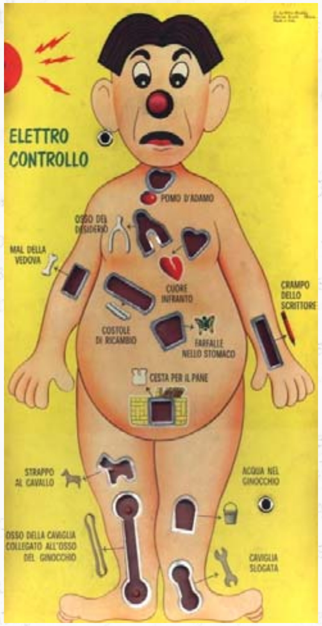
Looking inside Gomory

Matteo Fischetti, DEI University of Padova

(joint work with Egon Balas and Arrigo Zanette)



Gomory cuts

- Modern branch-and-cut MIP methods heavily based on **Gomory cuts**, used to reduce the number of branching nodes needed to reach 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.

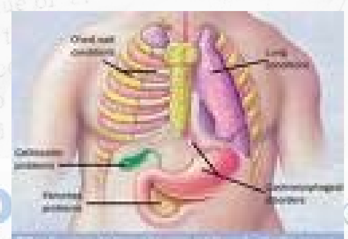
- From the cutting plane point of view, however, the cure is even worse than the disease—it hides the **trouble source**!

The pure cutting plane dimension

- The purpose of our project is to 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

- First step: **Gomory's fractional cuts (FGCs)**, for two reasons:
  - simplest to generate, and
  - when expressed in the structural variables, all their coefficients are **integer** → easier to work with them and to assess how nice or weird they are (**numerically more stable** than GMI cuts)

- **This talk**: looking inside the chest of FGC convergent method



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!

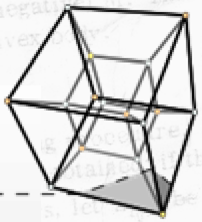
- Intrinsically different from the recent works on the first closure by Fischetti and Lodi (Chvatal-Gomory) and Balas and Saxena (GMI/split closure) where separation is **decoupled** from optimization

- **Subtle side effects!** The FGC

  $$2\,x_1 \quad - x_2 + \quad 3\,x_3 <= 5$$

  can work much better than its GMI (dominating but numerically less stable) counterpart

  $$2.272727272\mathbf{6}\,x_1 \; - \; x_2 + 3.181818181\mathbf{5}\,x_3 <= 5$$



12th Combinatorial Optimization Workshop
Centre Paul Langevin, Aussois, France, 7-11. January 2008
Organized by M. Jünger, F. Liebling, D. Naddef, W. Pulleyblank, G. Reinelt, G. Rinaldi and L. Wolsey

Cuts and Pivots: an entangled pair

- 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 aimed 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

- Keep a clean tableau → deal with numerical issues …
- Avoid cutting weird optimal vertices → **??**

Exploit dual degeneracy!

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

Unfortunately, the highly-correlated sequence of reoptimization pivots performed by a generic LP solver leads invariably to an **uncontrolled growth** of the basis determinant → method out of control after just a few iterations!

Dura lex, sed lex …

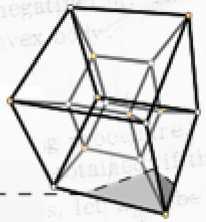
- In his proof of convergence, Gomory used the **lexicographic simplex** to cope with degeneracy

- The dual lexicographic simplex is a **modified version** of the dual simplex algorithm:
  – instead of considering the minimization of the objective function $x_0 = c^T x$
  – one is interested in the **lexicographic minimization** of the entire solution vector $(x_0, x_1, \ldots, x_n)$

- Rigid pivoting rules with ratio tests involving vectors → cumbersome and **slow/unstable implementation**

- Useful in theory for convergence proofs, but …

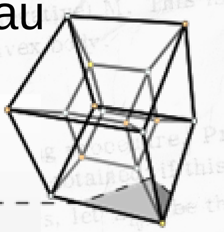
  … also **in practice**?



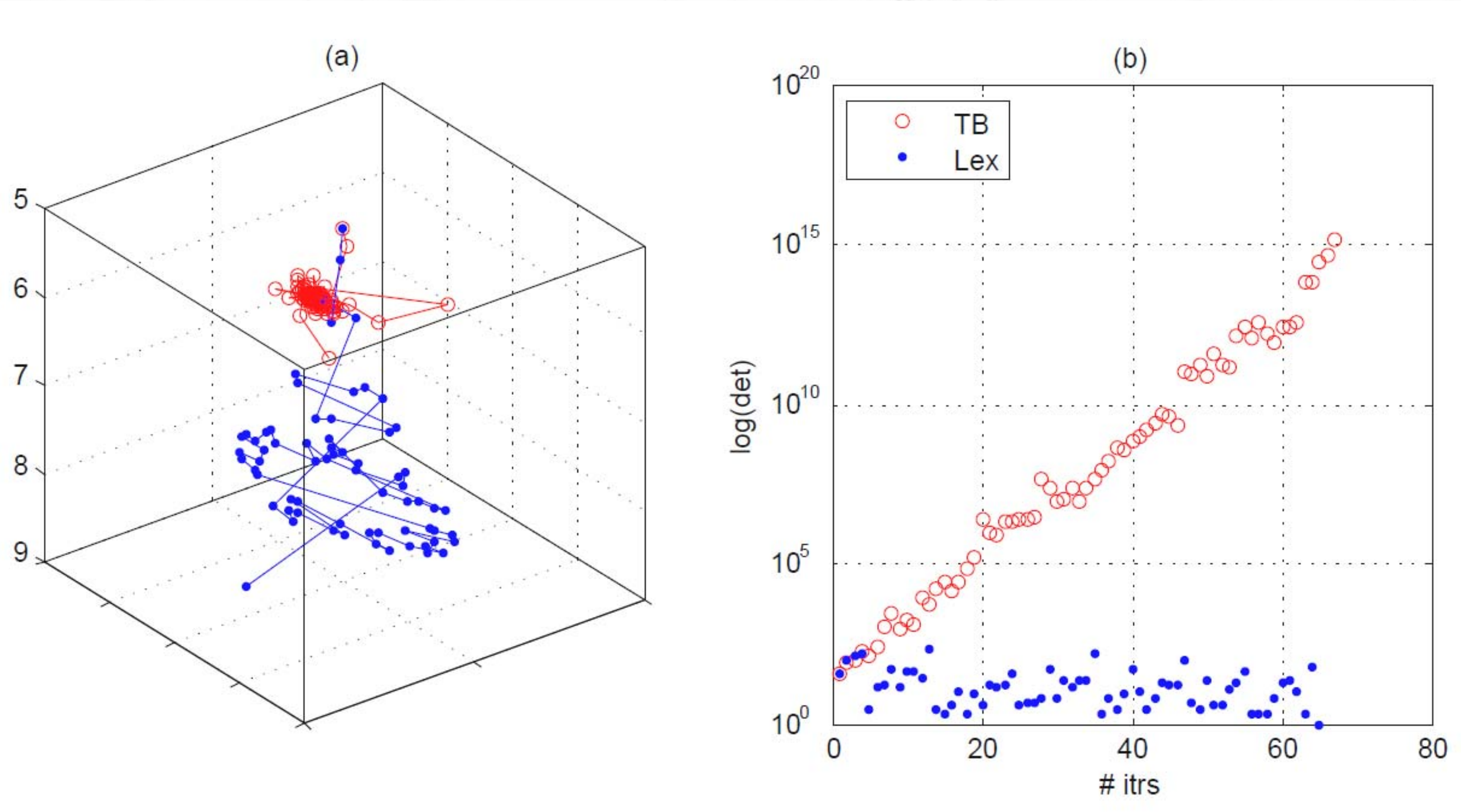
Lex-FGC: fractional vertices (stein15)

Cut#	x0	x1	x2	x3	x4	x5	x6	x7	…
00)	5.000	0.333	0.333	0.333	0.333	0.333	0.333	0.333	…
...									
05)	7.000	0.000	0.000	0.333	0.333	0.333	0.666	0.666	…
06)	7.000	0.000	0.071	0.785	0.428	0.500	0.428	0.214	…
07)	7.000	0.000	1.000	0.333	0.333	0.666	0.333	0.666	…
08)	7.000	0.030	0.969	0.696	0.606	0.121	0.303	0.272	…
09)	7.000	1.000	0.000	0.333	0.666	0.666	0.333	0.666	…
10)	7.000	1.000	0.025	0.743	0.230	0.435	0.538	0.333	…
11)	7.000	1.000	1.000	0.095	0.285	0.571	0.619	0.238	…
12)	7.000	1.000	1.000	1.000	0.333	0.333	0.333	0.333	…
13)	7.068	0.931	0.931	0.862	0.310	0.413	0.379	0.275	…
14)	8.000	0.000	0.000	0.000	0.000	0.000	1.000	…	…
...									
64)	8.000	1.000	1.000	1.000	1.000	1.000	1.000	0.500	…
65)	8.013	0.973	0.973	0.947	0.907	0.572	0.868	0.171	…
66)	9.000	0.000	0.000	0.000	0.000	0.000	1.000	1.000	… all integer!

- Each FGC removes **"enough fractionality"** of current x^*
- A consequence of the nice **"sign pattern"** of lex-optimal tableau



Textbook vs Lex (stein15)



(left) x* trajectories (vertical axis: lower bound)

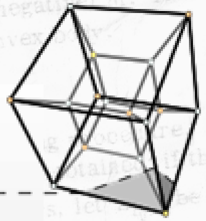
(right) basis temperature = log(det(B))

A reliable and fast lex. implementation

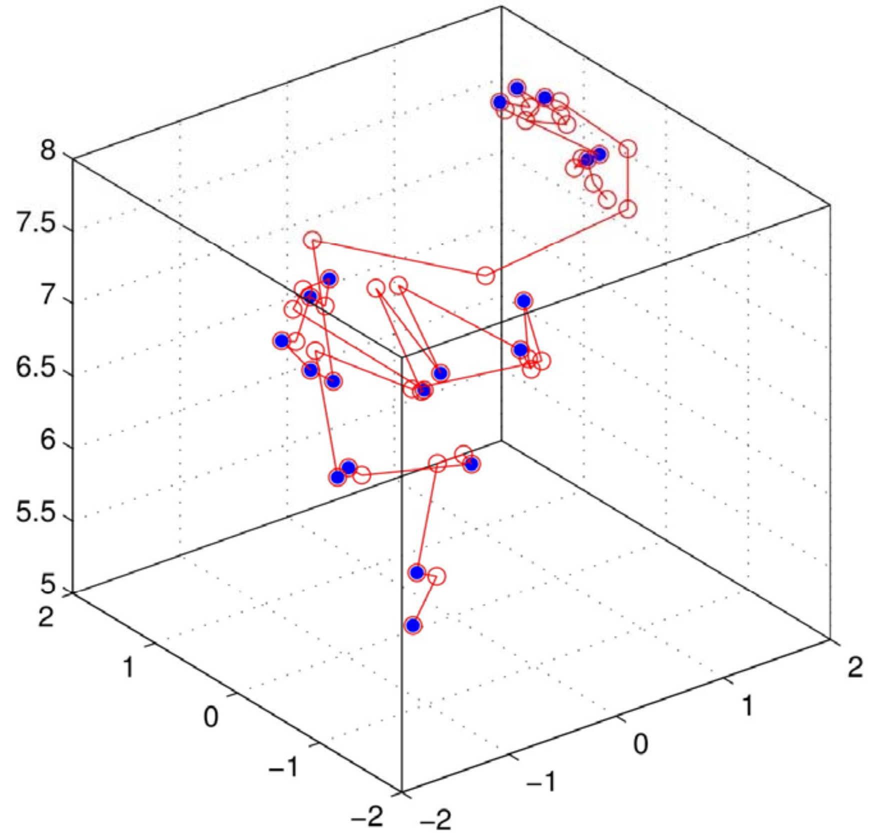
- Based on the use of a black-box LP solver
  - Step 0. Optimize x_0 --> optimal value x^*_0
  - Step 1. Fix $x_0 = x^*_0$, and optimize x_1 --> optimal value x^*_1
  - Step 2. Fix also $x_1 = x^*_1$, and optimize x_2 --> optimal value x^*_2
  - ...
- **Simple, but … does it work?**

Forget! … just a nightmare!

- <u>Clever version:</u> at each step, instead of adding the equation $x_j = x^*_j$
- … fix out of the basis all the nonbasic variables with nonzero reduced cost
- **→ sequence of fast (and clean) reoptimizations on smaller and smaller degeneracy subspaces, leading to the required lex-optimal tableau**
- → to be compared with Balas-Perregaard L&P pivoiting…



Role of lex reoptimization (stein15)



Heuristic pivoting rules

- Alternative pivoting rules to **mimic** the lexicographic dual simplex
- Useful to try to highlight the crucial properties that allow the lexicographic method to produce stable Gomory cuts

- **HEUR1**

  Just a truncated lex. method on the arguably most-important variable

  – After the addition of a FGC, lex. minimize (x_0, x_i) where x^*_i is the basic fractional variable generating the FGC

- **HEUR2**

  Try to select optimal vertices where the previously generated FGCs are slack (having a cut slack into the basis avoids it appears in the cut-generation row and hence reduces cut correlation)

  – After the addition of the i-th FGC with slack variable s_i (say), try to keep all slacks $s_1,\ldots,s_i$ inside the opt. basis by lex. minimizing

$$(x_0, -s_i, \ldots, -s_2, -s_1)$$

Experiments

- Pure ILP instances from MIPLIB 3 and MIPLIB 2003 (none solved by previous pure cutting plane methods based on FGCs)

- Input data is assumed to be integer. Cuts also derived from the **objective function** tableau row, as prescribed by Gomory's proof of convergence.

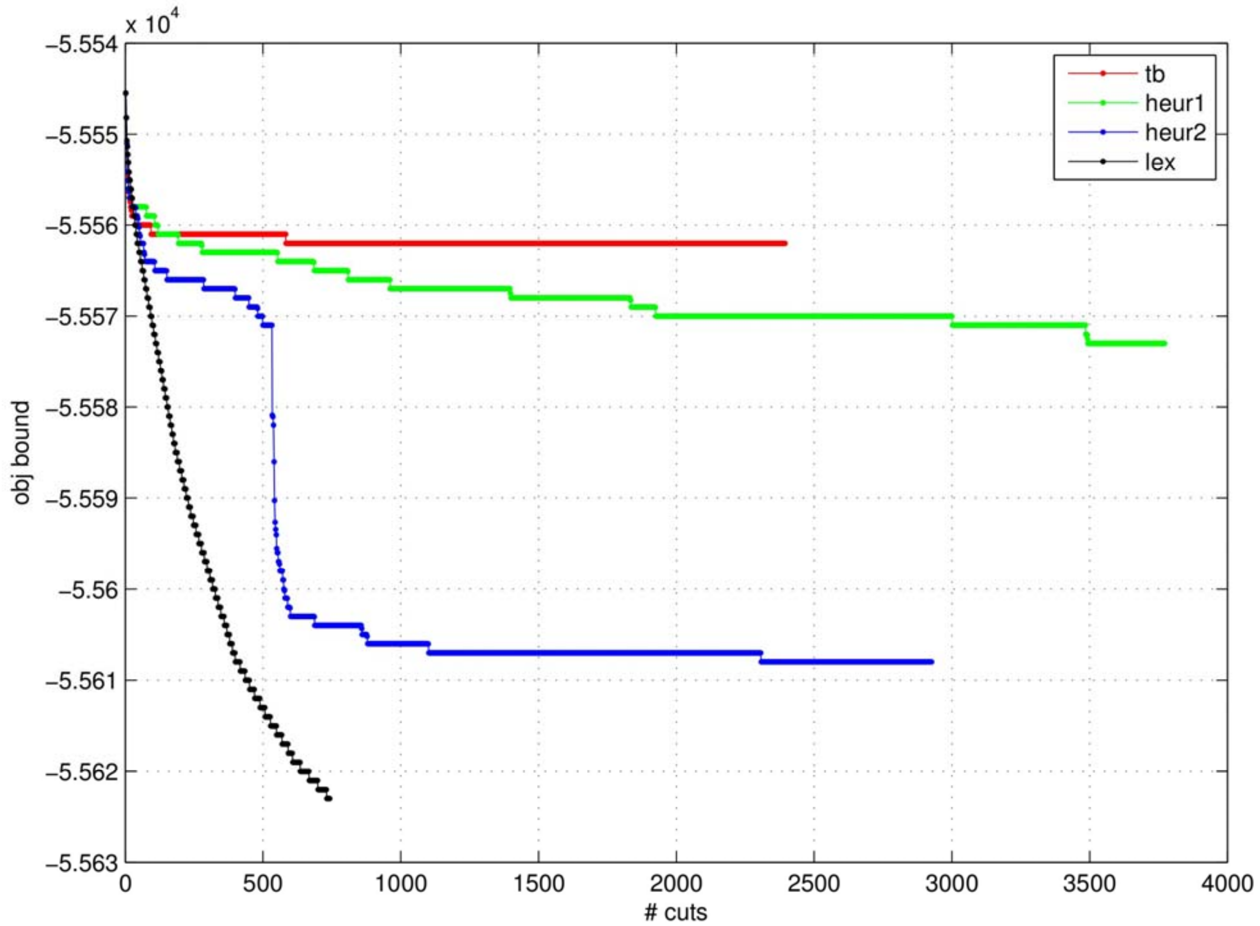
- Once a FGC is generated, we put it in its **all-integer form** in the space of the structural variables.

- **Numerical stabilization**: we use a threshold of 0.1 to test whether a coefficient is integer or not:
  – a coefficient with fractional part smaller than 0.1 is **rounded** to its nearest integer
  – cuts with larger fractionality are viewed as unreliable and hence **discarded**.

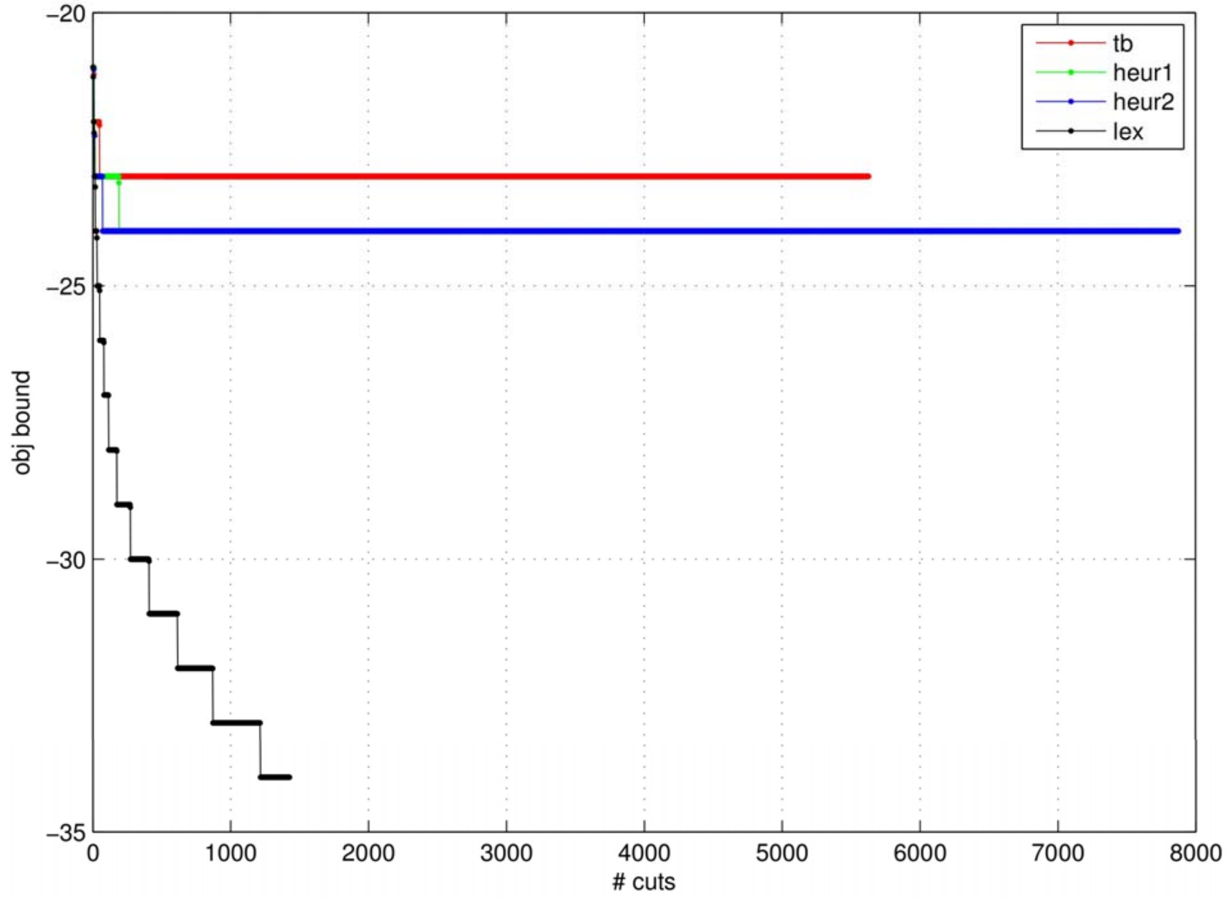
Experiments (one cut at a time)

- First set of experiments addressed the **single-cut** version of Gomory's algorithm
- Runs on a PC Intel Core 2 Q6600, 2.40GHz, with a **time limit of 1 hour** of CPU time and a **memory limit of 2GB** for each instance

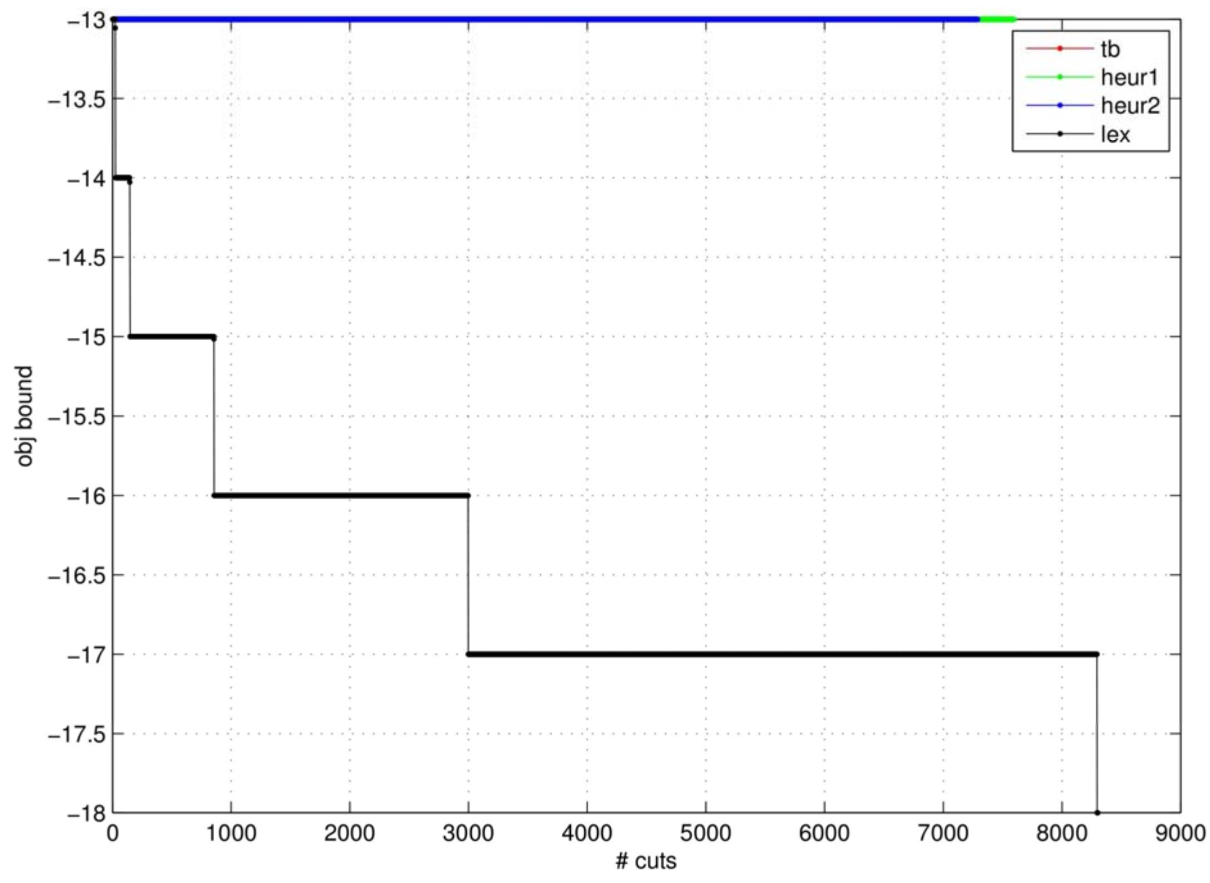
Instance air04



Instance bm23



Instance stein27

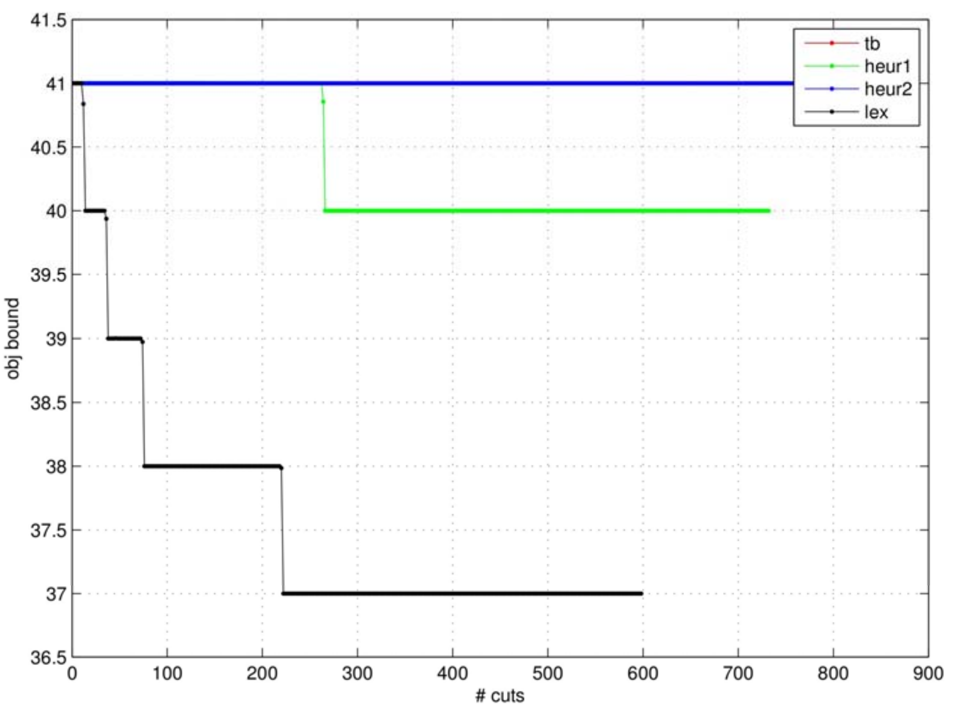


Multi-cut version

Problem	Textbook	Itrs	Cuts	Time	ClGap	Coeff.	Det.	κ	Lex	Itrs	Cuts	Time	ClGap	Coeff.	Det.	κ
air04	M	35	11912	1209.41	11.23	2.4e+03	6.9e+11	9.3e+09	tL	150	47995	3606.61	**19.21**	7.3e+03	2.2e+09	3.9e+11
air05	M	30	9648	955.09	5.11	7.1e+02	4.2e+11	4.9e+09	tL	261	76263	3598.20	**14.00**	1.5e+03	4.9e+10	9.6e+10
bm23	E	144	2168	9.33	5.28	2.3e+15	5.5e+30	8.7e+24	O	659	8298	1.67	**100.00**	3.7e+02	4.6e+10	3e+14
cap6000	tL	979	11693	3614.59	10.34	1.4e+09	7e+15	4.3e+20	E	966	6769	2110.12	**24.66**	4.5e+06	5.1e+09	1.6e+15
hard_ks100	tL	1950	10770	3603.03	100.00	5e+04	2.2e+05	1.5e+12	O	98	439	0.63	**100.00**	5e+05	1.5e+04	2.8e+09
hard_ks9	O	269	1678	0.30	100.00	7.8e+04	1.7e+05	6.9e+09	O	139	614	0.12	**100.00**	3e+04	3.6e+04	2.9e+10
krob200	O	44	2017	153.92	100.00	1.6e+02	1e+06	9.4e+07	O	17	282	22.18	**100.00**	7.2	1e+06	3.7e+04
l152lav	M	525	31177	2715.78	40.59	5.3e+03	8e+08	1.6e+10	O	681	22364	206.82	**100.00**	1.8e+04	6.7e+05	2.4e+10
lin318	O	15	250	12.82	100.00	7	3.4e+07	9.7e+05	O	19	416	124.27	**100.00**	17	3.2e+07	1.8e+08
lseu	E	149	3710	34.35	44.58	1.7e+14	9.8e+33	2e+19	O	1330	18925	6.62	**100.00**	9.6e+03	1.2e+14	2.6e+14
manna81	O	1	270	8.88	100.00	1	5.2e+92	3.7e+06	O	2	280	29.58	**100.00**	1	1.7e+97	3.7e+06
mitre	M	94	30016	1262.24	85.52	8.9e+08	Inf	4.2e+20	tL	232	20972	3619.30	**97.83**	6.2e+06	Inf	9.1e+13
mzzv11	M	33	17936	1735.42	38.56	4.9e+02	Inf	2.6e+11	tL	61	25542	3739.41	**40.60**	1.8e+02	2.8e+51	8.5e+11
mzzv42z	M	62	14664	1515.50	33.80	3e+06	Inf	6.5e+14	tL	61	14830	3616.85	**25.50**	8.5e+02	1e+38	8.7e+12
p0033	M	1529	23328	2493.35	81.53	1.8e+14	1.1e+34	2.2e+24	O	87	1085	0.14	**100.00**	4.5e+02	1.6e+13	6e+14
p0201	tL	1332	55408	3602.21	19.32	2e+12	2e+34	2.5e+26	E	27574	629004	1119.71	**74.32**	2.3e+05	3.6e+22	7.4e+11
p0548	M	825	35944	2223.35	**48.98**	1.3e+09	2.4e+137	2e+24	E	461	27067	131.69	47.50	4.8e+06	7.1e+135	5.5e+14
p2756	M	740	19925	2423.48	78.86	4.5e+12	Inf	1.3e+27	E	642	14645	509.72	**79.09**	2.4e+05	Inf	1.1e+13
pipex	M	2929	49391	1921.78	51.25	1.3e+14	3.3e+27	4e+28	O	50000	488285	188.11	**74.18**	2.8e+05	8.5e+11	1e+14
protfold	M	7	5526	1058.84	8.76	4.6e+06	Inf	5.1e+13	tL	159	38714	3607.08	**45.26**	30	1.5e+32	1e+07
sentoy	M	1953	34503	2586.38	18.25	5.1e+14	5.3e+43	7e+30	O	5331	68827	25.85	**100.00**	7.6e+04	3.2e+34	3.9e+14
seymour	tL	18	11748	7517.40	21.67	3.3e+08	1.3e+159	1.4e+16	tL	87	48339	3610.62	**26.89**	78	1e+23	3.6e+07
stein15	E	116	3265	9.50	20.92	2.5e+15	2.4e+26	3.2e+28	O	64	676	0.14	**100.00**	15	2e+02	2.3e+06
stein27	E	57	2399	20.01	0.00	1.7e+15	2.8e+36	8.3e+18	O	3175	37180	19.72	**100.00**	7.5e+02	1.9e+04	1.9e+05
timtab1-int	M	231	73188	585.72	23.59	1.6e+11	Inf	2.1e+21	cL	3165	1000191	841.56	**50.76**	3.5e+06	Inf	1.7e+16

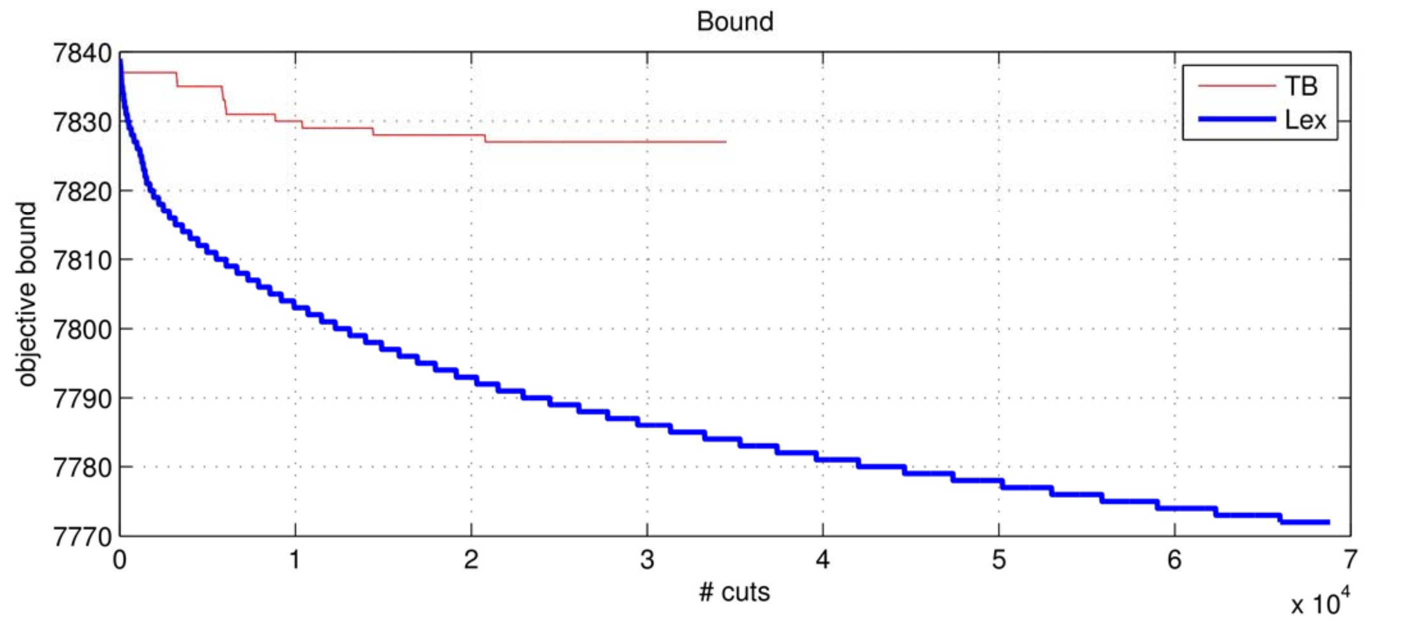
Table 1: Comparison between textbook and lexicographic implementation of Gomory's algorithm (multi-cut version)

Instance protfold



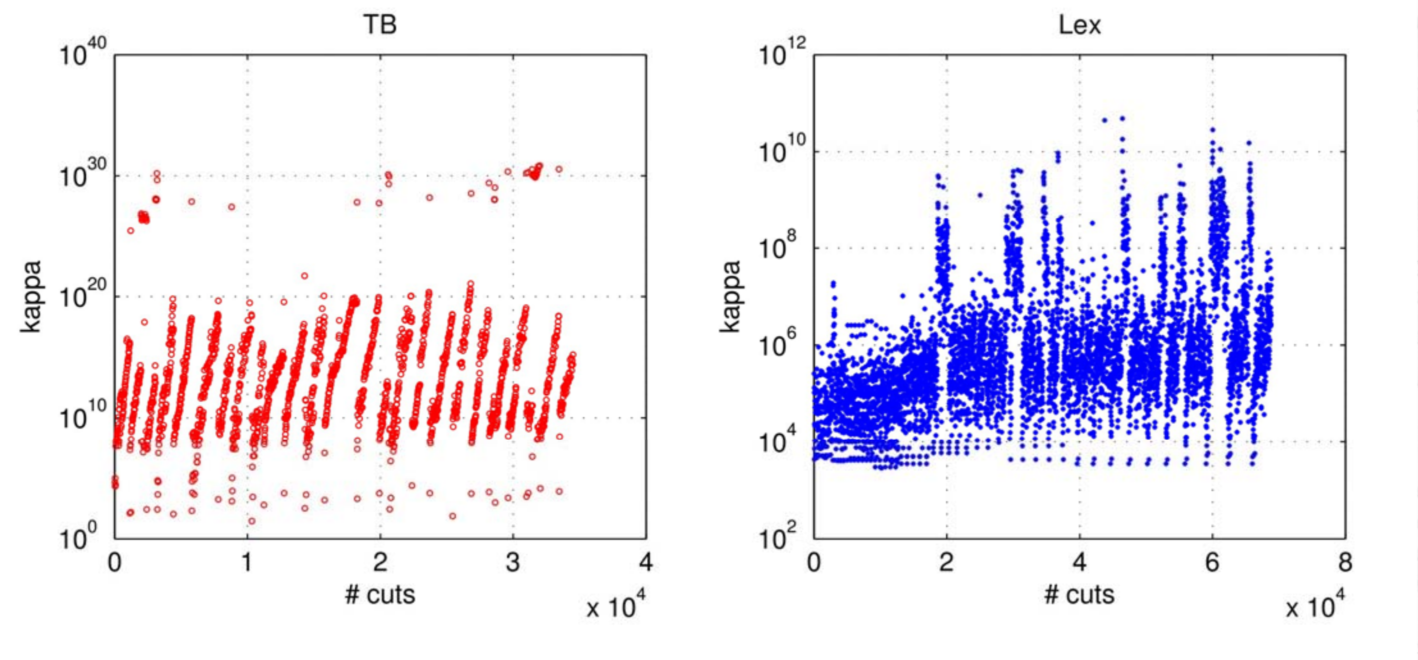
- Very hard protein-folding instance (max; opt=31; 9x2 days using tuned Xpress2006b)
- **Cplex**: after 14 days, memory overflow (3GB) after 4,000,000 nodes; best bound 36
- **lexFGC**: after 14 days, about 3,000,000 cuts in 9,500 rounds; best bound 34, still alive and kicking (max 100 MB memory)

Looking inside the chest: sentoy



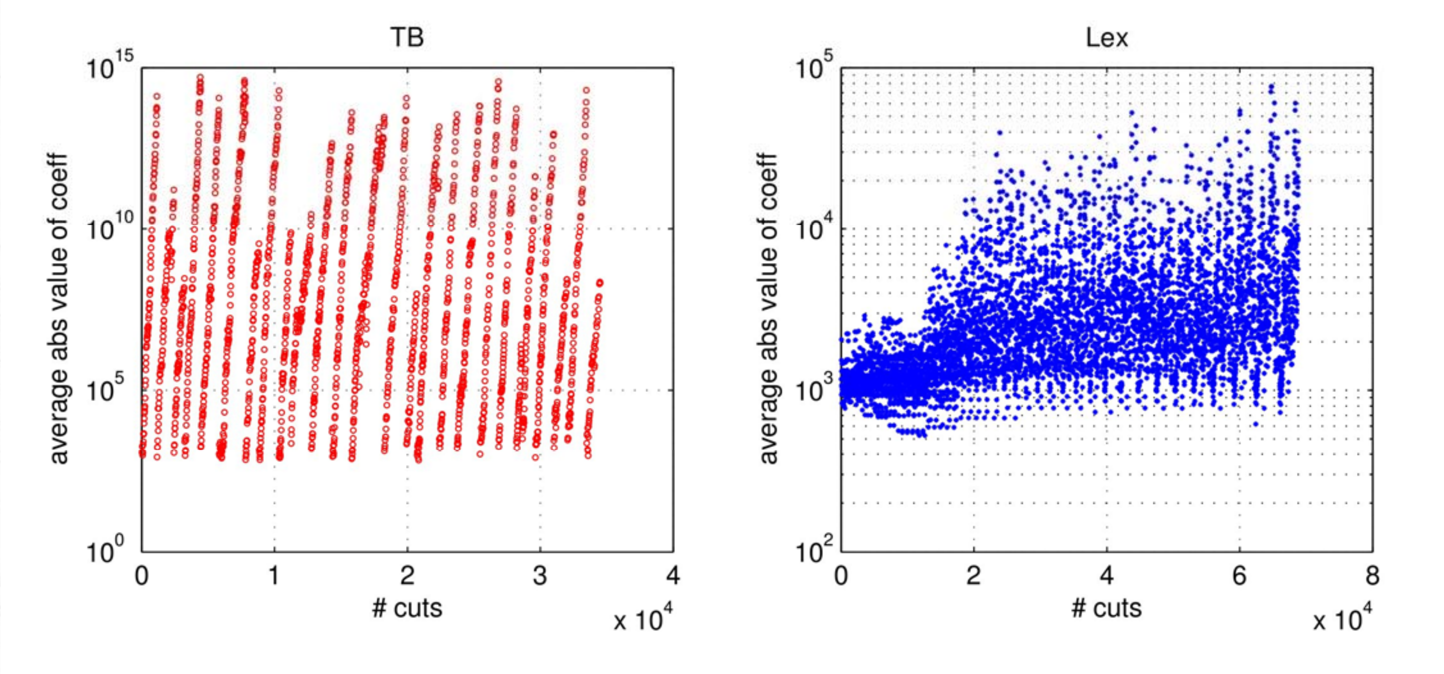
Upper bound (max problem; multi-cut version)

Looking inside the chest: sentoy



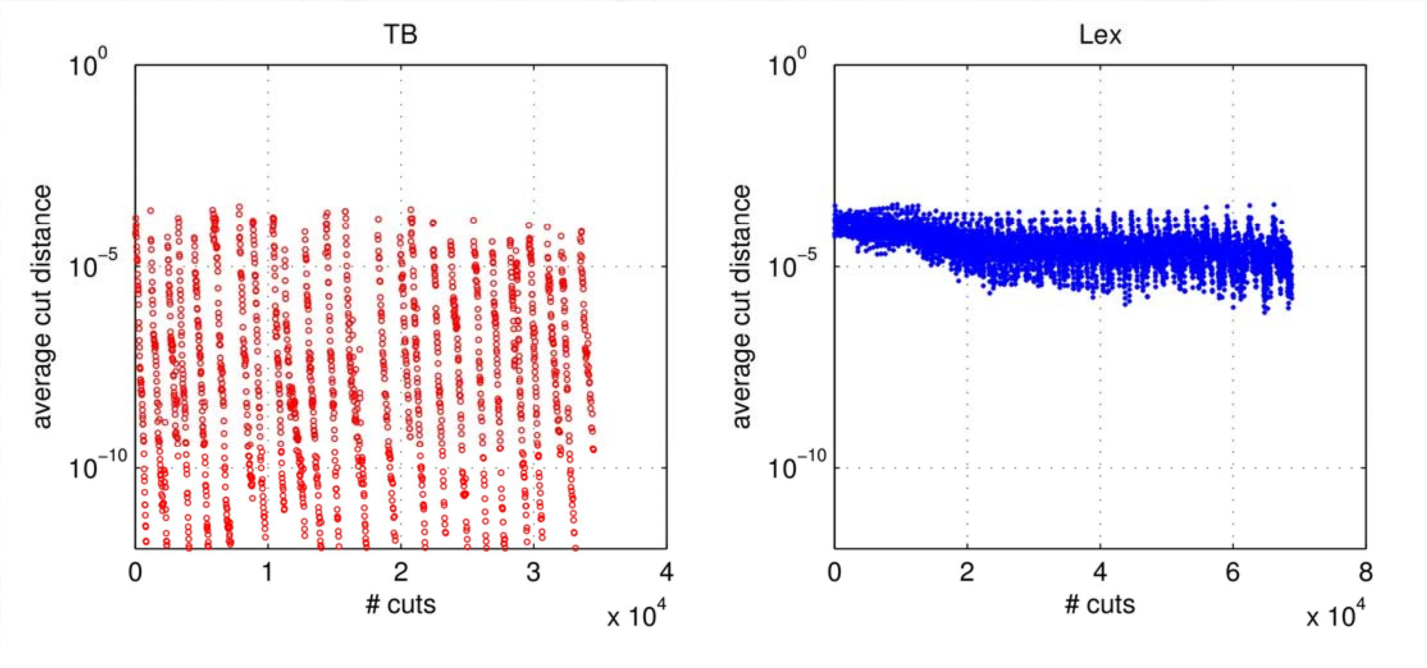
Condition number of the optimal basis

Looking inside the chest: sentoy

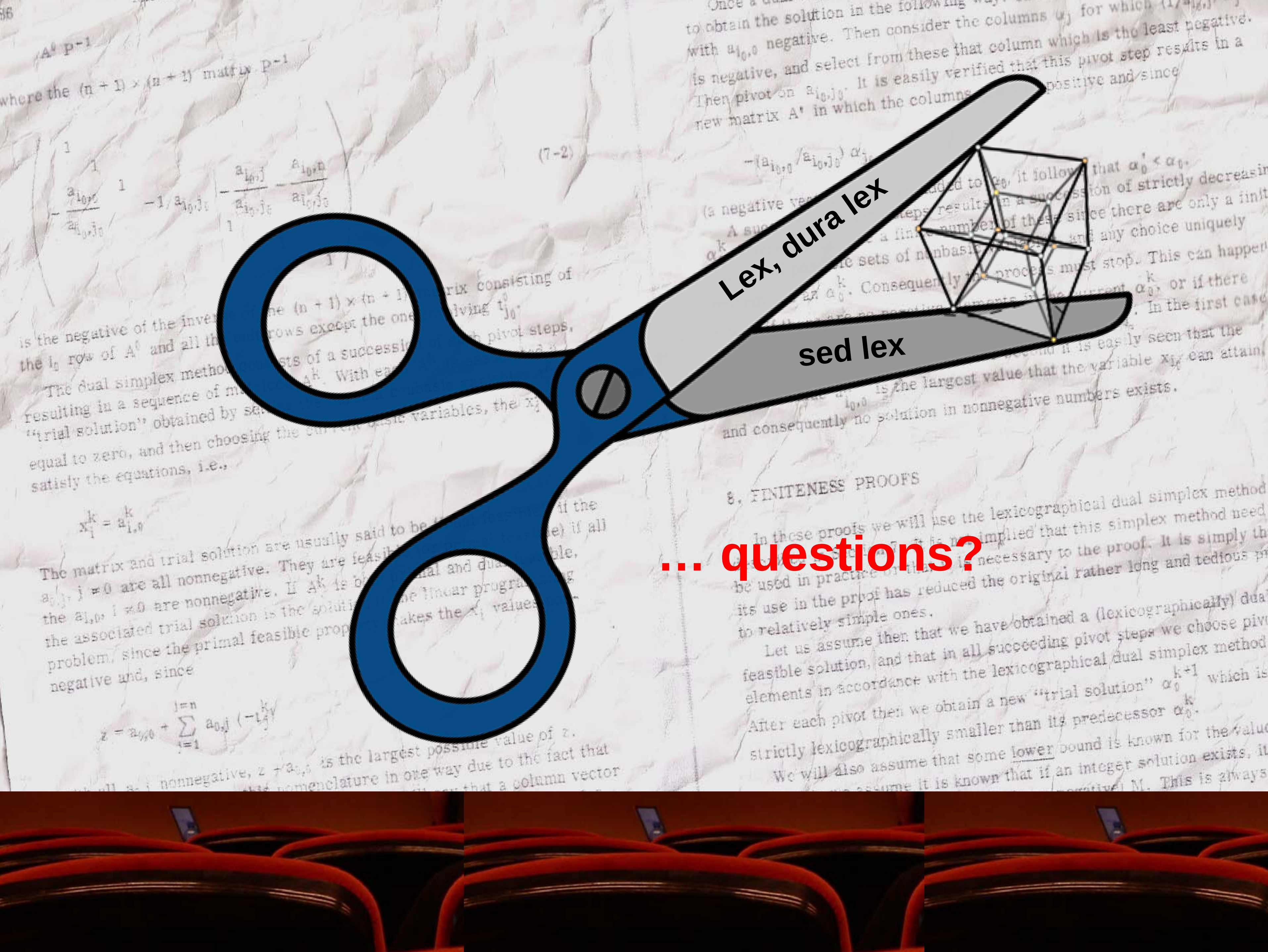


Average absolute value of cut coefficients

Looking inside the chest: sentoy



Avg. geometric distance of x^* from the FGC


Lex, dura lex
sed lex
… questions?