Embedding Formulations and Complexity for Unions of Polyhedra

Juan Pablo Vielma
Massachusetts Institute of Technology


Supported by NSF grant CMMI-1351619
(Linear) Mixed 0-1 Integer Formulations

- Modeling Finite Alternatives = Unions of Polyhedra

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

\[ \text{gr} (f) = \bigcup_{i=1}^{n} P_i \]
(Linear) Mixed 0-1 Integer Formulations

- Modeling Finite Alternatives = Unions of Polyhedra

\[
\min \sum_{j=1}^{m} f_j(x_j, y_j) \\
\text{s.t.} \\
(x, y) \in X
\]

\[\text{gr} \left( f \right) = \bigcup_{i=1}^{n} P_i\]
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

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

\begin{align*}
A^i x^i & \leq 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{align*}

$\mathcal{H}$-formulation

\begin{align*}
\sum_{i=1}^{n} \sum_{v \in \text{ext}(P_i)} v \lambda^i_v &= x \\
\sum_{v \in \text{ext}(P_i)} \lambda^i_v &= y_i \quad \forall i \\
\sum_{i=1}^{n} y_i &= 1 \\
\lambda^i & \in \mathbb{R}^{\text{ext}(P_i)}_+ \\
y & \in \{0, 1\}^n
\end{align*}

$\mathcal{V}$-formulation

- Convex Hull (Sharp) = LP relaxation projects to $\text{conv} \left( \bigcup_{i=1}^{n} P_i \right)$

- 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

\[
\sum_{i=1}^{n} \sum_{v \in \text{ext}(P_i)} v \lambda^i_v = x \\
\sum_{v \in \text{ext}(P_i)} \lambda^i_v = y_i \quad \forall i \\
\sum_{i=1}^{n} y_i = 1 \\
\lambda^i \in \mathbb{R}^{\text{ext}(P_i)}_+ \\
y \in \{0, 1\}^n
\]

\(\mathcal{V}\)-formulation

- **Convex Hull** (Sharp) = LP relaxation projects to \(\text{conv} \left( \bigcup_{i=1}^{n} P_i \right)\)

- **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
  \[ P_i = \{ x \in \mathbb{R}^d : A^*x \leq b^i \} \]

- Lee and Wilson late ‘90s
  \[ V := \bigcup_{i=1}^{n} \text{ext} \,(P_i) \]
  \[ \sum_{v \in V} v \lambda_v = x \]
  \[ \sum_{v \in V} \lambda_v = 1 \]
  \[ \lambda_v \leq \sum_{i: v \in \text{ext}(P_i)} y_i \]
  \[ \sum_{i=1}^{n} y_i = 1 \]
  \[ y \in \{0, 1\}^n, \quad \lambda \in \mathbb{R}_+^V \]

Sometimes

- Convex Hull (Sharp) = LP relaxation projects to $\text{conv} \left( \bigcup_{i=1}^{n} P_i \right)$

- 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

\[ V := \bigcup_{i=1}^{n} \text{ext}(P_i) \]

\[
\sum_{v \in V} v \lambda_v = x
\]

\[
\sum_{v \in V} \lambda_v = 1
\]

\[
\lambda_v \leq \sum_{i : v \in \text{ext}(P_i)} y_i
\]

\[
\sum_{i=1}^{n} y_i = 1
\]

\[ y \in \{0, 1\}^n, \quad \lambda \in \mathbb{R}^V_+ \]

\[ \text{\textbf{V-formulation}} \]

- **Convex Hull** (Sharp) = LP relaxation projects to $\text{conv}\left( \bigcup_{i=1}^{n} P_i \right)$
- **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**
  \[ V := \bigcup_{i=1}^{n} \text{ext} (P_i) \]
  \[ \sum_{v \in V} u \lambda_v = x \]
  \[ \sum_{v \in V} \lambda_v = 1 \]
  \[ \lambda_v \leq \sum_{i : v \in \text{ext}(P_i)} y_i \]
  \[ \sum_{i=1}^{n} y_i = 1 \]
  \[ y \in \{0,1\}^n, \quad \lambda \in \mathbb{R}_+^V \]

- **Lee and Wilson late ‘90s**

- **Convex Hull (Sharp)** = LP relaxation projects to \( \text{conv} \left( \bigcup_{i=1}^{n} P_i \right) \)

- **Integral (Locally Ideal)** – LP relaxation has integral extreme points (y)

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

- Balas, Blair and Jeroslow late '80s

- Convex Hull (Sharp) = LP relaxation projects to $\text{conv} \left( \bigcup_{i=1}^{n} P_i \right)$

- Integral (Locally Ideal) = LP relaxation has integral extreme points ($y$)

- Lee and Wilson late '90s

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

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

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

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

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

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

$V$-formulation

Embedding Formulations
Projected Formulation for Univariate Functions

\[ \sum_{j=1}^{4} d_j \lambda_{d_j} = x, \quad \sum_{j=1}^{4} f(d_j) \lambda_{d_j} = z \]

\[ \sum_{j=1}^{4} \lambda_{d_j} = 1, \quad \lambda_{d_j} \geq 0 \]

\[ \sum_{i=1}^{3} y_i = 1, \quad y \in \{0, 1\}^3 \]

\[ \lambda_{d_1} \leq y_1, \quad \lambda_{d_2} \leq y_1 + y_2 \]

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

- **Convex Hull, but not Integral**
- **Branching is very ineffective (unbalanced B&B tree)**
  - \( y_{i_0} = 1 \quad \Rightarrow \quad y_i = 0 \quad \forall i \neq i_0 \)
  - \( y_{i_0} = 0 \) does not imply much (anything)
Projected Formulation for Univariate Functions

\[
\begin{align*}
\sum_{j=1}^{4} d_j \lambda_{d_j} &= x, \\
\sum_{j=1}^{4} f(d_j) \lambda_{d_j} &= z \\
\sum_{j=1}^{4} \lambda_{d_j} &= 1, \\
\lambda_{d_j} &\geq 0 \\
\sum_{i=1}^{3} y_i &= 1, & y &\in \{0,1\}^3 \\
\lambda_{a_1} &\leq y_1, \\
\lambda_{a_2} &\leq y_1 + y_2 \\
\lambda_{a_3} &\leq y_2 + y_3, \\
\lambda_{a_4} &\leq y_3
\end{align*}
\]

- One solution = SOS2 branching (Beale and Tomlin ’70):
  - \( \lambda_{d_i} = 0 \) \( \forall i \leq i_0 - 1 \)
  - \( \lambda_{d_i} = 0 \) \( \forall i \geq i_0 + 1 \)
Projected Formulation for Univariate Functions

\[ \sum_{j=1}^{4} d_j \lambda_j = x, \quad \sum_{j=1}^{4} f(d_j) \lambda_j = z \]

\[ \sum_{j=1}^{4} \lambda_j = 1, \quad \lambda_j \geq 0 \]

\[ \sum_{i=1}^{3} y_i = 1, \quad y \in \{0, 1\}^3 \]

\[ \lambda_{d_1} \leq y_1, \quad \lambda_{d_2} \leq y_1 + y_2 \]

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

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

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

\[y \in \{0, 1\}^n, \quad y \in \mathbb{R}_+^n\]

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

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

• 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

Embedding Formulations
Different Encodings = Different Branching

- **Unary encoding**: \( \{ h^i \}_{i=1}^n = \{ e^i \}_{i=1}^n \)

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

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

\[\Rightarrow y_i = w_i\]
Different Encodings = Different Branching

• Binary encoding: \( \{ h^i \}_{i=1}^n = \{0, 1\}^{\log_2 n} \)

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

\[
\begin{pmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0
\end{pmatrix}
\]

\[ y = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}, \quad w \in \{0, 1\}^3 \]

\( w_1 = 1 \) \quad \checkmark \quad w_1 = 0

\( y_i = 0 \quad \forall i \geq 5 \) \quad \checkmark \quad y_i = 0 \quad \forall i \leq 4 \]
Discrete Alternatives to Unions of Polyhedra

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

\[
\sum_{i=1}^{n} \sum_{\nu \in \text{ext}(P_i)} \nu \lambda_{\nu}^i = x \\
\sum_{\nu \in \text{ext}(P_i)} \lambda_{\nu}^i = y_i \quad \forall i \\
\sum_{i=1}^{n} y_i = 1 \\
\lambda^i \in \mathbb{R}_{+}^{\text{ext}(P_i)} \\
y \in \{0, 1\}^n
\]
Discrete Alternatives to Unions of Polyhedra

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

\[
\sum_{i=1}^{n} \sum_{v \in \text{ext}(P_i)} v \lambda_v^i = x \\
\sum_{i=1}^{n} \sum_{v \in \text{ext}(P_i)} \lambda_v^i = 1 \quad \forall i \\
\sum_{i=1}^{n} h^i \sum_{v \in \text{ext}(P_i)} \lambda_v^i = w \\
\lambda^i \in \mathbb{R}_{+}^{\text{ext}(P_i)} \\
w \in \{0, 1\}^k
\]

- 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

![Performance Chart]

- Multivariate functions: **Embedding Binary** is 6 times faster than **Extended Binary**
Embedding Formulations: Strong Projected Formulations
Polyhedra as MIP Formulations

\[ \lambda \in \bigcup_{i=1}^{n} P_i, \quad P_i = \{ \lambda \in \mathbb{R}^d : A^i \lambda \leq b^i \} \]

\[ Q = \left\{ (\lambda, y) \in \mathbb{R}^d \times \mathbb{R}^n : \begin{cases} A\lambda \leq \sum_{i=1}^{n} b^i y_i \\ 1 = \sum_{i=1}^{n} y_i \\ y_i \geq 0 \\ y \in \mathbb{Z}^n \end{cases} \right\} \]

\[ (\lambda, e^i) \in Q \iff \lambda \in P_i \]
Embedding Formulations for Union of Polyhedra

• **Projected** MIP formulation of \( \lambda \in \bigcup_{i=1}^{n} P_i \subseteq \mathbb{R}^V: \)
  
  – Encoding \( \left\{ h^i \right\}_{i=1}^{n} \subseteq \{0, 1\}^k, \quad h^i \neq h^j \)
  
  – Polyhedron \( Q \subseteq \mathbb{R}^V \times \mathbb{R}^k, \) s.t.
    
    \[ (\lambda, h^i) \in Q \iff \lambda \in P_i \]

• **Embedding formulation** = strongest polyhedron:

  \[ Q = \text{conv} \left( \bigcup_{i=1}^{n} P_i \times \left\{ h^i \right\} \right) \]

  For unary encoding:
  
  \[ h^i = e^i \]

Cayley Polytope \[ \rightarrow \]

Cayley Embedding
Embedding Formulations for Union of Polyhedra

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

• **Embedding formulation** = strongest polyhedron:

  \[
  Q = \text{conv} \left( \bigcup_{i=1}^{n} P_i \times \{ h^i \} \right)
  \]

  \[\text{size}(Q) := \# \text{ of facets of } Q \text{ (usually function of } n)\]
Binary v/s Unary Encodings

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

• Unary better than Binary?
  • Formulation contains convex hull through projection:
    • \( \text{Proj}_\lambda (Q) = \text{conv} \left( \bigcup_{i=1}^{n} P_i \right) \)
    • \( \text{size} (\text{Proj}_\lambda (Q)) \leq \left( \frac{\text{size}(Q)}{\text{size}(Q) - k - 1} \right) \)
  • 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

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

- Binary better than Unary?
  - Formulation contains Minkowski sum through sections:
    - For unary encoding
      \[
      \left( \lambda, \frac{1}{n} \sum_{i=1}^{n} e^i \right) \in Q \iff \lambda \in \frac{1}{n} P_1 + \ldots + \frac{1}{n} P_n
      \]
  - Unary encoding formulation can be large even if convex hull is simple
  - Binary encoding seems to only contain partial sums of \( \log_2 n \) polytopes
Simple Case: Combinatorial Part of $\mathcal{V}$-formulation

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

$$\lambda \in \bigcup_{i=1}^n P_i \iff \begin{cases} \sum_{v \in V} v\lambda_v = x \\ \sum_{v \in V} \lambda_v = 1 \\ \lambda_v \leq \sum_{i : v \in \text{ext}(P_i)} y_i \\ \sum_{i=1}^n y_i = 1 \\ y \in \{0, 1\}^n, \quad \lambda \in \mathbb{R}_+^V \end{cases}$$
Simple Case: Combinatorial Part of $\mathcal{V}$-formulation

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

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

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

Embedding Formulations
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: \( n = 2^k \)
  
  \[-\{h^i\}_{i=1}^n = \{0, 1\}^k, \quad H := \{h^i - h^{i+1}\}_{i=1}^{n-1} \subseteq \{-1, 0, 1\}^k\]

  – # facets = twice the # of linear hyperplanes spanned by \( H \)

• Gray code: \( \{h^i - h^{i+1}\}_{i=1}^{n-1} \equiv \{e^i\}_{i=1}^k \)

  – # hyperplanes: \( k = \log_2 n \), # facets \( \leq 2\log_2 n + n \)

• One kind of Anti-Gray code: \( \{h^i - h^{i+1}\}_{i=1}^{n-1} \supseteq \{-1, 1\}^k \)

  – # 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):
  \[
  \left(2\sqrt{\frac{n}{2}}\right) + \left(\sqrt{\frac{n}{2} + 1}\right)^2
  \]
  Non-negativity

- Size of best binary formulation for union jack triangulation is at most (V. and Nemhauser ’08):
  \[
  4 \log_2 \sqrt{n/2} + 2 + \left(\sqrt{n/2} + 1\right)^2
  \]
  Non-negativity
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 $\leftarrow$ binary vector
- More vectors than triangles
  - Ind. Branch $\neq$ 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

<table>
<thead>
<tr>
<th>Triangle</th>
<th>$1011$</th>
<th>$1010$</th>
<th>$0010$</th>
<th>$0000$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1101$</td>
<td>$1111$</td>
<td>$0111$</td>
<td>$0110$</td>
<td></td>
</tr>
</tbody>
</table>

Embedding Formulations 26 / 24
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

Trivial upper Bound of \( 2 \binom{n-1}{k-1} \)

Histogram of Formulation Size

Unary Encoding

Gray Binary Encoding

Embedding Formulations
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

\[ \lambda(i,j) \geq 0 \quad i, j \in [m + 1] \]

\[ \lambda_{d_i}^r = \sum_{j=1}^{m+1} \lambda_{(i,j)} \]

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

Univariate Gray Code Formulation
Beyond Union Jack: Part I = Gray Code for Grid

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

\[ \lambda_{d_i}^r = \sum_{j=1}^{m+1} \lambda_{(i,j)} \]

\[ \lambda_{d_i}^c = \sum_{i=1}^{m+1} \lambda_{(i,j)} \]

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

\[(\lambda^c, y^c) \in Q \subseteq \mathbb{R}^m \times \mathbb{R}^k \]
Beyond Union Jack: Part I = Gray Code for Grid

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

\[ \lambda_{d_{i}}^{r} = \sum_{j=1}^{m+1} \lambda_{(i,j)} \]

\[ \lambda_{d_{i}}^{c} = \sum_{i=1}^{m+1} \lambda_{(i,j)} \]

\( m = 2^k \)

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

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

\[ 4 \log_2 m \]
1. Add “Dual” Triangulation
2. Color vertices following diagonal arcs:
   - Keep color for original arcs
   - Change color for dual arcs
3. Add binary $y^t_1$ and constraints:
   \[
   \sum_{(i,j) \text{ colored red}} \lambda_{(i,j)} \leq y^t_1 \quad \text{and} \quad \sum_{(i,j) \text{ colored blue}} \lambda_{(i,j)} \leq 1 - y^t_1
   \]
4. May need to repeat coloring once more
Independent Branching = Embedding + Redundancy

• Triangle $\leftarrow$ binary vector
• More vectors than triangles
  – Ind. Branch $\neq$ 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