Finally Wide Split Cuts in a MIP solver?

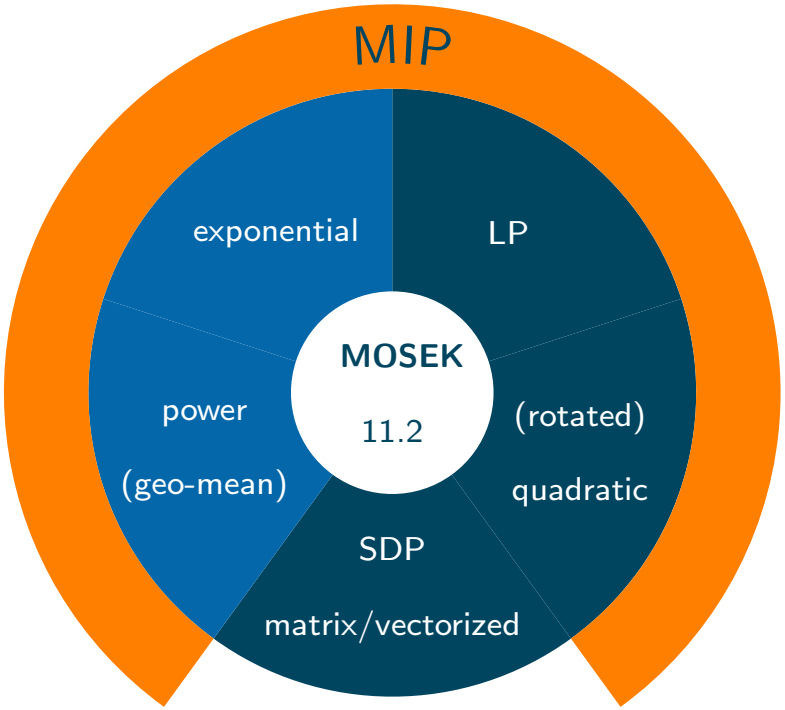
24th IFORS conference
July 13th, 2026
Vienna, Austria

Sven Wiese

www.mosek.com

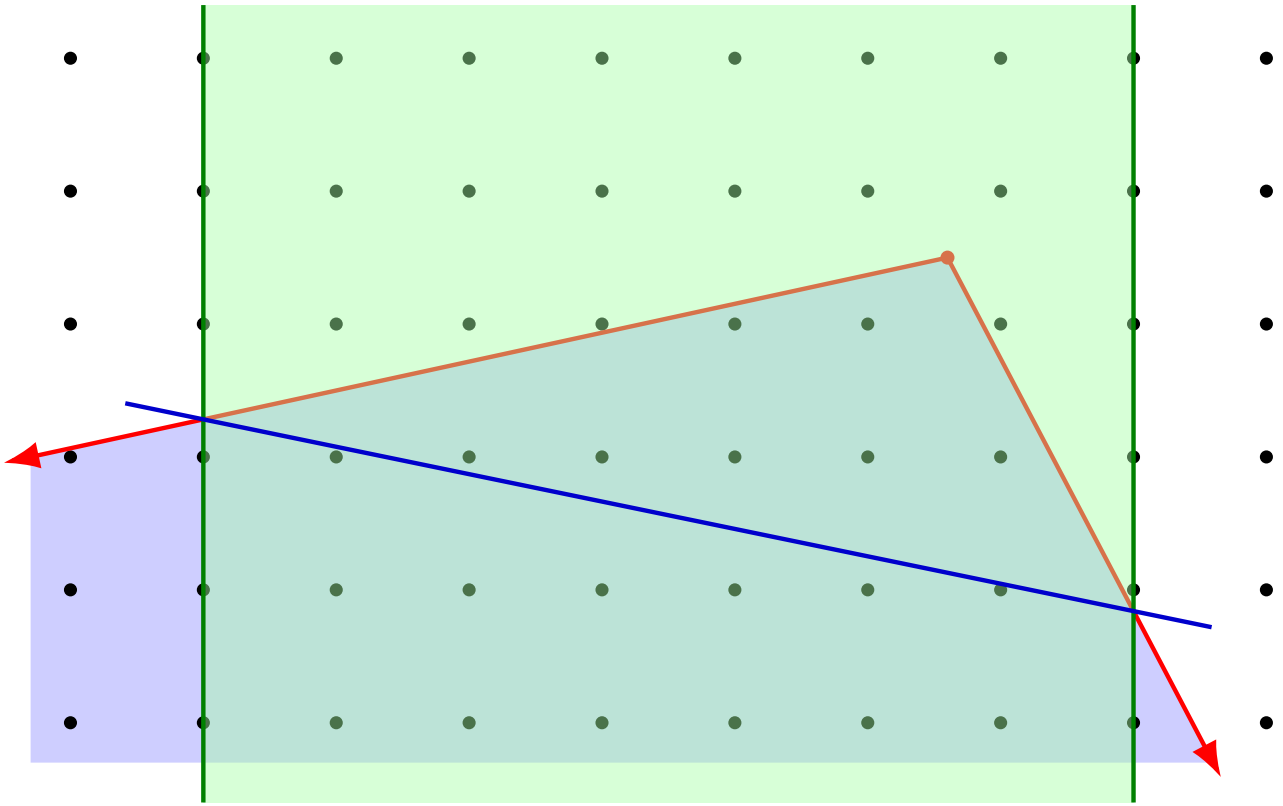
What's inside **MOSEK**?

Convex Cones!



- Simplex / Interior-point algorithms for continuous problems.
- (Conic) Branch-and-Cut / Outer-approximation algorithms for discrete problems.

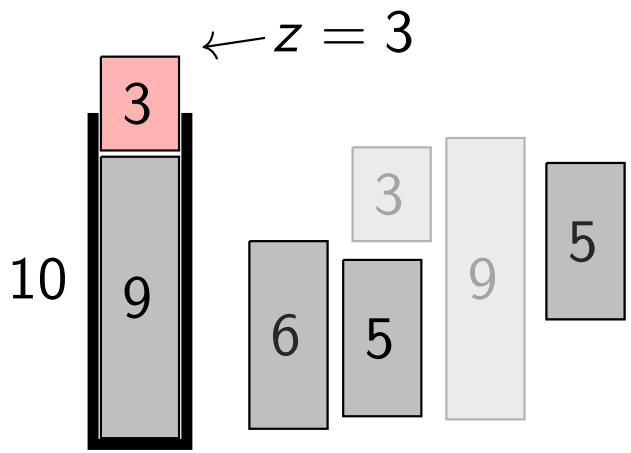
2015: First experiments with Wide Split Cuts



Wider split disjunctions lead to stronger cuts!

Are there valid wide split disjunctions in MIPs?

The Lazy Bureaucrat Problem (Furini, Ljubić, Sinnl, 2015):



$$\min_{(x,z)\in\{0,1\}^n\times\mathbb{Z}_+} \left\{ c^\top x : w^\top x \le C,\ w^\top x + z \ge C+1,\ z \le w_j + Mx_j \right\}$$

$\Longrightarrow$ It is enough to consider $z \in \{w_j \mid j = 1, \ldots, n\}$.

$\Longrightarrow$ $z \le w_j$ OR $z \ge w_{j+1}$ are valid (wide) split disjunctions.

Cut separation in the Bureaucrat Problem

- 10 rounds of cuts
- w/o: separate (ordinary) split cut for every basic variable
- w: separate split cut (possibly wide) for every basic variable

	% gap closed		#cuts	
instance	w/o	w	w/o	w
10000-4-10-75	34.31	42.21	55	32
10000-4-30-25	22.37	29.34	51	42
10000-4-100-75	9.21	12.53	139	62
10000-4-10-25	32.97	34.28	46	27
10000-4-500-50	1.10	6.17	170	137
10000-4-20-50	16.58	22.69	38	43
1000-4-20-25	38.73	100.00	48	9
1000-4-10-50	35.99	68.49	37	31
⋮	⋮	⋮	⋮	⋮
mean	19.81	31.95	89.92	49.16

2026: Can we do this in a general-purpose MIP solver?

- work in progress -

- Bureaucrat Problem: split validity comes from special problem knowledge.
- Can we detect valid wide split disjunctions in general MIPs?
- I.e., can we detect holes in some variables' domains, both discrete or continuous?
- Can we exploit the hole information in Branch-and-Cut?
- ... in a MIP solver like **MOSEK**?

Detecting holes I: Multiple-choice structure

If we have

$$y = \sum_{j=1}^{n} a_j x_j, \qquad \sum_{j=1}^{n} x_j = 1, \qquad x_j \in \{0, 1\}$$

then the domain of y is $\{a_1, \ldots, a_n\}$.

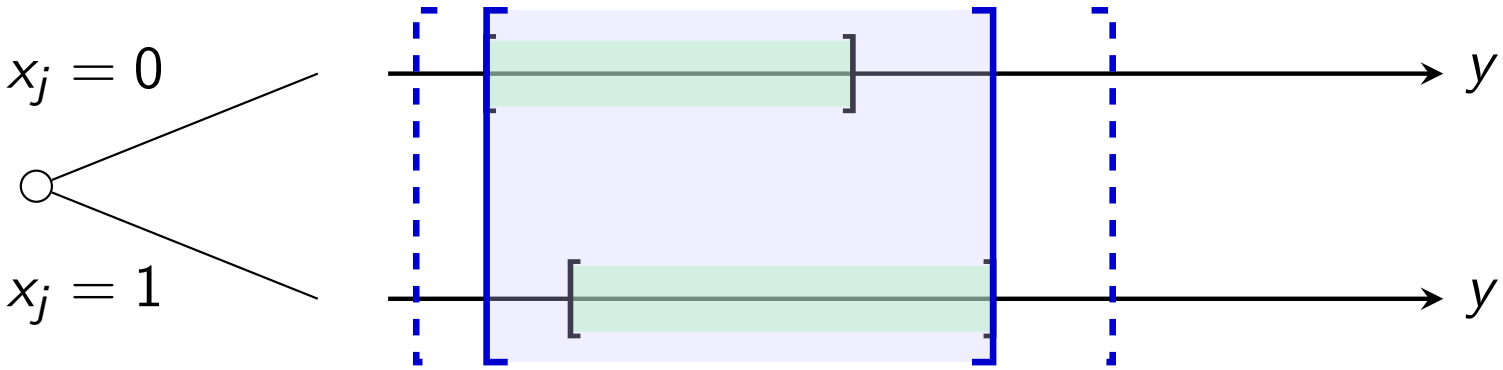
In particular,

$$y \leq a_j \text{ OR } y \geq a_{j+1}$$

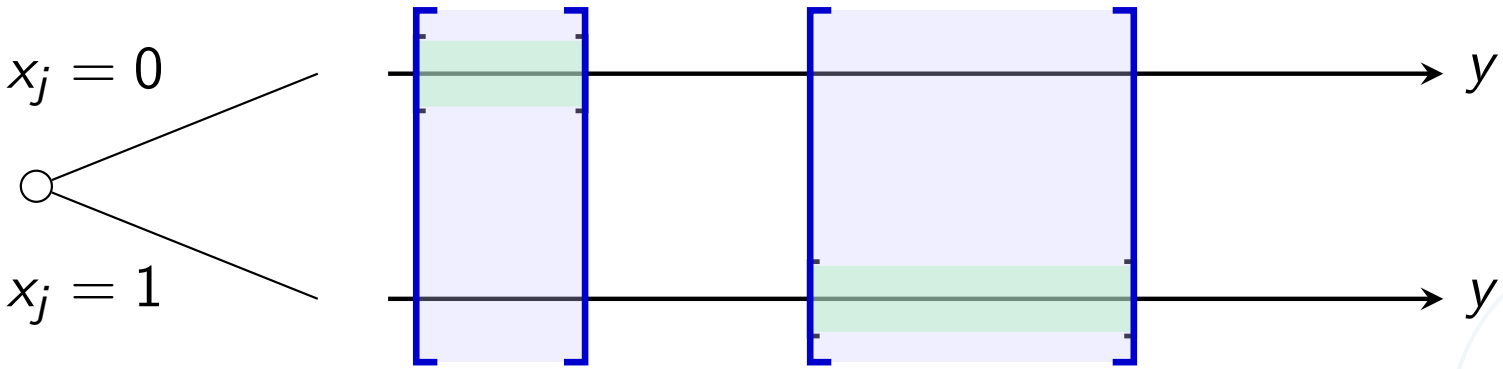
are valid disjunctions, i.e., (a_j, a_{j+1}) are domain holes.

Detecting holes II: Probing

Probing: gain information from setting either $x_j = 0$ or $x_j = 1$ and the resulting bound reductions.



Probing may reveal stronger global domains!



Probing may also reveal domain holes!

Detecting holes III: expression enumeration

Assume:

- $x_{j_0} = \sum_{j=1}^{n} a_j x_j$
- We can enumerate all possible values $D = \{v_1, \ldots, v_k\}$ of r.h.s.

Then x_{j_0} has *discrete* domain D.

Dual view:

- $c_{j_0} \leq 0$
- $x_{j_0} \leq \sum_{j=1}^{n} a_{ij} x_j \ \forall i = 1, \ldots, m$
- We can enumerate all possible values D_i of each r.h.s.

Then we may set the domain of x_{j_0} to $\bigcup_{i=1,\ldots,m} D_i$.

Are these detections possible?

- Internal test set with ~ 550 instances where **MOSEK**'s presolve detects holes.
- 246 are from the MIPLIB 2017 collection set ($\sim 23\%$).
- With loose limits (max expression length ≤ 25) detection times are ~ 0.05 sec. in geo. mean.
  - Some outliers up to 250 sec.
  - 85th percentile is ~ 0.01 sec.

Finally Wide Split Cuts then?

- as before: cut strategies w/o vs. w

	% gap closed		#cuts	
instance	w/o	w	w/o	w
drayage-25-2	20.60	25.51	376	569
aflow40b	14.82	17.70	253	607
gasprod2-1	3.67	4.34	1263	1267
10000-4-20-50	0.36	6.63	36037	11361
lr1dr02vc05v8a-t360	1.12	0.44	640	641
neos-3355120-tarago	23.97	14.80	1720	1649
⋮	⋮	⋮	⋮	⋮
mean	5.02	4.73	326.42	360.55

- Very heterogeneous results over the whole test set!
- $\Longrightarrow$ Likely need techniques specific to problem / hole type?

More hole exploitation I: "Redundancy bounds"

Example: $293 \leq x_0 \leq 585$, $c_0 = 1$, only downlock is

$$x_0 \geq 293 + 200x_1 + 200x_2, \quad x_1, x_2 \in \{0, 1\}.$$

Enumeration of r.h.s $= \{293, 493, 693\}$ = valid dual domain of x_0.

- $x_0 = 693$ is infeasible
- $\implies x_0 \in \{293, 493\}$
- $\implies x_0 \leq 493$

$\implies$ Few problems with reduction in $\#nnz$ by up to 13%.

More hole exploitation II: Hole-aware heuristics

- Some heuristics tentatively fix variables to some values.
- So never fix a variable to a value in a hole!
- $\Longrightarrow$ Slight increase in success rates of Shift-and-Propagate and Feasibility Jump heuristics.
- What about RINS, RENS, ...?

More hole exploitation III: Hole Propagation

- After branching and subsequent propagation: $\tilde{l} \leq y \leq \tilde{u}$.
- If $\tilde{u}$ lies in hole (h_l, h_u), strengthen the new bound: $y \leq h_l$.

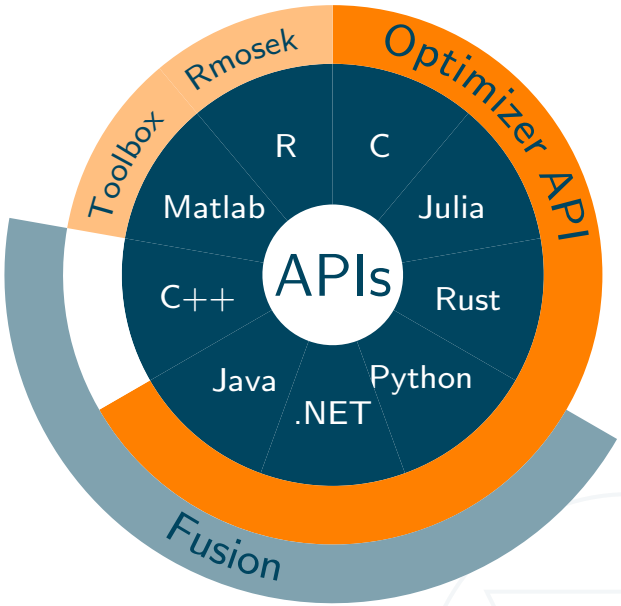
	hole propagation			
	models	affected	time	nodes
$\geq$ 0 sec.	1116	534	1.01	1.00
$\geq$ 10 sec.	589	378	1.00	0.99
$\geq$ 100 sec.	310	223	0.99	0.95
$\geq$ 1000 sec.	146	108	0.83	0.81

- Slightly worse on average, some drastically positive outliers...
- $\Longrightarrow$ Again: discriminate by hole type?

Further information on **MOSEK**

- Documentation at `mosek.com/documentation/`
  - Modeling cook book / cheat sheet.
  - White papers.
  - Manuals for interfaces.
  - Notebook collection.

- Tutorials and more at `github.com/MOSEK/`



mosek