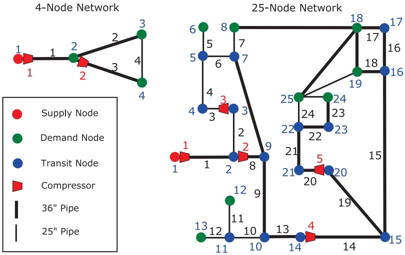
Building advanced MIP formulations combinatorially

Application: Gas network optimization

- Min-cost flow from producers to consumers [Martin 2006]
- *Wrinkle*: Gas pressure drops from friction in pipes
- Pressure loss is nonlinear: $p_v = f(p_u, q_{u,v})$

(p_u = pressure at node u, $q_{u,v}$ = gas flow volume on edge (u, v))



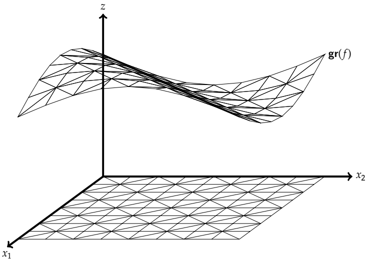
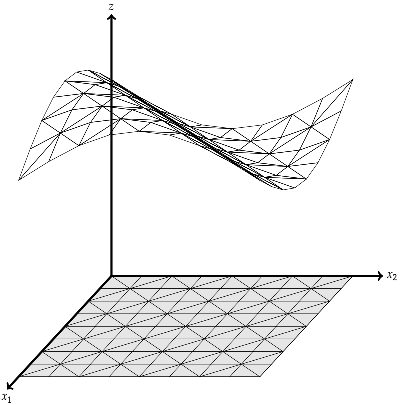
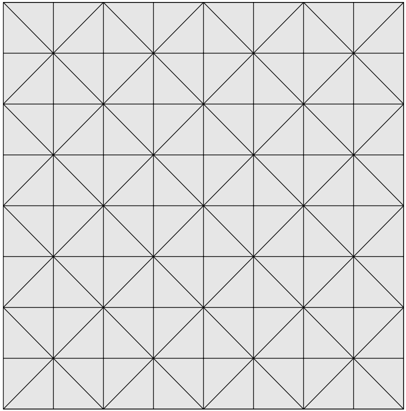


Figure from Wu et al. (2017).

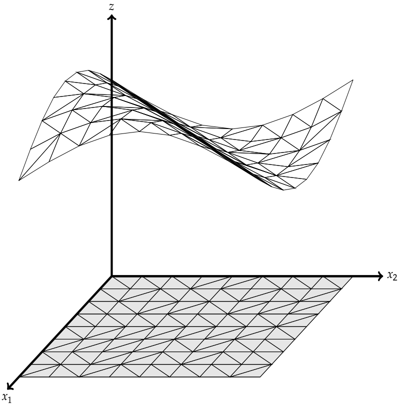
Bivariate piecewise linear functions

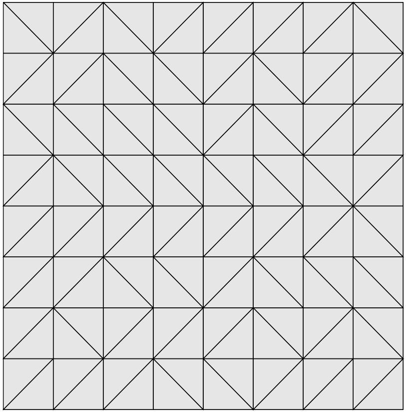




Union Jack triangulation

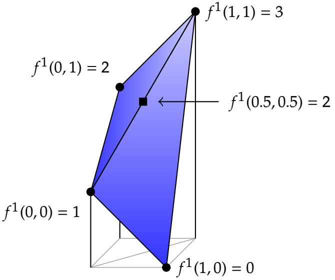
Bivariate piecewise linear functions





An unstructured triangulation...

Bivariate piecewise linear functions

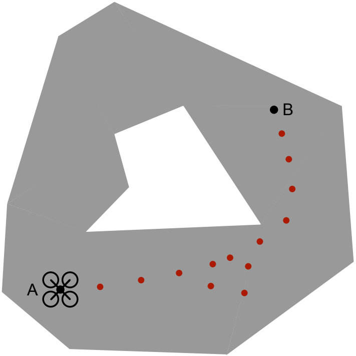


...because triangulation affects values piecewise linear function takes

Application: Obstacle avoidance

- Goal: Navigate quadcopter from point A to point B
- For Ω = domain − obstacles [Prodan 2012, 2016]:

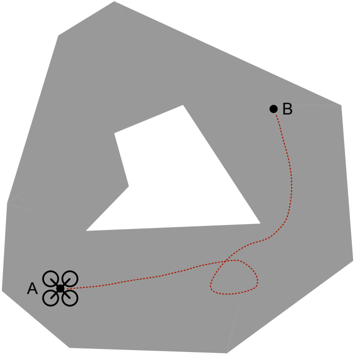
$$x^i \in \Omega \quad \forall i \in \{1, \ldots, m\}$$



Application: Obstacle avoidance

- Goal: Navigate quadcopter from point A to point B
- For Ω = domain − obstacles [Deits 2015,Kuindersma 2016]:

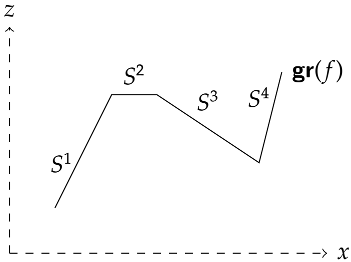
$$\forall t \in [0,1]: \quad x(t) \in \Omega$$



Nonconvex optimization using mixed-integer programming

1. Write as a *disjunctive set*:

$$\mathbf{gr}(f) = \bigcup_{i=1}^{d} S^i \subseteq \mathbb{R}^n$$

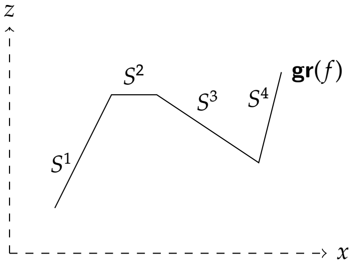


Nonconvex optimization using mixed-integer programming

1. Write as a *disjunctive set*:

$$\mathbf{gr}(f) = \bigcup_{i=1}^{d} S^i \subseteq \mathbb{R}^n$$

2. Introduce integer variables $y \in \mathbb{Z}^r$



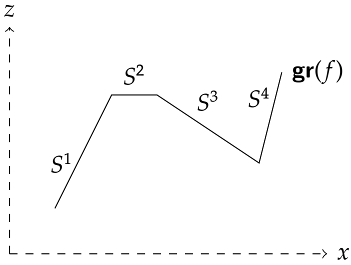
Nonconvex optimization using mixed-integer programming

1. Write as a *disjunctive set*:

$$\mathbf{gr}(f) = \bigcup_{i=1}^{d} S^i \subseteq \mathbb{R}^n$$

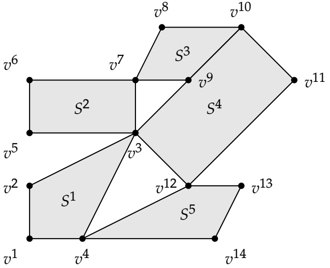
2. Introduce integer variables $y \in \mathbb{Z}^r$
3. Build LP relaxation $Q \subseteq \mathbb{R}^{n+r}$ so:

$$\mathrm{Proj}_{1:n}\left(Q \cap (\mathbb{R}^n \times \mathbb{Z}^r)\right) = \bigcup_{i=1}^{d} S^i$$



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

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

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

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

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

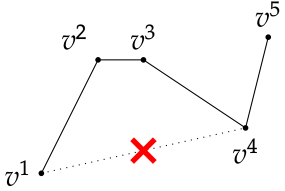
Formulating the SOS2 constraint

- 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, 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)

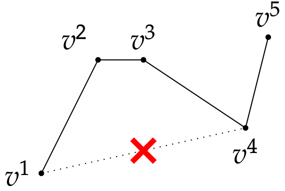
Formulating the SOS2 constraint

- 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)

Warm-up: Revisiting the Log formulation

- Recall that Log for 4 segments is:

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

Warm-up: Revisiting the $\texttt{Log}$ formulation

- Recall that $\texttt{Log}$ for 4 segments is:

$$
\begin{aligned}
y_1 = 1 \Longrightarrow \lambda_1 = \lambda_5 = 0, &\quad \text{and} \quad y_1 = 0 \Longrightarrow \lambda_3 = 0 \\
y_2 = 1 \Longrightarrow \lambda_1 = \lambda_2 = 0, &\quad \text{and} \quad y_2 = 0 \Longrightarrow \lambda_4 = \lambda_5 = 0 \\
(\lambda, y) \in \Delta^5 &\times \{0,1\}^2.
\end{aligned}
$$

Warm-up: Revisiting the Log formulation

- Recall that Log for 4 segments is:

$$y_1 = 1 \Longrightarrow \lambda_1 = \lambda_5 = 0, \quad \text{and} \quad y_1 = 0 \Longrightarrow \lambda_3 = 0$$
$$y_2 = 1 \Longrightarrow \lambda_1 = \lambda_2 = 0, \quad \text{and} \quad y_2 = 0 \Longrightarrow \lambda_4 = \lambda_5 = 0$$
$$(\lambda, y) \in \Delta^5 \times \{0,1\}^2.$$

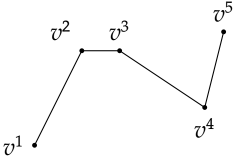
Definition (Vielma 2011)

Given $A^j, B^j \subseteq [n]$, an *independent branching formulation* is

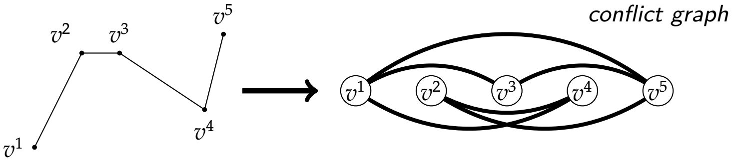
$$\sum_{v \in A^j} \lambda_v \leq y_j, \quad \sum_{v \in B^j} \lambda_v \leq 1 - y_j, \quad y_j \in \{0,1\} \quad \forall j \in [t].$$

- Strong (ideal), and size = $O(t)$
- Ad-hoc formulations for SOS2, "Union Jack" [Vielma 2011]

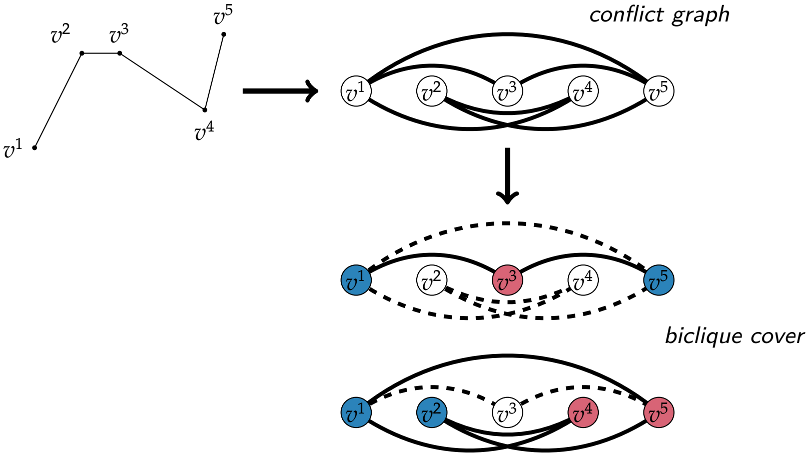
A combinatorial way to build formulations



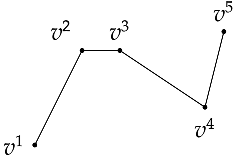
A combinatorial way to build formulations

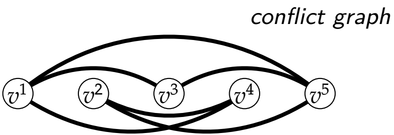


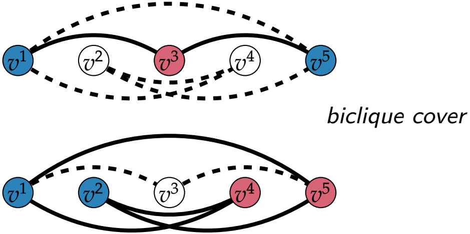
A combinatorial way to build formulations



A combinatorial way to build formulations

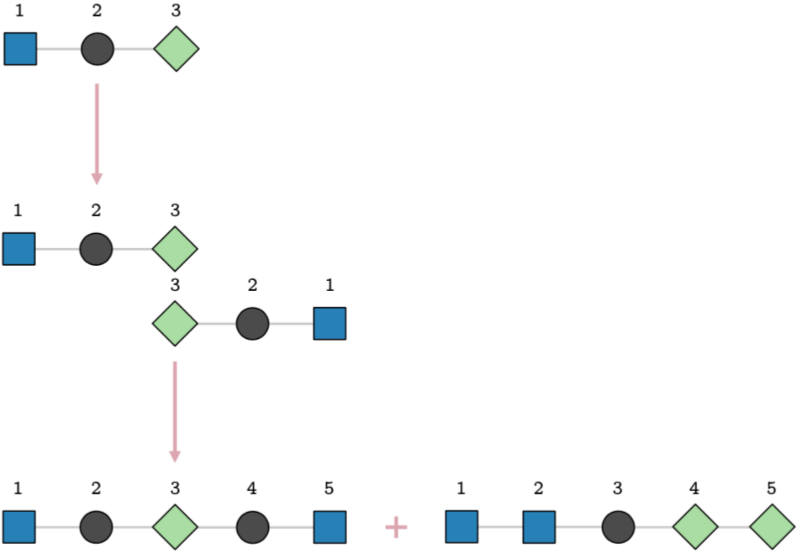




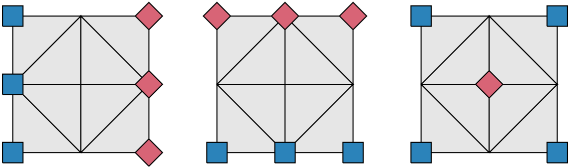


$$\lambda_1 + \lambda_5 \le 1 - y_1$$
$$\lambda_3 \le y_1$$
$$\lambda_1 + \lambda_2 \le 1 - y_2$$
$$\lambda_4 + \lambda_5 \le y_2$$
$$(\lambda, y) \in \Delta^5 \times \{0,1\}^2$$

A recursive construction for SOS2



Union Jack triangulation

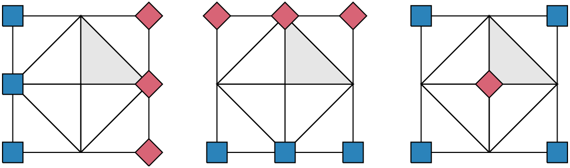


Level 1 Level 2 Level 3

$$
\begin{array}{llll}
\text{Level 1:} & \lambda_{(0,0)} + \lambda_{(0,1)} + \lambda_{(0,2)} \le y_1, & \lambda_{(2,0)} + \lambda_{(2,1)} + \lambda_{(2,2)} \le 1 - y_1 \\
\text{Level 2:} & \lambda_{(0,0)} + \lambda_{(1,0)} + \lambda_{(2,0)} \le y_2, & \lambda_{(0,2)} + \lambda_{(1,2)} + \lambda_{(2,2)} \le 1 - y_2 \\
\text{Level 3:} & \lambda_{(0,0)} + \lambda_{(0,2)} + \lambda_{(0,2)} + \lambda_{(2,2)} \le y_3, & \lambda_{(1,1)} \le 1 - y_3
\end{array}
$$

$$y \in \{0,1\}^3$$

Union Jack triangulation

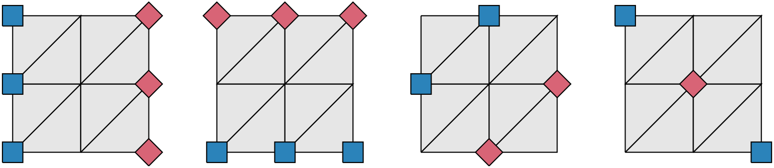


$$
\begin{array}{lll}
\text{Level 1:} & \lambda_{(0,0)} + \lambda_{(0,1)} + \lambda_{(0,2)} \leq 0, & \lambda_{(2,0)} + \lambda_{(2,1)} + \lambda_{(2,2)} \leq 1 \\
\text{Level 2:} & \lambda_{(0,0)} + \lambda_{(1,0)} + \lambda_{(2,0)} \leq 0, & \lambda_{(0,2)} + \lambda_{(1,2)} + \lambda_{(2,2)} \leq 1 \\
\text{Level 3:} & \lambda_{(0,0)} + \lambda_{(0,2)} + \lambda_{(0,2)} + \lambda_{(2,2)} \leq 0, & \lambda_{(1,1)} \leq 1
\end{array}
$$

$$y = (0, 0, 0)$$

- What happens when we fix y?

Another triangulation

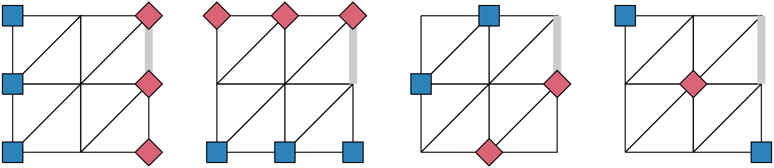


$$
\begin{array}{llll}
\text{Level 1:} & \lambda_{(0,0)} + \lambda_{(0,1)} + \lambda_{(0,2)} \le y_1, & \lambda_{(2,0)} + \lambda_{(2,1)} + \lambda_{(2,2)} \le 1 - y_1 \\
\text{Level 2:} & \lambda_{(0,0)} + \lambda_{(1,0)} + \lambda_{(2,0)} \le y_2, & \lambda_{(0,2)} + \lambda_{(1,2)} + \lambda_{(2,2)} \le 1 - y_2 \\
\text{Level 3:} & \lambda_{(0,1)} + \lambda_{(1,2)} \le y_3, & \lambda_{(1,0)} + \lambda_{(2,1)} \le 1 - y_3 \\
\text{Level 4:} & \lambda_{(0,2)} + \lambda_{(2,0)} \le y_4, & \lambda_{(1,1)} \le 1 - y_4
\end{array}
$$

$$y \in \{0,1\}^4$$

- SOS2 on x + SOS2 on y + "triangle selection" with y_3 and y_4

Another triangulation



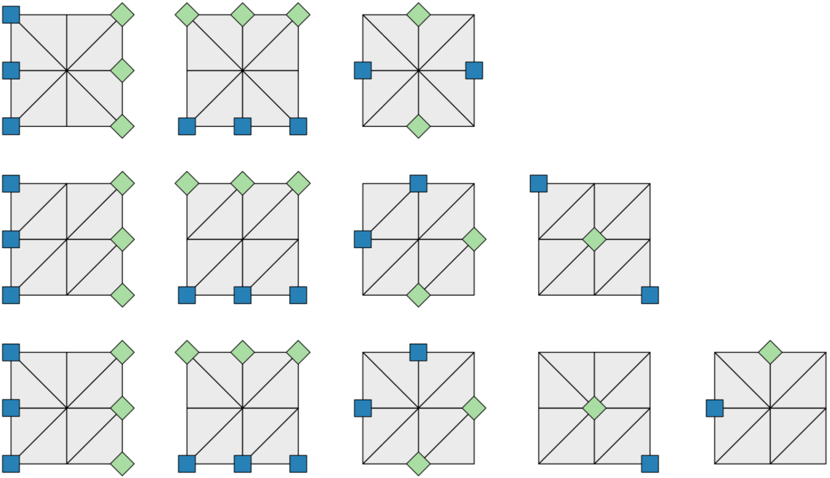
Level 1 Level 2 Level 3 Level 4

$$
\begin{array}{lll}
\text{Level 1:} & \lambda_{(0,0)} + \lambda_{(0,1)} + \lambda_{(0,2)} \leq 0, & \lambda_{(2,0)} + \lambda_{(2,1)} + \lambda_{(2,2)} \leq 1 \\
\text{Level 2:} & \lambda_{(0,0)} + \lambda_{(1,0)} + \lambda_{(2,0)} \leq 0, & \lambda_{(0,2)} + \lambda_{(1,2)} + \lambda_{(2,2)} \leq 1 \\
\text{Level 3:} & \lambda_{(0,1)} + \lambda_{(1,2)} \leq 0, & \lambda_{(1,0)} + \lambda_{(2,1)} \leq 1 \\
\text{Level 4:} & \lambda_{(0,2)} + \lambda_{(2,0)} \leq 1, & \lambda_{(1,1)} \leq 0
\end{array}
$$

$$y = (0, 0, 0, 1)$$

- What happens when we fix y? Not equal to embedding...

Even more triangulations



What is **independent branching**?

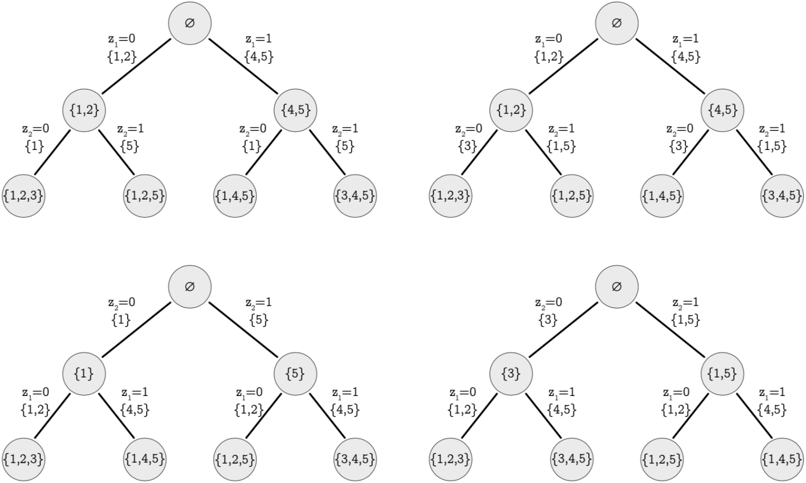
$$
\begin{aligned}
\lambda_1 + \lambda_2 \le z_1, \qquad & \lambda_4 + \lambda_5 \le 1 - z_1 \\
\lambda_3 \le z_2, \qquad & \lambda_1 + \lambda_5 \le 1 - z_2 \\
(\lambda, z) \in \Delta^5 & \times \{0,1\}^2.
\end{aligned}
$$

Log formulation for SOS2 on 5 components

$$
\begin{gathered}
\lambda_1 = \gamma_1^{\{1,2\}}, \quad \lambda_2 = \gamma_2^{\{1,2\}} + \gamma_2^{\{2,3\}}, \lambda_3 = \gamma_3^{\{2,3\}} + \gamma_3^{\{3,4\}}, \\
\lambda_4 = \gamma_4^{\{3,4\}} + \gamma_4^{\{4,5\}}, \quad \lambda_5 = \gamma_5^{\{4,5\}} \\
\gamma_1^{\{1,2\}} + \gamma_2^{\{1,2\}} + \gamma_2^{\{2,3\}} + \gamma_3^{\{2,3\}} + \gamma_3^{\{3,4\}} + \gamma_4^{\{3,4\}} + \gamma_4^{\{4,5\}} + \gamma_5^{\{4,5\}} = 1 \\
\gamma_1^{\{1,2\}} + \gamma_2^{\{1,2\}} + \gamma_2^{\{2,3\}} + \gamma_3^{\{2,3\}} = z_1 \\
\gamma_1^{\{1,2\}} + \gamma_2^{\{1,2\}} + \gamma_3^{\{3,4\}} + \gamma_4^{\{3,4\}} = z_2 \\
\gamma_v^S \ge 0 \quad \forall v, S \\
(\lambda, z) \in \Delta^5 \times \{0,1\}^2
\end{gathered}
$$

Disaggregated Log formulation (i.e. encoding of SOS2 with binary encoding) for SOS2 on 5 components

What is **independent branching**?



(Left) DLog, (Right) Log.
(Top) Branch on z_1 first, (Bottom) branch on z_2 second.

Independent branching formulations: Existence

- Do independent branching formulations exist for any constraint?
- An *infeasible set* $S \subseteq [n]$ is one for which

$$S \nsubseteq T^i \quad \forall i$$

(i.e. $\lambda = \frac{1}{|S|} \sum_{v \in S} \mathbf{e}^v \notin \bigcup_i P(T^i)$ is infeasible)

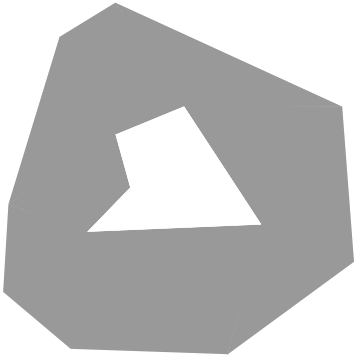
Theorem

An independent branching formulation exists if and only if each minimal infeasible set has cardinality two.

- What does this mean for...
  - SOS1?
  - SOS2?
  - Grid triangulations?
  - Cardinality constraints? ($\mathscr{T} = \{T \subseteq [n] : |T| = k\}$)

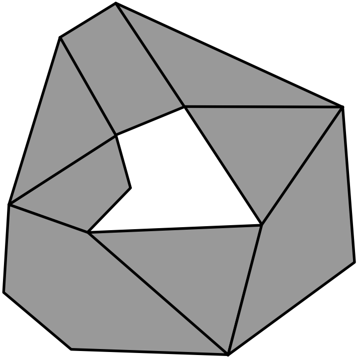
Obstacle avoidance

- Partition Ω into union of polyhedra (no *internal vertices*)



Obstacle avoidance

- Partition Ω into union of polyhedra (no *internal vertices*)

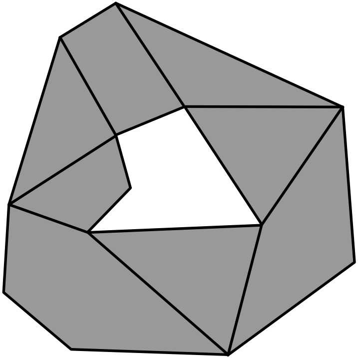


Obstacle avoidance

- Partition Ω into union of polyhedra (no *internal vertices*)

Theorem

An independent branching formulation exists if and only if each infeasible triplet S (i.e. $|S| = 3$, $S \nsubseteq T^i$ for each i) is not minimal.

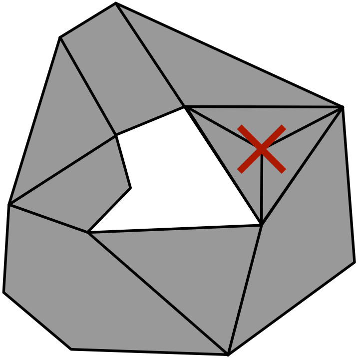


Obstacle avoidance

- Partition Ω into union of polyhedra (no *internal vertices*)

Theorem

An independent branching formulation exists if and only if each infeasible triplet S (i.e. $|S| = 3$, $S \nsubseteq T^i$ for each i) is not minimal.



- Consequence: Representability can be checked efficiently, easy to guarantee

Independent branching formulations

- *Conflict graph*: $\mathscr{G}^c = ([n], E)$, where

$$E = \left\{\{u,v\} \in [n]^2 : \{u,v\} \not\subseteq T^i \text{ for each } i\right\}$$

- *Biclique cover* for $\mathscr{G}^c$: $\{(A^j, B^j)\}_{j=1}^t$ where $E = \bigcup_{j=1}^t (A^j \times B^j)$

Theorem (H. and Vielma 2016)

If an independent branching formulation exists for $\bigcup_{i=1}^d P(T^i)$, then

$$\sum_{v \in A^j} \lambda_v \leq y_j, \quad \sum_{v \in B^j} \lambda_v \leq 1 - y_j, \quad y_j \in \{0,1\} \quad \forall j \in [t]$$

is an independent branching formulation if and only if $\{(A^j, B^j)\}_{j=1}^t$ is a biclique cover for $\mathscr{G}^c$.

Towards general construction methods

- Covering with stars:

$$A^v = \{v\}, \qquad B^v = \{u \in [n] : \{u, v\} \in E\} \quad \forall v \in [n]$$

Towards general construction methods

- Covering with stars:

$$A^v = \{v\}, \qquad B^v = \{u \in [n] : \{u,v\} \in E\} \quad \forall v \in [n]$$

- Graph products: for $\{G^i = (J^i, E^i)\}_i$,

$$\prod_i G^i \equiv \left(\prod_i J^i, \left\{ \{u,v\} : \exists i \text{ s.t. } \{u_i, v_i\} \in E^i \right\} \right).$$

Towards general construction methods

- Covering with stars:

$$A^v = \{v\}, \qquad B^v = \{u \in [n] : \{u, v\} \in E\} \quad \forall v \in [n]$$

- Graph products: for $\{G^i = (J^i, E^i)\}_i$,

$$\prod_i G^i \equiv \left(\prod_i J^i, \left\{ \{u, v\} : \exists i \text{ s.t. } \{u_i, v_i\} \in E^i \right\} \right).$$

  - SOS2 –> grid discretization of hypercube

Towards general construction methods

- Covering with stars:

$$A^v = \{v\}, \qquad B^v = \{u \in [n] : \{u, v\} \in E\} \quad \forall v \in [n]$$

- Graph products: for $\{G^i = (J^i, E^i)\}_i$,

$$\prod_i G^i \equiv \left(\prod_i J^i, \left\{ \{u, v\} : \exists i \text{ s.t. } \{u_i, v_i\} \in E^i \right\} \right).$$

    - SOS2 –> grid discretization of hypercube

- Graph unions: for $\{G^i = (J^i, E^i)\}_i$,

$$\bigcup_i G^i \equiv \left(\bigcup_i J^i, \bigcup_i E^i \right).$$

Towards general construction methods

- Covering with stars:

$$A^v = \{v\}, \qquad B^v = \{u \in [n] : \{u, v\} \in E\} \quad \forall v \in [n]$$

- Graph products: for $\{G^i = (J^i, E^i)\}_i$,

$$\prod_i G^i \equiv \left(\prod_i J^i, \left\{ \{u, v\} : \exists i \text{ s.t. } \{u_i, v_i\} \in E^i \right\} \right).$$

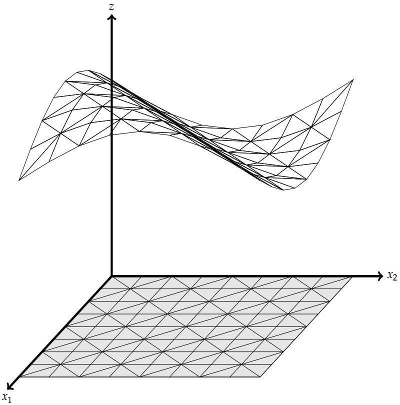
  - SOS2 –> grid discretization of hypercube

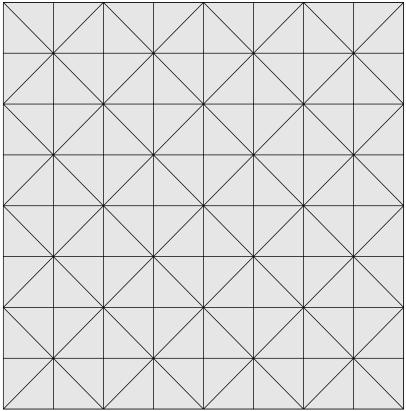
- Graph unions: for $\{G^i = (J^i, E^i)\}_i$,

$$\bigcup_i G^i \equiv \left(\bigcup_i J^i, \bigcup_i E^i \right).$$

  - Grid discretization –> grid triangulation

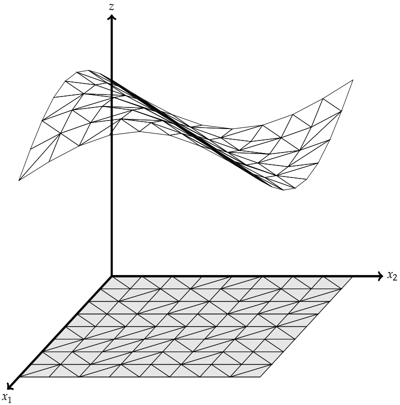
Bivariate piecewise linear functions

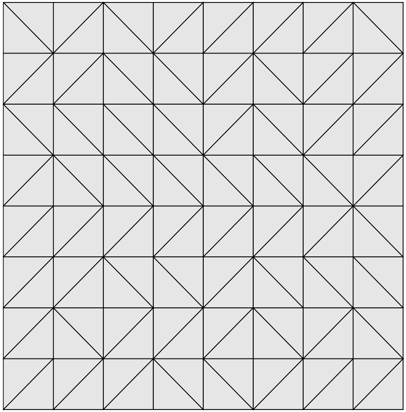




Union Jack triangulation

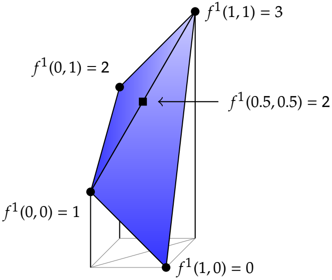
Bivariate piecewise linear functions

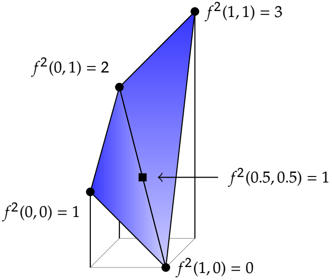




An unstructured triangulation...

Bivariate piecewise linear functions





...because triangulation affects values piecewise linear function takes

How small can we make t?

1. Union Jack triangulation:
   - $t \approx \log_2(\text{\# of breakpoints}) + 1$ [Vielma 2011]

How small can we make t?

1. Union Jack triangulation:
   - $t \approx \log_2(\text{\# of breakpoints}) + 1$ [Vielma 2011]
2. General triangulation:
   - $t = \text{\# of breakpoints}$ (covering with stars)

How small can we make t?

1. Union Jack triangulation:
   - $t \approx \log_2(\text{\# of breakpoints}) + 1$ [Vielma 2011]
2. General triangulation:
   - $t = \text{\# of breakpoints}$ (covering with stars)

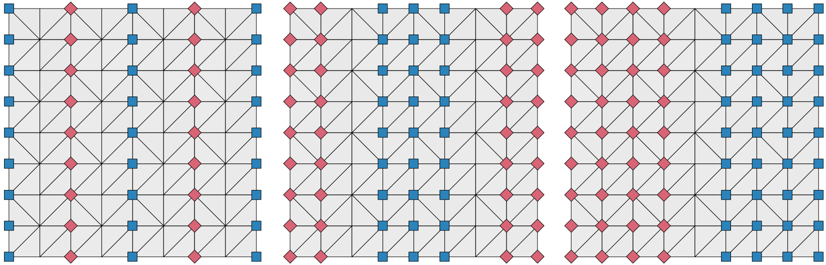
   Theorem

   There exists an independent branching formulation for any grid triangulation of size $t \approx \log_2(\text{\# breakpoints}) + 9$.

   - Optimal up to a (small) constant additive factor

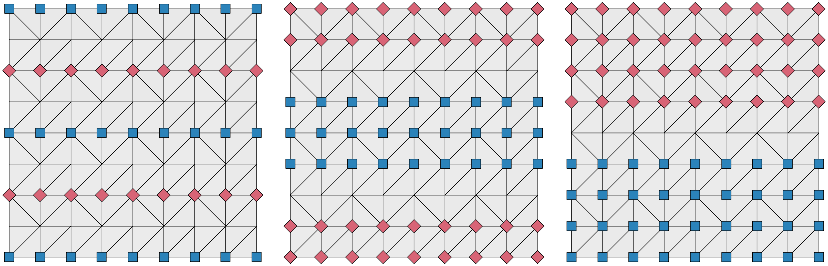
9-Stencil formulation: Stage 1a

- Aggregated SOS2 along x direction
- Separated edges between vertices that are “far apart” in x direction
- Needs $\lceil \log_2$(# breakpoints in x direction)$\rceil$ levels (variables)



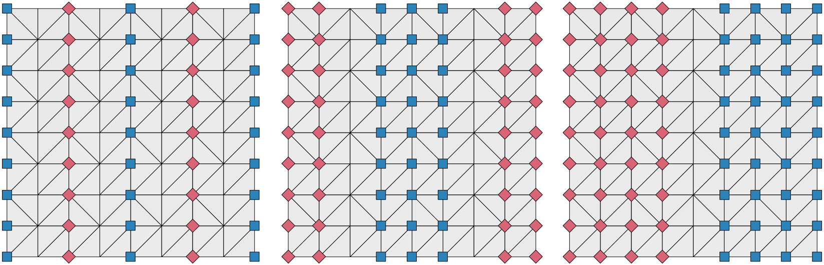
9-Stencil formulation: Stage 1b

- Aggregated SOS2 along y direction
- Separated edges between vertices that are "far apart" in y direction
- Needs $\lceil \log_2$(# breakpoints in y direction)$\rceil$ levels (variables)



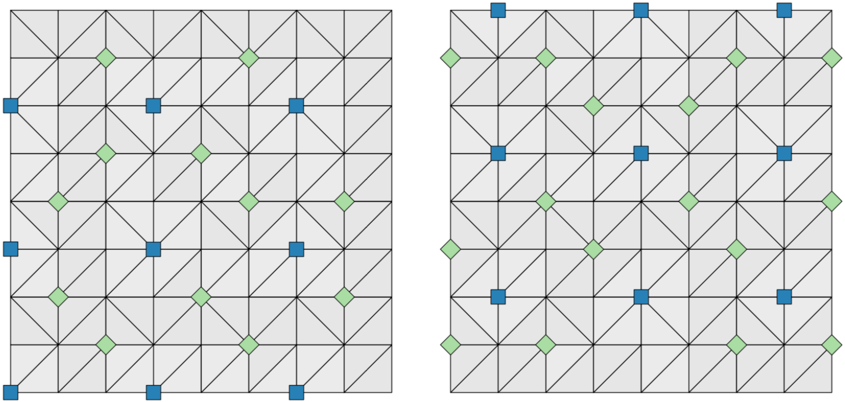
Stencil formulation: Stage 1a

- Aggregated SOS2 along x direction
- Separated edges between vertices that are "far apart" in x direction
- Needs $\lceil \log_2(\text{\# breakpoints in } x \text{ direction}) \rceil$ levels (variables)



9-Stencil formulation: Stage 2

- Aggregate stars on 3×3 grid
- Needs 9 levels (variables)



What about higher dimensions?

- Combinatorial approach generalizes to arbitrarily high dimensions

Theorem (H. and Vielma 2016)

There exists a biclique cover for the conflict graph of any d-dimensional grid triangulation of size $t \approx \log_2(\text{\# of breakpoints}) + 3^d$.

- Best previously known approach has $t =$ # of breakpoints
- Standard triangulation of $17 \times 17 \times 17$ grid:

$$t = 39 \text{ vs. } t = 4,913$$

How small can we make t?

1. Union Jack triangulation:
    - $t \approx \log_2(\text{\# of breakpoints}) + 1$ [Vielma 2011]

How small can we make t?

1. Union Jack triangulation:
   - $t \approx \log_2$(# of breakpoints) $+ 1$ [Vielma 2011]
2. General triangulation:
   - $t =$ # of breakpoints (covering with stars)

How small can we make t?

1. Union Jack triangulation:
   - $t \approx \log_2(\text{\# of breakpoints}) + 1$ [Vielma 2011]
2. General triangulation:
   - $t = \text{\# of breakpoints}$ (covering with stars)

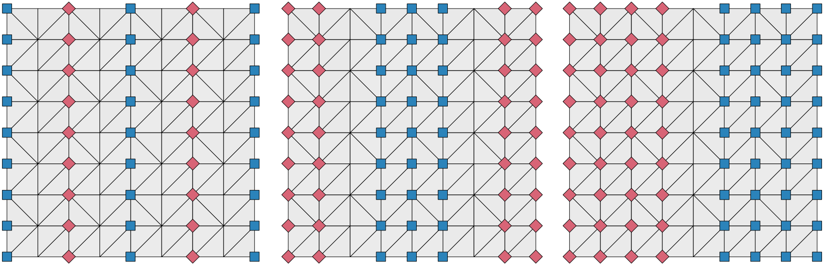
   Theorem

   There exists an independent branching formulation for any grid triangulation of size $t \approx \log_2(\text{\# breakpoints}) + 6$.

   - Optimal up to a (small) constant additive factor

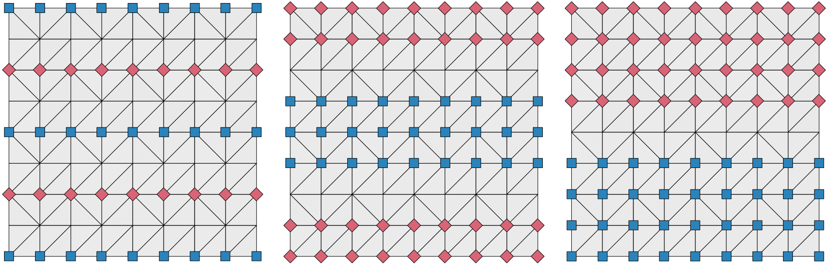
Stencil formulation: Stage 1a

- Aggregated SOS2 along x direction
- Separated edges between vertices that are "far apart" in x direction
- Needs $\lceil \log_2$(# breakpoints in x direction)$\rceil$ levels (variables)



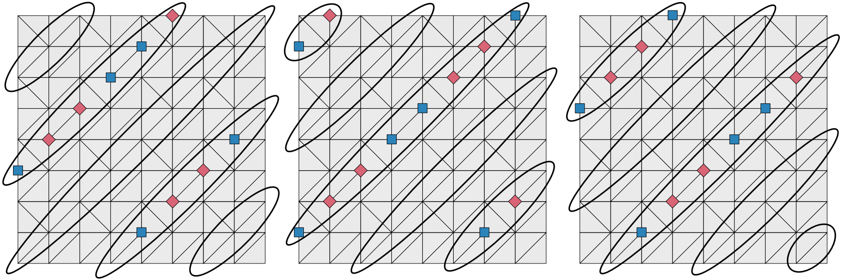
Stencil formulation: Stage 1b

- Aggregated SOS2 along y direction
- Separated edges between vertices that are "far apart" in y direction
- Needs $\lceil \log_2(\text{\# breakpoints in } y \text{ direction}) \rceil$ levels (variables)



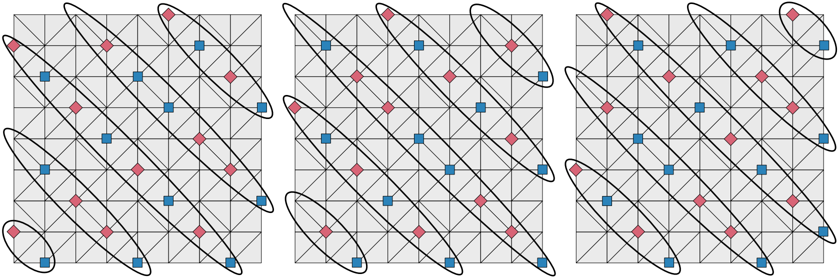
Stencil formulation: Stage 2a

- Separate all edges along diagonal lines
- Can aggregate diagonal lines that are “far apart”
- Needs 3 levels (variables)



Stencil formulation: Stage 2b

- Separate all edges along anti-diagonal lines
- Can aggregate anti-diagonal lines that are "far apart"
- Needs 3 levels (variables)



Bicommodity transportation problem

- Two commodities routed through same transportation network
- Cost of each arc is nonseparable function of two commodities
- Joint capacity constraint for each arc

$$
\begin{aligned}
\min_{x} \quad & \sum_{i \in S} \sum_{j \in D} f_{i,j}(x^1_{i,j}, x^2_{i,j}) \\
\text{s.t.} \quad & \sum_{j \in D} x^k_{i,j} = s^k_i \quad \forall i \in S, k \in \{1,2\} \\
& \sum_{i \in S} x^k_{i,j} = d^k_j \quad \forall j \in D, k \in \{1,2\} \\
& x^1_{i,j} + x^2_{i,j} \le C_{i,j} \quad \forall i \in S, j \in D \\
& x^1, x^2 \ge 0 \quad \forall i \in S, j \in D
\end{aligned}
$$

Bivariate piecewise linear functions: Computational experiments

N	Metric	`MC`	`CC`	`DLog`	`Stencil`
4	Mean (s)	1.4	1.5	0.9	0.4
	Win	0	0	0	100
8	Mean (s)	39.3	97.2	12.6	2.7
	Win	0	0	0	100
16	Mean (s)	1370.9	1648.1	352.8	24.6
	Fail	53	66	6	0
	Win	0	0	0	80
32	Mean (s)	1800.0	1800.0	1499.6	133.5
	Fail	80	80	50	0
	Win	0	0	0	80

Solve time (in seconds, with CPLEX v12.7.0). Functions have N^2 pieces, fixed network $|S| = |D| = 5$.

- New `Stencil` formulation is the fastest on every instance
- $>$10x speedup on average for medium/large instances
- Previous approaches could not solve 50 of 80 largest instances

Bivariate piecewise linear functions: Computational experiments

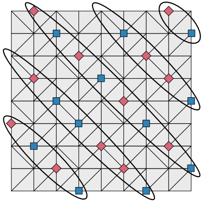
N	Metric	MC	CC	DLog	Stencil
4	Mean (s)	1.4	1.5	0.9	0.4
	Win	0	0	0	100
8	Mean (s)	39.3	97.2	12.6	2.7
	Win	0	0	0	100
16	Mean (s)	1370.9	1648.1	352.8	24.6
	Fail	53	66	6	0
	Win	0	0	0	80
32	Mean (s)	1800.0	1800.0	1499.6	133.5
	Fail	80	80	50	0
	Win	0	0	0	80

Solve time (in seconds, with CPLEX v12.7.0). Functions have N pieces, fixed network $|S| = |D| = 5$.

- New `Stencil` formulation is the fastest on every instance
- $>$10x speedup on average for medium/large instances
- Previous approaches could not solve 50 of 80 largest instances

Bivariate piecewise linear functions: Computational tools

- Formulations even more complicated now



- Even choosing the triangulation is nontrivial!
- PiecewiseLinearOpt.jl chooses triangulation, builds formulation for you

```
using JuMP, PiecewiseLinearOpt, CPLEX
m = Model(solver=CplexSolver())
@variable(m, x)
@variable(m, y)
z = piecewiselinear(m, x, y, 0:0.1:1, 0:0.1:1, (u,v) ->
 ↪  exp(u+v))
```

More crazy constructions...

The SOSk constraint

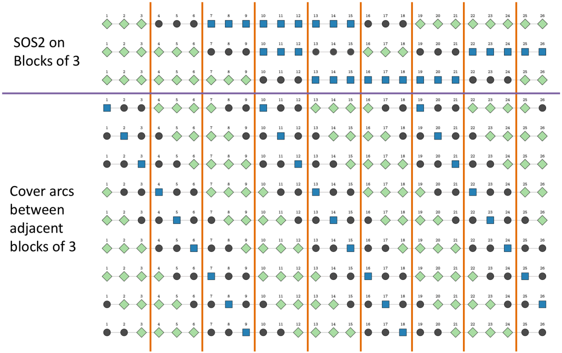
- $\lambda \in \Delta^N$: at most k nonzeros, consecutive on ordering

The SOSk constraint

- $\lambda \in \Delta^N$: at most k nonzeros, consecutive on ordering
- Simple construction with $N/2$ levels

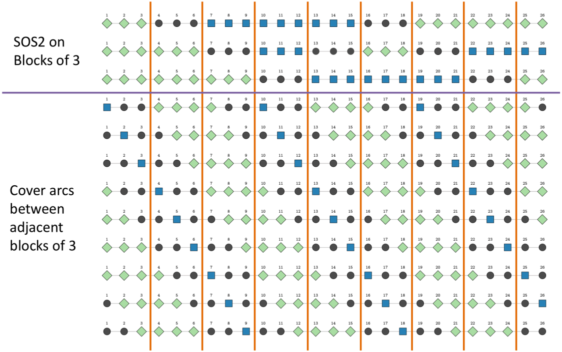
The SOSk constraint

- $\lambda \in \Delta^N$: at most k nonzeros, consecutive on ordering
- Simple construction with $N/2$ levels



The SOSk constraint

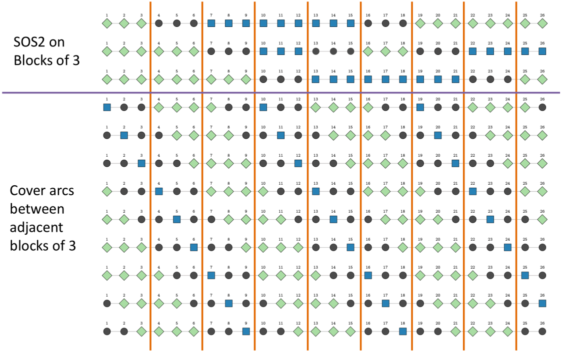
- $\lambda \in \Delta^N$: at most k nonzeros, consecutive on ordering
- Simple construction with $N/2$ levels



- Scales like $\log_2(N/k) + \mathscr{O}(k)$

The SOSk constraint

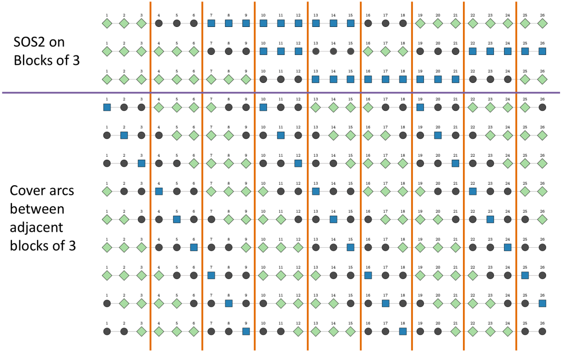
- $\lambda \in \Delta^N$: at most k nonzeros, consecutive on ordering
- Simple construction with $N/2$ levels



- Scales like $\log_2(N/k) + \mathcal{O}(k)$
- Independent branching lowerbound of k...tight when $k = \omega(\log(N))$

The SOSk constraint

- $\lambda \in \Delta^N$: at most k nonzeros, consecutive on ordering
- Simple construction with $N/2$ levels



- Scales like $\log_2(N/k) + \mathcal{O}(k)$
- Independent branching lowerbound of k...tight when $k = \omega(\log(N))$
- Standard MIP formulation lowerbound: $\mathcal{O}(\log(N))$

Colorings for grid triangulations

Theorem (Cornaz and Fonlupt (2006))

Take the graph $G = (J, \bar{E})$, along with some edge subset $\bar{F} \subseteq \bar{E}$. Define $V(\bar{F}) = \bigcup\{\{u, v\} \in \bar{F}\}$ as all nodes incident to $\bar{F}$, and take $F = (\bar{F} * \bar{F}) \backslash \bar{E}$ as all pairs incident to the edges $V(\bar{F})$ not contained in $\bar{E}$. Define both $E' = F \cup \bar{F}$ and $p : E' \to \{0, 1\}$ such that $p(e) = \mathbf{1}[e \in \bar{E}]$. Finally, take $\mathscr{C}(E')$ as the family of all cycles in G'. Then the following are equivalent:

1. *There exists a biclique (A, B) of G covering $(V(\bar{F}), \bar{F})$.*
2. *For all $C \in \mathscr{C}(E')$, $\sum_{u \in C} p(u)$ is even.*
3. *There exists some $f : V(\bar{F}) \to \{0, 1\}$ such that*
   - *$f(u) = f(v)$ for all $\{u, v\} \in F$,*
   - *$f(u) \neq f(v)$ for all $\{u, v\} \in \bar{F}$, and*
   - *$\big(\{u \in V(\bar{F}) : f(u) = 0\}, \{u \in V(\bar{F}) : f(u) = 1\}\big)$ is a biclique of G covering $(V(\bar{F}), \bar{F})$.*

Colorings for grid triangulations

- Consequence: cover certain triangulations with 1 or 2 additional levels
- Subsumes Union Jack construction