lambda_{d_j} \geq 0$$

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

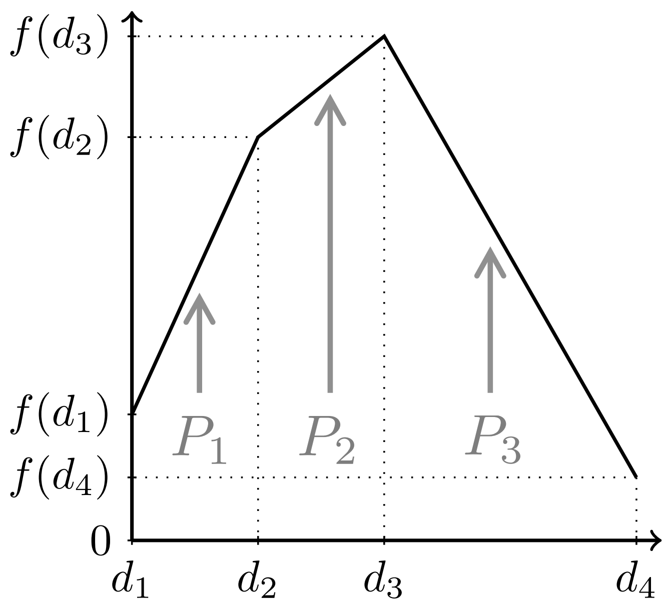
$$\lambda_{d_1} \leq y_1, \qquad \lambda_{d_2} \leq y_1 + y_2$$

$$\lambda_{d_3} \leq y_2 + y_3, \quad \lambda_{d_4} \leq y_3$$

- Convex Hull, but not Integral
- Branching is very ineffective (unbalanced B&B tree)
  - $y_{i_0} = 1 \qquad \Rightarrow \qquad y_i = 0 \quad \forall i \neq i_0$

    $y_{i_0} = 0$ does not imply much (anything)

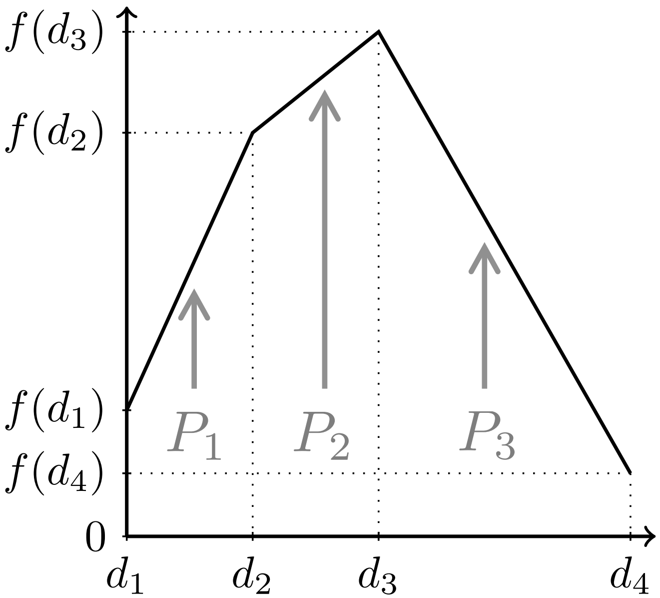
Projected Formulation for Univariate Functions



$$
\begin{aligned}
&\sum\nolimits_{j=1}^{4} d_j \lambda_{d_j} = x, && \sum\nolimits_{j=1}^{4} f(d_j)\lambda_{d_j} = z \\
&\sum\nolimits_{j=1}^{4} \lambda_{d_j} = 1, && \lambda_{d_j} \geq 0 \\
&\cancel{\sum\nolimits_{i=1}^{3} y_i = 1,} && \cancel{y \in \{0,1\}^3} \\
&\cancel{\lambda_{d_1} \leq y_1,} && \cancel{\lambda_{d_2} \leq y_1 + y_2} \\
&\cancel{\lambda_{d_3} \leq y_2 + y_3,} && \cancel{\lambda_{d_4} \leq y_3}
\end{aligned}
$$

- One solution = SOS2 branching (Beale and Tomlin ’70):
  - $\lambda_{d_i} = 0 \quad \forall i \leq i_0 - 1$
  - $\lambda_{d_i} = 0 \quad \forall i \geq i_0 + 1$

Projected Formulation for Univariate Functions



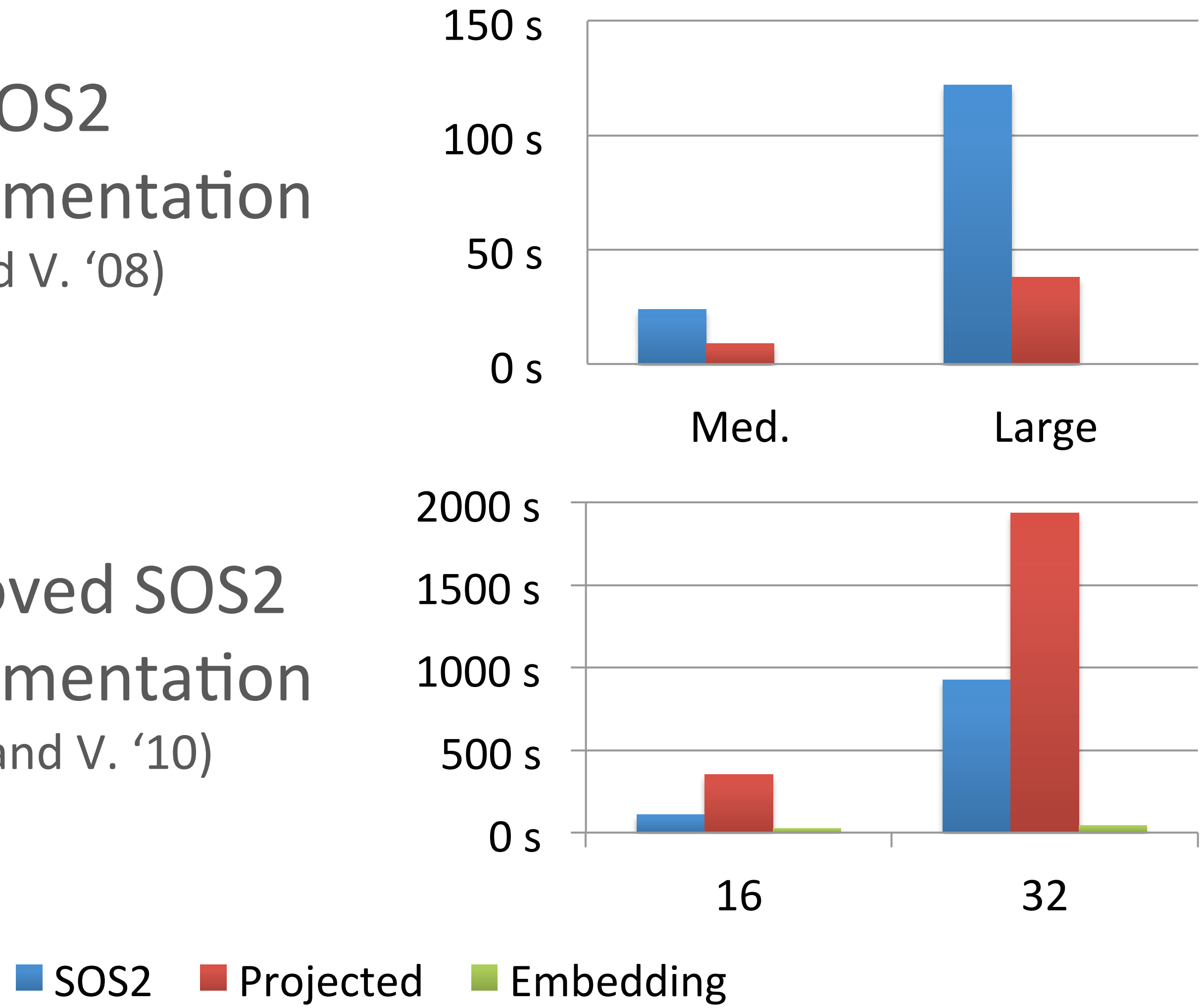
$$\sum_{j=1}^{4} d_j \lambda_{d_j} = x, \qquad \sum_{j=1}^{4} f(d_j)\lambda_{d_j} = z$$
$$\sum_{j=1}^{4} \lambda_{d_j} = 1, \qquad \lambda_{d_j} \geq 0$$
$$\sum_{i=1}^{3} y_i = 1, \qquad y \in \{0,1\}^3$$
$$\lambda_{d_1} \leq y_1, \qquad \lambda_{d_2} \leq y_1 + y_2$$
$$\lambda_{d_3} \leq y_2 + y_3, \qquad \lambda_{d_4} \leq y_3$$

- One solution = SOS2 branching (Beale and Tomlin ’70):
  - $\lambda_{d_i} = 0 \quad \forall i \leq i_0 - 1$ $\qquad y_i = 0 \quad \forall i \leq i_0 - 1$
  - $\lambda_{d_i} = 0 \quad \forall i \geq i_0 + 1$ $\qquad y_i = 0 \quad \forall i \geq i_0$

MIP Formulations v/s Specialized Branching

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

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



Encoding Formulations:
The Role of Binary Variables

Encodings to Induce Specialized Branching

- Discrete alternatives ($P_i = \{v^i\}$) :

$$\sum_{i=1}^{n} y_i v^i = x, \quad \sum_{i=1}^{n} y_i = 1$$
$$y \in \{0,1\}^n$$

Encodings to Induce Specialized Branching

- Discrete alternatives ($P_i = \{v^i\}$) :

$$\sum_{i=1}^{n} y_i v^i = x, \quad \sum_{i=1}^{n} y_i = 1$$

$$\cancel{y \in \{0,1\}^n} \quad y \in \mathbb{R}_+^n$$

$$\sum_{i=1}^{n} y_i h^i = w, \quad w \in \{0,1\}^k$$

- Pick $\{h^i\}_{i=1}^{n} \subseteq \{0,1\}^k$, $\quad h^i \neq h^j$

Encoding

- Li and Lu '09, Adams and Henry '11, V. and Nemhauser '08 for $k = \log_2 n$. Also in the folklore, e.g. Sommer, TIMS '72

Different Encodings = Different Branching

- **Unary** encoding : $\left\{h^i\right\}_{i=1}^n = \left\{e^i\right\}_{i=1}^n$

$$\sum_{i=1}^{8} y_i = 1, \quad y \in \mathbb{R}_+^8$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} y = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \end{pmatrix}, \quad w \in \{0,1\}^8$$

$$\Rightarrow y_i = w_i$$

Different Encodings = Different Branching

- **Binary** encoding : $\left\{h^i\right\}_{i=1}^{n} = \{0,1\}^{\log_2 n}$

$$\sum_{i=1}^{8} y_i = 1, \quad y \in \mathbb{R}^8_+$$

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array}$$
$$\begin{pmatrix} \boxed{1 & 1 & 1 & 1} & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} y = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}, \quad w \in \{0,1\}^3$$

$$\boxed{w_1 = 1} \quad \vee \quad w_1 = 0$$
$$\Updownarrow$$
$$\boxed{y_i = 0 \quad \forall i \geq 5} \quad \vee \quad y_i = 0 \quad \forall i \leq 4$$

Discrete Alternatives to Unions of Polyhedra

Adapt extended $\mathcal{V}$-formulation:

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{v \in \operatorname{ext}(P_i)} v\lambda_v^i &= x \\
\sum_{v \in \operatorname{ext}(P_i)} \lambda_v^i &= y_i \ \forall i \\
\sum_{i=1}^{n} y_i &= 1 \\
\lambda^i &\in \mathbb{R}_+^{\operatorname{ext}(P_i)} \\
y &\in \{0,1\}^n
\end{aligned}$$

Discrete Alternatives to Unions of Polyhedra

Adapt extended $\mathcal{V}$-formulation:

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

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

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

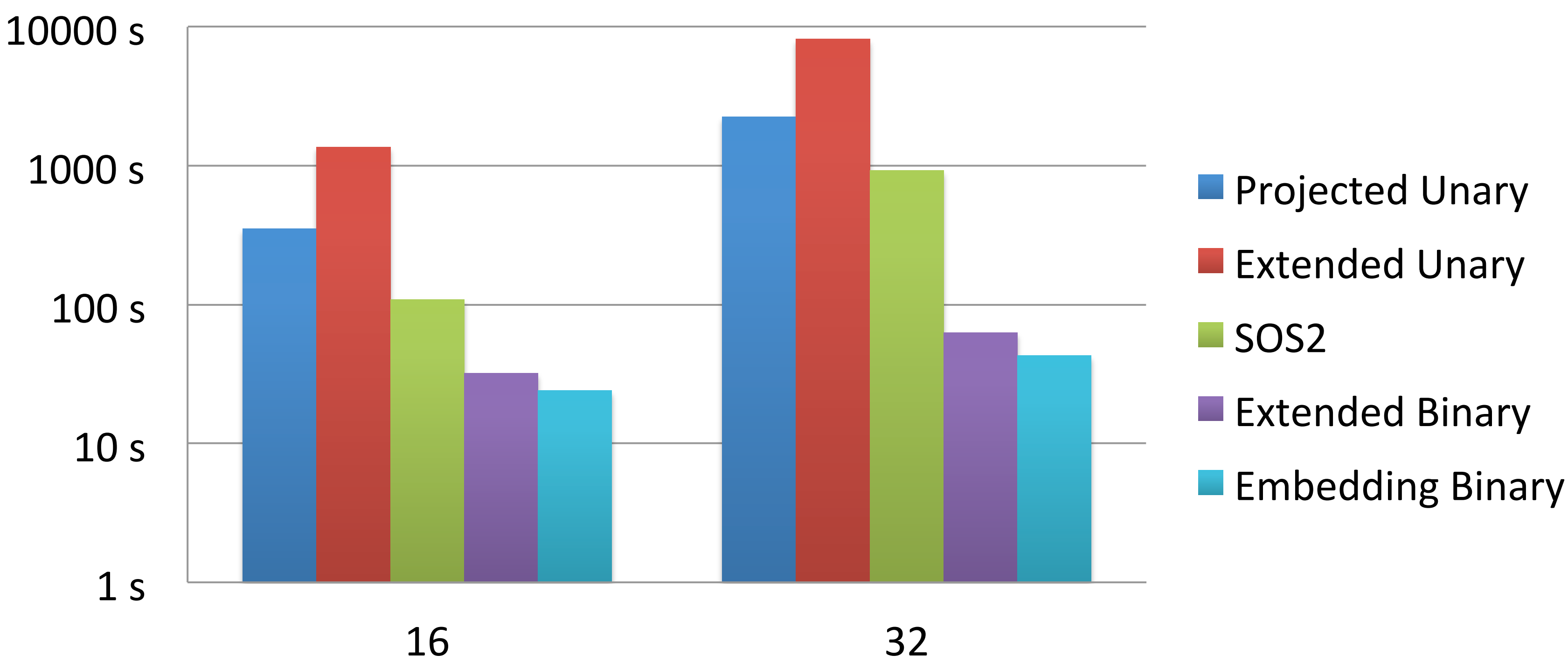
$$\lambda^i \in \mathbb{R}_+^{\operatorname{ext}(P_i)}$$

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

- V., Ahmed and Nemhauser 2010; Yıldız and V. 2013; V. 2014

Performance for Univariate Functions

- Results from Nemhauser, Ahmed and V. ’10 using CPLEX 11



- Multivariate functions : **Embedding Binary** is **6** times faster than **Extended Binary**

Embedding Formulations: Strong Projected Formulations

Polyhedra as MIP Formulations

$$\lambda \in \bigcup_{i=1}^{n} P_i, \qquad P_i = \left\{\lambda \in \mathbb{R}^d \;:\; A^i \lambda \leq b^i\right\}$$

$$Q = \left\{ (\lambda, y) \in \mathbb{R}^d \times \mathbb{R}^n \;:\; \begin{array}{r l} A\lambda \leq & \sum_{i=1}^{n} b^i y_i \\ 1 = & \sum_{i=1}^{n} y_i \\ y_i \geq & 0 \end{array} \right\}$$

$$y \in \mathbb{Z}^n$$

$$\left(\lambda, e^i\right) \in Q \quad \Leftrightarrow \quad \lambda \in P_i$$

Embedding Formulations for Union of Polyhedra

- **Projected** MIP formulation of $\lambda \in \bigcup_{i=1}^{n} P_i \subseteq \mathbb{R}^V$:
  - Encoding $\{h^i\}_{i=1}^n \subseteq \{0,1\}^k, \quad h^i \neq h^j$
  - Polyhedron $Q \subseteq \mathbb{R}^V \times \mathbb{R}^k$, s.t.

$$(\lambda, h^i) \in Q \quad \Leftrightarrow \quad \lambda \in P_i$$

- **Embedding formulation** = strongest polyhedron:

Cayley Polytope $\longrightarrow$

$$Q = \operatorname{conv}\Big(\underbrace{\bigcup_{i=1}^{n} P_i \times \{h^i\}}_{\text{Cayley Embedding}}\Big)$$

For unary encoding:

$$h^i = e^i$$

Embedding Formulations for Union of Polyhedra

- **Projected** MIP formulation of $\lambda \in \bigcup_{i=1}^{n} P_i \subseteq \mathbb{R}^V$:
  - Encoding $\{h^i\}_{i=1}^n \subseteq \{0,1\}^k, \quad h^i \neq h^j$
  - Polyhedron $Q \subseteq \mathbb{R}^V \times \mathbb{R}^k$, s.t.

$$(\lambda, h^i) \in Q \quad \Leftrightarrow \quad \lambda \in P_i$$

- **Embedding formulation** = strongest polyhedron:

$$Q = \operatorname{conv}\left(\bigcup_{i=1}^{n} P_i \times \{h^i\}\right)$$

$$\text{size}\,(Q) := \#\text{ of facets of } Q \text{ (usually function of } n)$$

Binary v/s Unary Encodings

$$Q = \mathrm{conv}\left(\bigcup_{i=1}^{n} P_i \times \left\{h^i\right\}\right), \quad \left\{h^i\right\}_{i=1}^{n} \subseteq \left\{0,1\right\}^k$$

- Unary better than Binary ?
  - Formulation contains convex hull through **projection**:
    - $\mathrm{Proj}_\lambda\left(Q\right) = \mathrm{conv}\left(\bigcup_{i=1}^{n} P_i\right)$
    - $\mathrm{size}\left(\mathrm{Proj}_\lambda\left(Q\right)\right) \leq \binom{\mathrm{size}(Q)}{\mathrm{size}(Q)-k-1}$
  - **Binary encoding** has $k = \log_2 n$:
    - Size of projection is at most quasipolynomial in size of formulation
  - **Unary encoding** has $k = n$:
    - Size of projection can be exponential in size of formulation

Binary v/s Unary Encodings

$$Q = \operatorname{conv}\left(\bigcup_{i=1}^{n} P_i \times \{h^i\}\right), \quad \{h^i\}_{i=1}^{n} \subseteq \{0,1\}^k$$

- Binary better than Unary?
  - Formulation contains Minkowski sum through **sections**:
    - For **unary encoding**

$$\left(\lambda, \frac{1}{n}\sum_{i=1}^{n} e^i\right) \in Q \quad \Leftrightarrow \quad \lambda \in \frac{1}{n}P_1 + \ldots + \frac{1}{n}P_n$$

    - **Unary encoding** formulation can be large even if convex hull is simple(x)
    - **Binary encoding** seems to only contain partial sums of $\log_2 n$ polytopes

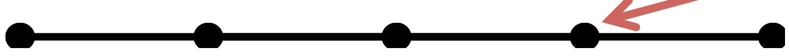
Simple Case: Combinatorial Part of $\mathcal{V}$-formulation

- $\Delta^V := \left\{ \lambda \in \mathbb{R}^V_+ \; : \; \sum_{v \in V} \lambda_v = 1 \right\}, \operatorname{ext}\left(P_i\right) = T_i \subseteq V$
- $P_i = \left\{ \lambda \in \Delta^V \; : \; \lambda_v \leq 0 \quad \forall v \notin T_i \right\}$

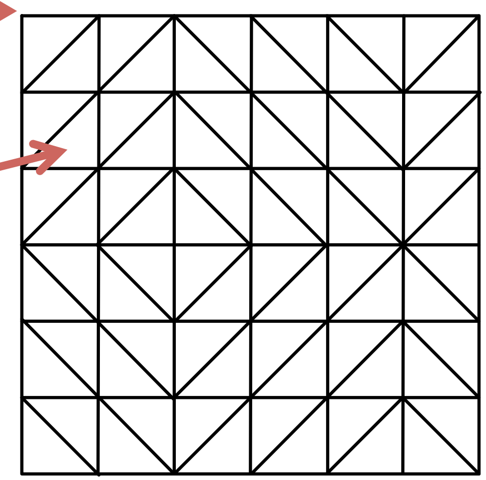
$$
\lambda \in \bigcup_{i=1}^n P_i \qquad \Leftrightarrow \qquad
\boxed{
\begin{aligned}
\sum_{v \in V} v \lambda_v &= x \\
\sum_{v \in V} \lambda_v &= 1 \\
\lambda_v &\leq \sum_{i : v \in \operatorname{ext}(P_i)} y_i \\
\sum_{i=1}^n y_i &= 1 \\
y \in \{0,1\}^n, \quad & \lambda \in \mathbb{R}^V_+
\end{aligned}
}
$$

Simple Case: Combinatorial Part of $\mathcal{V}$-formulation

- $\Delta^V := \left\{ \lambda \in \mathbb{R}_+^V : \sum_{v \in V} \lambda_v = 1 \right\}, \operatorname{ext}(P_i) = T_i \subseteq V$
- $P_i = \left\{ \lambda \in \Delta^V : \lambda_v \leq 0 \quad \forall v \notin T_i \right\}$



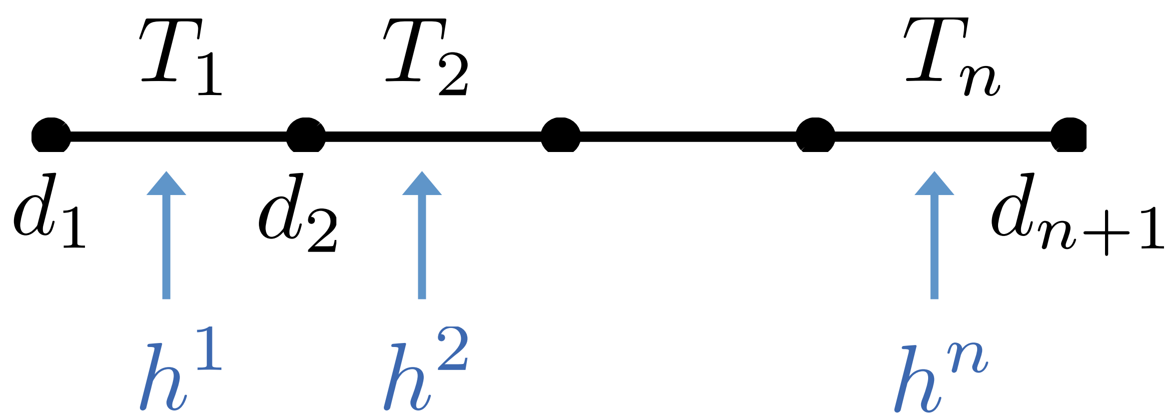
$$\lambda \in \bigcup_{i=1}^n P_i$$



- $\operatorname{conv}\left(\bigcup_{i=1}^n P_i \right) = \Delta^V$

Message 1: The Devil is in the Detail

- Choice of binary encoding is crucial



Formulation Size for Univariate case

- Simple facets: $\lambda_v \geq 0$
  - Only sometimes are facets
  - "Zero" computational cost and at most n of them
- All other facets: $\sum_{v \in V} \alpha_v \lambda_v \leq \sum_{i=1}^{k} \beta_i y_i$
- Unary encoding (Padberg, Lee and Wilson, early 00's):
  - $2n$ facets $(2n + 2$ including bounds$)$
- Binary encoding with Gray code (V. and Nemhauser, 08, 11):
  - $\log_2 n$ facets $(\leq 2 \log_2 n + n$ including bounds$)$

High Binary Complexity? Gray v/s Anti-Gray

- Assumption: $n = 2^k$
  - $\left\{h^i\right\}_{i=1}^{n} = \{0,1\}^k, \quad H := \left\{h^i - h^{i+1}\right\}_{i=1}^{n-1} \subseteq \{-1,0,1\}^k$
  - # facets =twice the # of **linear** hyperplanes spanned by $\boldsymbol{H}$
- Gray code: $\left\{h^i - h^{i+1}\right\}_{i=1}^{n-1} \equiv \left\{e^i\right\}_{i=1}^{k}$
  - # hyperplanes : $k = \log_2 n$, # facets $\leq 2\log_2 n + n$
- One kind of Anti-Gray code: $\left\{h^i - h^{i+1}\right\}_{i=1}^{n-1}$ "$\supseteq$" $\{-1,1\}^k$
  - # hyperplanes = # affine hyperplanes spanned by $\{0,1\}^{k-1}$
  - Using believed growth rate (e.g. Aichholzer and Auremacher '96):
    - # facets = $\Theta\left(n^{\log_2 n}\right)$

Message 2 : Binary Encoding = Smaller Formulation

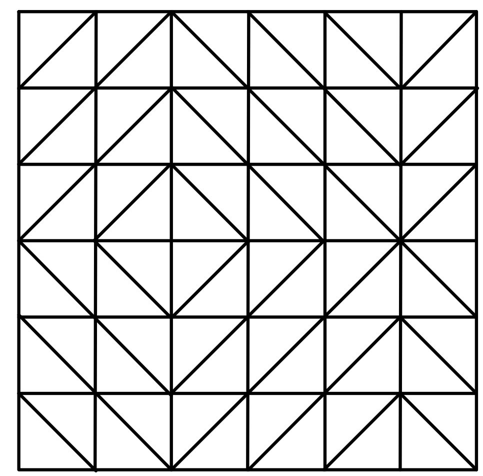
- Size of unary formulation is at least (Lee and Wilson '01):

$$\binom{2\sqrt{n/2}}{\sqrt{n/2}} + \underbrace{\left(\sqrt{n/2}+1\right)^2}_{\text{Non-negativity}}$$

- Size of best binary formulation for union jack triangulation is at most (V. and Nemhauser '08):

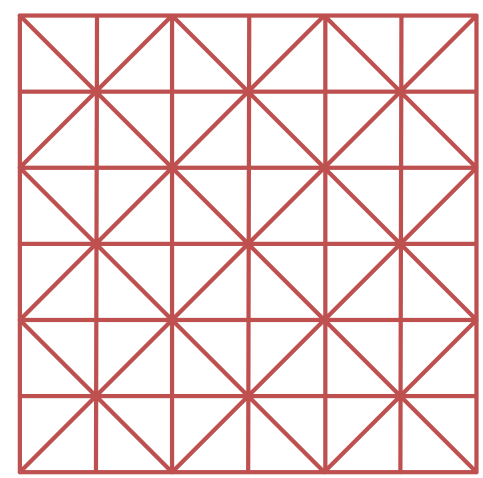
$$4\log_2\sqrt{n/2} + 2 + \underbrace{\left(\sqrt{n/2}+1\right)^2}_{\text{Non-negativity}}$$

$n = 2m^2$



m

m



Beyond Union Jack: Exploit Redundancy

- Embedding-like formulation for triangulations with "even degree outside the boundary"

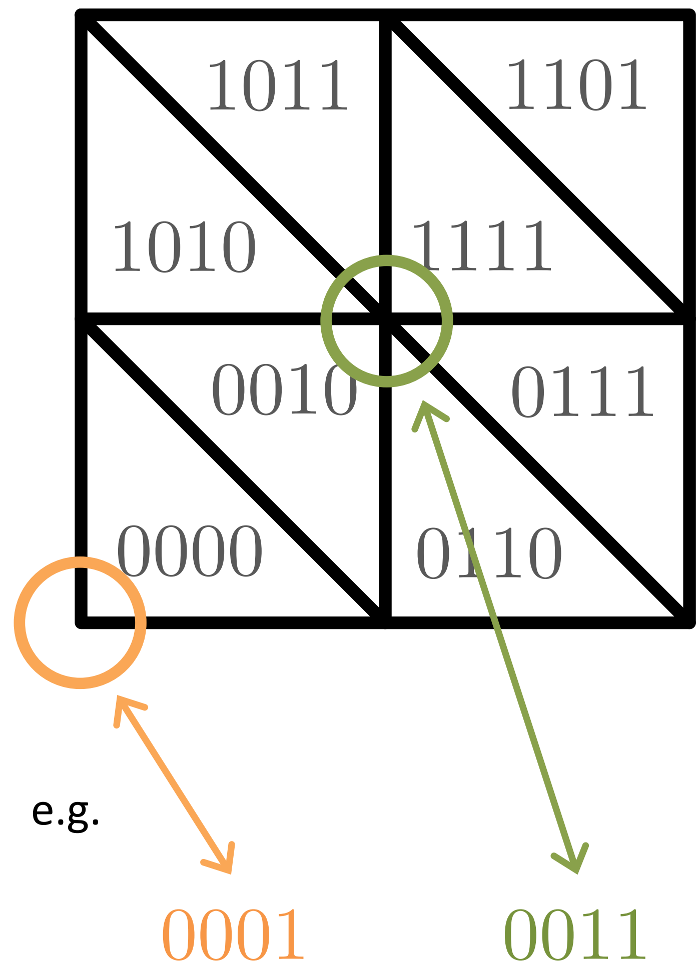
•Formulation size at most two larger than for union jack:

$$4\log_2\sqrt{n/2}+4+\left(\sqrt{n/2}+1\right)^2$$

•Formulation fits **independent branching** framework (V. and Nemhauser '08)

Independent Branching = Embedding + Redundancy

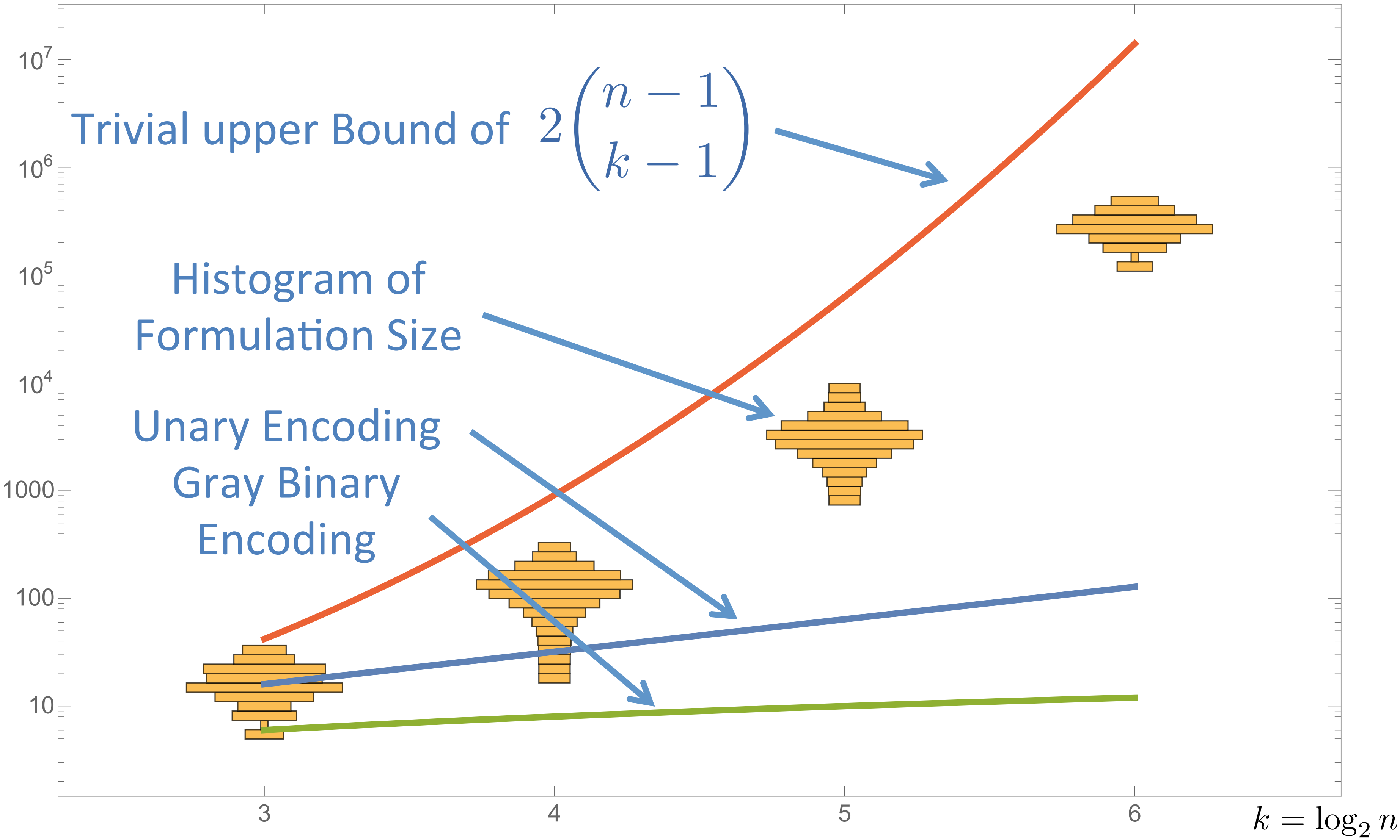
- Triangle $\longleftarrow$ binary vector
- More vectors than triangles
  - Ind. Branch ≠ Embedding
  - Embedding size is larger (17)
- Ind. Branching solution:
  - Add redundant single-vertex polytopes with remaining 8 binary vectors
- Unary cannot reduce size through redundancy



Summary

- Embedding Formulations = Systematic procedure
  - Encoding can significantly affect size
  - Redundancy can help for binary encodings
- Complexity of Union of Polyhedra beyond convex hull
  - Embedding Complexity (Integral Formulation)
  - MIP formulation complexity
- More on encoding properties
  - All gray codes yield the same size, but not combinatorially equivalent polytopes
- Can help discover strong (non-integral) formulations
  - Facility layout problem (Huchette, Dey, V. '14)

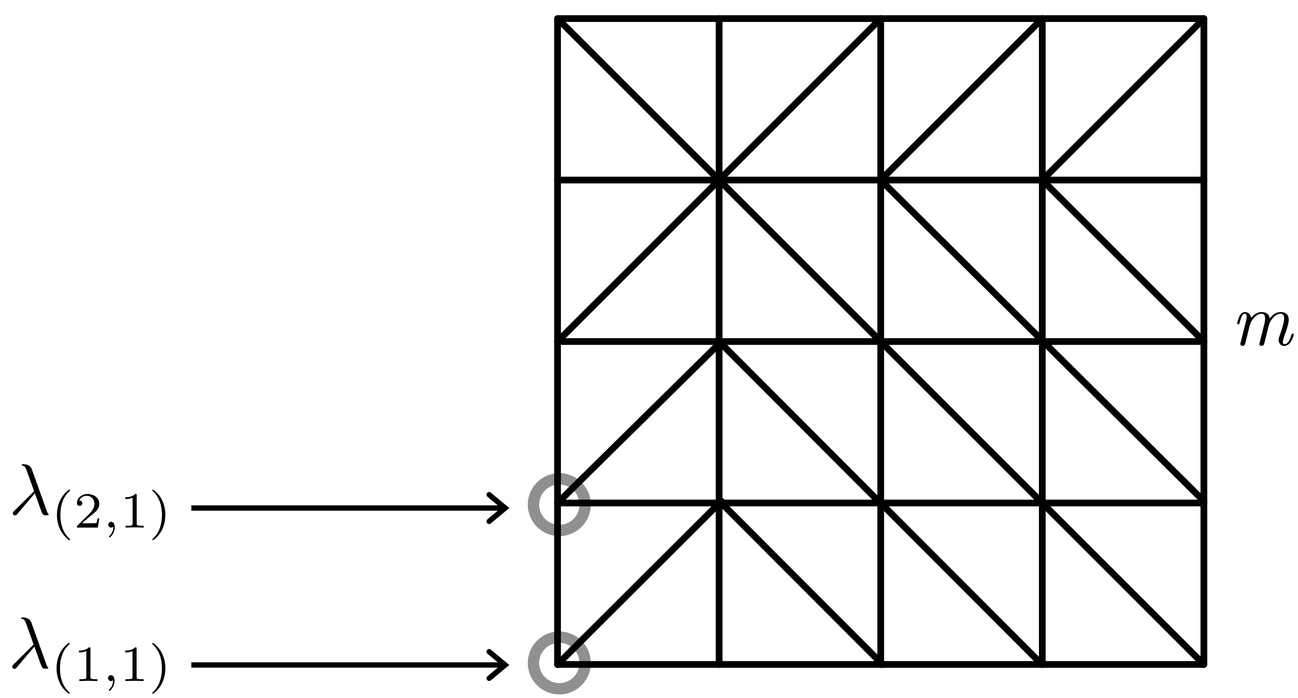
Formulation Size for all Binary Encodings



Beyond Union Jack: Part I = Gray Code for Grid

$\lambda_{(i,j)} \geq 0 \quad i,j \in [m+1]$

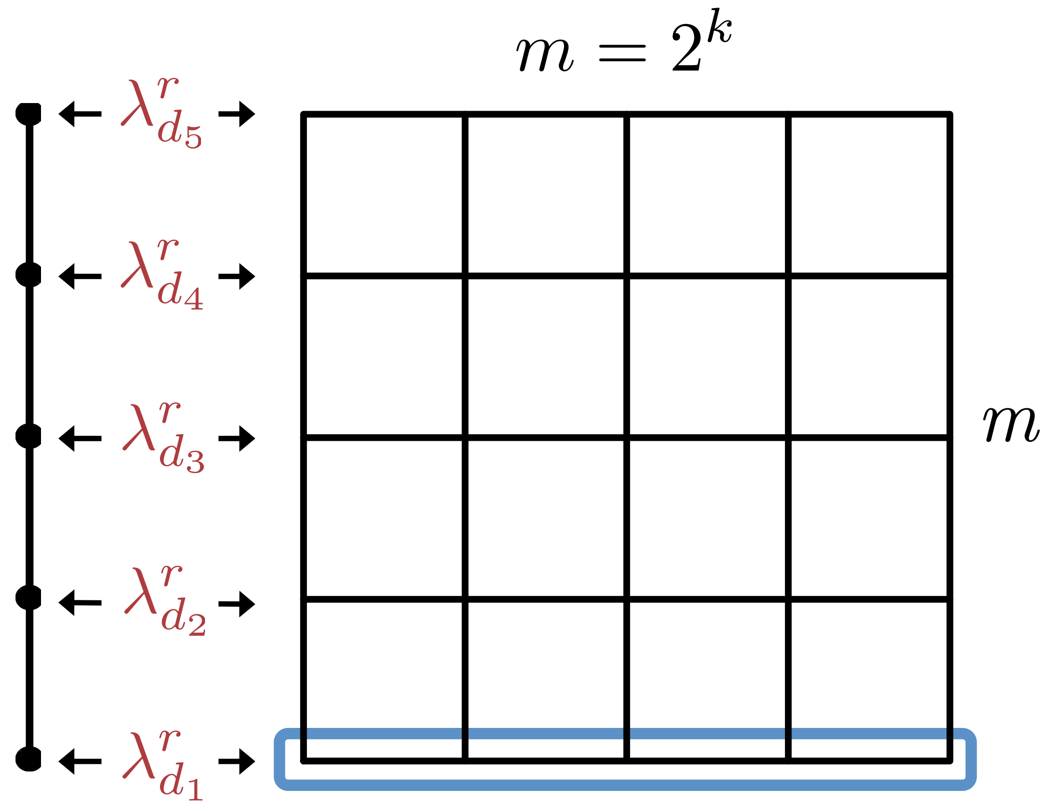
$m = 2^k$



Beyond Union Jack: Part I = Gray Code for Grid

$$\lambda_{(i,j)} \geq 0 \quad i,j \in [m+1]$$

$$\lambda^r_{d_i} = \sum_{j=1}^{m+1} \lambda_{(i,j)}$$



$$(\lambda^r, y^r) \in Q \subseteq \mathbb{R}^m \times \mathbb{R}^k$$

Univariate Gray Code Formulation

Beyond Union Jack: Part I = Gray Code for Grid

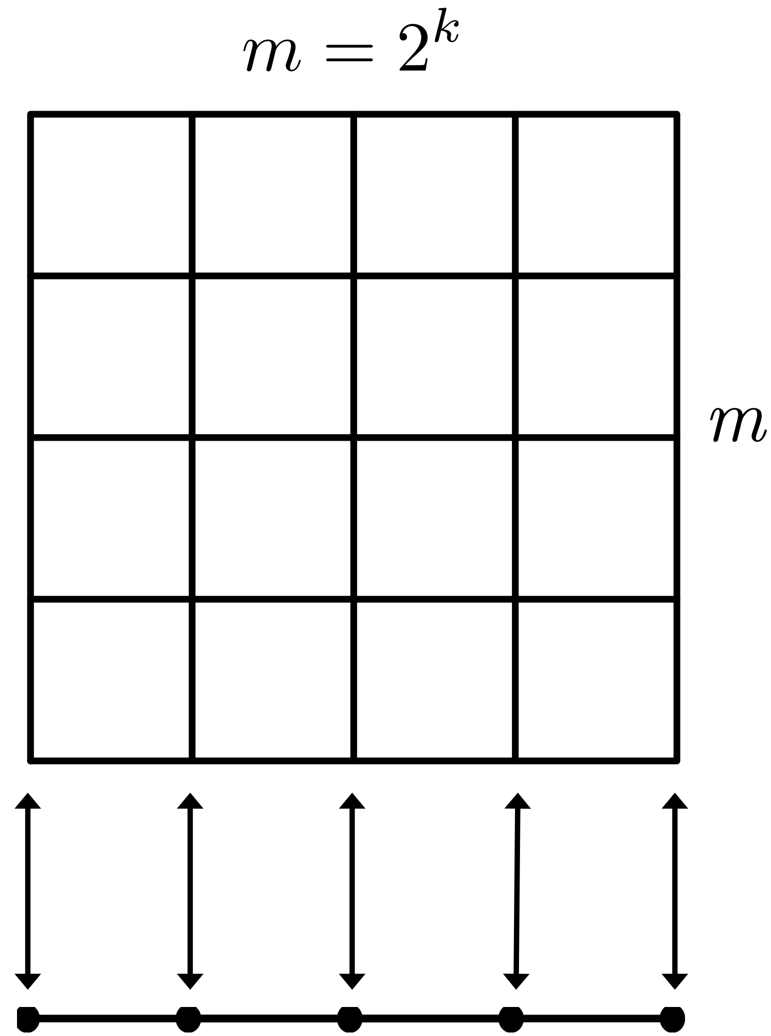
$$\lambda_{(i,j)} \geq 0 \quad i, j \in [m+1]$$

$$\lambda^r_{d_i} = \sum_{j=1}^{m+1} \lambda_{(i,j)}$$

$$\lambda^c_{d_i} = \sum_{i=1}^{m+1} \lambda_{(i,j)}$$

$$(\lambda^r, y^r) \in Q \subseteq \mathbb{R}^m \times \mathbb{R}^k$$

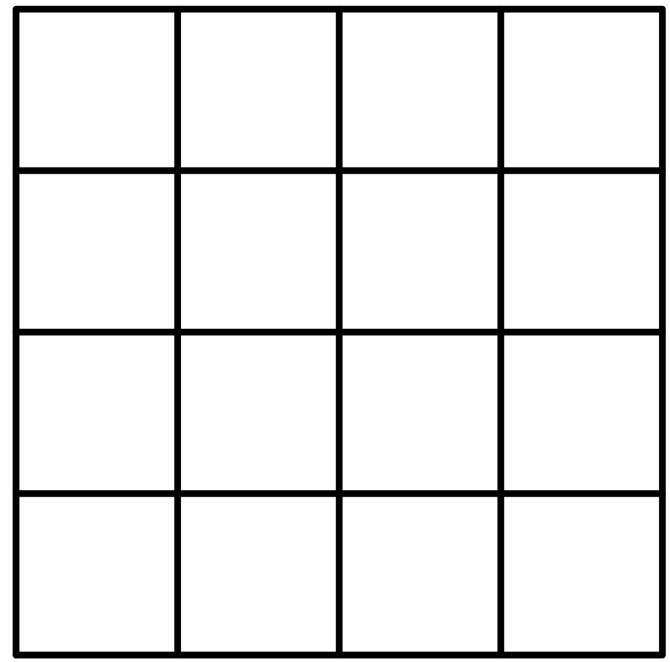
$$(\lambda^c, y^c) \in Q \subseteq \mathbb{R}^m \times \mathbb{R}^k$$



Beyond Union Jack: Part I = Gray Code for Grid

$$\lambda_{(i,j)} \geq 0 \quad i,j \in [m+1]$$

$$m = 2^k$$



m

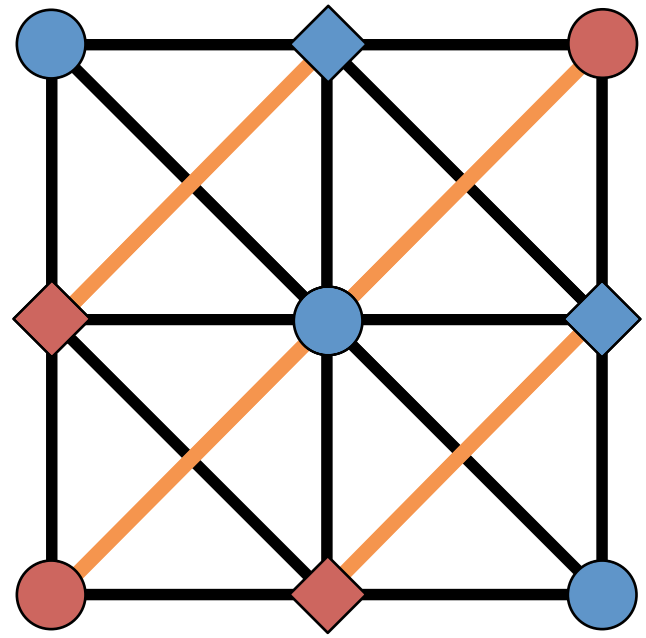
$$\lambda^r_{d_i} = \sum_{j=1}^{m+1} \lambda_{(i,j)}$$

$$\lambda^c_{d_i} = \sum_{i=1}^{m+1} \lambda_{(i,j)}$$

$$\left.\begin{array}{l} (\lambda^r, y^r) \in Q \subseteq \mathbb{R}^m \times \mathbb{R}^k \\ (\lambda^c, y^c) \in Q \subseteq \mathbb{R}^m \times \mathbb{R}^k \end{array}\right\} \; 4 \log_2 m$$

Beyond Union Jack: Part II = Selecting Triangles

1. Add "Dual" Triangulation
2. Color vertices following diagonal arcs:
   - Keep color for original arcs
   - Change color for dual arcs
3. Add binary y_1^t and constraints:



$$\sum_{(i,j) \text{ colored red}} \lambda_{(i,j)} \leq y_1^t \quad \text{and} \quad \sum_{(i,j) \text{ colored blue}} \lambda_{(i,j)} \leq 1 - y_1^t$$

4. May need to repeat coloring once more

Independent Branching = Embedding + Redundancy

- Triangle $\longleftarrow$ binary vector
- More vectors than triangles
  – Ind. Branch ≠ Embedding
  – Embedding size is larger (17)
- Ind. Branching solution:
  – Add redundant single-vertex polytopes with remaining 8 binary vectors
- Unary cannot reduce size through redundancy