New Geometric Algorithms in Discrete Optimization

Jesús A. De Loera, UC Davis

new results on several papers joint work with (subsets of):

M Köppe (UC Davis),
U. Rothblum & S. Onn (Technion Haifa),
R. Hemmecke & R. Weismantel (U. Magdeburg)

November 1, 2009

Le Menu

- Appetizer: Motivation
  - Review: Geometric Algorithms in LINEAR Discrete Optimization.
  - Question: How about NON-LINEAR Constraints or Objective function?
- Main Course: Discrete Optimization with Non-linear Objective Function.
  - Barvinok's Algorithm.
  - Graver Bases.
- Dessert: Closing Comments.

Le Menu

- Appetizer: Motivation
  - Review: Geometric Algorithms in LINEAR Discrete Optimization.
  - Question: How about NON-LINEAR Constraints or Objective function?
- Main Course: Discrete Optimization with Non-linear Objective Function.
  - Barvinok's Algorithm.
  - Graver Bases.
- Dessert: Closing Comments.

Le Menu

- Appetizer: Motivation
  - Review: Geometric Algorithms in LINEAR Discrete Optimization.
  - Question: How about NON-LINEAR Constraints or Objective function?
- Main Course: Discrete Optimization with Non-linear Objective Function.
  - Barvinok's Algorithm.
  - Graver Bases.
- Dessert: Closing Comments.

Le Menu

- Appetizer: Motivation
  - Review: Geometric Algorithms in LINEAR Discrete Optimization.
  - Question: How about NON-LINEAR Constraints or Objective function?
- Main Course: Discrete Optimization with Non-linear Objective Function.
  - Barvinok's Algorithm.
  - Graver Bases.
- Dessert: Closing Comments.

Le Menu

- Appetizer: Motivation
  - Review: Geometric Algorithms in LINEAR Discrete Optimization.
  - Question: How about NON-LINEAR Constraints or Objective function?
- Main Course: Discrete Optimization with Non-linear Objective Function.
  - Barvinok's Algorithm.
  - Graver Bases.
- Dessert: Closing Comments.

Le Menu

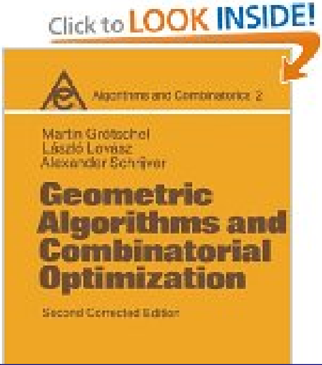
- Appetizer: Motivation
  - Review: Geometric Algorithms in LINEAR Discrete Optimization.
  - Question: How about NON-LINEAR Constraints or Objective function?
- Main Course: Discrete Optimization with Non-linear Objective Function.
  - Barvinok's Algorithm.
  - Graver Bases.
- Dessert: Closing Comments.

a Historical Overture...

Overview LINEAR Discrete Optimization



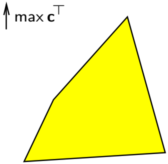
Efficient computation with Convex Sets & Lattices $\Longleftrightarrow$ Efficient Optimization



At the beginning of time there was ILP...

Linear programs

$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \end{array}$$



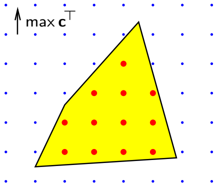
Special integer programs

$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$

Matrix A is SPECIAL!

Integer linear programs

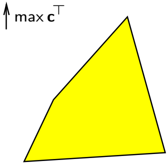
$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$



At the beginning of time there was ILP...

Linear programs

$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \end{aligned}$$



Ellipsoid Method

LP is polynomial-time solvable (L. Khachiyan)

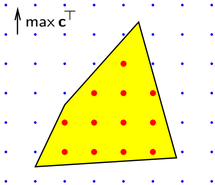
Special integer programs

$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$

Matrix A is SPECIAL!

Integer linear programs

$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$

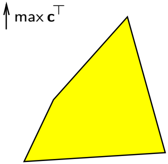


Hard

(NP-hard)

Linear programs

$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \end{array}$$



Ellipsoid Method

LP is polynomial-time solvable (L. Khachiyan)

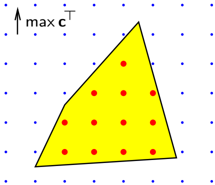
Special integer programs

$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$

Matrix A is SPECIAL!

Integer linear programs

$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$



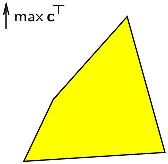
Hard

(NP-hard)

At the beginning of time there was ILP...

Linear programs

$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{array}$$



Ellipsoid Method

LP is polynomial-time solvable (L. Khachiyan)

Special integer programs

$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$

Matrix A is SPECIAL!

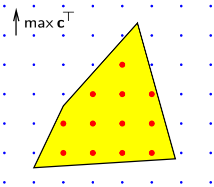
Lattice Basis Reduction

Fixed dimension ILPs (Lenstra)

Network flow ILPs (Frank-Tardos)

Integer linear programs

$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$



Hard

(NP-hard)

QUESTION:

How can we handle NON-LINEAR

Constraints and Objective functions???

Reality is NON-LINEAR

Non-linear Integer Optimization

$$\max/\min\ f(x_1, \ldots, x_d)$$

$$\text{subject to } g_j(x_1, \ldots, x_d) \leq 0,$$

for $j = 1 \ldots s$, and with

with x_i integer

- Here f and g_j's could be NON LINEAR polynomials.
- Model has huge power to fit various situations, but
- WHAT CAN BE PROVED IN THIS GENERAL CONTEXT??

Prior Work

- **Theorem:** For fixed number of variables AND convex polynomials f, g_i problem can be solved in polynomial time

  (Khachiyan and Porkolab, 2000)
- Nevertheless, the problem is INCREDIBLY HARD

  Theorem It is UNDECIDABLE already when f, g_i's are arbitrary polynomials (Jeroslow, 1979).
- EVEN WORSE

  Proposition: It undecidable even with number of variables=10.

We need to make further assumptions to prove something...

Reality is NON-LINEAR

Non-linear Integer Optimization

$$\max/\min\ f(x_1, \dots, x_d)$$
$$\text{subject to } g_j(x_1, \dots, x_d) \leq 0,$$

for $j = 1 \dots s$, and with

with x_i integer

- Here f and g_j's could be NON LINEAR polynomials.
- Model has huge power to fit various situations, but
- WHAT CAN BE PROVED IN THIS GENERAL CONTEXT??

Prior Work

- **Theorem:** For fixed number of variables AND convex polynomials f, g_i problem can be solved in polynomial time

  (Khachiyan and Porkolab, 2000)
- Nevertheless, the problem is INCREDIBLY HARD

  Theorem It is UNDECIDABLE already when f, g_i's are arbitrary polynomials (Jeroslow, 1979).
- EVEN WORSE

  Proposition: It undecidable even with number of variables=10.

We need to make further assumptions to prove something...

Reality is NON-LINEAR

Non-linear Integer Optimization

$$\max/\min f(x_1,\ldots,x_d)$$

$$\text{subject to } g_j(x_1,\ldots,x_d) \leq 0,$$

for $j = 1\ldots s$, and with

with x_i integer

- Here f and g_j's could be NON LINEAR polynomials.
- Model has huge power to fit various situations, but
- WHAT CAN BE PROVED IN THIS GENERAL CONTEXT??

Prior Work

- **Theorem:** For fixed number of variables AND convex polynomials f, g_i problem can be solved in polynomial time

  (Khachiyan and Porkolab, 2000)
- Nevertheless, the problem is INCREDIBLY HARD

  Theorem It is UNDECIDABLE already when f,g_i's are arbitrary polynomials (Jeroslow, 1979).
- EVEN WORSE

  Proposition: It undecidable even with number of variables=10.

We need to make further assumptions to prove something...

Reality is NON-LINEAR

Non-linear Integer Optimization

$$\max/\min \ f(x_1, \dots, x_d)$$

$$\text{subject to } g_j(x_1, \dots, x_d) \leq 0,$$

for $j = 1 \dots s$, and with

with x_i integer

- Here f and g_j's could be NON LINEAR polynomials.
- Model has huge power to fit various situations, but
- WHAT CAN BE PROVED IN THIS GENERAL CONTEXT??

Prior Work

- **Theorem:** For fixed number of variables AND convex polynomials f, g_i problem can be solved in polynomial time

  (Khachiyan and Porkolab, 2000)
- Nevertheless, the problem is INCREDIBLY HARD

  Theorem It is UNDECIDABLE already when f,g_i's are arbitrary polynomials (Jeroslow, 1979).
- EVEN WORSE

  Proposition: It undecidable even with number of variables=10.

We need to make further assumptions to prove something...

Reality is NON-LINEAR

Non-linear Integer Optimization

$$\max/\min f(x_1,\ldots,x_d)$$

$$\text{subject to } g_j(x_1,\ldots,x_d) \leq 0,$$

for $j = 1 \ldots s$, and with

with x_i integer

- Here f and g_j's could be NON LINEAR polynomials.
- Model has huge power to fit various situations, but
- WHAT CAN BE PROVED IN THIS GENERAL CONTEXT??

Prior Work

- **Theorem:** For fixed number of variables AND convex polynomials f, g_i problem can be solved in polynomial time

  (Khachiyan and Porkolab, 2000)
- Nevertheless, the problem is INCREDIBLY HARD

  Theorem It is UNDECIDABLE already when f,g_i's are arbitrary polynomials (Jeroslow, 1979).
- EVEN WORSE

  Proposition: It undecidable even with number of variables=10.

We need to make further assumptions to prove something...

How about polyhedral constraints non-linear objective??

Let f be a multivariate polynomial function,

$$\begin{aligned} \max \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{Ax} \leq \mathbf{b} \end{aligned}$$

Special programs

$$\begin{aligned} \max \quad & \mathbf{f}(x) \\ \text{s.t.} \quad & \mathbf{Ax} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$

Matrix A is SPECIAL!

$$\begin{aligned} \max \quad & \mathbf{f}(x) \\ \text{s.t.} \quad & \mathbf{Ax} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$

How about polyhedral constraints non-linear objective??

Let f be a multivariate polynomial function,

$$\begin{aligned} \max \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{aligned}$$

Hard
(NP-hard)

Special programs

$$\begin{aligned} \max \quad & \mathbf{f}(x) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$

Matrix A is SPECIAL!

$$\begin{aligned} \max \quad & \mathbf{f}(x) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$

Hard
(NP-hard)

How about polyhedral constraints non-linear objective??

Let f be a multivariate polynomial function,

$$\begin{array}{ll} \max & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \end{array}$$

Hard
(NP-hard)

Special programs

$$\begin{array}{ll} \max & \mathbf{f}(x) \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$

Matrix A is SPECIAL!

$$\begin{array}{ll} \max & \mathbf{f}(x) \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$

Hard
(NP-hard)

Let f be a multivariate polynomial function,

$$\begin{array}{ll} \max & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \end{array}$$

Hard
(NP-hard)

Special programs

$$\begin{array}{ll} \max & \mathbf{f}(x) \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$

Matrix A is SPECIAL!

$$\begin{array}{ll} \max & \mathbf{f}(x) \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$

Hard
(NP-hard)

???

RESULTS

Two Algorithms for Non-Linear Optimization over the Lattice Points of Polyhedra

How about polyhedral constraints non-linear objective??

THEOREM

Let $P = \{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ be a convex polytope. Let f is a multivariate polynomial of total degree D with integer coefficients and non-negative over $P \cap \mathbf{Z}^d$. Then, given the problem

$$\begin{aligned} \max \quad & f(x_1, \ldots, x_d) \\ \text{subject to} \quad & (x_1, \ldots, x_d) \in P \cap \mathbf{Z}^d, \end{aligned}$$

- maximizing an arbitrary degree-4 polynomial f for two variable problems is NP-hard.
- For every $\epsilon > 0$, there exists an algorithm $\mathcal{A}_\epsilon$ with running time polynomial in the input size and $1/\epsilon$, which computes an approximation $\mathbf{x}_\epsilon$ with

$$|f(\mathbf{x}_\epsilon) - f(\mathbf{x}^{\max})| \leq \epsilon f(\mathbf{x}^{\max}).$$

  where $\mathbf{x}^{\max}$ denotes an optimal solution of the optimization problem.

Note: Compare to Integer Linear Programming solvable in polynomial time (H. W. Lenstra Jr, 1983)

How about polyhedral constraints non-linear objective??

THEOREM

Let $P = \{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ be a convex polytope. Let f is a multivariate polynomial of total degree D with integer coefficients and non-negative over $P \cap \mathbf{Z}^d$. Then, given the problem

$$\begin{aligned} \max \quad & f(x_1, \ldots, x_d) \\ \text{subject to} \quad & (x_1, \ldots, x_d) \in P \cap \mathbf{Z}^d, \end{aligned}$$

- maximizing an arbitrary degree-4 polynomial f for two variable problems is NP-hard.
- For every $\epsilon > 0$, there exists an algorithm $\mathcal{A}_\epsilon$ with running time polynomial in the input size and $1/\epsilon$, which computes an approximation $\mathbf{x}_\epsilon$ with

$$|f(\mathbf{x}_\epsilon) - f(\mathbf{x}^{\max})| \leq \epsilon f(\mathbf{x}^{\max}).$$

  where $\mathbf{x}^{\max}$ denotes an optimal solution of the optimization problem.

Note: Compare to Integer Linear Programming solvable in polynomial time (H. W. Lenstra Jr, 1983)

How about polyhedral constraints non-linear objective??

THEOREM

Let $P = \{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ be a convex polytope. Let f is a multivariate polynomial of total degree D with integer coefficients and non-negative over $P \cap \mathbf{Z}^d$. Then, given the problem

$$\begin{aligned} \text{max} \quad & f(x_1, \ldots, x_d) \\ \text{subject to} \quad & (x_1, \ldots, x_d) \in P \cap \mathbf{Z}^d, \end{aligned}$$

- maximizing an arbitrary degree-4 polynomial f for two variable problems is NP-hard.
- For every $\epsilon > 0$, there exists an algorithm $\mathcal{A}_\epsilon$ with running time polynomial in the input size and $1/\epsilon$, which computes an approximation $\mathbf{x}_\epsilon$ with

$$|f(\mathbf{x}_\epsilon) - f(\mathbf{x}^{\max})| \leq \epsilon f(\mathbf{x}^{\max}).$$

where $\mathbf{x}^{\max}$ denotes an optimal solution of the optimization problem.

Note: Compare to Integer Linear Programming solvable in polynomial time (H. W. Lenstra Jr, 1983)

How about polyhedral constraints non-linear objective??

THEOREM

Let $P = \{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ be a convex polytope. Let f is a multivariate polynomial of total degree D with integer coefficients and non-negative over $P \cap \mathbf{Z}^d$. Then, given the problem

$$\begin{aligned} \max \quad & f(x_1, \ldots, x_d) \\ \text{subject to} \quad & (x_1, \ldots, x_d) \in P \cap \mathbf{Z}^d, \end{aligned}$$

- maximizing an arbitrary degree-4 polynomial f for two variable problems is NP-hard.
- For every $\epsilon > 0$, there exists an algorithm $\mathcal{A}_\epsilon$ with running time polynomial in the input size and $1/\epsilon$, which computes an approximation $\mathbf{x}_\epsilon$ with

$$|f(\mathbf{x}_\epsilon) - f(\mathbf{x}^{\max})| \leq \epsilon f(\mathbf{x}^{\max}).$$

  where $\mathbf{x}^{\max}$ denotes an optimal solution of the optimization problem.

Note: Compare to Integer Linear Programming solvable in polynomial time (H. W. Lenstra Jr, 1983)

Integer Optimization over 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,

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

N-fold systems DO appear in applications! Transportation problems with fixed number of suppliers are examples!

Theorem

Fix two 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, an integer vectors b, a cost vector c, and a convex function f, solves the corresponding n-fold integer programming problem.

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

Integer Optimization over 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,

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

N-fold systems DO appear in applications! Transportation problems with fixed number of suppliers are examples!

Theorem

Fix two 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, an integer vectors b, a cost vector c, and a convex function f, solves the corresponding n-fold integer programming problem.

$$\max\{f(cx) : \; [A, B]^{(n)}x = b, \; x \in \mathbf{N}^{nq}\} \quad .$$

Integer Optimization over 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,

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

N-fold systems DO appear in applications! Transportation problems with fixed number of suppliers are examples!

Theorem

Fix two 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, an integer vectors b, a cost vector c, and a convex function f, solves the corresponding n-fold integer programming problem.

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

Key Message

We need creative representations

of the Lattice points of polyhedra

We need COMBINATORIAL GEOMETRY...

A FIRST ALGORITHM: Barvinok's generating functions

Generating functions for Lattice Points

Think of the lattice points as monomials, e.g., $(7, 4, -3)$ is $z_1^7 z_2^4 z_3^{-3}$. Define

$$\text{For } K \subset \mathbf{R}^d \quad g_K(z) := \sum_{\alpha \in K \cap \mathbf{Z}^d} z_1^{\alpha_1} z_2^{\alpha_2} \ldots z_n^{\alpha_n}.$$

EXAMPLE: for the line segment $P = [0, M]$,
$g_P(z) = z^0 + z^1 + z^2 + z^3 + \ldots z^M = \frac{1-z^{M+1}}{1-z} \quad \text{for } z \neq 1$

Theorem (Alexander Barvinok, 1994)

Let P be a polyhedron, given by rational inequalities, of fixed dimension d. There is a polynomial-time algorithm for computing function

$$g_P(z_1, \ldots, z_d) = \sum_{(\alpha_1,\ldots,\alpha_d) \in P \cap \mathbf{Z}^d} z_1^{\alpha_1} \cdots z_d^{\alpha_d} = \sum_{\alpha \in P \cap \mathbf{Z}^d} \mathbf{z}^\alpha$$

in the form of a rational function.

Corollary

In particular,

$$N = |P \cap \mathbf{Z}^d| = g_P(\mathbf{1})$$

can be computed in polynomial time (in fixed dimension).

A FIRST ALGORITHM: Barvinok's generating functions

Generating functions for Lattice Points

Think of the lattice points as monomials, e.g., $(7, 4, -3)$ is $z_1^7 z_2^4 z_3^{-3}$. Define

$$\text{For } K \subset \mathbf{R}^d \quad g_K(z) := \sum_{\alpha \in K \cap \mathbf{Z}^d} z_1^{\alpha_1} z_2^{\alpha_2} \dots z_n^{\alpha_n}.$$

EXAMPLE: for the line segment $P = [0, M]$,
$g_P(z) = z^0 + z^1 + z^2 + z^3 + \dots z^M = \frac{1-z^{M+1}}{1-z}$ for $z \neq 1$

Theorem (Alexander Barvinok, 1994)

Let P be a polyhedron, given by rational inequalities, of fixed dimension d. There is a polynomial-time algorithm for computing function

$$g_P(z_1, \dots, z_d) = \sum_{(\alpha_1,\dots,\alpha_d) \in P \cap \mathbf{Z}^d} z_1^{\alpha_1} \cdots z_d^{\alpha_d} = \sum_{\alpha \in P \cap \mathbf{Z}^d} \mathbf{z}^\alpha$$

in the form of a rational function.

Corollary

In particular,

$$N = |P \cap \mathbf{Z}^d| = g_P(\mathbf{1})$$

can be computed in polynomial time (in fixed dimension).

A FIRST ALGORITHM: Barvinok's generating functions

Generating functions for Lattice Points

Think of the lattice points as monomials, e.g., $(7, 4, -3)$ is $z_1^7 z_2^4 z_3^{-3}$. Define

$$\text{For } K \subset \mathbf{R}^d \quad g_K(z) := \sum_{\alpha \in K \cap \mathbf{Z}^d} z_1^{\alpha_1} z_2^{\alpha_2} \ldots z_n^{\alpha_n}.$$

EXAMPLE: for the line segment $P = [0, M]$,
$g_P(z) = z^0 + z^1 + z^2 + z^3 + \ldots z^M = \frac{1-z^{M+1}}{1-z}$ for $z \neq 1$

Theorem (Alexander Barvinok, 1994)

Let P be a polyhedron, given by rational inequalities, of fixed dimension d. There is a polynomial-time algorithm for computing function

$$g_P(z_1, \ldots, z_d) = \sum_{(\alpha_1, \ldots, \alpha_d) \in P \cap \mathbf{Z}^d} z_1^{\alpha_1} \cdots z_d^{\alpha_d} = \sum_{\alpha \in P \cap \mathbf{Z}^d} \mathbf{z}^\alpha$$

in the form of a rational function.

Corollary

In particular,

$$N = |P \cap \mathbf{Z}^d| = g_P(\mathbf{1})$$

can be computed in polynomial time (in fixed dimension).

A FIRST ALGORITHM: Barvinok's generating functions

Generating functions for Lattice Points

Think of the lattice points as monomials, e.g., $(7, 4, -3)$ is $z_1^7 z_2^4 z_3^{-3}$. Define

$$\text{For } K \subset \mathbf{R}^d \quad g_K(z) := \sum_{\alpha \in K \cap \mathbf{Z}^d} z_1^{\alpha_1} z_2^{\alpha_2} \dots z_n^{\alpha_n}.$$

EXAMPLE: for the line segment $P = [0, M]$,
$g_P(z) = z^0 + z^1 + z^2 + z^3 + \dots z^M = \frac{1-z^{M+1}}{1-z}$ for $z \neq 1$

Theorem (Alexander Barvinok, 1994)

Let P be a polyhedron, given by rational inequalities, of fixed dimension d. There is a polynomial-time algorithm for computing function

$$g_P(z_1, \dots, z_d) = \sum_{(\alpha_1, \dots, \alpha_d) \in P \cap \mathbf{Z}^d} z_1^{\alpha_1} \cdots z_d^{\alpha_d} = \sum_{\alpha \in P \cap \mathbf{Z}^d} \mathbf{z}^\alpha$$

in the form of a rational function.

Corollary

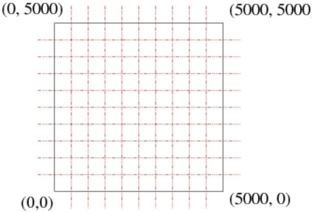
In particular,

$$N = |P \cap \mathbf{Z}^d| = g_P(\mathbf{1})$$

can be computed in polynomial time (in fixed dimension).

Example

Let P be the square with vertices $V_1 = (0, 0)$, $V_2 = (5000, 0)$, $V_3 = (5000, 5000)$, and $V_4 = (0, 5000)$.



The generating function $f(P)$ has over 25,000,000 monomials,
$f(P) = 1 + z_1 + z_2 + z_1^1 z_2^2 + z_1^2 z_2 + \cdots + z_1^{5000} z_2^{5000}$,

But it can be written using only four rational functions

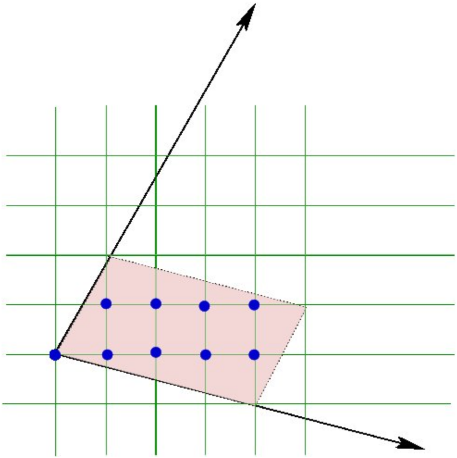
$$\frac{1}{(1-z_1)(1-z_2)} + \frac{{z_1}^{5000}}{(1-{z_1}^{-1})(1-z_2)} + \frac{{z_2}^{5000}}{(1-{z_2}^{-1})(1-z_1)} + \frac{{z_1}^{5000}{z_2}^{5000}}{(1-{z_1}^{-1})(1-{z_2}^{-1})}$$

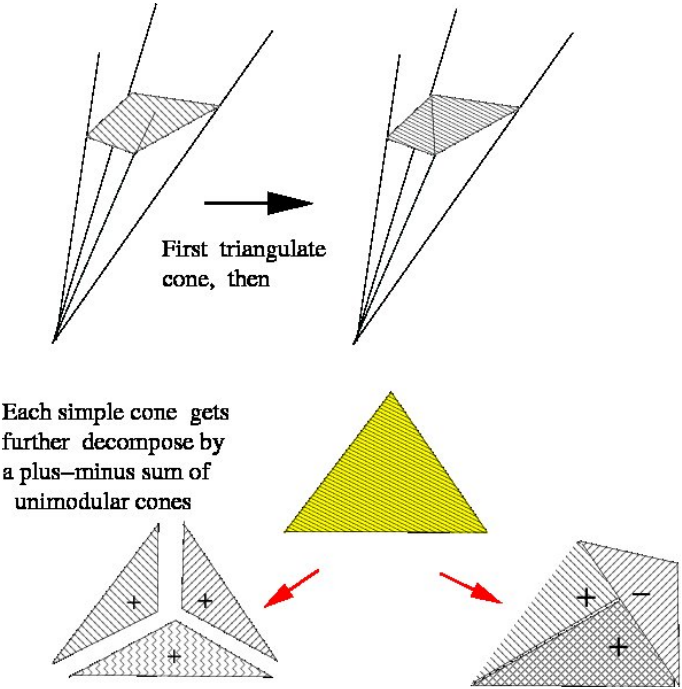
Also, $f(tP, z)$ is

$$\frac{1}{(1-z_1)(1-z_2)} + \frac{{z_1}^{5000 \cdot t}}{(1-{z_1}^{-1})(1-z_2)} + \frac{{z_2}^{5000 \cdot t}}{(1-{z_2}^{-1})(1-z_1)} + \frac{{z_1}^{5000 \cdot t}{z_2}^{5000 \cdot t}}{(1-{z_1}^{-1})(1-{z_2}^{-1})}$$

EXAMPLE: we have $d = 2$ and $c_1 = (1, 2)$, $c_2 = (4, -1)$. We have:

$$f(K) = \frac{z_1^4 z_2 + z_1^3 z_2 + z_1^2 z_2 + z_1 z_2 + z_1^4 + z_1^3 + z_1^2 + z_1 + 1}{(1 - z_1 z_2^2)(1 - z_1^4 z_2^{-1})}.$$




First triangulate
cone, then
Each simple cone gets
further decompose by
a plus–minus sum of
unimodular cones
+
+
+
+
−
+

Differential operators on generating functions

The Euler differential operator $\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)$ maps:

$$g(z) = \sum_{j=0}^{D} g_j z^j \quad \longmapsto \quad z\frac{\mathrm{d}}{\mathrm{d}z} g(z) = \sum_{j=0}^{D} (j \cdot g_j) z^j$$

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + z^4$$

Apply differential operator:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 2z^2 + 3z^3 + 4z^4$$

Apply differential operator again:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 4z^2 + 9z^3 + 16z^4$$

Differential operators on generating functions

The Euler differential operator $\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)$ maps:

$$g(z) = \sum_{j=0}^{D} g_j z^j \quad \longmapsto \quad z\frac{\mathrm{d}}{\mathrm{d}z} g(z) = \sum_{j=0}^{D} (j \cdot g_j) z^j$$

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + z^4$$

Apply differential operator:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 2z^2 + 3z^3 + 4z^4$$

Apply differential operator again:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 4z^2 + 9z^3 + 16z^4$$

Differential operators on generating functions

The Euler differential operator $\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)$ maps:

$$g(z) = \sum_{j=0}^{D} g_j z^j \quad \longmapsto \quad z\frac{\mathrm{d}}{\mathrm{d}z} g(z) = \sum_{j=0}^{D} (j \cdot g_j) z^j$$

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + z^4 = \frac{1}{1-z} - \frac{z^5}{1-z}$$

Apply differential operator:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 2z^2 + 3z^3 + 4z^4 = \frac{1}{(1-z)^2} - \frac{-4z^5 + 5z^4}{(1-z)^2}$$

Apply differential operator again:

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g_P(z) = 1z^1 + 4z^2 + 9z^3 + 16z^4 = \frac{z + z^2}{(1-z)^3} - \frac{25z^5 - 39z^6 + 16z^7}{(1-z)^3}$$

Differential operators on generating functions

Lemma

$$f(x_1,\dots,x_d)=\sum_{\boldsymbol\beta} c_{\boldsymbol\beta}\mathbf{x}^{\boldsymbol\beta}\in \mathbf{Z}[x_1,\dots,x_d]$$

can be converted to a differential operator

$$D_f=f\left(z_1\frac{\partial}{\partial z_1},\dots,z_d\frac{\partial}{\partial z_d}\right)=\sum_{\boldsymbol\beta} c_{\boldsymbol\beta}\left(z_1\frac{\partial}{\partial z_1}\right)^{\beta_1}\cdots\left(z_d\frac{\partial}{\partial z_d}\right)^{\beta_d}$$

which maps

$$g(\mathbf{z})=\sum_{\boldsymbol\alpha\in S}\mathbf{z}^{\boldsymbol\alpha}\quad\longmapsto\quad (D_f g)(\mathbf{z})=\sum_{\boldsymbol\alpha\in S} f(\boldsymbol\alpha)\mathbf{z}^{\boldsymbol\alpha}.$$

Theorem

Let $g_P(\mathbf{z})$ be the Barvinok generating function of the lattice points of P. Let f be a polynomial in $\mathbf{Z}[x_1,\dots,x_d]$ of maximum total degree D.
We can compute, in time polynomial in D and the size of the input data, a Barvinok rational function representation $g_{P,f}(\mathbf{z})$ for $\sum_{\boldsymbol\alpha\in P\cap\mathbf{Z}^d} f(\boldsymbol\alpha)\mathbf{z}^{\boldsymbol\alpha}$.

Differential operators on generating functions

Lemma

$$f(x_1, \ldots, x_d) = \sum_{\boldsymbol{\beta}} c_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}} \in \mathbf{Z}[x_1, \ldots, x_d]$$

can be converted to a differential operator

$$D_f = f\left(z_1 \frac{\partial}{\partial z_1}, \ldots, z_d \frac{\partial}{\partial z_d}\right) = \sum_{\boldsymbol{\beta}} c_{\boldsymbol{\beta}} \left(z_1 \frac{\partial}{\partial z_1}\right)^{\beta_1} \ldots \left(z_d \frac{\partial}{\partial z_d}\right)^{\beta_d}$$

which maps

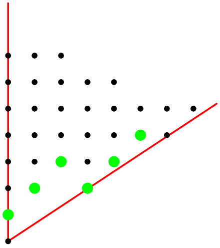
$$g(\mathbf{z}) = \sum_{\boldsymbol{\alpha} \in S} \mathbf{z}^{\boldsymbol{\alpha}} \quad \longmapsto \quad (D_f g)(\mathbf{z}) = \sum_{\boldsymbol{\alpha} \in S} f(\boldsymbol{\alpha}) \mathbf{z}^{\boldsymbol{\alpha}}.$$

Theorem

Let $g_P(\mathbf{z})$ be the Barvinok generating function of the lattice points of P. Let f be a polynomial in $\mathbf{Z}[x_1, \ldots, x_d]$ of maximum total degree D.
We can compute, in time polynomial in D and the size of the input data, a Barvinok rational function representation $g_{P,f}(\mathbf{z})$ for $\sum_{\boldsymbol{\alpha} \in P \cap \mathbf{Z}^d} f(\boldsymbol{\alpha}) \mathbf{z}^{\boldsymbol{\alpha}}$.

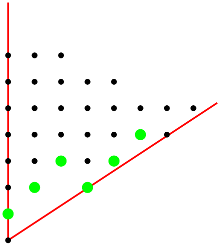
A SECOND ALGORITHM: Graver Bases

- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b,\ x \geq 0\}$. We will use Computational Geometry and Algebra.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$. u is *conformally smaller* than v, denoted $u \sqsubset v$, if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for $i = 1, \ldots, n$.
  Eg: $(3, -2, -8, 0, 8) \sqsubset (4, -3, -9, 0, 9)$, incomparable to $(-4, -3, 9, 1, -8)$.



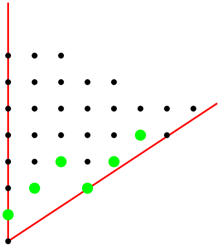
A SECOND ALGORITHM: Graver Bases

- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b,\ x \geq 0\}$. We will use Computational Geometry and Algebra.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$. u is *conformally smaller* than v, denoted $u \sqsubseteq v$, if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for $i = 1, \ldots, n$.
  Eg: $(3, -2, -8, 0, 8) \sqsubseteq (4, -3, -9, 0, 9)$, incomparable to $(-4, -3, 9, 1, -8)$.

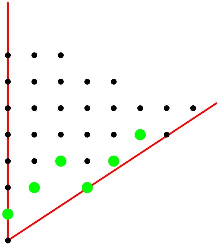


A SECOND ALGORITHM: Graver Bases

- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b, \ x \geq 0\}$. We will use Computational Geometry and Algebra.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$. u is *conformally smaller* than v, denoted $u \sqsubset v$, if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for $i = 1, \ldots, n$.
  Eg: $(3, -2, -8, 0, 8) \sqsubset (4, -3, -9, 0, 9)$, incomparable to $(-4, -3, 9, 1, -8)$.



- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b,\ x \geq 0\}$. We will use Computational Geometry and Algebra.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$. u is *conformally smaller* than v, denoted $u \sqsubset v$, if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for $i = 1, \ldots, n$.
  Eg: $(3, -2, -8, 0, 8) \sqsubset (4, -3, -9, 0, 9)$, incomparable to $(-4, -3, 9, 1, -8)$.



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

- **Example:** If $A = [1\ 2\ 1]$ then its Graver basis is

$$\pm\{[2,-1,0],[0,-1,2],[1,0,-1],[1,-1,1]\}$$

.

- Equivalent to the computation of several Hilbert bases computations.

- The fastest algorithm to compute Graver bases is based on a completion and project-and-lift method (Got Groebner bases?). Implemented in 4ti2 (by R. Hemmecke and P. Malkin).

- 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

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

.

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

- The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A.
- **Example:** If $A = [1\ 2\ 1]$ then its Graver basis is

$$\pm\{[2,-1,0],[0,-1,2],[1,0,-1],[1,-1,1]\}$$

.

- Equivalent to the computation of several Hilbert bases computations.
- The fastest algorithm to compute Graver bases is based on a completion and project-and-lift method (Got Groebner bases?). Implemented in 4ti2 (by R. Hemmecke and P. Malkin).
- 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

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

.

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

- The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A.
- **Example:** If $A = [1\ 2\ 1]$ then its Graver basis is

$$\pm\{[2,-1,0],[0,-1,2],[1,0,-1],[1,-1,1]\}$$

.

- Equivalent to the computation of several Hilbert bases computations.
- The fastest algorithm to compute Graver bases is based on a completion and project-and-lift method (Got Groebner bases?). Implemented in `4ti2` (by R. Hemmecke and P. Malkin).
- 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

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

.

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

- The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A.
- **Example:** If $A = [1\ 2\ 1]$ then its Graver basis is

$$\pm\{[2,-1,0],[0,-1,2],[1,0,-1],[1,-1,1]\}$$

.

- Equivalent to the computation of several Hilbert bases computations.
- The fastest algorithm to compute Graver bases is based on a completion and project-and-lift method (Got Groebner bases?). Implemented in `4ti2` (by R. Hemmecke and P. Malkin).
- 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

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

.

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

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

- **Example:** If $A = [1\ 2\ 1]$ then its Graver basis is

$$\pm\{[2,-1,0],[0,-1,2],[1,0,-1],[1,-1,1]\}$$

.

- Equivalent to the computation of several Hilbert bases computations.

- The fastest algorithm to compute Graver bases is based on a completion and project-and-lift method (Got Groebner bases?). Implemented in `4ti2` (by R. Hemmecke and P. Malkin).

- 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

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

.

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

- The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A.
- **Example:** If $A = [1\ 2\ 1]$ then its Graver basis is

$$\pm\{[2,-1,0],[0,-1,2],[1,0,-1],[1,-1,1]\}$$

.

- Equivalent to the computation of several Hilbert bases computations.
- The fastest algorithm to compute Graver bases is based on a completion and project-and-lift method (Got Groebner bases?). Implemented in `4ti2` (by R. Hemmecke and P. Malkin).
- 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

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

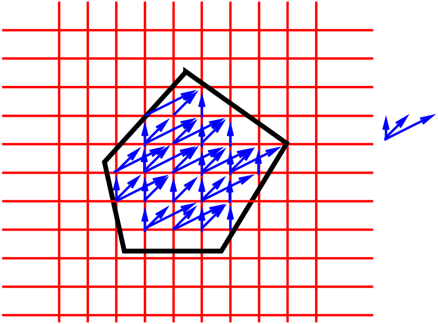
.

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

- For a fixed cost vector c, we can visualize a Graver basis of an integer program by creating a graph!!
- Here is how to construct it, consider

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

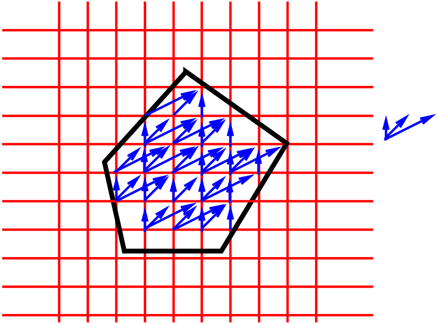
.
Nodes are lattice points in $L(b)$ and the Graver basis elements give directed edges departing from each lattice point $u \in L(b)$.



- For a fixed cost vector c, we can visualize a Graver basis of an integer program by creating a graph!!
- Here is how to construct it, consider

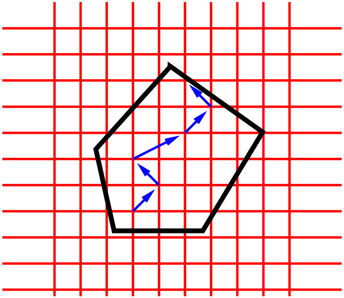
$$L(b) := \{x \mid Ax = b, x \geq 0, \ x \in \mathbf{Z}^n\}$$

.
Nodes are lattice points in $L(b)$ and the Graver basis elements give directed edges departing from each lattice point $u \in L(b)$.



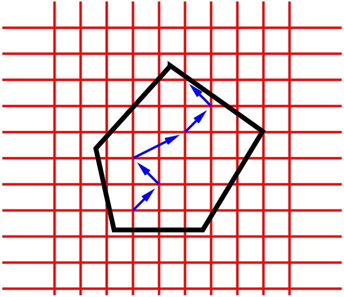
GOOD NEWS: Test Sets and Augmentation Method

- A TEST SET is a finite collection of integral vectors with the property that every feasible non-optimal solution of an integer program can be improved by adding a vector in the test set.



- **Theorem** [J. Graver 1975] Graver bases for A can be used to solve the augmentation problem Given $A \in \mathbf{Z}^{m \times n}$, $x \in \mathbf{N}^n$ and $c \in \mathbf{Z}^n$, either find an improving direction $g \in \mathbf{Z}^n$, namely one with $x - g \in \{y \in \mathbf{N}^n : Ay = Ax\}$ and $cg > 0$, or assert that no such g exists.

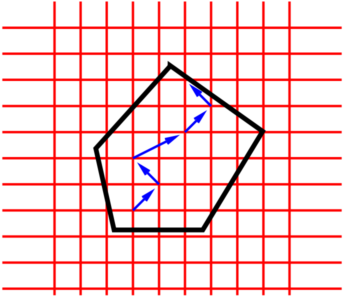
- A TEST SET is a finite collection of integral vectors with the property that every feasible non-optimal solution of an integer program can be improved by adding a vector in the test set.



- **Theorem** [J. Graver 1975] Graver bases for A can be used to solve the augmentation problem Given $A \in \mathbf{Z}^{m \times n}$, $x \in \mathbf{N}^n$ and $c \in \mathbf{Z}^n$, either find an improving direction $g \in \mathbf{Z}^n$, namely one with $x - g \in \{y \in \mathbf{N}^n \,:\, Ay = Ax\}$ and $cg > 0$, or assert that no such g exists.

GOOD NEWS: Test Sets and Augmentation Method

- A TEST SET is a finite collection of integral vectors with the property that every feasible non-optimal solution of an integer program can be improved by adding a vector in the test set.



- **Theorem** [J. Graver 1975] Graver bases for A can be used to solve the augmentation problem Given $A \in \mathbf{Z}^{m \times n}$, $x \in \mathbf{N}^n$ and $c \in \mathbf{Z}^n$, either find an improving direction $g \in \mathbf{Z}^n$, namely one with $x - g \in \{y \in \mathbf{N}^n : Ay = Ax\}$ and $cg > 0$, or assert that no such g exists.

- Graver test sets can be exponentially large even in fixed dimension! In general NP-hard to compute.
- OUR NEW RESULTS: There are useful cases where Graver bases become very manageable!!!
- **Key Lemma** Fix any pair of integer matrices $A \in \mathbf{Z}^{r \times q}$ and $B \in \mathbf{Z}^{s \times q}$. 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.
- **Idea** [Aoki-Takemura, Santos-Sturmfels, Hosten-Sullivant] For every pair of integer matrices $A \in \mathbf{Z}^{r \times q}$ and $B \in \mathbf{Z}^{s \times q}$, there exists a constant $g(A, B)$ such that for all n, the Graver basis of $[A, B]^{(n)}$ consists of vectors with at most $g(A, B)$ the number nonzero components.

- Graver test sets can be exponentially large even in fixed dimension! In general NP-hard to compute.
- OUR NEW RESULTS: There are useful cases where Graver bases become very manageable!!!
- **Key Lemma** Fix any pair of integer matrices $A \in \mathbf{Z}^{r \times q}$ and $B \in \mathbf{Z}^{s \times q}$. 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.
- **Idea** [Aoki-Takemura, Santos-Sturmfels, Hosten-Sullivant] For every pair of integer matrices $A \in \mathbf{Z}^{r \times q}$ and $B \in \mathbf{Z}^{s \times q}$, there exists a constant $g(A,B)$ such that for all n, the Graver basis of $[A,B]^{(n)}$ consists of vectors with at most $g(A,B)$ the number nonzero components.

- Graver test sets can be exponentially large even in fixed dimension! In general NP-hard to compute.
- OUR NEW RESULTS: There are useful cases where Graver bases become very manageable!!!
- **Key Lemma** Fix any pair of integer matrices $A \in \mathbf{Z}^{r \times q}$ and $B \in \mathbf{Z}^{s \times q}$. 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.
- **Idea** [Aoki-Takemura, Santos-Sturmfels, Hosten-Sullivant] For every pair of integer matrices $A \in \mathbf{Z}^{r \times q}$ and $B \in \mathbf{Z}^{s \times q}$, there exists a constant $g(A, B)$ such that for all n, the Graver basis of $[A, B]^{(n)}$ consists of vectors with at most $g(A, B)$ the number nonzero components.

- Graver test sets can be exponentially large even in fixed dimension! In general NP-hard to compute.
- OUR NEW RESULTS: There are useful cases where Graver bases become very manageable!!!
- **Key Lemma** Fix any pair of integer matrices $A \in \mathbf{Z}^{r \times q}$ and $B \in \mathbf{Z}^{s \times q}$. 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.
- **Idea** [Aoki-Takemura, Santos-Sturmfels, Hosten-Sullivant] For every pair of integer matrices $A \in \mathbf{Z}^{r \times q}$ and $B \in \mathbf{Z}^{s \times q}$, there exists a constant $g(A, B)$ such that for all n, the Graver basis of $[A, B]^{(n)}$ consists of vectors with at most $g(A, B)$ the number nonzero components.

Proof by Example

Consider the matrices $A = [1\ 1]$ and $B = I_2$. The Graver complexity of the pair A, B is $g(A, B) = 2$.

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

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

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

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

Conclusions and Future work

- Traditional Methods are not sufficient to solve all current integer optimization models, even the simple linear ones!
- There is demand to solve NON-LINEAR optimization problems, not just model things linearly anymore...
- In fact more is needed: Mixed variables, multi objective, stochastics, etc... are necessary for having better models.
- Tools from Discrete and Computational Geometry are bound to play a stronger role in the foundations of new algorithmic tools!
- Not just the foundations need to be studied, new software is beginning to appear that uses all these ideas: `4ti2`, `LattE`.

Conclusions and Future work

- Traditional Methods are not sufficient to solve all current integer optimization models, even the simple linear ones!
- There is demand to solve NON-LINEAR optimization problems, not just model things linearly anymore...
- In fact more is needed: Mixed variables, multi objective, stochastics, etc... are necessary for having better models.
- Tools from Discrete and Computational Geometry are bound to play a stronger role in the foundations of new algorithmic tools!
- Not just the foundations need to be studied, new software is beginning to appear that uses all these ideas: `4ti2`, `LattE`.

Conclusions and Future work

- Traditional Methods are not sufficient to solve all current integer optimization models, even the simple linear ones!
- There is demand to solve NON-LINEAR optimization problems, not just model things linearly anymore...
- In fact more is needed: Mixed variables, multi objective, stochastics, etc... are necessary for having better models.
- Tools from Discrete and Computational Geometry are bound to play a stronger role in the foundations of new algorithmic tools!
- Not just the foundations need to be studied, new software is beginning to appear that uses all these ideas: `4ti2`, `LattE`.

Conclusions and Future work

- Traditional Methods are not sufficient to solve all current integer optimization models, even the simple linear ones!
- There is demand to solve NON-LINEAR optimization problems, not just model things linearly anymore...
- In fact more is needed: Mixed variables, multi objective, stochastics, etc... are necessary for having better models.
- Tools from Discrete and Computational Geometry are bound to play a stronger role in the foundations of new algorithmic tools!
- Not just the foundations need to be studied, new software is beginning to appear that uses all these ideas: `4ti2`, `LattE`.

Conclusions and Future work

- Traditional Methods are not sufficient to solve all current integer optimization models, even the simple linear ones!
- There is demand to solve NON-LINEAR optimization problems, not just model things linearly anymore...
- In fact more is needed: Mixed variables, multi objective, stochastics, etc... are necessary for having better models.
- Tools from Discrete and Computational Geometry are bound to play a stronger role in the foundations of new algorithmic tools!
- Not just the foundations need to be studied, new software is beginning to appear that uses all these ideas: `4ti2`, `LattE`.

Thank you

Gracias