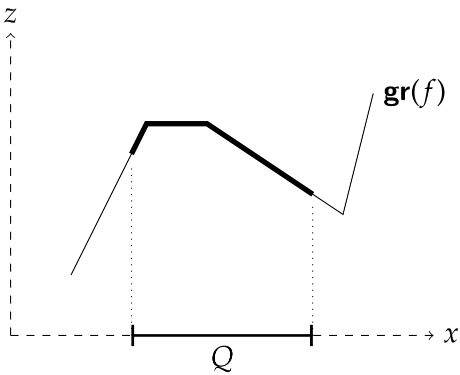
Building advanced MIP formulations for branching

Univariate piecewise linear functions

Want to optimize over the *graph* of a nonconvex function:

$$\mathbf{gr}(f) = \left\{(x, f(x)) : x \in D\right\}$$



Univariate piecewise linear functions

Want to optimize over the *graph* of a nonconvex function:

$$\mathbf{gr}(f) = \Big\{(x, f(x)) : x \in D\Big\}$$

$$\begin{aligned}
\min_x \quad & \sum_{i \in S} \sum_{j \in D} f_{i,j}(x_{i,j}) \\
\text{s.t.} \quad & \sum_{j \in D} x_{i,j} = s_i && \forall i \in S \\
& \sum_{i \in S} x_{i,j} = d_j && \forall j \in D \\
& x_{i,j} \geq 0 && \forall i \in S, j \in D
\end{aligned}$$

Univariate piecewise linear functions

Want to optimize over the *graph* of a nonconvex function:

$$\mathbf{gr}(f) = \left\{(x, f(x)) : x \in D\right\}$$

$$\begin{aligned}
\min_x \quad & \sum_{i \in S} \sum_{j \in D} z_{i,j} & \\
\text{s.t.} \quad & \sum_{j \in D} x_{i,j} = s_i & \forall i \in S \\
& \sum_{i \in S} x_{i,j} = d_j & \forall j \in D \\
& x_{i,j} \geq 0 & \forall i \in S, j \in D \\
& (x_{i,j}, z_{i,j}) \in \mathbf{gr}(f_{i,j}) & \forall i \in S, j \in D
\end{aligned}$$

Application: Power systems

- Optimal power flow problem: Generate power to meet demand at buses (nodes) on network
- Surge of interest recently [Jabr 2006, Kocuk 2015, Low 2014, ...]
- Voltage at bus:

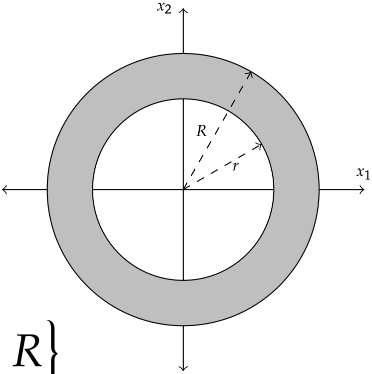
$$z = x_1 + x_2 \mathbf{i}$$

- Bounds on voltage magnitude:

$$1 - \epsilon \le |z| \le 1 + \epsilon$$

- Want x to lie in the *annulus*:

$$\mathscr{A} = \left\{ x \in \mathbb{R}^2 : r \le \|x\|_2 \le R \right\}$$



Application: Robotics

- Footstep planning problem in robotics [Deits 2014, Kuindersma 2016]: θ is rotation of body
- Angle determines feasible region for next step
- Model angle as

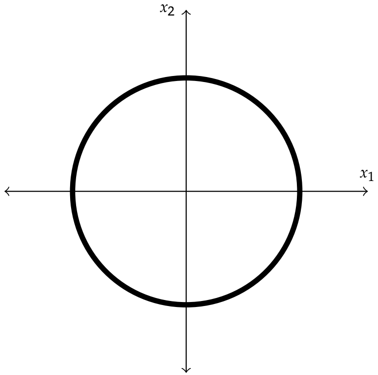
$$x = (\cos(\theta), \sin(\theta))$$

- Must satisfy identity

$$x_1^2 + x_2^2 = 1$$

- Want x to lie on the unit circle:

$$\mathscr{A} = \left\{ x \in \mathbb{R}^2 : \|x\|_2 = 1 \right\}$$



What do we want in a MIP formulation?

Performance correlates very strongly with certain properties:

1. Strength How closely does the relaxation Q approximate $\bigcup_{i=1}^{d} S^i$?

What do we want in a MIP formulation?

Performance correlates very strongly with certain properties:

1. Strength How closely does the relaxation Q approximate $\bigcup_{i=1}^{d} S^i$?
   - *Sharp* if: $\mathrm{Proj}_{1:n}(Q) = \mathrm{Conv}(\bigcup_{i=1}^{d} S^i)$

What do we want in a MIP formulation?

Performance correlates very strongly with certain properties:

1. Strength How closely does the relaxation Q approximate $\bigcup_{i=1}^{d} S^i$?
   - *Sharp* if: $\mathrm{Proj}_{1:n}(Q) = \mathrm{Conv}(\bigcup_{i=1}^{d} S^i)$
   - *Ideal* if: sharp and $\mathrm{ext}(Q) \subseteq \mathbb{R}^n \times \mathbb{Z}^r$

What do we want in a MIP formulation?

Performance correlates very strongly with certain properties:

1. Strength How closely does the relaxation Q approximate $\bigcup_{i=1}^{d} S^i$?
   - *Sharp* if: $\mathrm{Proj}_{1:n}(Q) = \mathrm{Conv}(\bigcup_{i=1}^{d} S^i)$
   - *Ideal* if: sharp and $\mathrm{ext}(Q) \subseteq \mathbb{R}^n \times \mathbb{Z}^r$
   - An ideal formulation is the strongest possible

What do we want in a MIP formulation?

Performance correlates very strongly with certain properties:

1. Strength How closely does the relaxation Q approximate $\bigcup_{i=1}^{d} S^i$?
   - *Sharp* if: $\mathrm{Proj}_{1:n}(Q) = \mathrm{Conv}(\bigcup_{i=1}^{d} S^i)$
   - *Ideal* if: sharp and $\mathrm{ext}(Q) \subseteq \mathbb{R}^n \times \mathbb{Z}^r$
   - An ideal formulation is the strongest possible
2. Size How many additional variables and constraints do we need?

What do we want in a MIP formulation?

Performance correlates very strongly with certain properties:

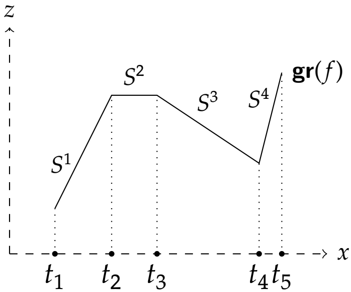
1. Strength How closely does the relaxation Q approximate $\bigcup_{i=1}^{d} S^i$?
   - *Sharp* if: $\mathrm{Proj}_{1:n}(Q) = \mathrm{Conv}(\bigcup_{i=1}^{d} S^i)$
   - *Ideal* if: sharp and $\mathrm{ext}(Q) \subseteq \mathbb{R}^n \times \mathbb{Z}^r$
   - An ideal formulation is the strongest possible
2. Size How many additional variables and constraints do we need?
3. Branching How does formulation change in branch-and-bound?

Univariate piecewise linear functions

Standard formulation #1: The `MC` method

$$(x, z) \in S^i \iff t_i \le x \le t_{i+1} \text{ and } z = a^i x + b^i$$

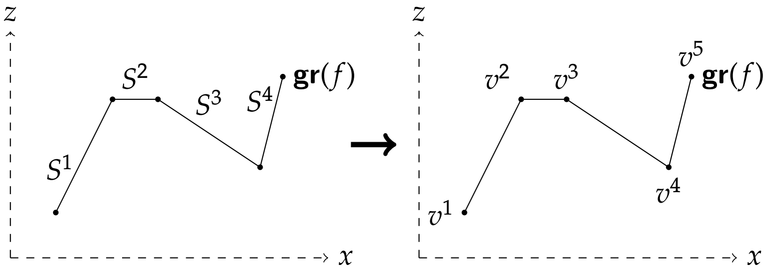
$$\begin{gathered}
(x, z) = \sum_{i=1}^{4} (x^i, z^i) \\
t_i y_i \le x^i \le t_{i+1} y_i \quad \forall i \in [\![4]\!] \\
z^i = a^i x^i + b^i y_i \quad \forall i \in [\![4]\!] \\
\sum_{i=1}^{4} y_i = 1 \\
(x, y, z) \in \mathbb{R} \times \{0,1\}^4 \times \mathbb{R} \\
(x^i, z^i) \in \mathbb{R} \times \mathbb{R} \quad \forall i \in [\![4]\!]
\end{gathered}$$



✓ As strong as possible (ideal)

✗ Not small (size=O(# segments))

Univariate piecewise linear functions

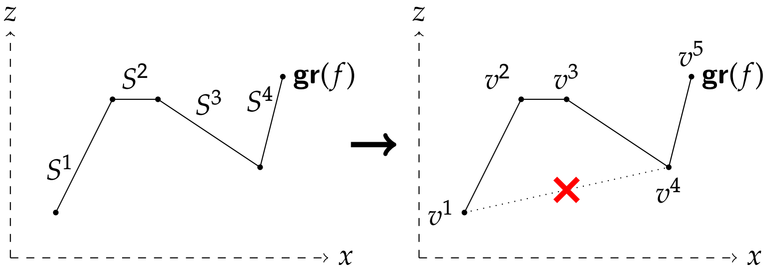


- Introduce λ_i variable for each breakpoint v^i

$$(x,z) \in \mathbf{gr}(f) \iff (x,z) = \sum_{i=1}^{d+1} v^i \lambda_i \text{ and } \lambda \text{ is SOS2}$$

- λ is SOS2 if: [Beale 1970, 1976]
  1. they are convex multipliers ($\lambda \in \Delta^{d+1}$ = unit simplex)
  2. $\text{support}(\lambda) \subseteq \{j, j+1\}$ for some j

Univariate piecewise linear functions

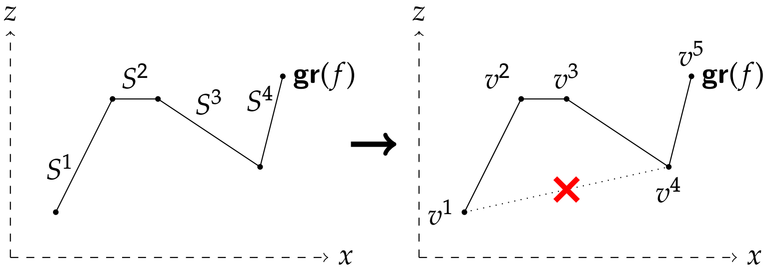


- Introduce λ_i variable for each breakpoint v^i

$$(x,z) \in \mathbf{gr}(f) \iff (x,z) = \sum_{i=1}^{d+1} v^i \lambda_i \text{ and } \lambda \text{ is SOS2}$$

- λ is SOS2 if: [Beale 1970, 1976]
  1. they are convex multipliers ($\lambda \in \Delta^{d+1}$ = unit simplex)
  2. $\text{support}(\lambda) \subseteq \{j, j+1\}$ for some j

Univariate piecewise linear functions



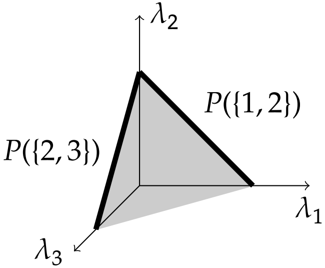
- Introduce λ_i variable for each breakpoint v^i

$$(x,z) \in \bigcup_{i=1}^{d} S^i \iff (x,z) = \sum_{i=1}^{d+1} v^i \lambda_i \text{ and } \lambda \in \bigcup_{i=1}^{d} P(\{i, i+1\})$$

- $P(T) = \{\lambda \in \Delta^{d+1} : \mathrm{support}(\lambda) \subseteq T\}$ (*face of the simplex*)

The SOS2 constraint

$$\lambda \in \bigcup_{i=1}^{d} P(\{i, i+1\})$$



1. Strip away problem data (values of v^i)
2. Formulate the SOS2 constraint on λ over the unit simplex Δ^{d+1}
3. Apply linear transformation $(x, z) = \sum_{i=1}^{d+1} v^i \lambda_i$

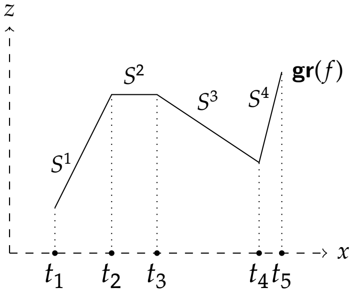
$$P(T) = \{\lambda \in \Delta^{d+1} : \text{support}(\lambda) \subseteq T\} (\textit{face of the simplex})$$

Univariate piecewise linear functions

Standard formulation #2: The CC method

$$(x,z) \in S^i \iff \text{support}(\lambda) \subseteq \{i, i+1\}$$

$$(x,z) = \sum_{j=1}^{5} \lambda_v (t_j, f(t_j))$$
$$\lambda_1 \leq y_1, \quad \lambda_5 \leq y_4$$
$$\lambda_j \leq y_{j-1} + y_j \quad \forall j \in [\![2,4]\!]$$
$$\sum_{j=1}^{4} y_j = 1$$
$$(x,z) \in \mathbb{R} \times \mathbb{R}$$
$$(\lambda, y) \in \Delta^5 \times \{0,1\}^4$$



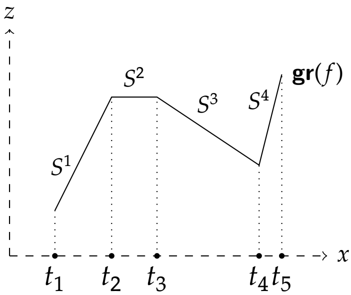
- ✗ Not strong (not ideal)
- ✗ Not small (size=O(# segments))

Univariate piecewise linear functions

Standard formulation #2: The CC method

$$(x,z) \in S^i \iff \text{support}(\lambda) \subseteq \{i, i+1\}$$

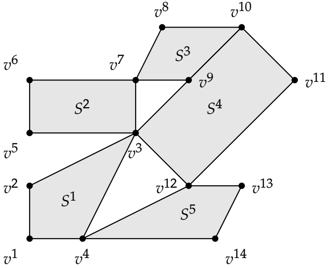
$$\begin{aligned}
\lambda_1 &\leq y_1 \\
\lambda_2 &\leq y_1 + y_2 \\
\lambda_3 &\leq y_2 + y_3 \\
\lambda_4 &\leq y_3 + y_4 \\
\lambda_5 &\leq y_4 \\
\sum_{j=1}^4 y_j &= 1 \\
(\lambda, y) &\in \Delta^5 \times \{0,1\}^4
\end{aligned}$$



✗ $\lambda = (1/2, 1/2, 0, 0, 0)$,
$y = (1/2, 0, 1/2, 0)$

Combinatorial disjunctive constraints

$$x \in \bigcup_{i=1}^{d} S^i$$



1. Strip away problem data (values of v^i)
2. Formulate the disjunctive constraint on λ over the unit simplex Δ^n
3. Apply linear transformation $x = \sum_{i=1}^{n} v^i \lambda_i$

$$P(T) = \{\lambda \in \Delta^{d+1} : \text{support}(\lambda) \subseteq T\} (\textit{face of the simplex})$$

Combinatorial disjunctive constraints

$$\lambda \in \bigcup_{i=1}^{d} P(T^i)$$

$$T^1 = \{1, 2, 3, 4\}$$
$$T^2 = \{3, 5, 6, 7\}$$
$$T^3 = \{7, 8, 9, 10\}$$
$$T^4 = \{3, 10, 11, 12\}$$
$$T^5 = \{4, 12, 13, 14\}$$

1. Strip away problem data (values of v^i)
2. Formulate the disjunctive constraint on λ over the unit simplex Δ^n
3. Apply linear transformation $x = \sum_{i=1}^{n} v^i \lambda_i$

$$P(T) = \{\lambda \in \Delta^{d+1} : \text{support}(\lambda) \subseteq T\} (\textit{face of the simplex})$$

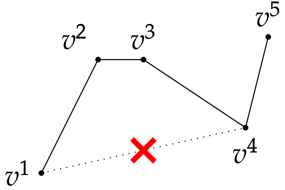
Formulating the SOS2 constraint

- MIP formulations (and direct methods) studied for decades

  [Balakrishnan 1989, Beale 1970, Croxton 2003, D'Ambrosio 2010, de Farias Jr. 2008, 2013, Dantzig 1960, Jeroslow 1984, 1985, Keha 2004, 2006, Lee 2001, Magnanti 2004, Markowitz 1957, Padberg 2000, Sherali 2001, Tomlin 1981, Wilson 1998, ...]

- Previous state-of-the-art Log formulation [Vielma 2010, 2011]

$$
\begin{aligned}
\lambda_1 + \lambda_5 &\leq 1 - y_1 \\
\lambda_3 &\leq y_1 \\
\lambda_1 + \lambda_2 &\leq 1 - y_2 \\
\lambda_4 + \lambda_5 &\leq y_2 \\
(\lambda, y) &\in \Delta^5 \times \{0,1\}^2
\end{aligned}
$$

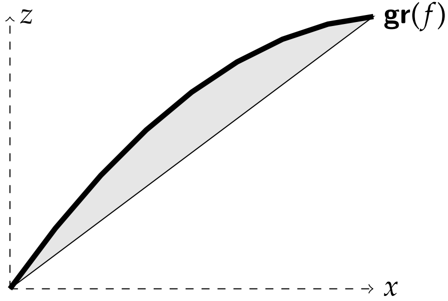


✓ *Strongest possible* (ideal)

✓ *Smallest possible* (size $= O(\log(d))$ with matching lower bounds)

- What about branching?

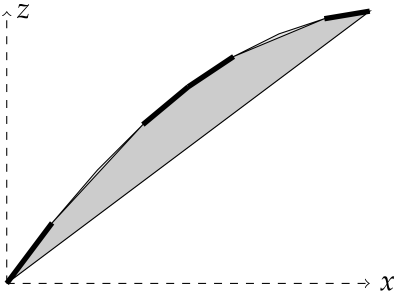
Branching and the Log formulation



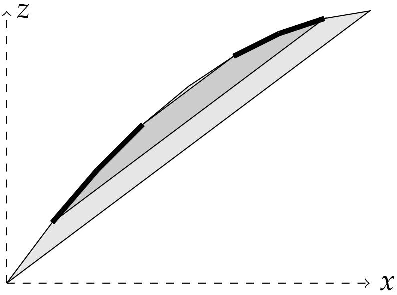
LP relaxation of any ideal formulation

- Log is as strong as possible (w.r.t. LP relaxation)
- What about after branching?

Branching and the Log formulation



LP relaxation + $y_1 \leq 0$



LP relaxation + $y_1 \geq 1$

- Branching $y_1 \leq 0$ restricts to $d/2$ segments of graph...
- ...but leaves LP relaxation essentially unchanged!

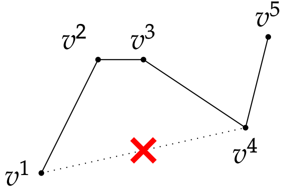
Formulating the SOS2 constraint

- MIP formulations (and direct methods) studied for decades

  [Balakrishnan 1989, Beale 1970, Croxton 2003, D'Ambrosio 2010, de Farias Jr. 2008, 2013, Dantzig 1960, Jeroslow 1984, 1985, Keha 2004, 2006, Lee 2001, Magnanti 2004, Markowitz 1957, Padberg 2000, Sherali 2001, Tomlin 1981, Vielma 2010, 2011, Wilson 1998, ...]

- `B-ZigZag` **formulation** [H. and Vielma 2017]

$$
\begin{aligned}
\lambda_3 + \lambda_4 + 2\lambda_5 &\le y_1 + y_2 \\
\lambda_2 + \lambda_3 + 2\lambda_4 + 2\lambda_5 &\ge y_1 + y_2 \\
\lambda_4 + \lambda_5 &\le y_2 \\
\lambda_3 + \lambda_4 + \lambda_5 &\ge y_2 \\
(\lambda, y) &\in \Delta^5 \times \{0,1\}^2
\end{aligned}
$$



✓ *Strongest possible* (ideal)

✓ *Smallest possible* (size $= O(\log(d))$ with matching lower bounds)

- What about branching?

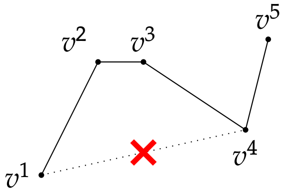
Formulating the SOS2 constraint

- MIP formulations (and direct methods) studied for decades

  [Balakrishnan 1989, Beale 1970, Croxton 2003, D'Ambrosio 2010, de Farias Jr. 2008, 2013, Dantzig 1960, Jeroslow 1984, 1985, Keha 2004, 2006, Lee 2001, Magnanti 2004, Markowitz 1957, Padberg 2000, Sherali 2001, Tomlin 1981, Vielma 2010, 2011, Wilson 1998, ...]

- **`ZigZag` formulation** [H. and Vielma 2017]

$$
\begin{aligned}
\lambda_3 + \lambda_4 + 2\lambda_5 &\le y_1 \\
\lambda_2 + \lambda_3 + 2\lambda_4 + 2\lambda_5 &\ge y_1 \\
\lambda_4 + \lambda_5 &\le y_2 \\
\lambda_3 + \lambda_4 + \lambda_5 &\ge y_2 \\
(\lambda, y) \in \Delta^5 \times \{0,1,2\} &\times \{0,1\}
\end{aligned}
$$

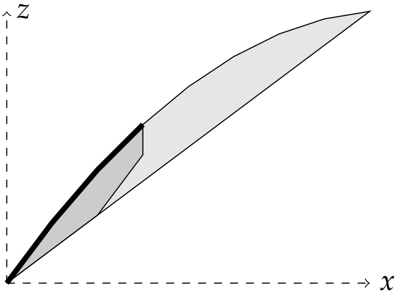


✓ *Strongest possible* (ideal)

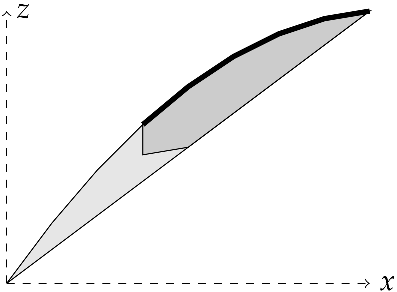
✓ *Smallest possible* (size $= O(\log(d))$ with matching lower bounds)

- What about branching?

Branching and ZigZag formulation



LP relaxation $+ \; y_1 \leq 1$



LP relaxation $+ \; y_1 \geq 2$

- "Incremental branching": Split x domain into "left" and "right"
- Both subproblems have substantially strengthened LP relaxations

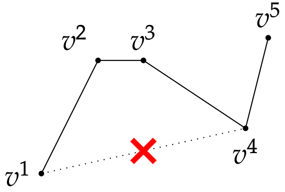
Formulating the SOS2 constraint

- MIP formulations (and direct methods) studied for decades

  [Balakrishnan 1989, Beale 1970, Croxton 2003, D'Ambrosio 2010, de Farias Jr. 2008, 2013, Dantzig 1960, Jeroslow 1984, 1985, Keha 2004, 2006, Lee 2001, Magnanti 2004, Markowitz 1957, Padberg 2000, Sherali 2001, Tomlin 1981, Vielma 2010, 2011, Wilson 1998, ...]

- **`ZigZag` formulation** [H. and Vielma 2017]

$$\begin{aligned}
\lambda_3 + \lambda_4 + 2\lambda_5 &\le y_1 \\
\lambda_2 + \lambda_3 + 2\lambda_4 + 2\lambda_5 &\ge y_1 \\
\lambda_4 + \lambda_5 &\le y_2 \\
\lambda_3 + \lambda_4 + \lambda_5 &\ge y_2 \\
(\lambda, y) \in \Delta^5 \times \{0,1,2\} &\times \{0,1\}
\end{aligned}$$



✓ *Strongest possible* (ideal)

✓ *Smallest possible* (size $= O(\log(d))$ with matching lower bounds)

✓ Incremental branching

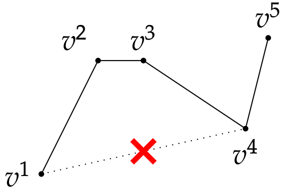
Formulating the SOS2 constraint

- MIP formulations (and direct methods) studied for decades

  [Balakrishnan 1989, Beale 1970, Croxton 2003, D'Ambrosio 2010, de Farias Jr. 2008, 2013, Dantzig 1960, Jeroslow 1984, 1985, Keha 2004, 2006, Lee 2001, Magnanti 2004, Markowitz 1957, Padberg 2000, Sherali 2001, Tomlin 1981, Vielma 2010, 2011, Wilson 1998, ...]

- `ZigZag` **formulation** [H. and Vielma 2017]

$$
\begin{aligned}
\lambda_3 + \lambda_4 + 2\lambda_5 &\leq y_1 \\
\lambda_2 + \lambda_3 + 2\lambda_4 + 2\lambda_5 &\geq y_1 \\
\lambda_4 + \lambda_5 &\leq y_2 \\
\lambda_3 + \lambda_4 + \lambda_5 &\geq y_2 \\
(\lambda, y) \in \Delta^5 \times \{0,1,2\} &\times \{0,1\}
\end{aligned}
$$

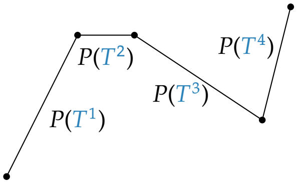


✓ *Strongest possible* (ideal)

✓ *Smallest possible* (size $= O(\log(d))$ with matching lower bounds)

✓ Incremental branching using a general-integer formulation!

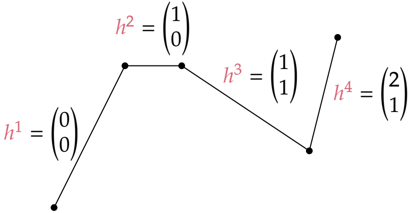
The embedding approach



Two ingredients:

1. The sets $\mathscr{T} = (T^i \subseteq [n])_{i=1}^d$ (correspond to faces of simplex; not in (x, z)-space!)

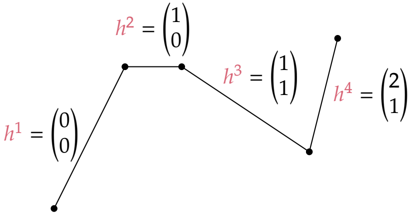
The embedding approach



Two ingredients:

1. The sets $\mathscr{T} = (T^i \subseteq [n])_{i=1}^d$ (correspond to faces of simplex; not in (x, z)-space!)
2. Unique *codes* $H = (h^i)_{i=1}^d \subset \mathbb{R}^r$ (also hole-free, in convex position)

The embedding approach



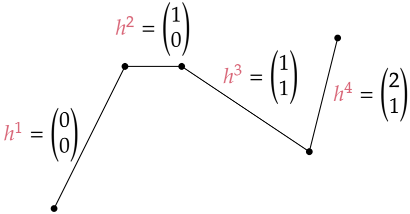
Two ingredients:

1. The sets $\mathscr{T} = (T^i \subseteq [n])_{i=1}^d$ (correspond to faces of simplex; not in (x, z)-space!)
2. Unique *codes* $H = (h^i)_{i=1}^d \subset \mathbb{R}^r$ (also hole-free, in convex position)

Build *embedding*:

$$\mathrm{Em}(\mathscr{T}, H) = \begin{pmatrix} P(T^1) \\ h^1 \end{pmatrix} \cup \begin{pmatrix} P(T^2) \\ h^2 \end{pmatrix} \cup \cdots \cup \begin{pmatrix} P(T^d) \\ h^d \end{pmatrix}$$

The embedding approach



Two ingredients:

1. The sets $\mathscr{T} = (T^i \subseteq [n])_{i=1}^d$ (correspond to faces of simplex; not in (x, z)-space!)
2. Unique *codes* $H = (h^i)_{i=1}^d \subset \mathbb{R}^r$ (also hole-free, in convex position)

Proposition (Vielma 2017)

$\mathrm{Conv}(\mathrm{Em}(\mathscr{T}, H))$ is an ideal formulation. Conversely, any non-extended ideal formulation implies the existence of some corresponding $\mathscr{T}$ and H.

The embedding approach

Two ingredients:

1. The sets $\mathscr{T} = (T^i \subseteq [n])_{i=1}^d$ (correspond to faces of simplex; not in (x,z)-space!)
2. Unique *codes* $H = (h^i)_{i=1}^d \subset \mathbb{R}^r$ (also hole-free, in convex position)

Proposition (Vielma 2017)

$\mathrm{Conv}(\mathrm{Em}(\mathscr{T}, H))$ is an ideal formulation. Conversely, any non-extended ideal formulation implies the existence of some corresponding $\mathscr{T}$ and H.

Two questions:

1. What is an inequality description for $\mathrm{Conv}(\mathrm{Em}(\mathscr{T}, H))$?
2. How do we select the codes H?

Geometric formulation construction

Theorem (H. and Vielma 2017a)

If $\mathscr{T}$ is path connected and H is in convex position, then $\mathrm{Conv}(\mathrm{Em}(\mathscr{T}, H))$ is

$$\sum_{v=1}^{n} \min_{s:v\in T^s}\{b \cdot h^s\}\lambda_v \leq b \cdot y \leq \sum_{v=1}^{n} \max_{s:v\in T^s}\{b \cdot h^s\}\lambda_v \quad \forall b \in B$$

$$(\lambda, y) \in \Delta^n \times \mathrm{aff}(H),$$

where B contains normal directions to all hyperplanes spanned by $C = \left\{h^j - h^i : T^i \cap T^j \neq \emptyset\right\}$ in $\mathrm{span}(C)$.

Geometric formulation construction

Theorem (H. and Vielma 2017a)

If $\mathscr{T}$ is path connected and H is in convex position, then $\mathrm{Conv}(\mathrm{Em}(\mathscr{T}, H))$ is

$$\sum_{v=1}^{n} \min_{s:v \in T^s} \{b \cdot h^s\} \lambda_v \leq b \cdot y \leq \sum_{v=1}^{n} \max_{s:v \in T^s} \{b \cdot h^s\} \lambda_v \quad \forall b \in B$$

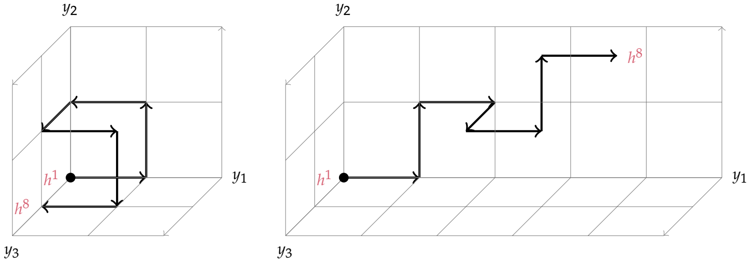
$$(\lambda, y) \in \Delta^n \times \mathrm{aff}(H),$$

where B contains normal directions to all hyperplanes spanned by $C = \left\{ h^j - h^i : T^i \cap T^j \neq \emptyset \right\}$ in $\mathrm{span}(C)$.

Crucial points:

1. $\#$ variables $=$ $\#$ of components of codes in H
2. $\#$ constraints $= 2 \times (\#$ hyperplanes$)$

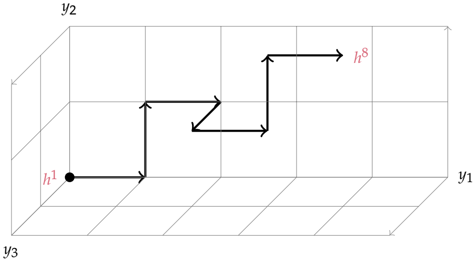
Geometric formulation construction



(Left) binary reflected Gray codes, (Right) `ZigZag` codes

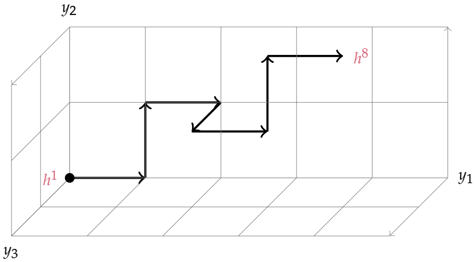
1. Ambient space $\mathbb{R}^{\log_2(d)} \implies \log_2(d)$ variables

Geometric formulation construction



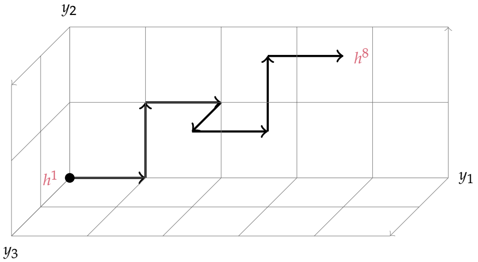
$$C = \left\{ h^j - h^i : T^i \cap T^j \neq \emptyset \right\}$$

Geometric formulation construction



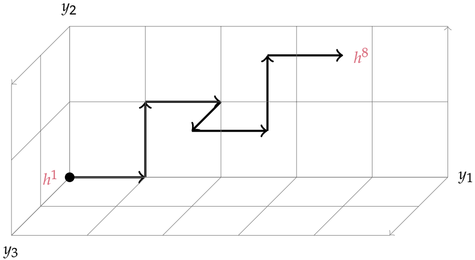
$$C = \left\{ h^{i+1} - h^{i} \right\}_{i=1}^{d-1}$$

Geometric formulation construction



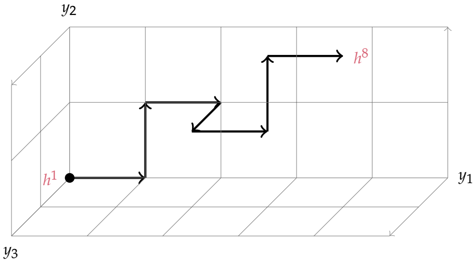
$$C = \left\{ \mathbf{e}^i \right\}_{i=1}^{\log_2(d)}$$

Geometric formulation construction



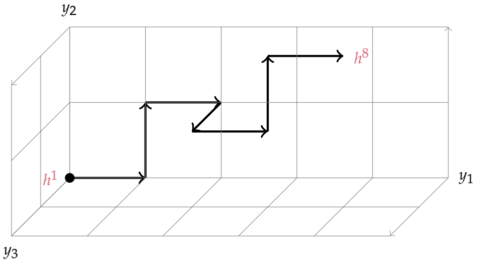
$$B = \text{normal directions to hyperplanes spanned by } C$$

Geometric formulation construction



$$B = \left\{\mathbf{e}^i\right\}_{i=1}^{\log_2(d)}$$

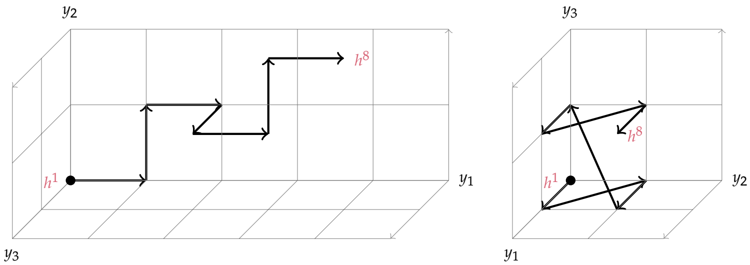
Geometric formulation construction



$$B = \left\{ \mathbf{e}^i \right\}_{i=1}^{\log_2(d)}$$

2. directions in C are axis-aligned $\Longrightarrow 2\log_2(d)$ constraints

Geometric formulation construction



(Left) `ZigZag` codes, (Right) `B-ZigZag` codes

- `B-ZigZag` = linear map of `ZigZag` to unit hypercube

Geometric formulation construction

Theorem (H. and Vielma 2017a)

If $\mathscr{T}$ is path connected and H is in convex position, then $\mathrm{Conv}(\mathrm{Em}(\mathscr{T}, H))$ is

$$\sum_{v=1}^{n} \min_{s: v \in T^s} \{b \cdot h^s\} \lambda_v \leq b \cdot y \leq \sum_{v=1}^{n} \max_{s: v \in T^s} \{b \cdot h^s\} \lambda_v \quad \forall b \in B$$

$$(\lambda, y) \in \Delta^n \times \mathrm{aff}(H),$$

where B contains normal directions to all hyperplanes spanned by $C = \left\{ h^j - h^i : T^i \cap T^j \neq \emptyset \right\}$ in $\mathrm{span}(C)$.

Crucial points:

1. # variables = # of components of codes in H
2. # constraints = 2 ×(# hyperplanes)

Proof sketch

- Take each facet $F = a \cdot \lambda \leq b \cdot y$ of $Q = \mathrm{Conv}(\mathrm{Em}(\mathscr{T}, H))$
  - Use $\sum_{v=1}^{n} \lambda_v = 1$ to remove any constant offset
- Extreme points easy to understand: $\mathrm{ext}(Q) = \{(\mathbf{e}^v, h^s) : v \in T^s\}$
- Take "directions in C that are orthogonal to b and support F":

$$\tilde{C} = \left\{h^j - h^i \in C : \exists v \text{ s.t. } (\mathbf{e}^v, h^i), (\mathbf{e}^v, h^j) \in F\right\}$$

Three cases to worry about:

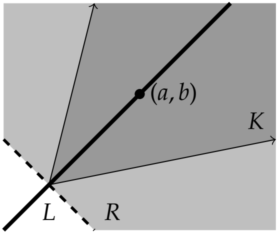
1. $\underline{\dim(\tilde{C}) = \dim(C)}$
   - $b \in \mathrm{span}(C)^\perp$, so F is a variable bound ($\lambda_v \geq 0$)
2. $\underline{\dim(\tilde{C}) = \dim(C) - 1}$
   - b is normal direction to hyperplane spanned by $\tilde{C}$ in $\mathrm{span}(C)$
   - General inequality facet, so

$$a \cdot \mathbf{e}^v = a_v = \min_{s : v \in T^s} b \cdot h^s \quad \forall v \in [n].$$

Proof sketch

3. $\underline{\dim(\tilde{C}) < \dim(C) - 1}$
   - Contradiction by finding a direction in $C \backslash \tilde{C}$ to "slide" F so that:
     1. maintains validity of F and
     2. strictly increases support of F w.r.t. $\operatorname{ext}(Q)$, but
     3. keeps F a proper face
   - Set of all coefficients satisfying these properties:

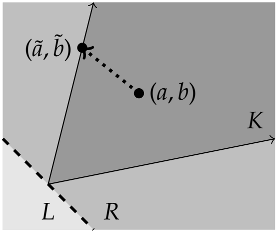
$$(\tilde{a}, \tilde{b}) \in \underbrace{(K = \text{polyhedral cone})}_{\text{feasibility}} \cap \underbrace{(L = \text{linear space})}_{\text{support of } F} \cap \underbrace{(R = \text{open halfspace})}_{\text{proper face}}$$



Proof sketch

3. $\underline{\dim(\tilde{C}) < \dim(C) - 1}$
   - Contradiction by finding a direction in $C \setminus \tilde{C}$ to "tilt" F so that:
     1. maintains validity of F and
     2. strictly increases support of F w.r.t. $\mathrm{ext}(Q)$, but
     3. keeps F a proper face
   - Set of all coefficients satisfying these properties:

$$(\tilde{a}, \tilde{b}) \in \underbrace{(K = \text{polyhedral cone})}_{\text{feasibility}} \cap \underbrace{(L = \text{linear space})}_{\text{support of } F} \cap \underbrace{(R = \text{open halfspace})}_{\text{proper face}}$$



Path connectivity of $\mathscr{T} \implies \dim(C) = \dim(H) \implies \dim(L) > 1$

Univariate formulations: Computational performance

d	Metric	MC	CC	SOS2	Inc	DLog	Log	B-ZigZag	ZigZag
6	Mean (s)	0.6	3.8	1.1	0.6	1.1	1.4	1.1	0.9
	Win	35	0	7	46	5	1	4	2
13	Mean (s)	3.0	71.2	4.5	1.7	4.6	4.4	2.4	2.6
	Win	11	0	9	47	11	0	15	7
28	Mean (s)	18.4	178.9	87.4	5.5	11.1	8.8	5.1	4.6
	Win	1	0	6	14	1	0	37	41
59	Mean (s)	348.7	541.0	664.3	17.1	19.1	16.3	9.8	9.3
	Win	0	0	0	0	0	0	41	59

Solve time (in seconds, with CPLEX v12.7.0). Functions have d pieces, fixed network $|S| = |D| = 10$.

- Log is (almost) strictly dominated by zig-zag formulations
- Zig-zag formulations fastest on all larger instances

Univariate formulations: Computational performance

d	Metric	MC	CC	SOS2	Inc	DLog	Log	B-ZigZag	ZigZag
6	Mean (s)	0.6	3.8	1.1	0.6	1.1	1.4	1.1	0.9
	Win	35	0	7	46	5	1	4	2
13	Mean (s)	3.0	71.2	4.5	1.7	4.6	4.4	2.4	2.6
	Win	11	0	9	47	11	0	15	7
28	Mean (s)	18.4	178.9	87.4	5.5	11.1	8.8	5.1	4.6
	Win	1	0	6	14	1	0	37	41
59	Mean (s)	348.7	541.0	664.3	17.1	19.1	16.3	9.8	9.3
	Win	0	0	0	0	0	0	41	59

Solve time (in seconds, with CPLEX v12.7.0). Functions have d pieces, fixed network $|S| = |D| = 10$.

- Log is (almost) strictly dominated by zig-zag formulations
- Zig-zag formulations fastest on all larger instances

Univariate functions: Computational tools

Here's the math ($d = 8$):

$$
\begin{aligned}
\min_{x \geq 0} \quad & \textstyle\sum_{i \in S} \sum_{j \in D} z_{i,j} \\
\text{s.t.} \quad & \textstyle\sum_{j \in D} x_{i,j} = s_i \quad \forall i \in S \\
& \textstyle\sum_{i \in S} x_{i,j} = d_j \quad \forall j \in D \\
& (x_{i,j}, z_{i,j}) \in \mathbf{gr}(f_{i,j}) \qquad \forall i \in S, j \in D
\end{aligned}
$$

Univariate functions: Computational tools

Here's the math ($d = 8$):

$$
\begin{aligned}
\min_{x \geq 0} \quad & \textstyle\sum_{i \in S} \sum_{j \in D} z_{i,j} \\
\text{s.t.} \quad & \textstyle\sum_{j \in D} x_{i,j} = s_i \quad \forall i \in S \\
& \textstyle\sum_{i \in S} x_{i,j} = d_j \quad \forall j \in D \\
& (x_{i,j}, z_{i,j}) = \textstyle\sum_{k=1}^{N+1} v_{i,j}^k \lambda_k^{i,j} && \forall i \in S, j \in D \\
& \lambda_3^{i,j} + \lambda_4^{i,j} + 2\lambda_5^{i,j} + 2\lambda_6^{i,j} + 3\lambda_7^{i,j} + 3\lambda_8^{i,j} + 4\lambda_9^{i,j} \leq y_1^{i,j} && \forall i \in S, j \in D \\
& \lambda_2^{i,j} + \lambda_3^{i,j} + 2\lambda_4^{i,j} + 2\lambda_5^{i,j} + 3\lambda_6^{i,j} + 3\lambda_7^{i,j} + 4\lambda_8^{i,j} + 4\lambda_9^{i,j} \geq y_1^{i,j} && \forall i \in S, j \in D \\
& \lambda_4^{i,j} + \lambda_5^{i,j} + \lambda_6^{i,j} + \lambda_7^{i,j} + 2\lambda_8^{i,j} + 2\lambda_9^{i,j} \leq y_2^{i,j} && \forall i \in S, j \in D \\
& \lambda_3^{i,j} + \lambda_4^{i,j} + \lambda_5^{i,j} + \lambda_6^{i,j} + 2\lambda_7^{i,j} + 2\lambda_8^{i,j} + 2\lambda_9^{i,j} \geq y_2^{i,j} && \forall i \in S, j \in D \\
& \lambda_6^{i,j} + \lambda_7^{i,j} + \lambda_8^{i,j} + \lambda_9^{i,j} \leq y_3^{i,j} \leq \lambda_5^{i,j} + \lambda_6^{i,j} + \lambda_7^{i,j} + \lambda_8^{i,j} + \lambda_9^{i,j} && \forall i \in S, j \in D \\
& (\lambda^{i,j}, y^{i,j}) \in \Delta^9 \times \{0,1,2,3,4\} \times \{0,1,2\} \times \{0,1\} && \forall i \in S, j \in D
\end{aligned}
$$

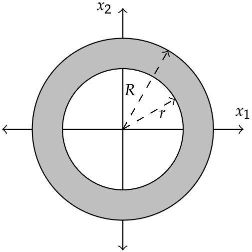
Now turn this into code.

Univariate functions: Computational tools

```
using JuMP, PiecewiseLinearOpt
model = Model()
@variable(model, x[i in S, j in D] >= 0)
for j in D
    @constraint(model, sum(x[i,j] for i in S) == d[j])
end
for i in S
    @constraint(model, sum(x[i,j] for j in D) == s[i])
end
for i in S, j in D
    z[i,j] = piecewiselinear(model, x[i,j], t[i,j],
     ↪  f[i,j], method=:ZigZag)
end
@objective(model, Min, sum(z))
solve(model)
```

Annulus constraints

$$\mathscr{A} = \left\{ x \in \mathbb{R}^2 : r \leq \|x\|_2 \leq R \right\}$$

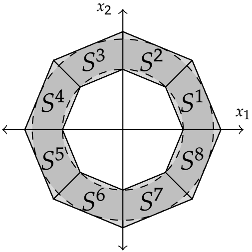


How do we optimize over the annulus?

- Not *mixed-integer convex representable*! [Lubin 2017]

Annulus constraints

$$\mathscr{A} \approx \hat{\mathscr{A}} = \bigcup_{i=1}^{d} S^i$$



Discretization proposed for OPF by [Foster 2013]

Theorem (H. and Vielma 2017a)

Both the reflected binary Gray and the `ZigZag` codes yield ideal formulations for $\hat{\mathscr{A}}$ that use $\lceil \log_2(d) \rceil$ integer variables and $O(\text{polylog}(d))$ general inequality constraints.

Zig-zag formulation for the annulus

$$C = \left\{ h^j - h^i : T^i \cap T^j \neq \varnothing \right\}$$

Zig-zag formulation for the annulus

$$C = \left\{h^{i+1} - h^{i}\right\}_{i=1}^{d-1} \cup \left\{h^{d} - h^{1}\right\}$$

Zig-zag formulation for the annulus

$$C = \left\{\mathbf{e}^k\right\}_{k=1}^{\log_2(d)} \cup \left\{\left(2^{\log_2(d)-1}, 2^{\log_2(d)-2}, \ldots, 2^0\right)\right\}$$

$H = (h^i)_{i=1}^{d}$ = `ZigZag` codes

Zig-zag formulation for the annulus

$$B = \left\{\mathbf{e}^k\right\}_{k=1}^{\log_2(d)} \cup \left\{\frac{1}{2^\ell}\mathbf{e}^k - \frac{1}{2^k}\mathbf{e}^\ell\right\}_{\{k,\ell\}\in[\log_2(d)]^2}$$

$H = (h^i)_{i=1}^d =$ `ZigZag` codes

Zig-zag formulation for the annulus

$$B = \left\{\mathbf{e}^k\right\}_{k=1}^{\log_2(d)} \cup \left\{\frac{1}{2^\ell}\mathbf{e}^k - \frac{1}{2^k}\mathbf{e}^\ell\right\}_{\{k,\ell\}\in[\log_2(d)]^2}$$

$$H = (h^i)_{i=1}^d = \texttt{ZigZag} \text{ codes}$$

$$\begin{gathered}
\sum_{i=1}^d \min\{h_k^{i-1}, h_k^i\}(\lambda_{2i-1} + \lambda_{2i}) \le z_k \quad \forall k \in [r] \\
\sum_{i=1}^d \max\{h_k^{i-1}, h_k^i\}(\lambda_{2i-1} + \lambda_{2i}) \ge z_k \quad \forall k \in [r] \\
\sum_{i=1}^d \min\left\{\frac{h_k^{i-1}}{2^\ell} - \frac{h_\ell^{i-1}}{2^k}, \frac{h_k^i}{2^\ell} - \frac{h_\ell^i}{2^k}\right\}(\lambda_{2i-1} + \lambda_{2i}) \le \frac{z_k}{2^\ell} - \frac{z_\ell}{2^k} \quad \forall\{k,\ell\} \in [r]^2 \\
\sum_{i=1}^d \max\left\{\frac{h_k^{i-1}}{2^\ell} - \frac{h_\ell^{i-1}}{2^k}, \frac{h_k^i}{2^\ell} - \frac{h_\ell^i}{2^k}\right\}(\lambda_{2i-1} + \lambda_{2i}) \ge \frac{z_k}{2^\ell} - \frac{z_\ell}{2^k} \quad \forall\{k,\ell\} \in [r]^2 \\
(\lambda, z) \in \Delta^{2d} \times \mathbb{R}^r.
\end{gathered}$$