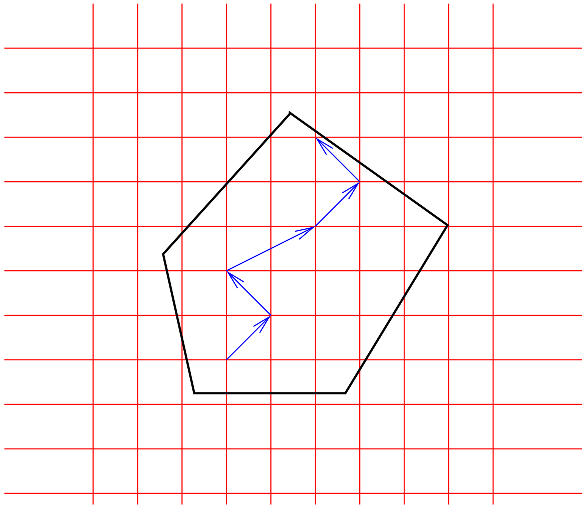
New Integer Programming Results Using Test Sets Techniques

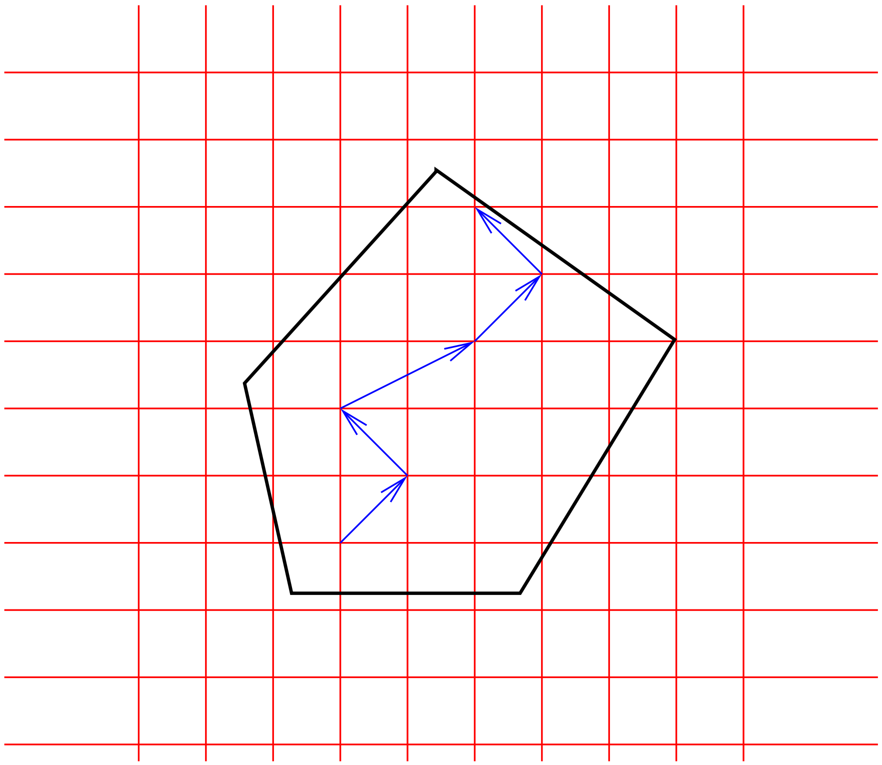
Jesús Antonio De Loera
University of California, Davis

Results joint work with subsets of the following people:
S. Onn, R. Hemmecke, U. Rothblum, and R. Weismantel



Test Sets and Augmentation Methods

For a integer program (or programs) $P_b = min\{cx : Ax = b,\ x \geq 0\}$ a test set is a finite set of integral vectors such that every feasible non-optimal solution can be improved by adding a vector from the test set.



Graver and Gröbner Bases

- The lattice $L(A) = \{x \in \mathbb{Z}^n : Ax = 0\}$ has a natural partial order. For $u, v \in \mathbb{Z}^n$ we say that u is *conformal* to v, denoted $u \sqsubset v$, if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for $i = 1, \ldots, n$, that is, u and v lie in the same orthant of $\mathbb{R}^n$ and each component of u is bounded by the corresponding component of v in absolute value.

- The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A.

- **Example:** If $A = [1 \quad 2 \quad 1]$ then its Graver basis is $\pm\{[2, -1, 0], [0, -1, 2], [1, 0, -1], [1, -1, 1]\}$.

- Graver bases for A can be used to solve the augmentation problem Given $A \in \mathbb{Z}^{m \times n}$, $x \in \mathbb{N}^n$ and $c \in \mathbb{Z}^n$, either find an improving direction $g \in \mathbb{Z}^n$, namely one with $x - g \in \{y \in \mathbb{N}^n : Ay = Ax\}$ and $cg > 0$, or assert that no such g exists.

- The fastest algorithm to compute Graver bases is based on a completion and project-and-lift method. Implemented in `4ti2`. Equivalent to the computation of minimal Hilbert bases.

- Graver bases contain, and generalize, the LP test set given by the circuits of the matrix A. Circuits contain all possible edges of polyhedra in the family

$$P(b) := \{x |\ Ax = b, x \geq 0\}$$

.

- For a fixed cost vector c, there exist a more sophisticated set of vectors, subset of the Graver basis of A, which gives a connected oriented graph with a unique sink, a Gröbner basis for A with respect to c.

- Originally discovered by Algebraic Geometers. There is an algebraic algorithm, Buchberger's algorithm, but there are now more specialized algorithms.

- We can visualize a Graver or a Gröbner basis of a family of lattice point sets:

$$L(b) := \{x|\ Ax = b, x \geq 0,\ x \in \mathbb{Z}^n\}$$

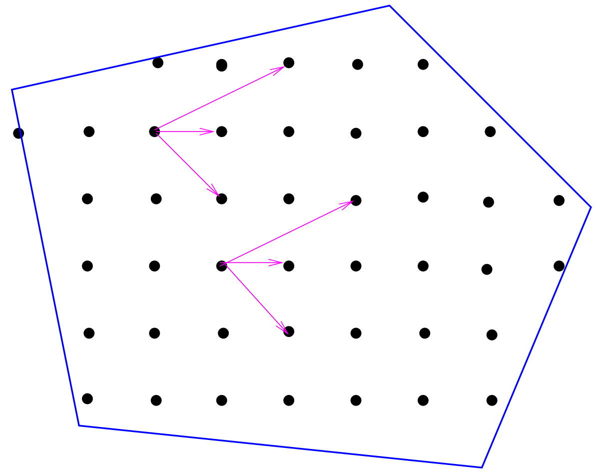
.

- The Graver basis elements are edges departing from each lattice point $u \in L(b)$.

  Theorem The Graver basis contains all edges for all integer hulls $conv(\{x|\ Ax = b, x \geq 0,\ x \in \mathbb{Z}^n\})$ as b changes.

- For a Gröbner basis, the edges receive an orientation, given by the cost vector c.

  Theorem: [Diaconis-Sturmfels] The graph whose vertices are lattice points and arrows are Gröbner basis elements is connected for all right-hand side b. The orientation induced by cost c has a unique sink.



PROBLEM In general, test sets can be exponentially large even in fixed dimension! People typically deal with them via a list of the whole test set. Hard to compute, you don't want to do this often.

SOLUTIONS Ideas how to make them "manageable". We present two such situations, where the Graver and Gröbner test sets become very very manageable.

Graver bases Application

N-fold Systems

Fix any pair of integer matrices A and B with the same number of columns, of dimensions $r \times q$ and $s \times q$, respectively. The n-fold matrix of the ordered pair A, B is the following $(s + nr) \times nq$ matrix,

$$[A, B]^{(n)} \quad := \quad (\mathbf{1}_n \otimes B) \oplus (I_n \otimes A) \quad = \quad \begin{pmatrix} B & B & B & \cdots & B \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix} \quad .$$

Theorem Fix any integer matrices A, B of sizes $r \times q$ and $s \times q$, respectively. Then there is a polynomial time algorithm that, given any n and any integer vectors b and c, solves the corresponding n-fold integer programming problem.

$$\min\{cx : \ [A, B]^{(n)} x = b, \ x \in \mathbb{N}^{nq}\} \quad .$$

Our proof

- **Lemma 1** There is a polynomial time algorithm that, given any matrix $A \in \mathbb{Z}^{m \times n}$ along with its Graver basis $G(A)$, and vectors $x \in \mathbb{N}^n$ and $c \in \mathbb{Z}^n$, solves the integer program $IP_A(b, c)$ with $b := Ax$.

  Lemma holds for any matrix, but its complexity bound depends on the size of the Graver basis which is part of the input.

- **Lemma 2** Fix any pair of integer matrices $A \in \mathbb{Z}^{r \times q}$ and $B \in \mathbb{Z}^{s \times q}$. Then there is a polynomial time algorithm that, given n, computes the Graver basis $G([A, B]^{(n)})$ of the n-fold matrix $[A, B]^{(n)}$. In particular, the cardinality and the bit size of $G([A, B]^{(n)})$ are bounded by a polynomial function of n.

- **Proof by Example:** Consider the matrices $A = [1\ 1]$ and $B = I_2$. The Graver complexity of the pair A, B is $g(A, B) = 2$. The 2-fold matrix and its Graver basis, consisting of two antipodal vectors only, are

$$[A,B]^{(2)} \;=\; \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix},\; G([A,B]^{(2)}) \;=\; \pm\begin{pmatrix} 1 & -1 & -1 & 1 \end{pmatrix}$$

By our theorem, the Graver basis of the 4-fold matrix

$$[A,B]^{(4)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

$$G([A,B]^{(4)}) = \pm \begin{pmatrix} 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \end{pmatrix}.$$

- **Lemma 3** Fix any pair of integer matrices $A \in \mathbb{Z}^{r\times q}$ and $B \in \mathbb{Z}^{s\times q}$. Then there is a polynomial time algorithm that, given n and demand vector $b \in \mathbb{N}^{s+nr}$, either finds a feasible solution $x \in \mathbb{N}^{nq}$ to the generalized n-fold integer programming problem (2), or asserts that no feasible solution exists.

Application 1: Convex Integer Optimization

Theorem For fixed d and matrices A, B, there is a polynomial oracle-time algorithm that given $n, b, w_1, \ldots, w_d$ and a convex function c, presented via a comparison oracle, solves the convex integer programming problem

$$\max\{c(w_1x, w_2x, \ldots, w_dx) : [A, B]^{(n)}x = b, x \geq 0\}$$

Lemma For every integer matrix A and every integer vector b, the Graver basis of A, $G(A)$, contains **all** edge-directions of the polyhedron $conv(\{x \text{ integer} : Ax = b,\ x \geq 0\})$.

Application 2: N-fold systems in the wild

An n-fold matrix:

$$A = \begin{bmatrix} 1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0 \\ 0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0 \\ 0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0 \\ 0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \end{bmatrix}$$

Does anyone recognize this matrix??

Transportation problems are N-fold systems!

It is from the classical Transportation problem.

68	119	26	7	220
20	84	17	94	215
15	54	14	10	93
5	29	14	16	64
108	286	71	127	

There are
1,225,914,276,768,514 such tables.

Multi-way Transportation Polytopes are N-fold systems too!

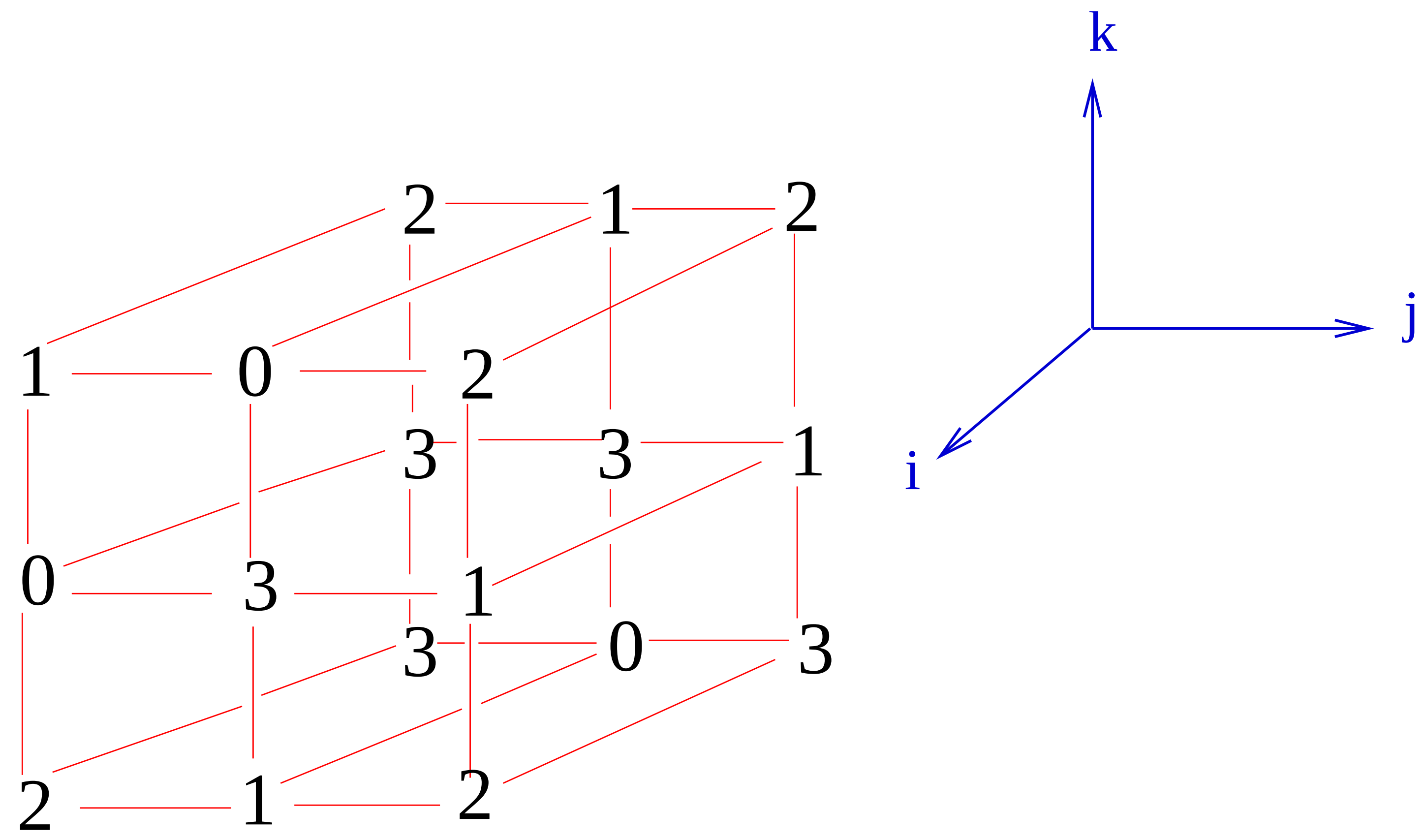
A d-**table of size** $(n_1, \ldots, n_d)$ is an $n_1 \times n_2 \times \cdots \times n_d$ array of nonnegative real numbers $v = (v_{i_1,\ldots,i_d})$, $1 \leq i_j \leq n_j$.

For $0 \leq m < d$, an m-**margin** of v is any of the $\binom{d}{m}$ possible m-tables obtained by summing the entries over all but m indices.

Example If $(v_{i,j,k})$ is a 3-table then its 0-marginal is $v_{+,+,+} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} v_{i,j,k}$, its 1-margins are $(v_{i,+,+}) = (\sum_{j=1}^{n_2} \sum_{k=1}^{n_3} v_{i,j,k})$ and likewise $(v_{+,j,+})$, $(v_{+,+,k})$, and its 2-margins are $(v_{i,j,+}) = (\sum_{k=1}^{n_3} v_{i,j,k})$ and likewise $(v_{i,+,k})$, $(v_{+,j,k})$.

Definition: A **multi-index transportation polytope** is the set of all real d-tables that satisfy a set of given margins.

Axial 3-Way transportation Polytopes



In this case we specify **plane-sums** in **three** possible directions.

We can formulate the 3-way integer transportation problem as an n-fold integer programs! Application of our theorem:

Theorem Fix any r, s. Then there is a polynomial time algorithm that, given l, integer objective vector c, and integer line-sums $(u_{i,j})$, $(v_{i,k})$ and $(w_{j,k})$, solves the integer transportation problem

$$\min\{\, cx \ :\ x \in \mathbb{N}^{r\times s\times l} \,,\, \sum_i x_{i,j,k} = w_{j,k} \,,\, \sum_j x_{i,j,k} = v_{i,k} \,,\, \sum_k x_{i,j,k} = u_{i,j} \,\}$$

The only known polynomial time algorithm for the corresponding line-sum integer transportation problem

Gröbner Bases Applications

Axial 3-way transportation polytopes are special

For the axial 3-way transportation polytope we are given 1-marginals $a = (a_1, \ldots, a_n)$, $b = (b_1, \ldots, b_m)$, and $c = (c_1, \ldots, c_k)$.

- (dimension and real feasibility easy) If $\sum a_i = \sum b_i = \sum c_i$, then polytope is non-empty and it has dimension $n \cdot m \cdot k - n - m - k + 2$.

- (integral feasibility easy) Given integral consistent marginals a, b, c, it is guaranteed the polytope contains an integral point.

  Moreover it can be found via a greedy algorithm in linear time!! This is done via the North-west Corner rule.

  For some special objective cost matrices this process gives an **optimal** integer solution! For example, Monge Matrices.

Theorem: [Sturmfels, Hosten-Sullivant] There is an easy (lexicographic) Gröbner basis (entries are 0,-1,1 only) generalizing the one we know classic 2-way transportation case.

0	0	0	0	0	0	0	0	0
0	0	−1	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	−1	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

MAIN RESULT

THEOREM: [JDL & S. Onn 2005] Any convex rational polytope

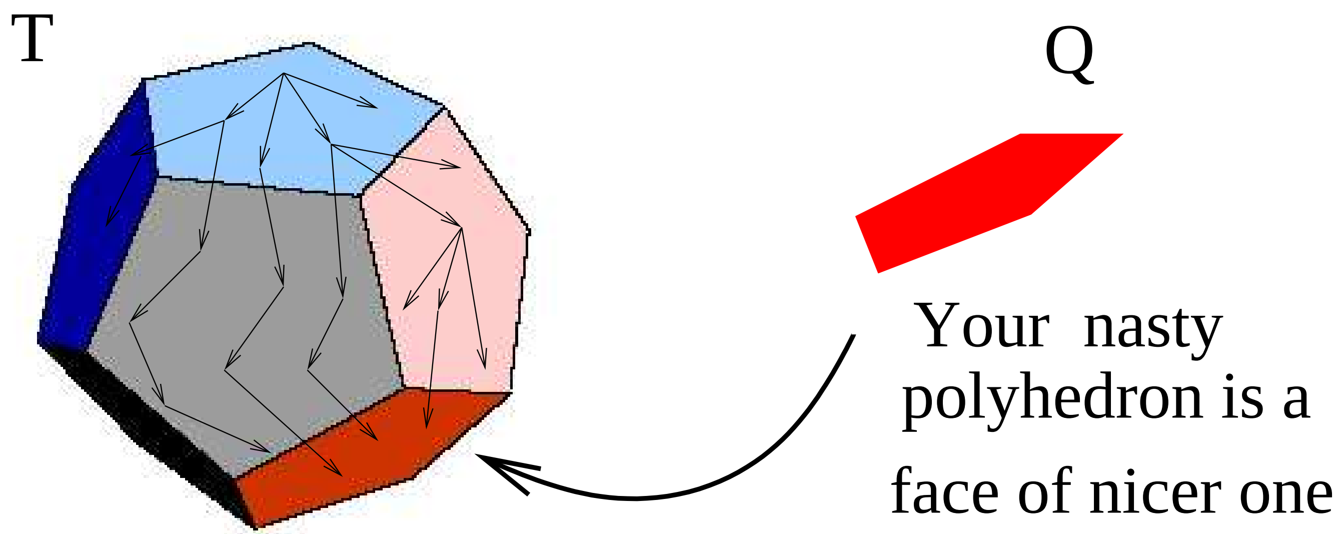
$$Q = \{y \in \mathbb{R}^n : \; Ay = b \; y \geq 0 \}$$

is polynomial-time representable as **a face** of a 3-way $(r \times c \times 3)$ transportation polytope with 1-margins:

$T = \{ x \in \mathbb{R}^{r\times c\times h}_{\geq 0} : \; \sum_{i,j} x_{i,j,k} = w_k \, , \; \sum_{j,k} x_{i,j,k} = v_i \, , \; \sum_{i,k} x_{i,j,k} = u_j \} \, .$

Why do we care?

The Geometry has Changed!



Wish: to find out whether there is a lattice point inside a polytope Q:
Yields two heuristics and one algorithm for feasibility of ILPs!

Heuristic One: Gröbner bases

- make Q a face of an axial transportation polytope T (main theorem). This face is given by particular entries are zero.

- Find a lattice point in T using the generalized Northwest-corner rule.

- Using the Gröbner basis make greedy aiming to make the entries that have to be zero, zero!

  If we succeed we have found point of Q!

Heuristic Two: Monge matrices

- make Q a face of an axial transportation polytope T (main theorem). This face is given by particular entries are zero.

- Try to find a Monge sequence with respect to a cost matrix that forces the entries that have to be zero (large costs on those entries).

  If we succeed we have found point of Q!

Algorithm: Reverse-Search

- make Q a face of an axial transportation polytope (main theorem). This face is given by particular entries are zero.

- Using the Gröbner basis make greedy moves we could in principle list of **all** lattice points inside the axial transportation polytope taking care to announce any point in Q.

- The list of lattice points can be done without stacks of memory (a single pointer) using Avis-Fukuda **reverse-search** algorithm.

  If no point of Q is found, we have a proof of infeasibility.