Quantum Integer Programming

47-779
Test-set methods - Graver basis

Quiz 2

Solve the following Integer Program using the BPT and the CT methods

$$\min_{\mathbf{x}} 2x_0 + 4x_1 + 4x_2 + 4x_3 + 4x_4 + 4x_5 + 5x_6 + 4x_7 + 5x_8 + 6x_9 + 5x_{10}$$

$$s.t. \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{x} \in \{0,1\}^{11}$$

Remember

- IP to Polynomials
- Polynomials to Ideal
- Gröbner of Ideal
- Interpretation of Gröbner
- Test-set

Food for thought: Would it be possible to solve via Gröbner basis?

$$\min \sum_i \exp(c_i x_i^2) \quad \text{or} \quad \min \sum_i \log(c_i + x_i)$$

$$A\mathbf{x} = b \qquad\qquad A\mathbf{x} = b$$

$$\mathbf{x} \in \{0,1\}^n \qquad\qquad \mathbf{x} \in \{0,1\}^n$$

Agenda

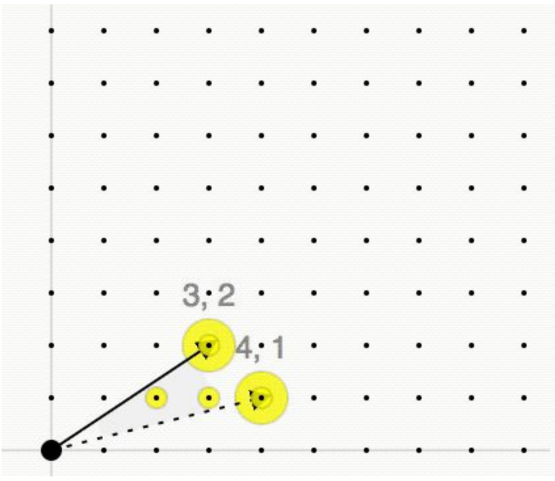
- Brief History of Test Sets: Graver, Neighborhood of the Origin, Groebner
- Hilbert Basis
- Integral Basis
- Graver Basis
- Graver and Grobner
- N-Fold Integer Program
- Comparison of Naive Graver Basis approach with IP solvers
- Take-home message

Acknowledgements: Material is based on the [Lecture by Maria Isabel Hartillo for IMUS-MSRI2016] and the [slides by Shmuel Onn] and De Loera, Jesús A., Raymond Hemmecke, and Matthias Köppe, eds. *Algebraic and geometric ideas in the theory of discrete optimization*. Society for Industrial and Applied Mathematics, 2012.

Hilbert basis

We will obtain the Hilbert basis from two different approaches.

$\mathcal{F} = \text{Cone} \cap \mathbb{Z}^2$



$\mathcal{H}$ in yellow

- A set $\mathcal{H} \subset \mathcal{F} \subset \mathbb{Z}^n$ is an integral generating set of $\mathcal{F}$ if for every $\mathbf{x} \in \mathcal{F}$ there exists $\{\mathbf{h}_1, \cdots, \mathbf{h}_k\} \in \mathcal{H}$ such that
  $$\mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{h}_i, \lambda_i \in \mathbb{Z}_+$$
  That integral generating set can be called an **integral basis** if it's minimal with respect to inclusion.
  The integer points is a polyhedral cone have a finite **integral basis** called the **Hilbert basis**
- Consider solving a homogeneous system of linear equations over the non-negative integers. Using the constraint matrix from our integer programs as such system we would like to find the **kernel** of that matrix.
  $$\ker(\mathbf{A}) = \{x \in \mathbb{Z}_+^n : \mathbf{A}\mathbf{x} = \mathbf{0}\}$$
  The subset of non-zero minimal elements of the **kernel**:
  $$\mathcal{H}_{\mathbf{A}} = \{\mathbf{x} \in \ker(\mathbf{A}) \setminus \{\mathbf{0}\}\} \text{ s.t. is minimal to inclusion}$$
  Is the **Hilbert basis** of the matrix $\mathbf{A}$

Integral basis

Slight generalization of **Hilbert basis** which can be used in a simplex-like integral basis algorithm to solve IPs by pivoting out elements until it becomes irreducible (no branching or cuts). **This method is efficient to verify optimality**

Table 1. Problems from the MIPLIB

	Columns	Rows	Iterations	Max. Size[a]	Time[b]
p0033	33	16	71	54	1
p0201	201	133	191	278	48
p0282	282	241	34	243	14
p0548	548	176	549	1424	610
lseu	28	89	329	310	38
cap6000	6000	2176	12	134	1592
mod008	319	6	636	848	409

[a] Maximum number of non-basic variables during the course of the algorithm.
[b] In seconds, on a Sun Enterprise 450 with 300 MHz.

*Results from 20 years ago!

Although finite, the test-set might be huge!

Promising incorporation within branch-and-cut methods.

[1] Haus, Utz-Uwe, Matthias Köppe, and Robert Weismantel. "The integral basis method for integer programming." Mathematical Methods of Operations Research 53.3 (2001): 353-361.

Graver basis - Definition

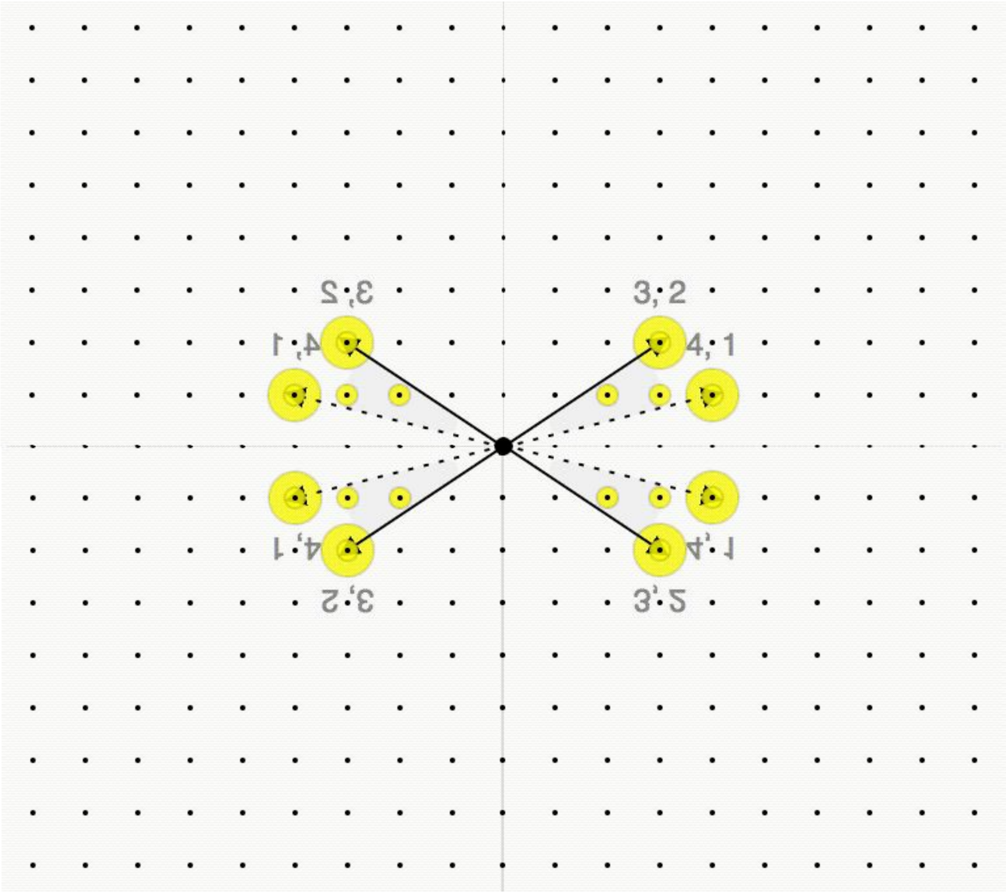
Given the constraint matrix $\mathbf{A}$
We denote $\mathbb{O}_j$ the j^{th} orthant of $\mathbb{R}^n\,(2^n)$
Then for each orthant we define

$$\mathcal{H}_j(\mathbf{A}) = \mathcal{H}(\mathbf{A}) \cap \mathbb{O}_j$$

The union of all these minimal **Hilbert basis**

$$\mathcal{G}(\mathbf{A}) = \bigcup_j \mathcal{H}_j(\mathbf{A})$$

Is denoted the **Graver basis**



Graver basis - Ordering

Notation

- $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ are conformal if $u_j v_j \geq 0, \forall j = \{1, \cdots, n\}$, meaning that both belong to the same orthant
  - Example: $\mathbf{a} = (1, -2, 0) \quad \mathbf{b} = (2, -1, 8) \quad \mathbf{c} = (-3, -4, 6)$
    $\mathbf{a}$ and $\mathbf{b}$ are conformal, but $\mathbf{c}$ is not conformal to $\mathbf{a}$ or $\mathbf{b}$
- For $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ we denote $\mathbf{u} \sqsubseteq \mathbf{v}$ if $\mathbf{u}$ and $\mathbf{v}$ are conformal and if $|u_j| \leq |v_j|, \forall j = \{1, \cdots, n\}$; meaning that both belong to the same orthant and $\mathbf{v}$ is farther from the origin than $\mathbf{u}$
  - Example: Neither $\mathbf{a} \not\sqsubseteq \mathbf{b}$ or $\mathbf{b} \not\sqsubseteq \mathbf{a}$, but $\mathbf{a} \sqsubseteq \mathbf{d}$ where $\mathbf{d} = (3, -4, 1)$

From the constraint matrix $\mathbf{A}$ we define the lattice

$$\mathcal{L}(\mathbf{A}) = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{0}, \mathbf{x} \in \mathbb{Z}^n\} \setminus \{\mathbf{0}\}$$

Of which the set of $\sqsubseteq$-minimal elements (which is finite) is the **Graver basis** $\mathcal{G}(A)$

- **Positive Sum Property**
  Every $\mathbf{z} \in \ker(\mathbf{A})$ has a $\sqsubseteq$-representation with respect to $\mathcal{G}(A)$

$$\mathbf{z} = \sum_i \alpha_i g_i, \quad \alpha_i \in \mathbb{Z}_+, \mathbf{g}_i \in \mathcal{G}(A), g_i \sqsubseteq \mathbf{z}$$

Graver and Gröbner basis

Using the CT method, let's remember that from an integer program with constraint matrix $\mathbf{A}$ we can compute a right-hand-side independent toric Ideal $\mathcal{I} \cap \mathbb{Q}[\mathbf{w}] = \mathcal{I}_{\mathbf{A}}$
We denote the Gröbner basis of that toric ideal with respect to an ordering given by the objective $\mathbf{c} \in \mathbb{Z}^n$ as $\mathcal{B}_{>_\mathbf{c}}(\mathbf{A}) = \mathcal{B}(\mathbf{A}, \mathbf{c})$

Now, let's define a **Universal Gröbner basis** as

$$\mathcal{U}(\mathbf{A}) = \bigcup_{c \in \mathbb{Z}^n} \mathcal{B}(\mathbf{A}, \mathbf{c})$$

The **Graver basis** contains, up to negating vector, the **Universal Gröbner basis**

$$\mathcal{U}(\mathbf{A}) = \bigcup_{c \in \mathbb{Z}^n} \mathcal{B}(\mathbf{A}, \mathbf{c}) \subseteq \mathcal{G}(\mathbf{A})$$

- Notice how the **Graver basis** is independent from the objective function
- In certain cases, the **Universal Gröbner basis** and the **Graver basis** are equal
  - If $\mathbf{A}$ is totally unimodular
  - If $\mathbf{A}$ is a Lawrence lifting matrix

Graver and Gröbner basis

Lawrence lifting

Consider the enlarged matrix $\Lambda(\mathbf{A}) = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{I}_n & \mathbf{I}_n \end{bmatrix}$

Its toric ideal is given by $\mathcal{I}_{\Lambda(\mathbf{A})} = \{\mathbf{x}^{\mathbf{u}^+}\mathbf{y}^{\mathbf{u}^-} - \mathbf{x}^{\mathbf{u}^-}\mathbf{y}^{\mathbf{u}^+} : \mathbf{u} \in \ker(\mathbf{A})\}$

Satisfies that: $\mathcal{G}(\Lambda(\mathbf{A})) = \mathcal{U}(\Lambda(\mathbf{A})) = \mathcal{B}(\Lambda(\mathbf{A}), \mathbf{c}), \mathbf{c}$ arbitrary

Therefore we can device an algorithm to compute Graver basis:

Input: $\mathbf{A} \in \mathbb{Z}^{m \times n}$

Output: $\mathcal{G}(\mathbf{A})$

Choose any term order $>$ on $\mathbb{Q}[\mathbf{x}, \mathbf{y}]$

Compute the reduced Gröbner basis $\mathcal{B}$ of $\mathcal{I}_{\Lambda(\mathbf{A})}$ with respect to $>$

Substitute $y_i \mapsto 1$ for any $\mathbf{g} \in \mathcal{B}$

Return $\mathcal{G}$

Graver and Gröbner basis

Lawrence augmentation

https://colab.research.google.com/github/bernalde/QuIP/blob/master/notebooks/Notebook%203%20-%20Graver%20basis.ipynb

Normal form algorithm

Normal form

Below is an algorithm to compute the Normal form $\mathbf{r}$ of an element $\mathbf{s} \in \mathcal{L}$ with respect to a set $\mathcal{G} \subset \mathcal{L}$ such that

$$\mathbf{r} = \text{normalForm}(\mathbf{s}, \mathcal{G}) \in \mathcal{L} \text{ s.t. } \mathbf{s} = \sum_i \alpha_i \mathbf{g}_i + \mathbf{r}, \alpha_i \in \mathbb{Z}_+, \mathbf{g}_i \in \mathcal{G}, \mathbf{r} \sqsubseteq \mathbf{s}; \mathbf{g}_i \not\sqsubseteq \mathbf{r}, \forall \mathbf{g}_i \in \mathcal{G}$$

Normal form Algorithm

Input: vector $\mathbf{s} \in \mathcal{L}$, set $\mathcal{G} \subset \mathcal{L}$

Output: vector $\mathbf{r} = \text{normalForm}(\mathbf{s}, \mathcal{G}) \in \mathcal{L}$

Initialize: $\mathbf{s} \mapsto \mathbf{r}$

WHILE $\exists \mathbf{g} \in \mathcal{G} \text{ s.t. } \mathbf{g} \sqsubseteq \mathbf{r}$

$\mathbf{r} - \mathbf{g} \mapsto \mathbf{r}$

Return $\mathbf{r}$

Notice that this procedure is extremely costly

Pottier's algorithm

Now with the Normal form we can compute the set of $\sqsubseteq$-minimal elements in a lattice $\mathcal{L} \setminus \{\mathbf{0}\}$ (**Graver basis** since we choose $\mathcal{L} = \ker(\mathbf{A})$)

input Generating initial set $F \subseteq \mathcal{L} = \ker \mathbb{N}(A)$
output Graver basis set $G \subseteq \mathcal{L} \setminus \{\mathbf{0}\}$
Initialize symmetric set: $G \leftarrow F \cup (-F)$
Generate S vector set: $C \leftarrow \bigcup_{f,g \in \mathcal{G}} \{f+g\}$
while $C \neq \emptyset$ **do**
  $\forall s \in C \quad : \quad r \leftarrow normalForm(s, \mathcal{G}) \quad and \quad C \leftarrow C \setminus \{s\}$
  if $r \neq 0$ **then**
    Update: $\mathcal{G} \leftarrow \mathcal{G} \cup \{r\} \quad and \quad C \leftarrow C \cup \{r + g_i\} \quad g_i \in \mathcal{G}$
  end if
end while
return $\mathcal{G}$

Drawback: The set $\mathcal{G}$ might contain many elements of $\mathcal{L}$ that are not $\sqsubseteq$-minimal

Project-and-lift algorithm

- Apply Pottier's algorithm to achieve Graver basis on a subset of all variables. All vectors in $\ker(\mathbf{A})$ (in particular all Graver basis elements) can be generated by increasing norm on these variables: **Project phase** $\pi : \mathbb{Z}^n \mapsto \mathbb{Z}^d$
- Apply Pottier's algorithm again, but to all variables.
  - Fewer sums $\mathbf{f} + \mathbf{g}$ have to be considered
    - $\mathbf{f}, \mathbf{g}$ should have the same sign pattern
  - Only the sums $\mathbf{f} + \mathbf{g}$ have to be considered if they fulfill upper bound conditions on the chosen variables

input set F of $\mathcal{L} \subseteq \mathbb{Z}^n$, such that $\pi(F)$ are the $\sqsubseteq$-minimal elements in $\pi(\mathcal{L}) \setminus \{\mathbf{0}\}$.
output set $G \subseteq \mathcal{L}$ containing all $\sqsubseteq$-minimal elements in $\mathcal{L} \setminus \{\mathbf{0}\}$.
$G \leftarrow F$.
$C \leftarrow \bigcup_{\mathbf{f},\mathbf{g} \in G} \{\mathbf{f}+\mathbf{g}\}$.
while $C \neq \emptyset$ **do**
  $\mathbf{s} \leftarrow$ an element in C with $\|\pi(\mathbf{s})\|_1 = \min\{\|\pi(\mathbf{t})\|_1 : \mathbf{t} \in C\}$.
  $C \leftarrow C \setminus \{\mathbf{s}\}$.
  if $\nexists \mathbf{v} \in G$ with $\mathbf{v} \sqsubseteq \mathbf{s}$ **then**
    $C \leftarrow C \cup \{\mathbf{s}+\mathbf{g} : \mathbf{g} \in G$ where $\pi(\mathbf{s})$ and $\pi(\mathbf{g})$ are sign compatible$\}$.
    $G \leftarrow G \cup \{\mathbf{s}\}$.
return G.

The most efficient implementation of this algorithm is available in the software [4ti2](4ti2).

Test-sets and valid objectives

Test-set

Given an integer linear program $\min_{\mathbf{x}} f(\mathbf{x})$ s.t. $\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n$ there exists a finite set denotes test-set $\mathcal{T} = \{\mathbf{t}^1, \cdots, \mathbf{t}^N\}$ that only depends on $\mathbf{A}$, that assures that a feasible solution nonoptimal point $\mathbf{x}_0$ satisfies for some $\alpha \in \mathbb{Z}_+$

- $f(\mathbf{x}_0 + \alpha \mathbf{t}^i) < f(\mathbf{x}_0)$
- $\mathbf{x}_0 + \alpha \mathbf{t}^i$ is feasible

For which objective functions $f(\mathbf{x})$?

- Separable convex minimization: $\sum_i f_i(\mathbf{c}_i^\top \mathbf{x})$ with f_i convex
- Convex integer maximization: $-f(\mathbf{W}\mathbf{x})$ where $\mathbf{W} \in \mathbb{Z}^{d \times n}$ and f convex
- Norm p minimization: $f(\mathbf{x}) = ||\mathbf{x} - \hat{\mathbf{x}}||_p$
- Quadratic minimization: $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{Q} \mathbf{x}$ where $\mathbf{Q}$ lies on the dual of the quadratic Graver cone of $\mathbf{A}$
  - this includes certain nonconvex $\mathbf{Q} \not\succeq 0$
- Polynomial minimization: $f(\mathbf{x}) = P(\mathbf{x})$ where P is a polynomial of degree d , that lies on cone $\mathcal{K}_d(\mathbf{A})$, dual of d^{th} degree Graver cone of $\mathbf{A}$

[1] Alghassi, Hedayat, Raouf Dridi, and Sridhar Tayur. "Graver bases via quantum annealing with application to non-linear integer programs." arXiv preprint arXiv:1902.04215 (2019).

Solution methods for Combinatorial Optimization

Current status and perspectives

Classical methods

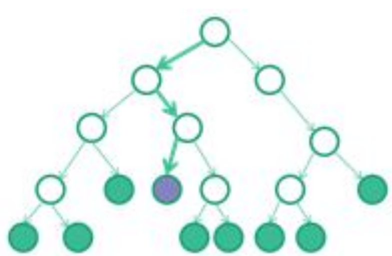
Methods based on divide-and-conquer

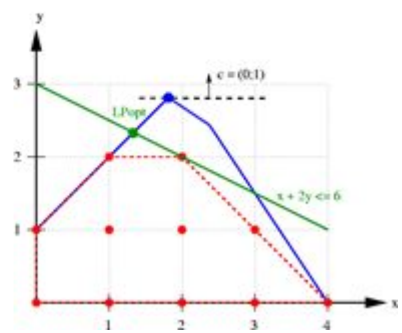
- Branch-and-Bound algorithms
- Harness advances in polyhedral theory
- With global optimality guarantees
- Very efficient codes available
- Exponential complexity

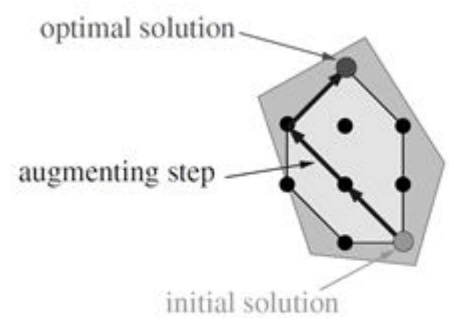
Not very popular classical methods

Methods based on test-sets

- Algorithms based on “augmentation”
- Use tools from algebraic geometry
- Global convergence guarantees
- Very few implementations out there
- Polynomial **oracle** complexity **once we have test-set**







[1] https://de.wikipedia.org/wiki/Branch-and-Cut
[2] Algebraic And geometric ideas in the theory of discrete optimization. De Loera, Hemmecke, Köppe. 2012

Test-set methods - Example

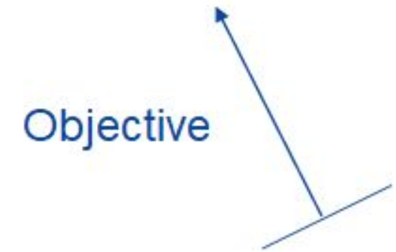
Primal method for Integer Programs

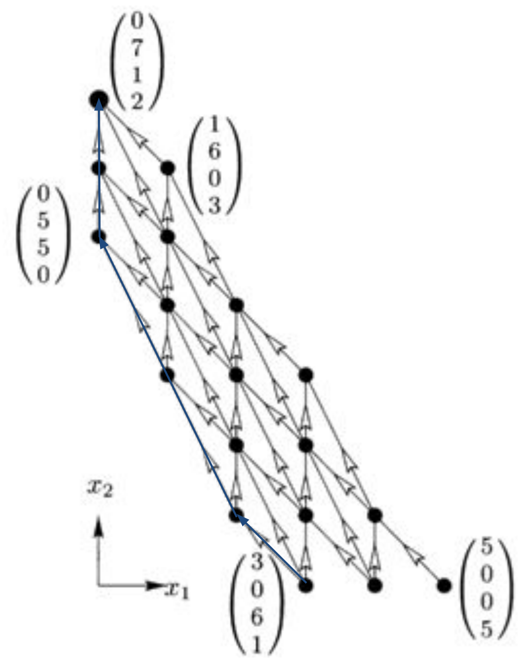
We require:

- An initial feasible solution
- An oracle to compare objective function
- The test-set (set of directions)
- Given a convex objective, the test set will point us a direction where to improve it, and if no improvement, we have the optimal solution.
- The test-set only depends on the constraints and can be computed for equality constraints with integer variables

Example

$$A\mathbf{x} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 10 \\ 15 \end{bmatrix}$$





[1] Gröbner Bases and Integer Programming, G. Ziegler. 1997

[2] Integer Programming (1st ed. 2014) by Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli

Test-set methods - Example

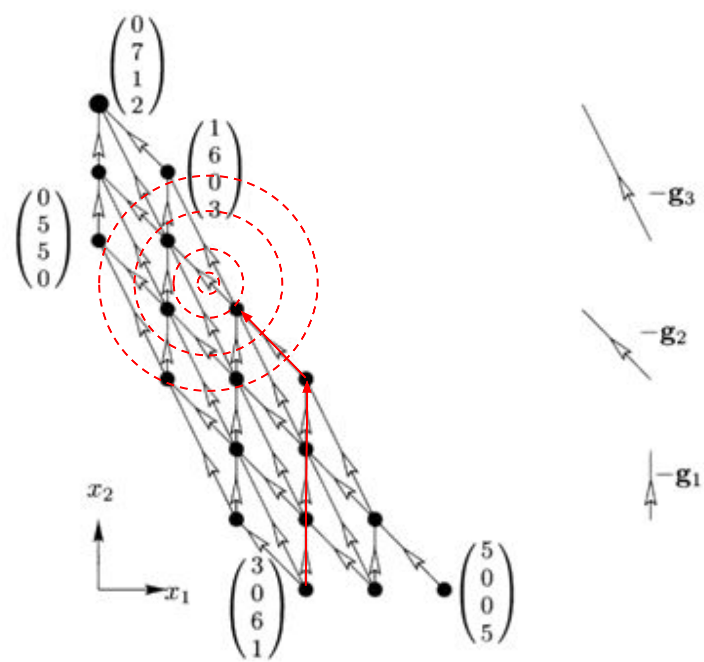
We require:

- An initial feasible solution
- An oracle to compare objective function
- The test-set (set of directions)
- Given a convex objective, the test set will point us a direction where to improve it, and if no improvement, we have the optimal solution.
- The test-set only depends on the constraints and can be computed for equality constraints with integer variables

Example

Objective

$$A\mathbf{x} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 10 \\ 15 \end{bmatrix}$$



[1] Gröbner Bases and Integer Programming, G. Ziegler. 1997

Graver-best augmentation

Graver-best augmentation algorithm;

Data: $A \in \mathbb{Z}^{m \times n}$, $\mathbf{b} \in \mathbb{Z}^m$ $\mathbf{l}$, $\mathbf{u} \in \mathbb{Z}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, a finite test set $\mathcal{T}$ for $IP_{A,\mathbf{b},\mathbf{l},\mathbf{u},f}$, a feasible solution $\mathbf{z}_0$ to $IP_{A,\mathbf{b},\mathbf{l},\mathbf{u},f}$

Result: a optimal solution $\mathbf{z}_{min}$ of $IP_{A,\mathbf{b},\mathbf{l},\mathbf{u},f}$;

while *There are* $\mathbf{t} \in \mathcal{T}, \alpha \in \mathbb{Z}_+$ *with* $\mathbf{z}_0 + \alpha\mathbf{t}$ *feasible and* $f(\mathbf{z}_0 + \alpha\mathbf{t}) < f(\mathbf{z}_0)$ **do**

Among all such pairs $\mathbf{t} \in \mathcal{T}$, $\alpha \in \mathbb{Z}_+$ choose one with $f(\mathbf{z}_0 + \alpha\mathbf{t})$ minimal;

$\mathbf{z}_0 = \mathbf{z}_0 + \alpha\mathbf{t}$;

end

return $\mathbf{z}_0$;

Graver-best augmentation

Graver-best augmentation

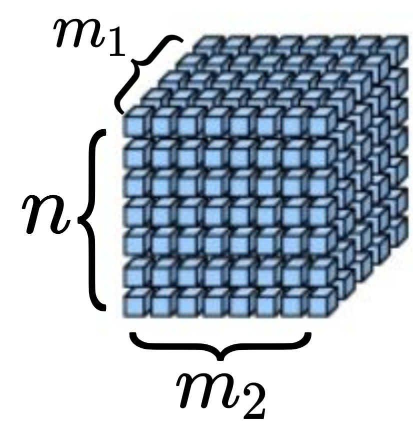
https://colab.research.google.com/github/bernalde/QuIP/blob/master/notebooks/Notebook%203%20-%20Graver%20basis.ipynb

N-fold integer programming

$$\min \{\mathbf{wx} \mid A\mathbf{x} = \mathbf{b},\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u},\ \mathbf{x} \in \mathbb{Z}^n\}$$

$$A = E^{(N)} = \underbrace{\begin{pmatrix} E_1 & E_1 & \cdots & E_1 \\ E_2 & 0 & \cdots & 0 \\ 0 & E_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E_2 \end{pmatrix}}_{n}$$

Multi-index transportation problem of Motzkin. Minimization of cost over $m_1 \times \cdots \times m_{k-1} \times n$ tables with given margins



Applications:

- Multi-commodity flows
- Privacy in statistical databases
- Closest strings determination

Properties:

- **Graver basis** can be computed in **poly-time**

[1] Onn, Shmuel. "Theory and Applications of n-fold Integer Programming." Mixed Integer Nonlinear Programming. Springer, New York, NY, 2012. 559-593.

Universality

- Consider the special for of the n-fold operator $A^{[n]}$ where $E_1 = I$, $E_2 = \mathbf{A}$

$$E^{(N)} = \begin{pmatrix} E_1 & E_1 & \cdots & E_1 \\ E_2 & 0 & \cdots & 0 \\ 0 & E_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E_2 \end{pmatrix}$$

- Now consider the n-fold product of the 1×3 matrix

$$A_{3\times 3} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} I_3 & I_3 & I_3 \\ A & O & O \\ O & A & O \\ O & O & A \end{pmatrix} = (1\ 1\ 1)^{(3)}$$

- Then every (non)-linear integer program

$$\min_{\mathbf{y}} f(\mathbf{y}) \text{ s.t. } \mathbf{A}\mathbf{y} = \mathbf{b}, \mathbf{y} \in \mathbb{Z}_+^n$$

- Can be lifted to a universal n-fold program

$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ s.t. } (1 \quad 1 \quad 1)^{[m][n]}\mathbf{x} = \mathbf{a}, \mathbf{x} \in \mathbb{Z}_+^{m\times n}$$

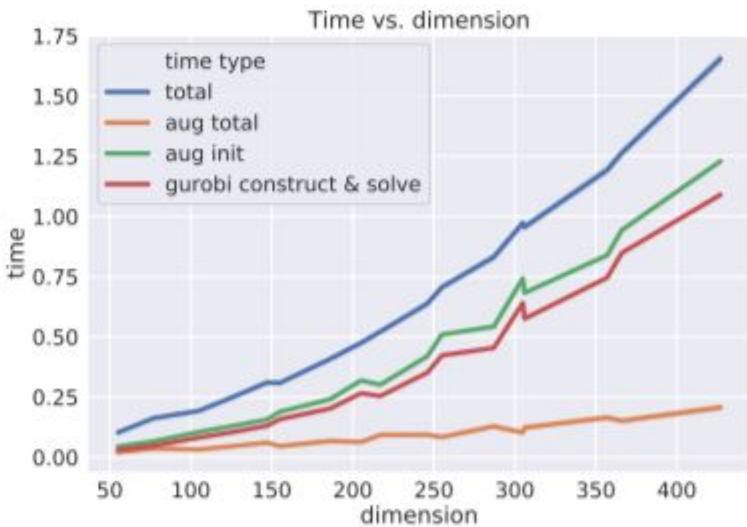
For a fixed m and variable n it is solvable via n-fold programming.

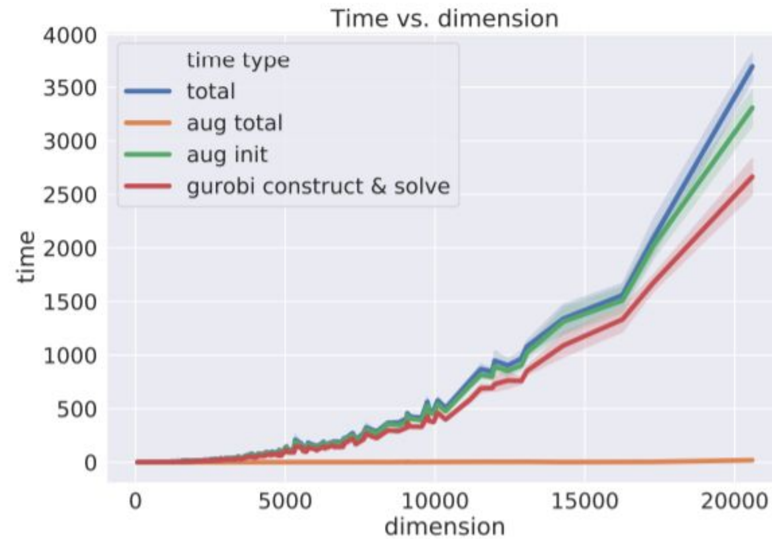
- Poly-time to compute Graver basis, poly-many augmentations to reach optimal solution.

[1] De Loera, Jesús A., and Shmuel Onn. "All linear and integer programs are slim 3-way transportation programs." SIAM Journal on Optimization 17.3 (2006): 806-821.

Graver vs. IP solvers in N-fold

Computing Graver basis for N-fold programs (although poly-time) is really complicated.
Here they used an approximate to compute the Graver basis (via IP actually).





- aug total is the augmentation procedure (it stays almost constant and grows slowly).
- aug init is their procedure to compute Graver basis.
- If they only had a way of computing Graver more efficiently! (Quantum or Analytical).
- Even for n-fold problems where computing the Graver basis can be done in poly-time, it is really hard (mainly because of memory)

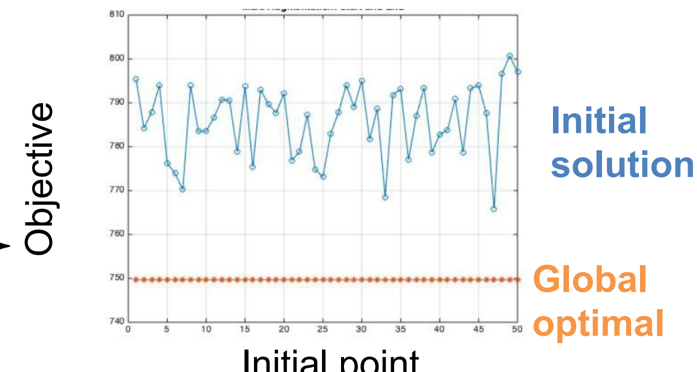
[1] https://arxiv.org/pdf/1802.09007.pdf

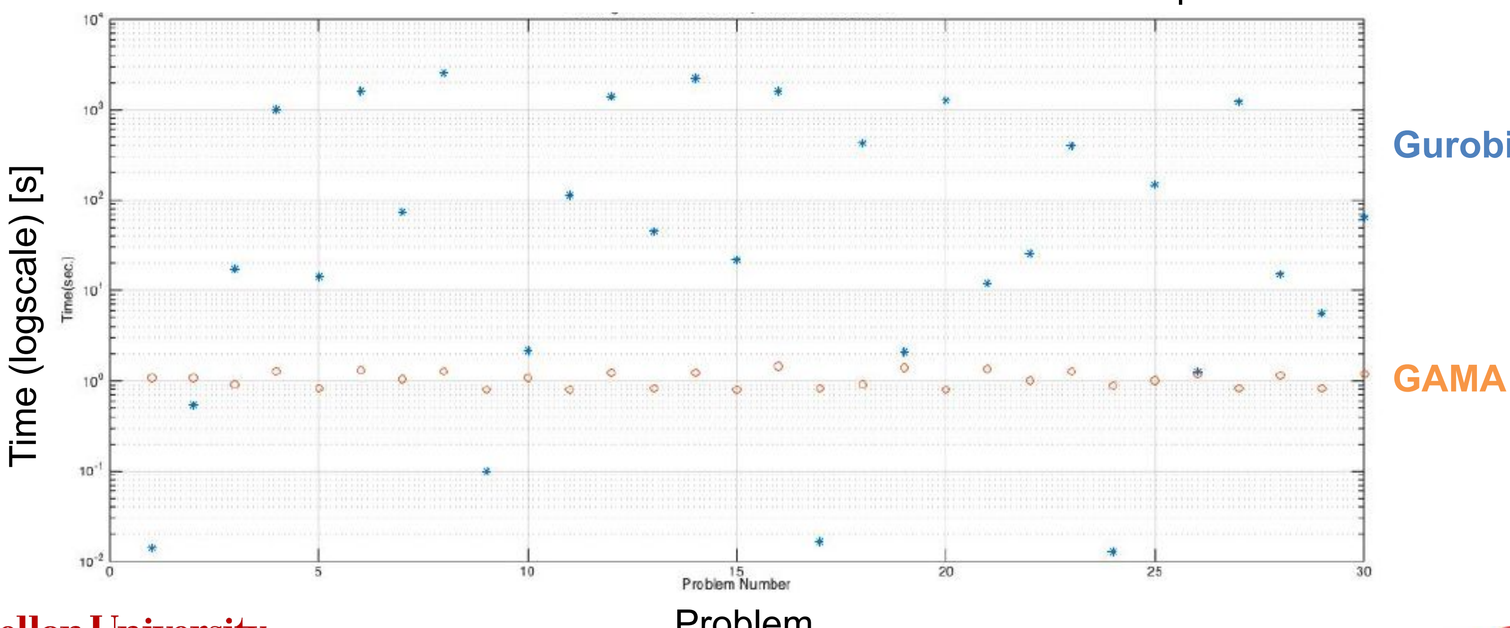
Graver vs. IP solvers in Quadratic cardinality boolean

$$\min\left\{\mathbf{c}^T\mathbf{x} + \mathbf{x}^T Q\mathbf{x} : \mathbf{1}_n^T\mathbf{x} = b\right\}$$

For this problem Graver basis can be computed analytically

- For convex case, regardless of initial point you converge to global optimal solution
- For nonconvex, you initialize at several points:
    - GAMA algorithm!





[1] https://arxiv.org/abs/1907.10930

Take-home message

- Computing test-sets classically is rough!
- Naive use of Graver Basis even in Ising/Quantum is not the goal
- Tailored formulations and decompositions that exploit the structure
- What can be done with “partial” Graver Basis?
- Are there special structured matrices that allow for systematic Graver Basis calculation?
- Best “use-cases” have complex objective functions (example: higher moments in Portfolio optimization) and/or highly non-convex or only Oracle calls that defy commercially available classical solvers