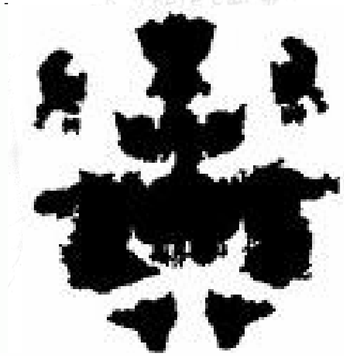
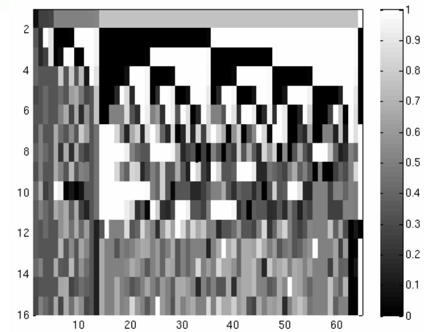
Pure Cutting Plane Methods for ILP: a computational perspective

Matteo Fischetti, DEI, University of Padova





Rorschach test for OR disorders: can you see the tree?

Outline

1. **Pure** cutting plane methods for ILPs: motivation
2. **Kickoff**: Gomory's method for ILPs (1958, fractional cuts)
3. **Bad** (expected) news: **very poor** if implemented naively
4. **Good** news: room for more **clever** implementations

Based on joint work with
Egon Balas and Arrigo Zanette

Motivation

- Modern branch-and-cut MIP methods are heavily based on **Gomory cuts** → reduce the number of branching nodes to optimality

- However, **pure** cutting plane methods based on Gomory cuts alone are typically **not used in practice**, due to their poor convergence properties

- Branching as a **symptomatic cure** to the well-known drawbacks of Gomory cuts — saturation, bad numerical behavior, etc.

- From the cutting plane point of view, however, the cure is even worse than the disease — it hides the **trouble source**!

The pure cutting plane dimension

- **Goal**: try to come up with a viable **pure** cutting plane method (i.e., one that is not knocked out by numerical difficulties)…
- … even if on most problems it will not be competitive with the branch-and-bound based methods
- This talk: **Gomory's fractional cuts (FGCs)**, for several reasons:
  - simple tableau derivation
  - reliable LP validity proof (runtime cut-validity certificate)
  - all integer coefficients → **numerically more stable** than their mixed-integer counterpart (GMIs)

Rules of the game: cuts from LP tableau

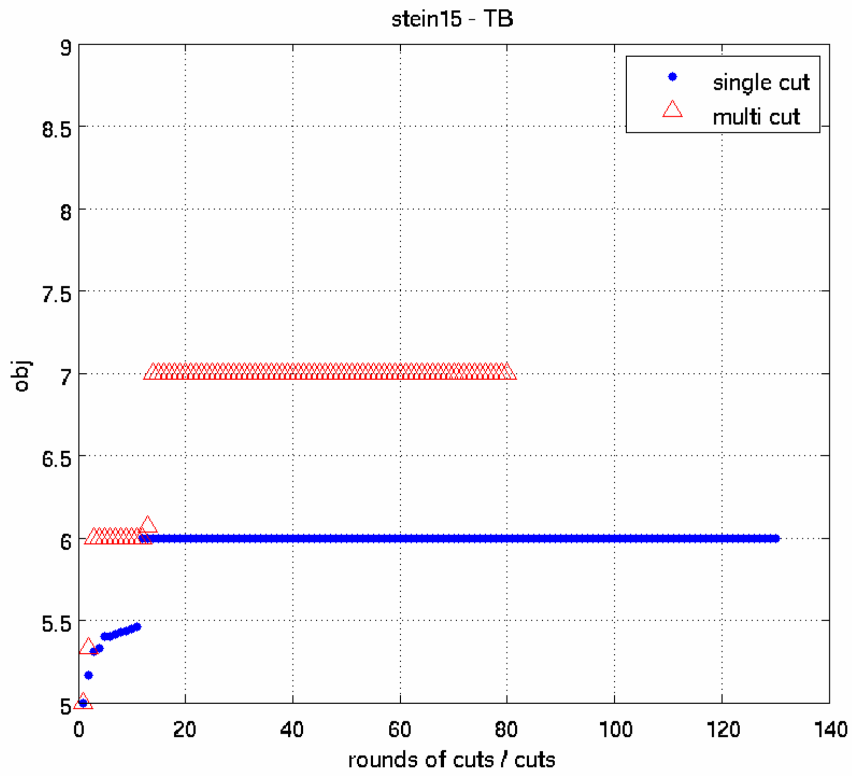
- **Main requirement:** reading (essentially for free) the FGCs directly from the optimal LP tableau

- Cut separation heavily **entangled** with LP reoptimization!

- Closed loop system (tableau-cut-tableau) without any control valve: **highly unstable**!

- Intrinsically different from the recent works on the first closure by F. & Lodi (Chvatal-Gomory closure) and Balas & Saxena and Dash, Gunluk & Lodi (GMI/split closure) where separation is an external black-box **decoupled** from LP reoptimization

Bad news: Stein15 (LP bound)



- Toy set covering instance from MIPLIB; LP bound = 5; ILP optimum = 8
- The **multi-cut** vers. generates **rounds of cuts** before each LP reopt.

The tip of the iceberg

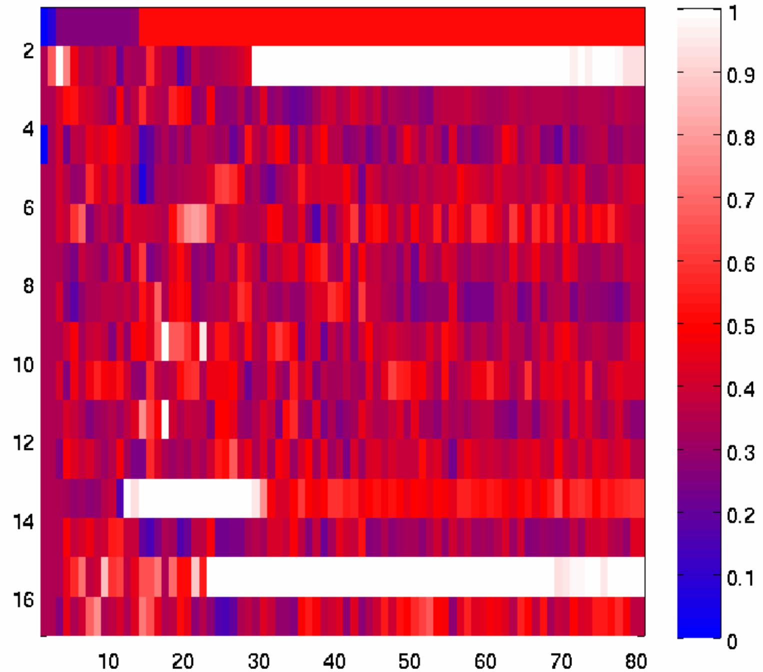
- Bound saturation is just the tip of the iceberg

- Let’s have a look under the sea…

… with our brand-new 3D glasses

Bad news: Stein15 (LP sol.s)

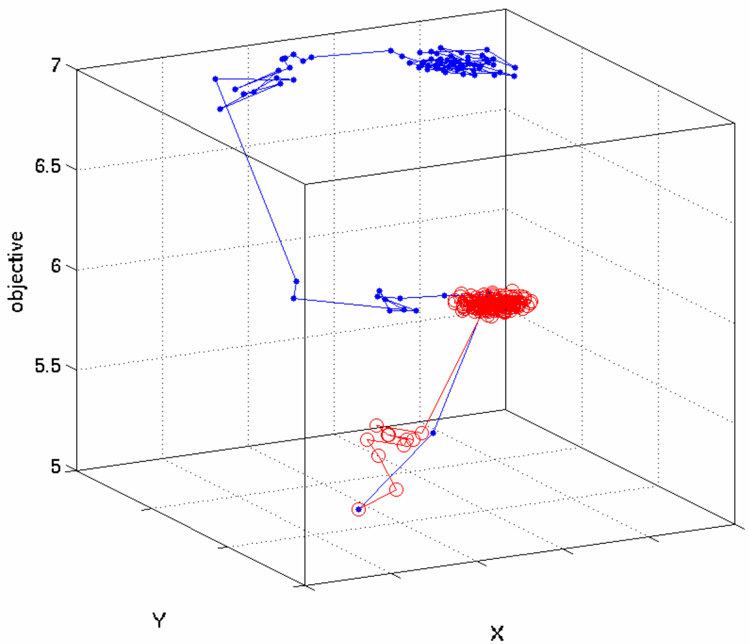
iter.	t=0	t=1	t=2	t
x_1	0.500	0.499	0.433	$x^*_1(t)$
x_2	0.333	0.333	0.111	$x^*_2(t)$
…	0.250	0.222	0.220	…
x_j	0.311	0.123	0.231	$x^*_j(t)$
…	0.171	0.196	0.201	…



Fractionality spectrography: color plot of the LP sol.s (muti-cut vers.)

- After few iterations, an almost-uniform red plot (very bad…)

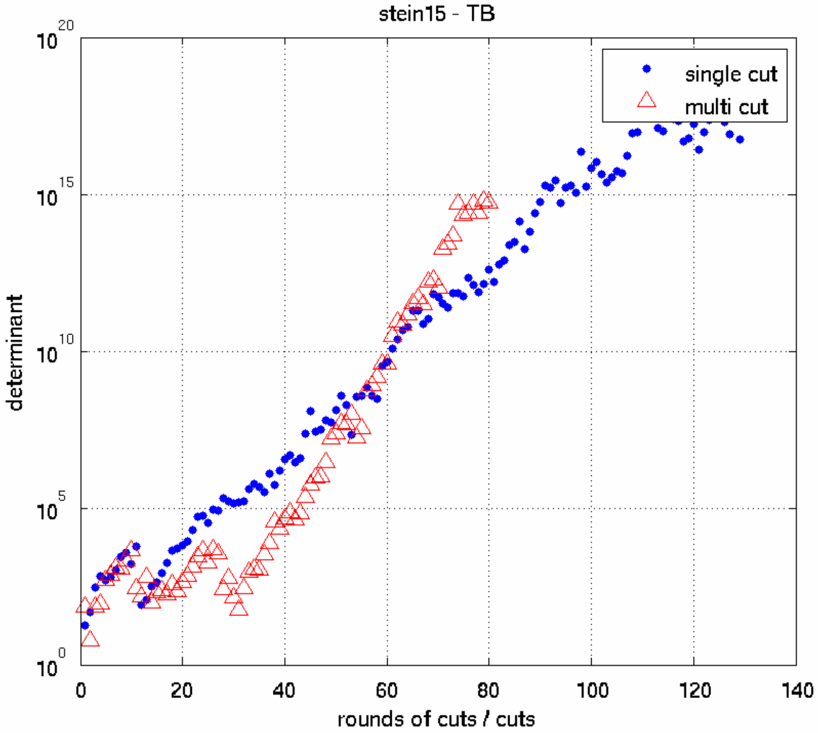
Bad news: Stein15 (LP sol.s)



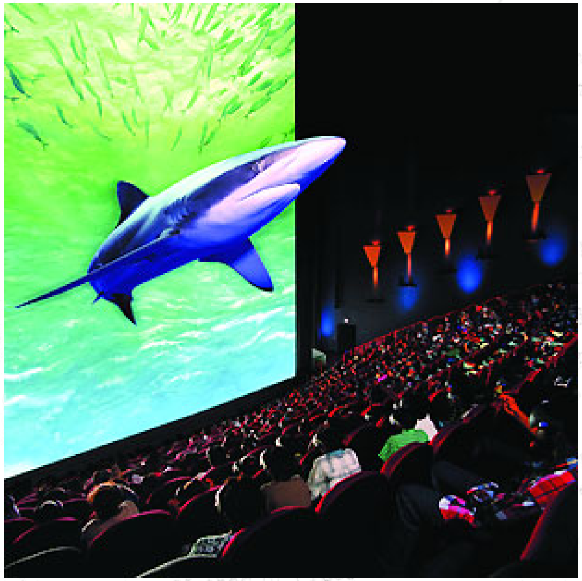
- Plot of the LP-sol. trajectories for **single-cut** (red) and **multi-cut** (blue) versions (multidimensional scaling)

- Both versions collapse after a while → no more fuel?

Bad news: Stein15 (determinants)



- **Too much fuel !!**



Cuts and Pivots

- Very long sequence of cuts that eventually lead to an optimal integer solution → cut **side effects** that are typically underestimated when just a few cuts are used within an enumeration scheme

- **A must!** Pivot strategies to keep the optimal tableau **clean** so as generate **clean** cuts in the next iterations

- In particular: avoid cutting LP optimal vertices with a **weird fractionality** (possibly due to numerical inaccuracy)

  → the corresponding LP basis has a large determinant (needed to describe the weird fractionality)

  → the tableau contains weird entries that lead to weaker and weaker Gomory cuts

Role of degeneracy

- Dual degeneracy is an intrinsic property of cutting plane methods
- It can play an important role and actually can **favor** the practical convergence of a cutting plane method…
- … provided that it is exploited to choose the cleanest LP solution (and tableau) among the equivalent optimal one

Unfortunately, by design, efficient LP codes work against us!

They are **so smart** in reducing the n. of dual pivots, and of course they stop immediately when primal feasibility is restored!

→ The new LP solution tends to be close to the previous one

→ Small changes in the LP solution imply large determinants

→ Large determinants imply unstable tableaux and shallow cuts

→ Shallow cuts induce smaller and smaller LP solution changes

→ Hopeless!

Dura lex, sed lex …

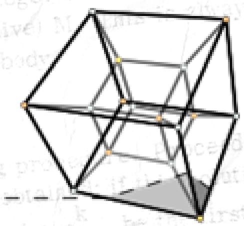
- In his proof of convergence, Gomory used the **lexicographic (dual) simplex** to cope with degeneracy → lex-minimize ($x_0 = c^T x, x_1, x_{2,} \dots, x_n$)

- Implementation: use a modern LP solver as a black box:
  - Step 0. Minimize x_0 --> optimal value x^*_0
  - Step 1. Fix $x_0 = x^*_0$, and minimize x_1 --> optimal value x^*_1
  - Step 2. Fix also $x_1 = x^*_1$, and minimize x_2 --> optimal value x^*_2
  - ...

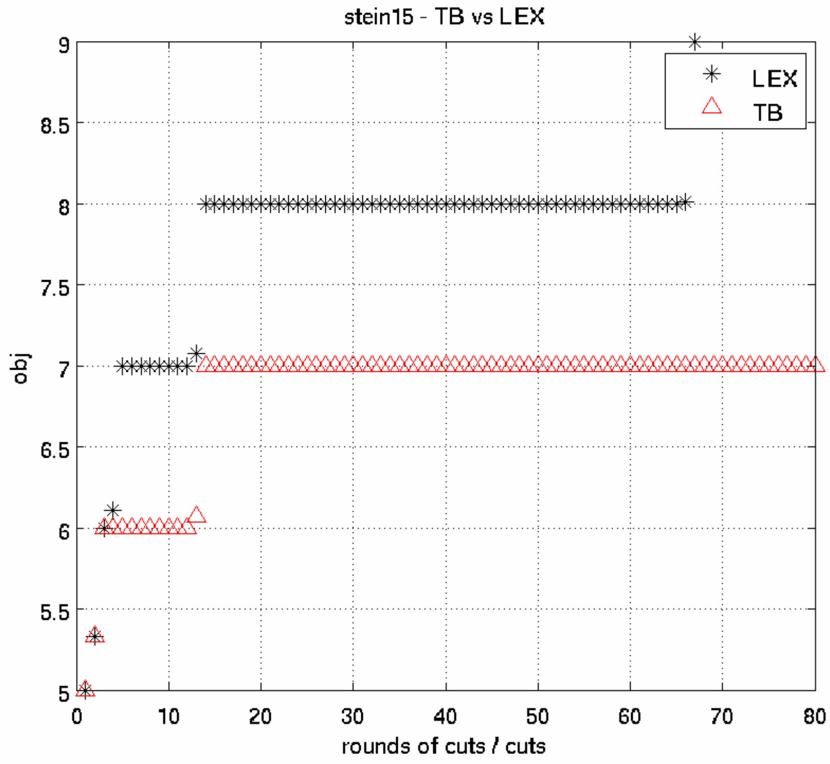
- **Key point:** at each step, instead of adding equation $x_j = x^*_j$ explicitly…
  … just fix out of the basis all the nonbasic var.s with nonzero reduced cost

- → Sequence of fast (and clean) reoptimizations on smaller and smaller degeneracy subspaces, leading to the required lex-optimal tableau

- Lex-min useful for the convergence proof, but … also in **practice**?



Good news #1: Stein15 (LP bound)

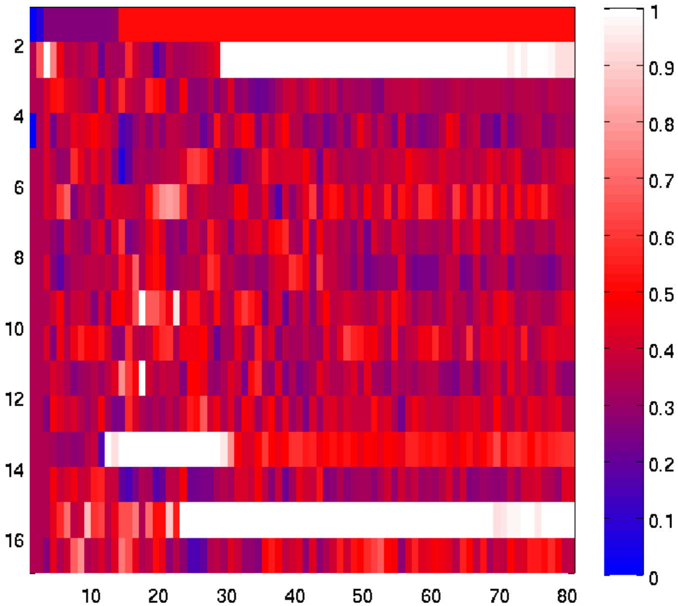


LP bound = **5**; ILP optimum = **8**

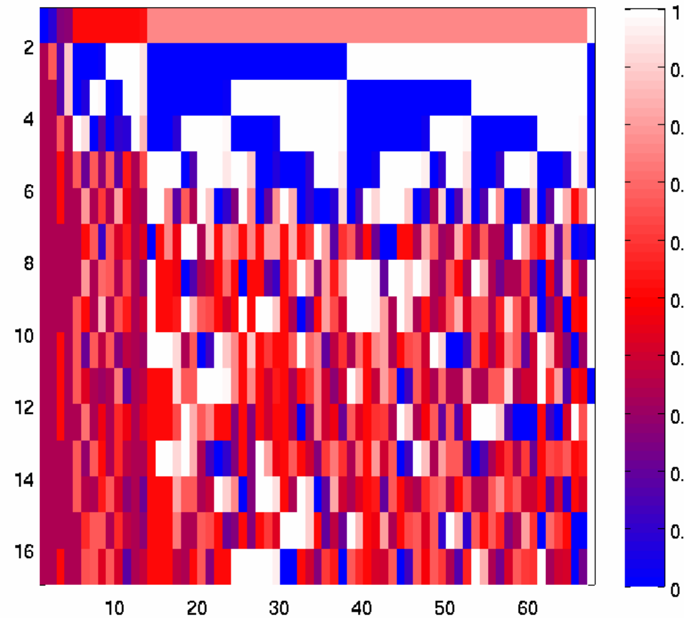
TB = "Text-Book" **multi-cut** vers. (as before)

LEX = single-cut with **lex-optimization**

Good news #1: Stein15 (LP sol.s)



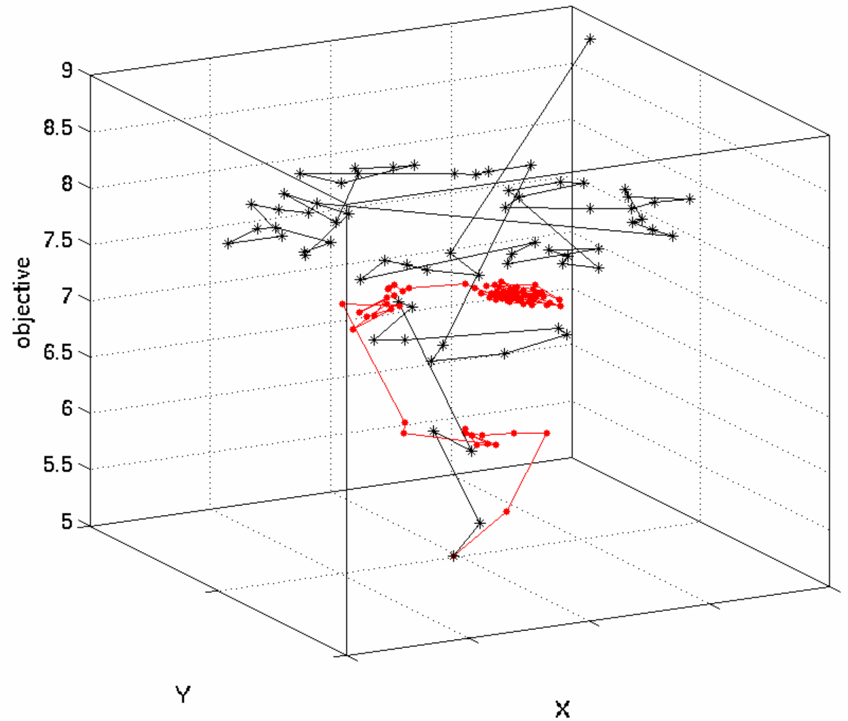
TB = multi-cut vers. (as before)



LEX = single-cut with lex-optimization

Fractionality spectrography

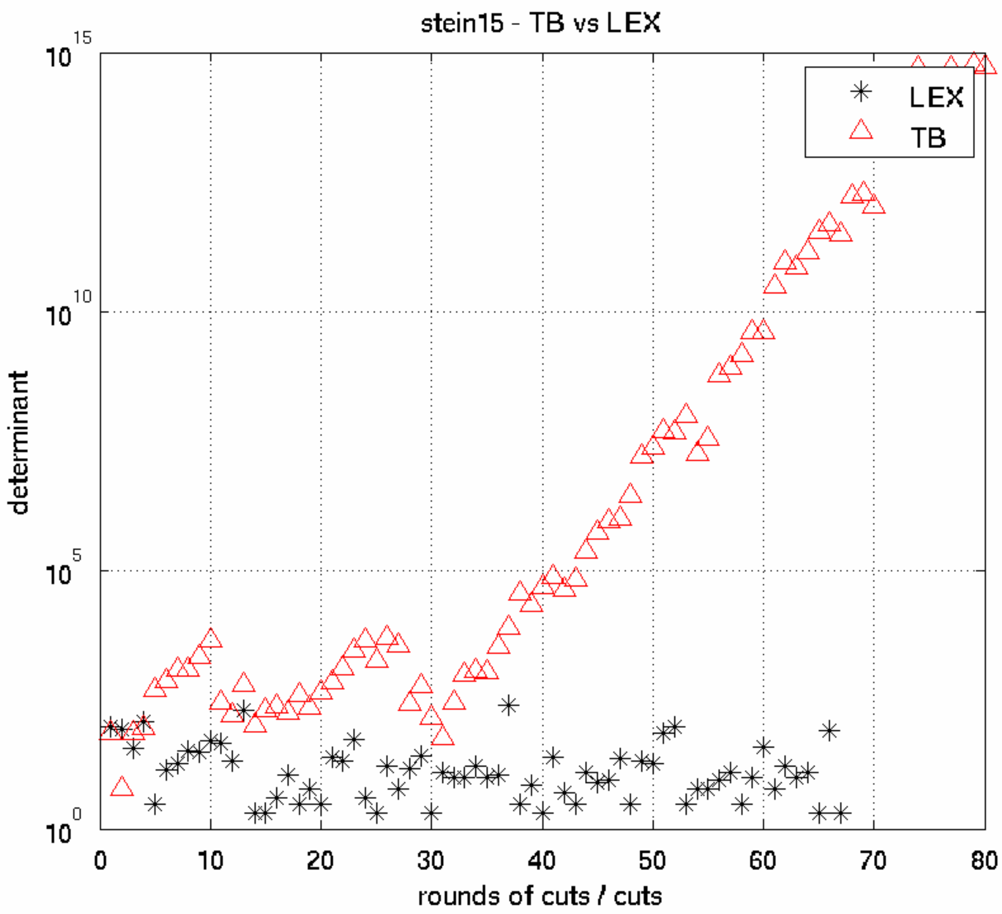
Good news #1: Stein15 (LP sol.s)



Plot of the LP-sol. trajectories for **TB** (**red**) and **LEX** (**black**) versions

(X,Y) = 2D representation of the x-space (multidimensional scaling)

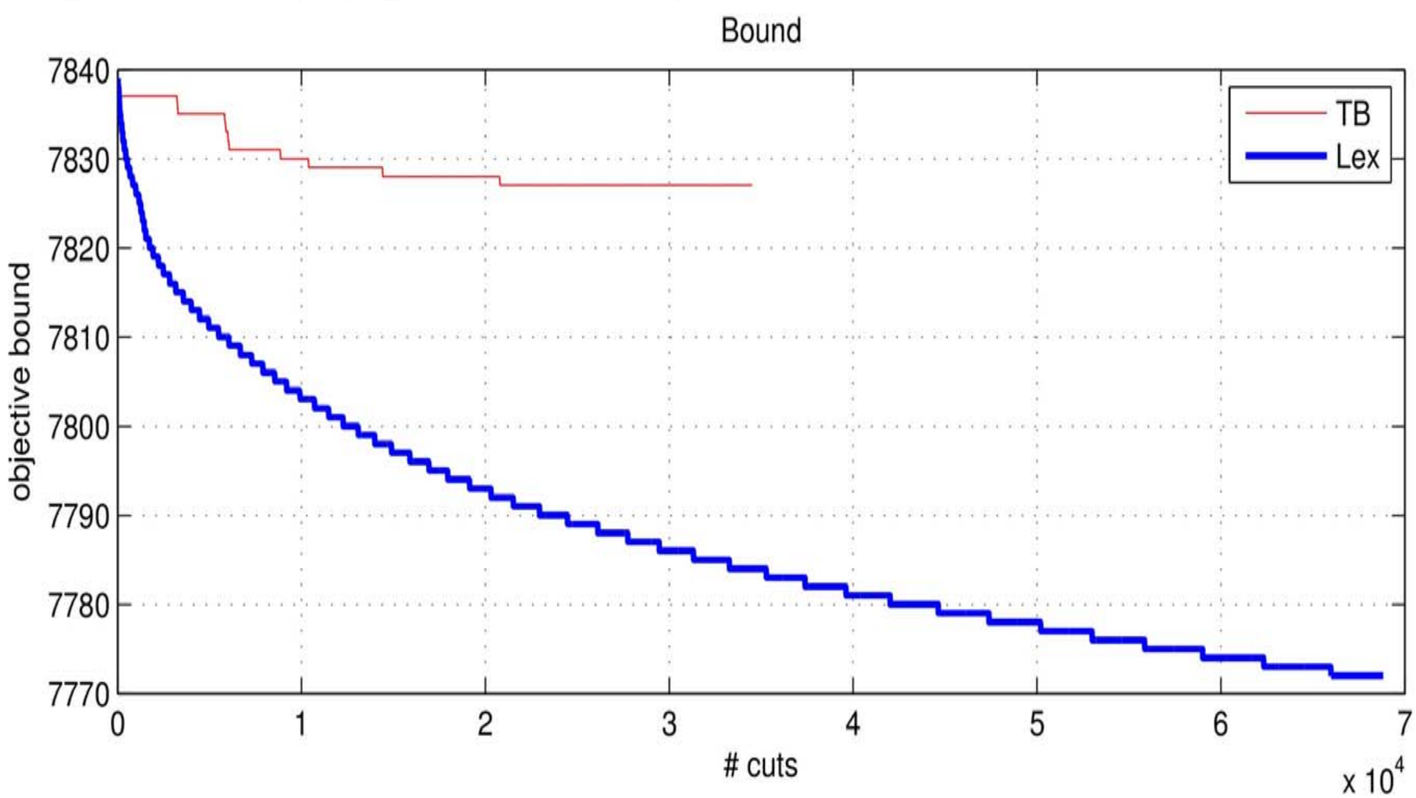
Good news #1: Stein15 (determinants)



TB = multi-cut vers. (as before)

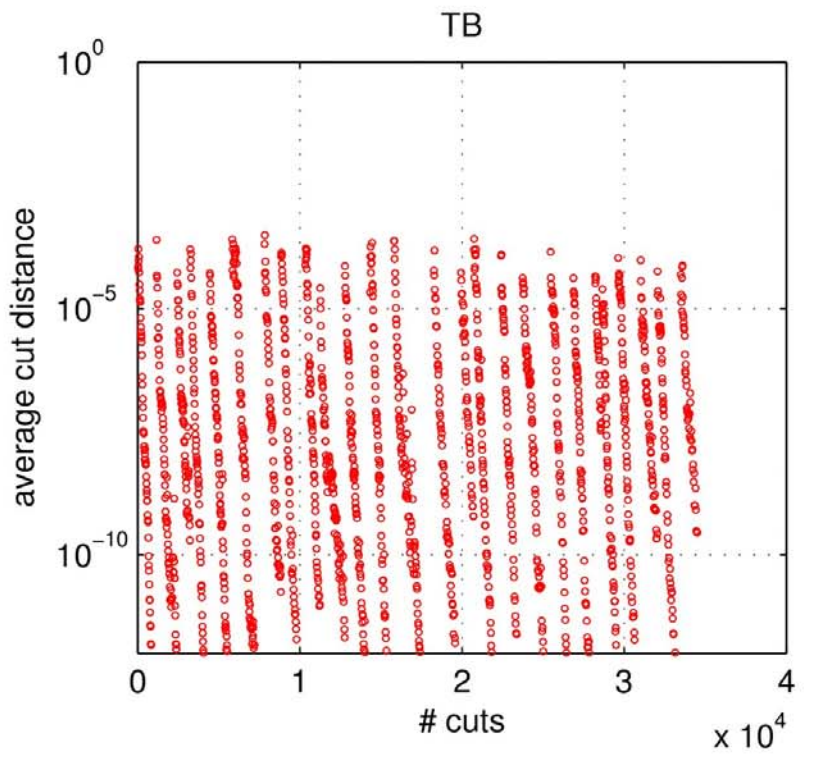
LEX = single-cut with lex-opt.

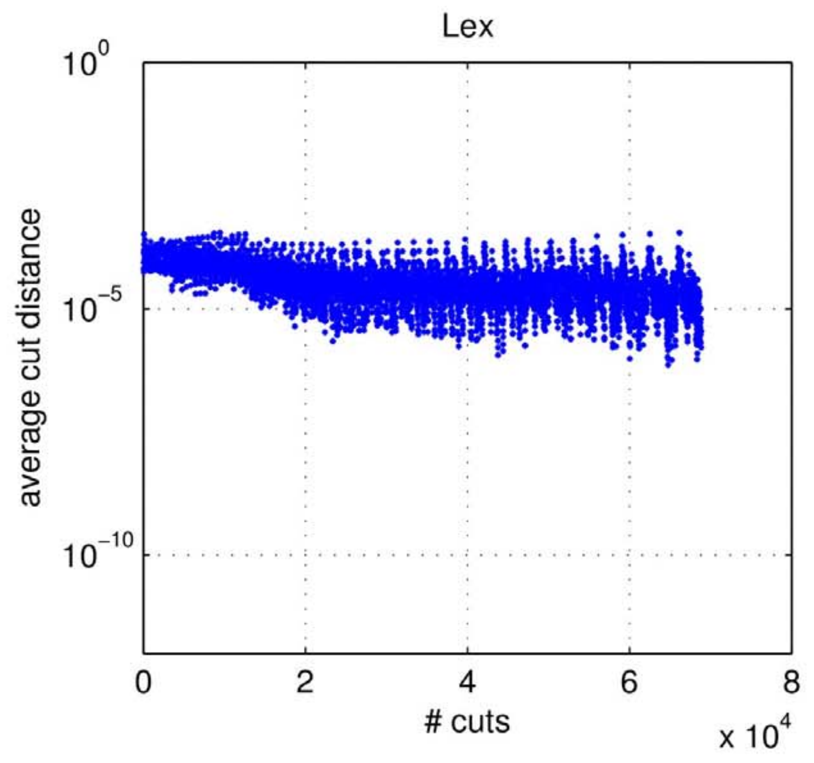
Good news #1: sentoy (max. problem)



TB = multi-cut vers. (as before) LEX = single-cut with lex-opt.

Good news #1: sentoy



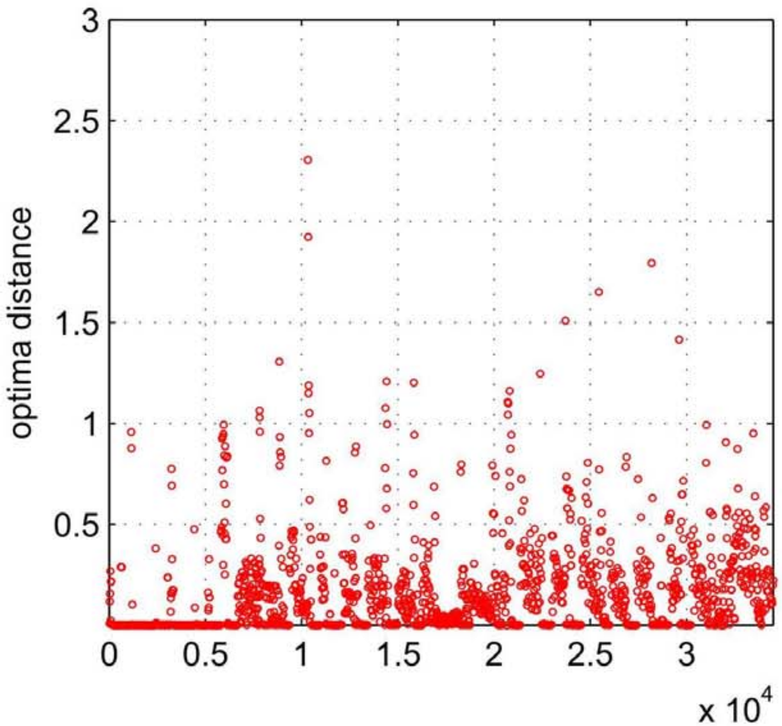


TB = multi-cut vers. (as before)

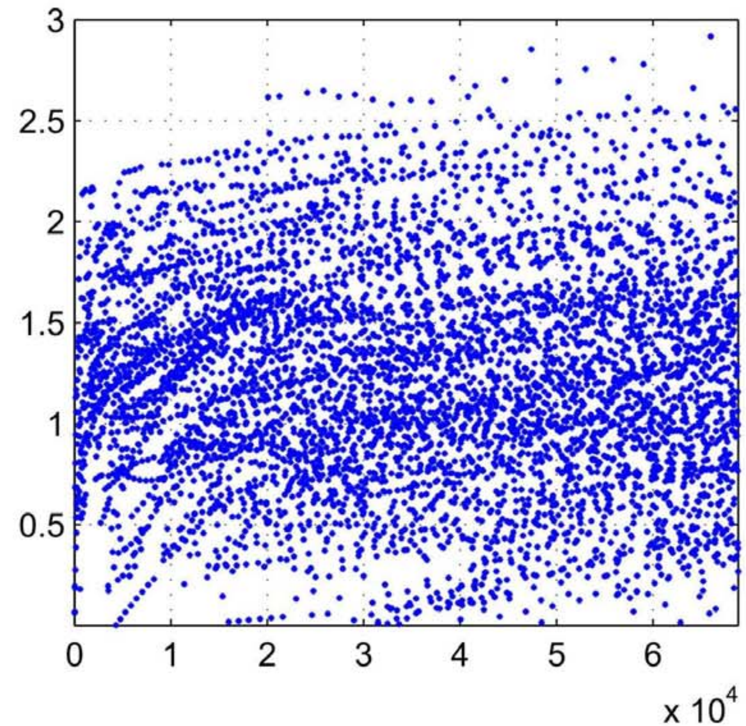
LEX = single-cut with lex-opt.

Avg. geometric distance of x* from the Gomory cut

Good news #1: sentoy



TB = multi-cut vers. (as before)

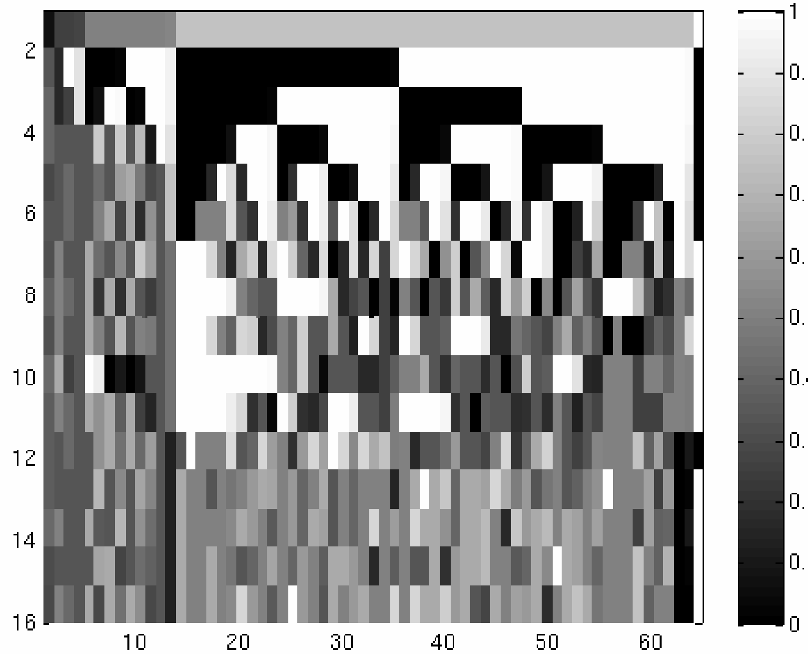


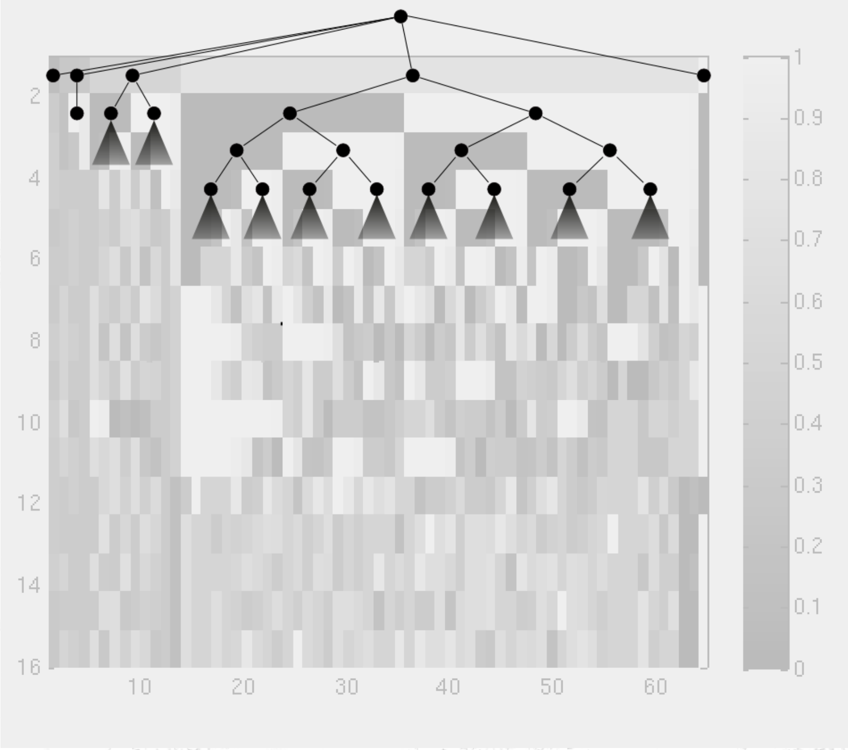
LEX = single-cut with lex-opt.

Avg. geometric distance between two consecutive optimal sol.s x*

Ok, it works … but WHY?

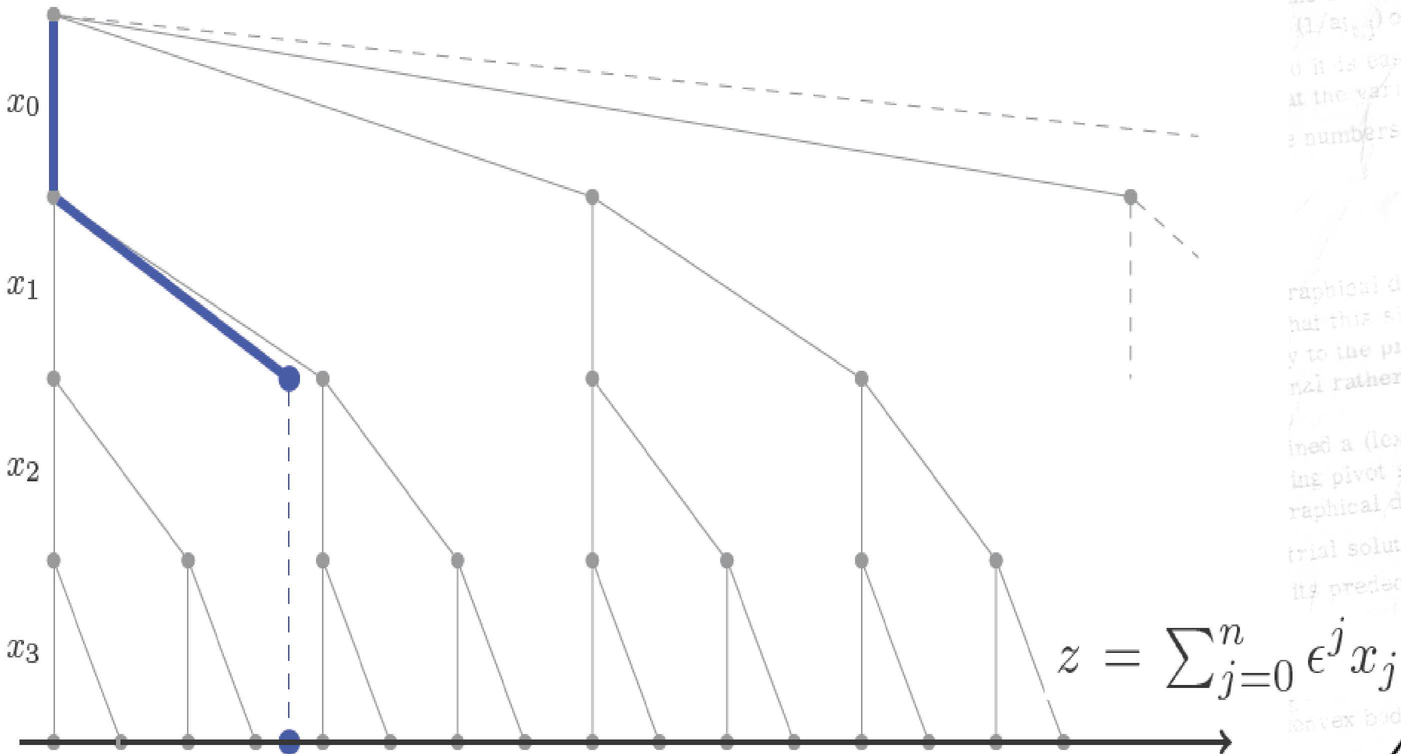
Enumerative interpretation of the Gomory method (Nourie & Venta, 1982)



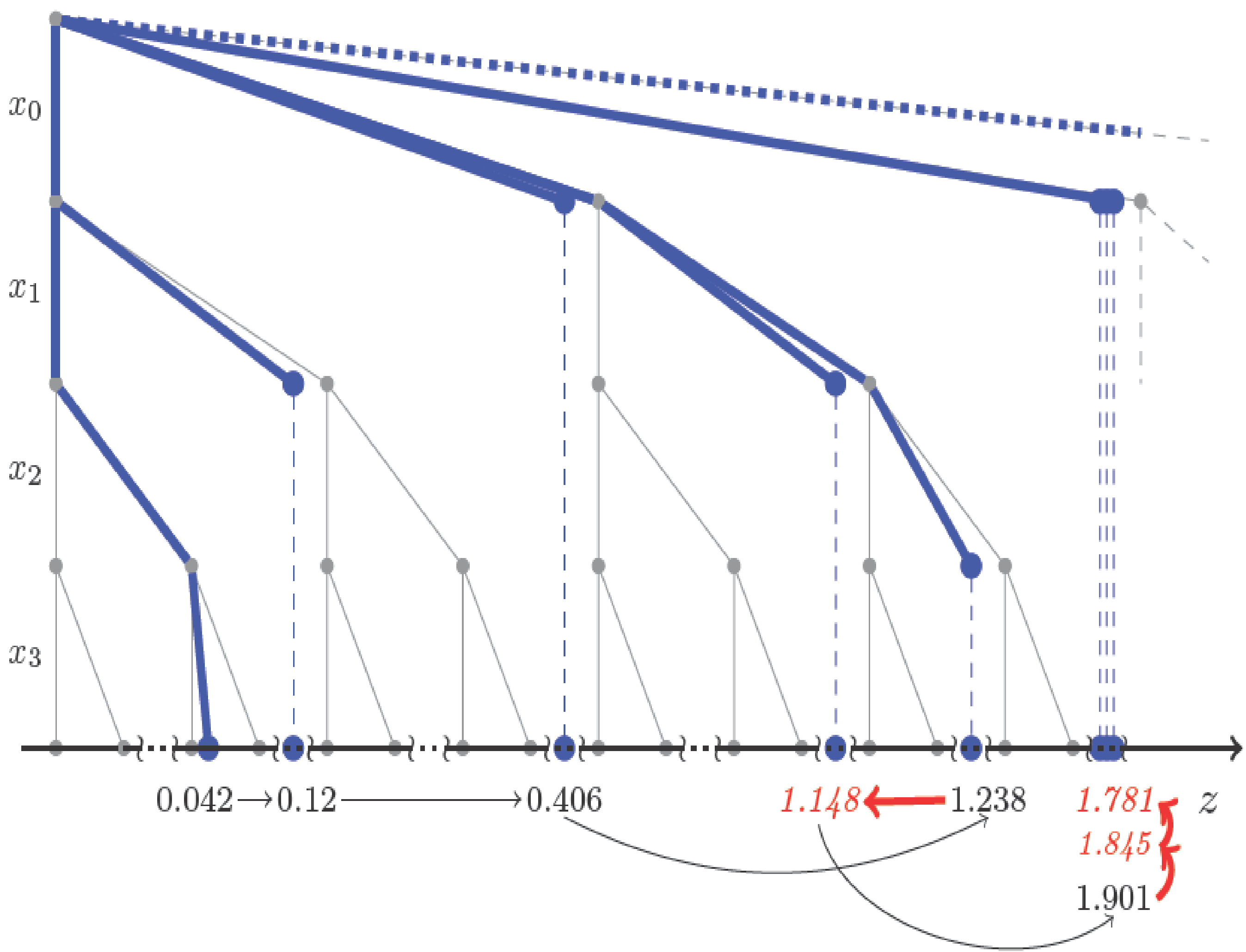


The underlying enumeration tree

- Any fractional solution x^* can be visualized on a lex-tree
- The structure of the tree is fixed (for a given lex-order of the var.s)
- Leaves correspond to integer sol.s of increasing lex-value (left to right)



The "bad" Gomory (TB = no lex)



lex-value *z* may decrease → risk of loop in case of naïve cut purging!

Nice "sign pattern" of lex-optimal tableau

					x_h						x_j			x_{10}			x_{25}		x_0
x^*_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	1
x^*_1	1	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-			0
x^*_5	0	1	0	0	0	0	0	0	0	0	0	0	-						0
x^*_k	0	0	1	0	0	0	0	0	0	0	-	-							0
x^*_8	0	0	0	1	0	0	0	0	0	-									0
x^*_h	0	0	0	0	1	0	0	0	-		+		+	+			+		0
x^*_{22}	0	0	0	0	0	1	-	-											0

increasing lex. order ↓

RHS ← basic var.s → ← nonbasic var.s →

Green row: nonbasic "+" var. x_j increases → a basic var x_k with $k < h$ increases

The key FGC property for convergence

Take the tableau row associated with the (lex) **first** fractional var. x^*_h

$$x_h + \sum_{j \in J^-} \overline{a}_{ij} x_j + \sum_{j \in J^+} \overline{a}_{ij} x_j = \overline{a}_{i0} \; (= x^*_h)$$

where $J^- = \{ j : \overline{a}_{ij} < 0 \}$ and $J^+ = \{ j : \overline{a}_{ij} > 0 \}$

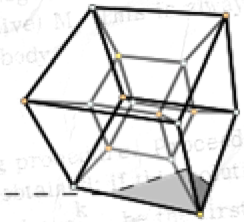
We want to lex-**increase** the optimal value → add a FGC in its **≥** form:

$$x_h + \sum_{j \in J^-} \lceil \overline{a}_{ij} \rceil x_j + \sum_{j \in J^+} \lceil \overline{a}_{ij} \rceil x_j \geq \lceil x^*_h \rceil$$

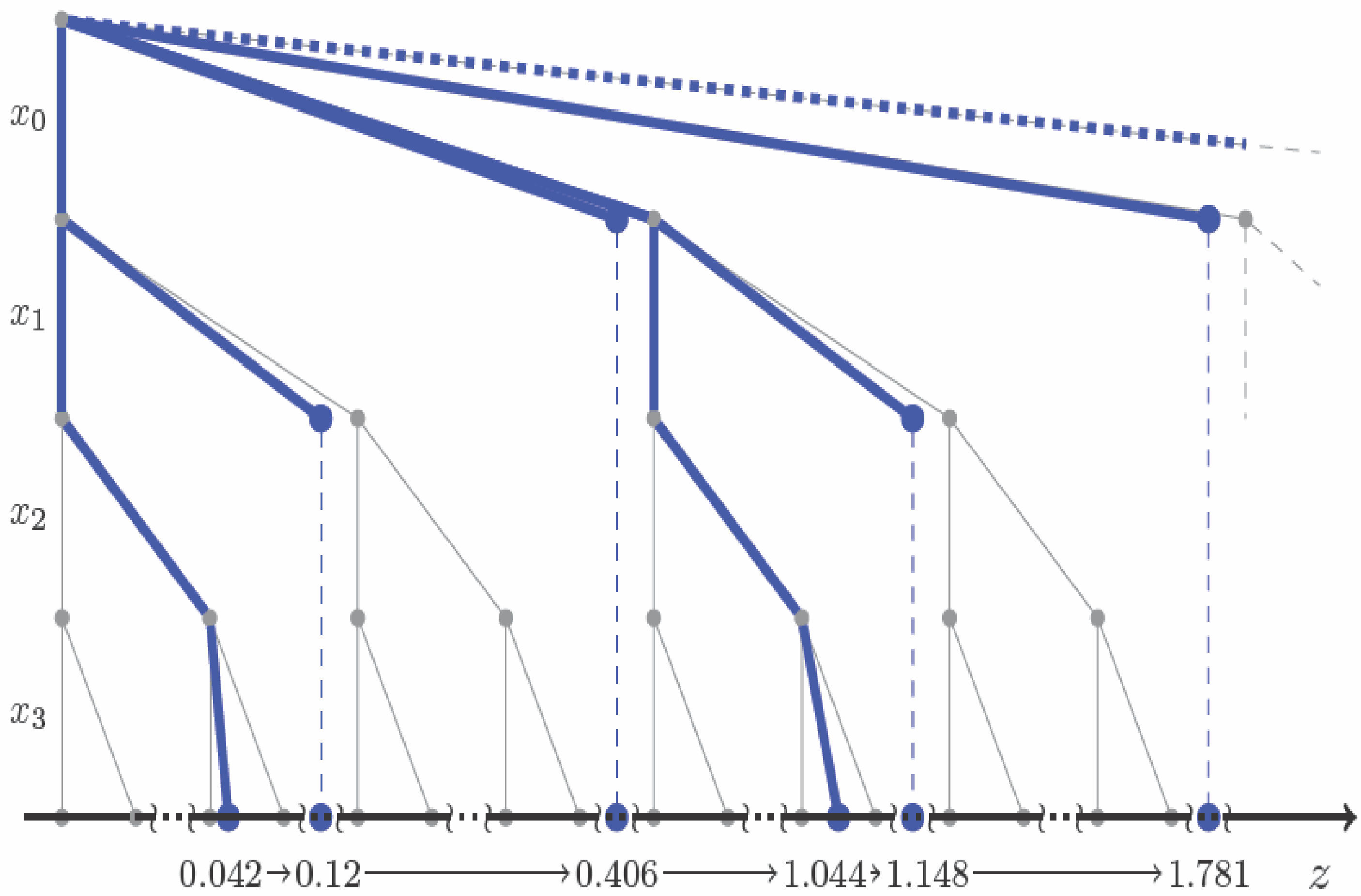
(a FGC in its ≤ form will not work!). Two cases for the new LP-opt. x

[BRANCH] $x_j = 0$ for all $j \in J^+$ → $x_h \geq \lceil \overline{a}_{i0} \rceil - \sum_{j \in J^-} \lceil \overline{a}_{ij} \rceil x_j \geq \lceil x^*_h \rceil$

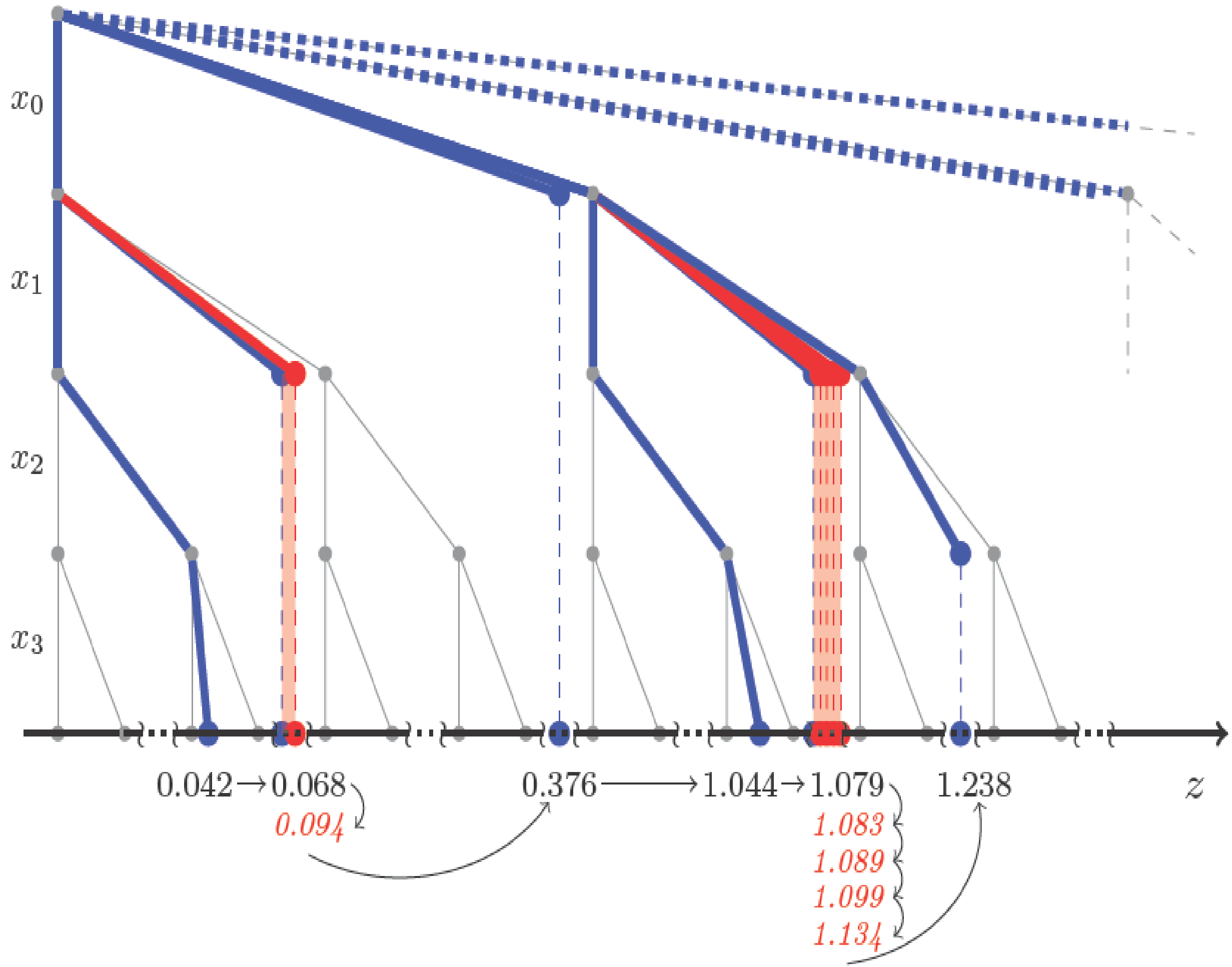
[BACKTRACK] otherwise, a "previous component" increases → BIG lex-increase



The “good” Gomory (lex & ≥)



A “still bad” Gomory (lex but ≤)



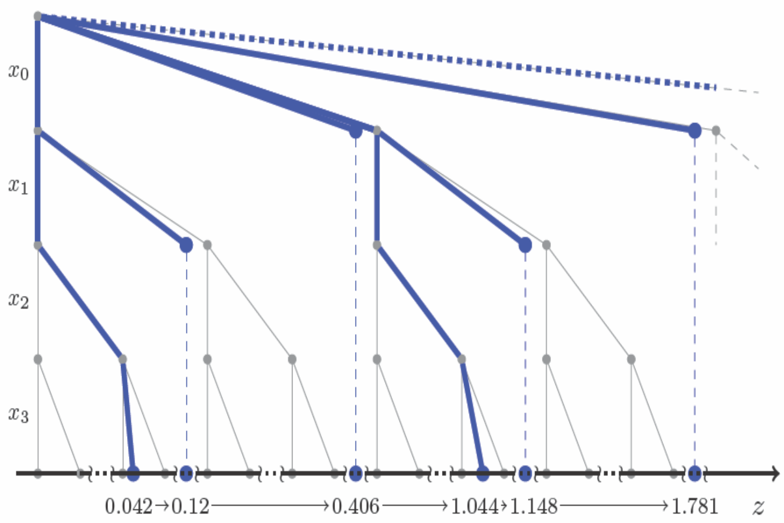
… slow sequence, but still **monotonically** lex-increasing (not enough for finite convergence)

Lessons learned

The Gomory method is **framed** within its enumerative cast

“Good” FGCs may allow for large backtracking steps, but they cannot modify the underlying tree

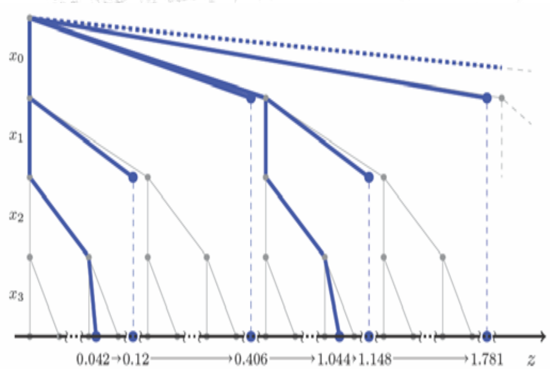
Inefficient depth-first branching on an unnatural variable order → branching even on **integer-valued variables!!**



Good news #2: lex on the fly

Facts:

- If x^*_h is the **first** fractional var. of the current lex-optimal LP sol., there is no harm in changing the lex sequence from position *h*
- Our lex-reoptimization method allows one to do this "**natively**", in an effective way
- The first fractional var. x^*_h plays the role of the **branching** var. in enumerative method
- One can borrow from enumerative methods any clever selection policy for the branching variable x^*_b (b for branching), and move this var. in the h-th position of the current lex-order → (hopefully) **no more branchings on integer variables!**



Variants: get rid of the obj. function

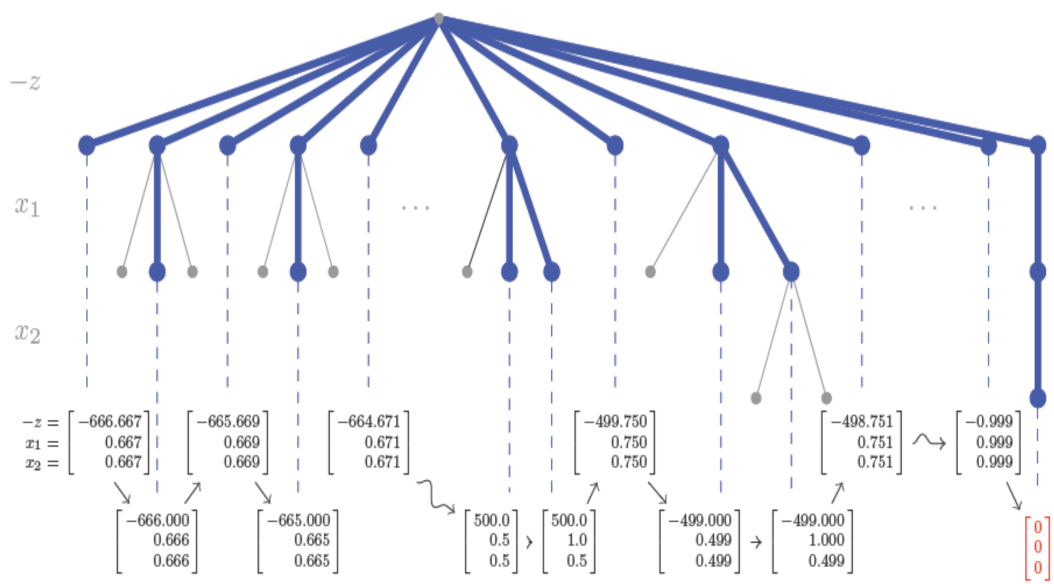
The first branching variable x_0 is the objective function → a very unnatural choice for an enumerative method!

In some cases, this choice forces Gomory's method to visit a same subtree several times (see e.g. the Cook-Kannan-Schrijver example below)

→ Try to get rid of the obj. function: use of invalid cuts (L-CP), binary search, etc. **BUT: are these still <u>pure</u> cutting plane methods ??**

$$\max\{y \ : \ x_1 + x_2 + y \le 2,\ y \le x_1,\ y \le x_2,\ x_1, x_2 \in \{0,1\},\ y \ge 0\}$$

Let z := 1000 y
z integer



Role of cuts & dynamic lex-order

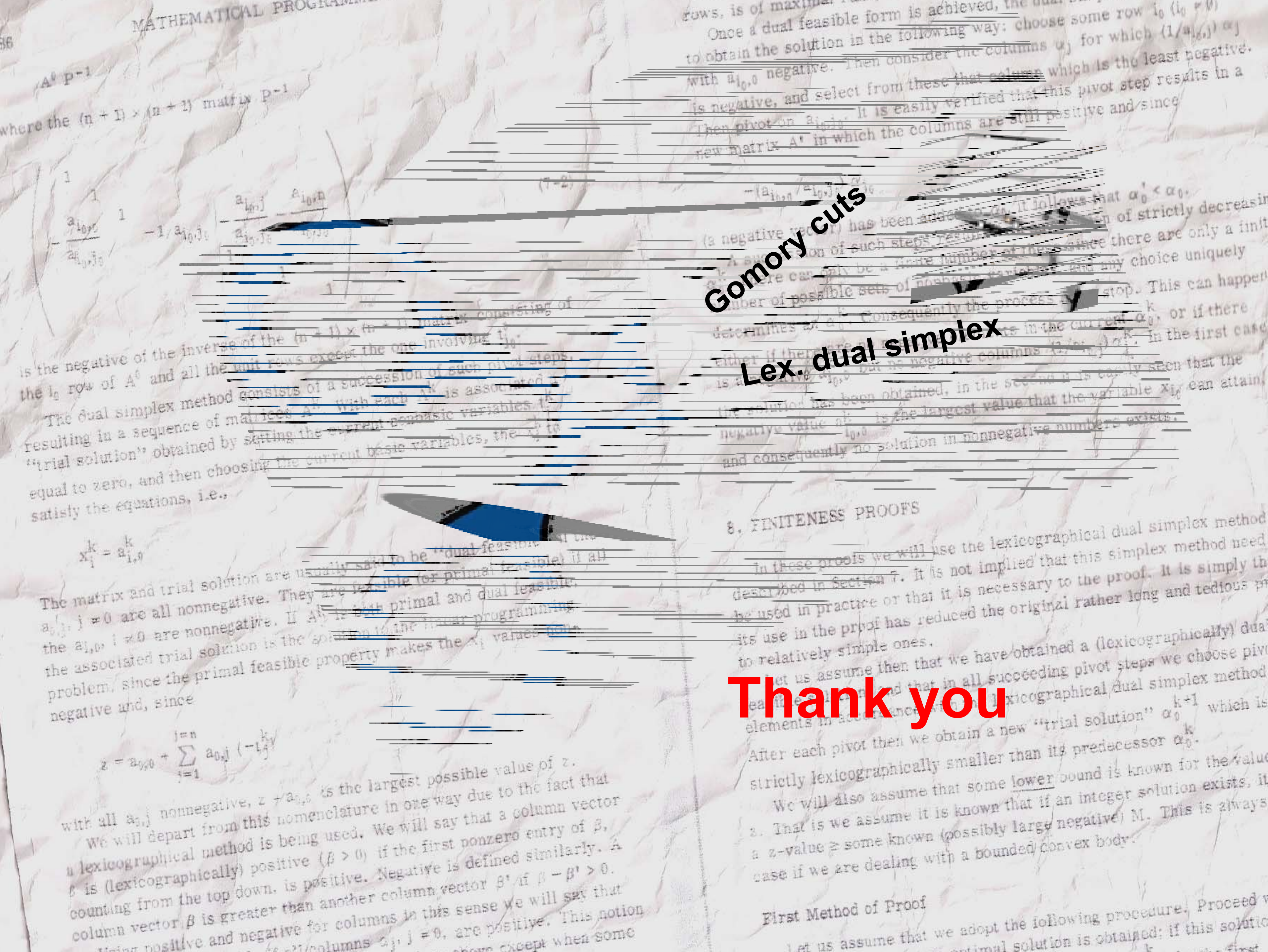
	L-CP		L-BB		L-CP.dyn		L-BB.dyn	
Instance	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes
bm23	2.65	2205	0.29	3372	0.96	**1013**	0.13	1358
hard_ks100	16.67	41303	7200	122828201	15.18	**24077**	4084.83	56872670
hard_ks9	0.07	164	0.01	274	0.05	**133**	0.01	228
l152lav	705.22	1111733	1876.69	3335512	7201.47	**634042**	1406.11	1354164
lseu	199.6	191256	1153.73	15089374	46.59	**23327**	12.38	109578
manna81	2991.28	142013	7200	3321404	7202.88	**8276**	7200	2692153
p0033	0.16	380	0.62	11538	0.17	**307**	0.83	12192
p0201	117.79	58101	59.07	357664	38.56	**11127**	6.51	26942
pipex	9.01	6474	1.31	13904	2.83	**2198**	0.72	7686
sentoy	114.56	103349	20.04	197510	2.39	**1460**	0.16	2038
stein15	0.1	123	0.03	418	0.11	**97**	0.02	260
stein27	6.58	4160	1.62	13260	6.47	**3551**	1.22	9210

L-CP and **L-B&B** work on the same underlying tree (L-CP exploiting FGCs)

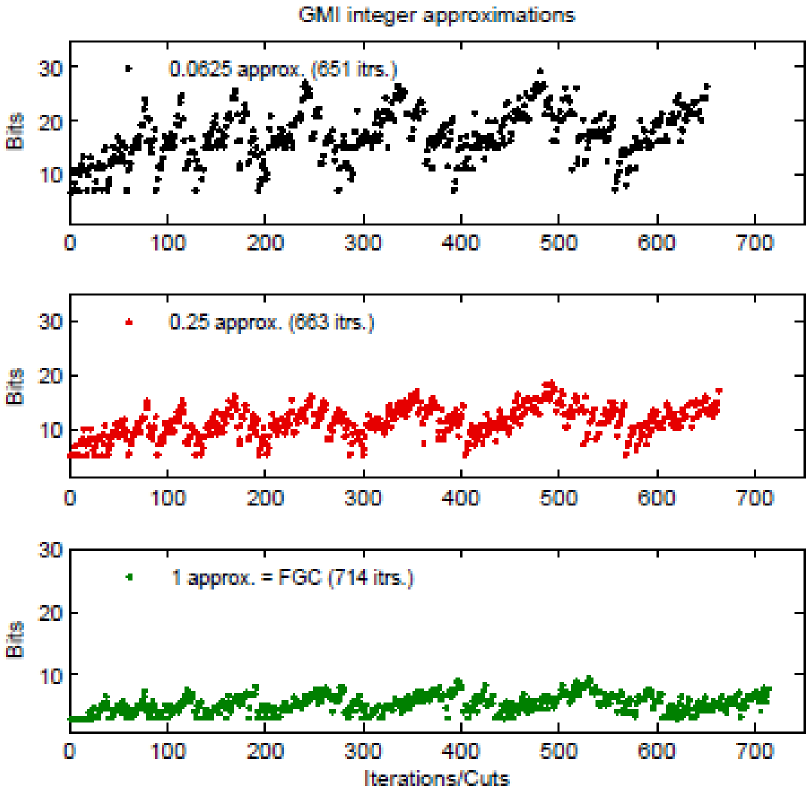
***.dyn** versions modify the lex-order on the fly (no branching on integer var.s)

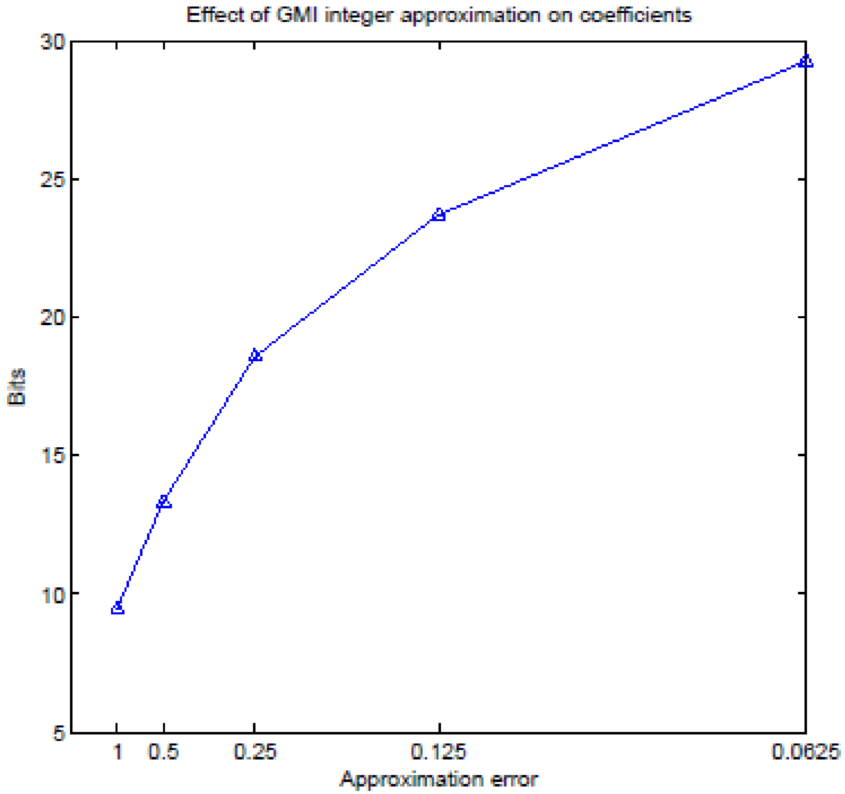
Computational tests

		Lex-FGC					Lex-FGC.dyn			
Instance		#Itr.s	Cuts	Time	Cl.Gap		#Itr.s	Cuts	Time	Cl.Gap
air4	T	434	140191	7207.51	25.19	T	559	178404	7218.26	**36.17**
air5	T	833	240266	7209.73	25.26	T	1119	321186	7209.53	**30.7**
bm23	O	660	8294	2	100	O	652	8047	0.99	**100**
cap6000	T	16043	107239	7201.43	32.64	T	76649	493654	7201.18	**83.16**
hard_ks100	O	99	483	0.43	**100**	O	809	4100	2.73	100
hard_ks9	O	141	609	0.22	100	O	130	542	0.08	**100**
krob200	O	41	1643	93.94	100	O	23	375	18.64	**100**
l152lav	O	744	25109	118.2	**100**	T	42184	1911790	7201.22	86.29
lin318	O	28	1001	200.08	100	O	27	848	67.79	**100**
lseu	O	9591	133589	53.24	100	O	4261	53522	11.75	**100**
manna81	O	12	280	19.05	100	O	12	280	12.63	**100**
mitre	T	565	116369	7209.36	90.83	O	786	132399	6125.11	**100**
mzzv11	T	16	14516	8533.94	30.86	T	28	23496	7409.5	**37.18**
mzzv42z	T	19	15264	7519.35	17.45	T	31	24081	7550.08	**23.14**
p0033	O	501	4421	1.2	**100**	O	1214	10622	2.12	100
p0201	T	190383	5471004	7201.06	87.3	T	259143	8984687	7201.05	**96.49**
p0548	T	129823	4832247	7201.06	**76.86**	C	12412	450566	110781.7	53.21
p2756	T	51454	673642	7201.06	**79.78**	E	134	7550	7.35	79.78
pipex	O	473607	4583701	1591.97	**100**	C	1081606	¿10000000	2219.38	97.51
protfold	T	144	57617	7253.09	**45.26**	T	253	94771	7203.1	36.13
sentoy	O	5338	68991	24.36	100	O	3170	40765	6.52	**100**
seymour	T	117	67931	7224.93	26.89	T	149	84533	7217.92	**32.11**
stein15	O	68	708	0.15	100	O	59	641	0.1	**100**
stein27	O	3134	35861	13.96	100	O	2250	27462	5.63	**100**
timtab	T	5193	1675111	7344.59	**50.44**	T	4090	1320383	7550.76	46.37


Gomory cuts
Lex. dual simplex
Thank you

Question: what about GMI cuts?





Bits required to represent the integer cut coeff.s when approximating GMI cuts (approx. error =1 for FGCs, approx. error = 0 for GMIs)

→GMI cuts appear numerically much more difficult to handle (at least, in a pure cutting plane context …)