Algebraic-Geometric ideas in Discrete Optimization

Jesús A. De Loera, UC Davis

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

M. Köppe & J. Lee (IBM),
U. Rothblum & S. Onn (Technion Haifa),
R. Hemmecke (T.Univ. Munich) & R. Weismantel (ETH Zürich)

November 5, 2011

Le Menu

- Appetizer: Challenges in Discrete Optimization and why the need for new tools...
- Main Dish: Some Algebraic-Geometric Algorithms in Optimization
  - Barvinok's Algorithm.
  - Graver Bases.
- Dessert: Closing Comments and Future directions.

Le Menu

- Appetizer: Challenges in Discrete Optimization and why the need for new tools...
- Main Dish: Some Algebraic-Geometric Algorithms in Optimization
  - Barvinok's Algorithm.
  - Graver Bases.
- Dessert: Closing Comments and Future directions.

Le Menu

- Appetizer: Challenges in Discrete Optimization and why the need for new tools...
- Main Dish: Some Algebraic-Geometric Algorithms in Optimization
  - Barvinok's Algorithm.
  - Graver Bases.
- Dessert: Closing Comments and Future directions.

Le Menu

- Appetizer: Challenges in Discrete Optimization and why the need for new tools...
- Main Dish: Some Algebraic-Geometric Algorithms in Optimization
    - Barvinok's Algorithm.
    - Graver Bases.
- Dessert: Closing Comments and Future directions.

Challenges in Discrete Optimization why need for new tools

(in particular from algebra, geometry and topology).

What is Discrete Optimization?

- A part of Applied Mathematics, its main problem: Given a finite set X, each of whose elements has an assigned cost, price or optimality criteria, find the cheapest such object.
- Problems come from bioinformatics, industrial engineering, management, operations planning, finances, any area where the best solution is required!
- History starts with WWII Initial work by Kantorovich (1939), T.C Koopmans (1941), von Neumann (1947), Dantzig (1950), Ford and Fulkerson (1956). Invention of linear programming and the simplex method.

What is Discrete Optimization?

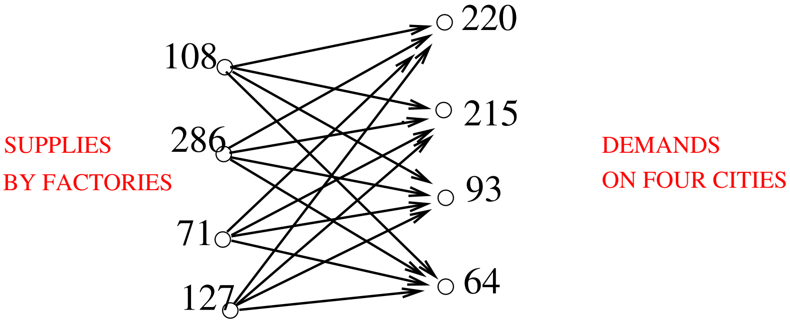
- A part of Applied Mathematics, its main problem: Given a finite set X, each of whose elements has an assigned cost, price or optimality criteria, find the cheapest such object.
- Problems come from bioinformatics, industrial engineering, management, operations planning, finances, any area where the best solution is required!
- History starts with WWII Initial work by Kantorovich (1939), T.C Koopmans (1941), von Neumann (1947), Dantzig (1950), Ford and Fulkerson (1956). Invention of linear programming and the simplex method.

What is Discrete Optimization?

- A part of Applied Mathematics, its main problem: Given a finite set X, each of whose elements has an assigned cost, price or optimality criteria, find the cheapest such object.
- Problems come from bioinformatics, industrial engineering, management, operations planning, finances, any area where the best solution is required!
- History starts with WWII Initial work by Kantorovich (1939), T.C Koopmans (1941), von Neumann (1947), Dantzig (1950), Ford and Fulkerson (1956). Invention of linear programming and the simplex method.

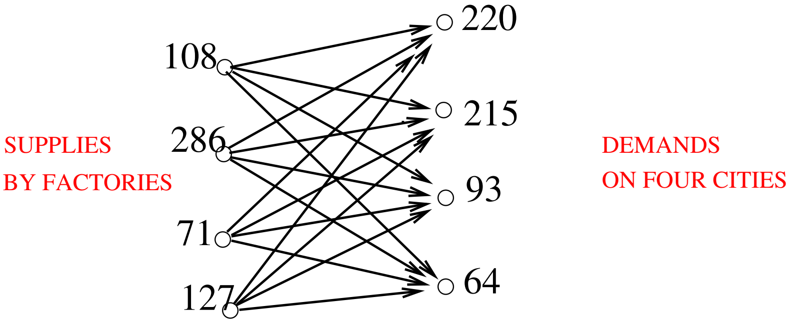
A Useful Example

- The Transportation problem: A company builds laptops in four factories, each with certain supply power. Four cities have laptop demands. There is a cost $c_{i,j}$ for transporting a laptop from factory i to city j. What is the best assignment of transport in order to minimize the cost?



A Useful Example

- The Transportation problem: A company builds laptops in four factories, each with certain supply power. Four cities have laptop demands. There is a cost $c_{i,j}$ for transporting a laptop from factory i to city j. What is the best assignment of transport in order to minimize the cost?



ILP model: equations and inequalities

- Let $x_{i,j}$ be a variable indicating number of laptops factory i provides to city j. $x_{i,j}$ can only take non-negative integer values, $x_{i,j} \geq 0$.
- Then Since factory i produces a_i laptops we have

$$\sum_{j=1}^{n} x_{i,j} = a_i, \text{ for all } i = 1, \ldots, n.$$

and since city j needs b_j laptops

$$\sum_{i=1}^{n} x_{i,j} = b_j, \text{ for all } j = 1, \ldots, n.$$

- Now we minimize $\sum c_{i,j} x_{i,j}$.

ILP model: equations and inequalities

- Let $x_{i,j}$ be a variable indicating number of laptops factory i provides to city j. $x_{i,j}$ can only take non-negative integer values, $x_{i,j} \geq 0$.
- Then Since factory i produces a_i laptops we have

$$\sum_{j=1}^{n} x_{i,j} = a_i, \text{ for all } i = 1, \ldots, n.$$

and since city j needs b_j laptops

$$\sum_{i=1}^{n} x_{i,j} = b_j, \text{ for all } j = 1, \ldots, n.$$

- Now we minimize $\sum c_{i,j} x_{i,j}$.

ILP model: equations and inequalities

- Let $x_{i,j}$ be a variable indicating number of laptops factory i provides to city j. $x_{i,j}$ can only take non-negative integer values, $x_{i,j} \geq 0$.
- Then Since factory i produces a_i laptops we have

$$\sum_{j=1}^{n} x_{i,j} = a_i, \text{ for all } i = 1, \ldots, n.$$

and since city j needs b_j laptops

$$\sum_{i=1}^{n} x_{i,j} = b_j, \text{ for all } j = 1, \ldots, n.$$

- Now we minimize $\sum c_{i,j} x_{i,j}$.

ILP model: equations and inequalities

- Let $x_{i,j}$ be a variable indicating number of laptops factory i provides to city j. $x_{i,j}$ can only take non-negative integer values, $x_{i,j} \geq 0$.
- Then Since factory i produces a_i laptops we have

$$\sum_{j=1}^{n} x_{i,j} = a_i, \text{ for all } i = 1, \ldots, n.$$

  and since city j needs b_j laptops

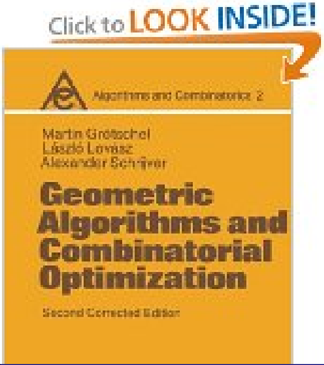
$$\sum_{i=1}^{n} x_{i,j} = b_j, \text{ for all } j = 1, \ldots, n.$$

- Now we minimize $\sum c_{i,j} x_{i,j}$.

Overview LINEAR Discrete Optimization



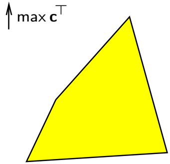
Efficient computation with Convex Sets & Lattices $\Longleftrightarrow$ Efficient Optimization



At the beginning there was...

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!

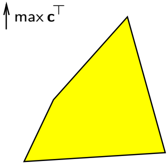
Integer programs

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

At the beginning there was...

Linear programs

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



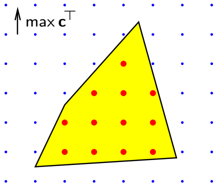
Special integer programs

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

Matrix A is SPECIAL!

Integer programs

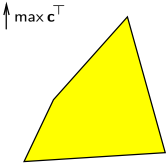
$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$



At the beginning there was...

Linear programs

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



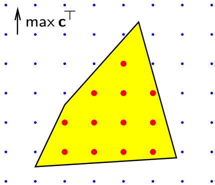
Special integer programs

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

Matrix A is SPECIAL!

Integer programs

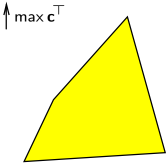
$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$



At the beginning there was...

Linear programs

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



Easy
(polynomial-time solvable)

Special integer programs

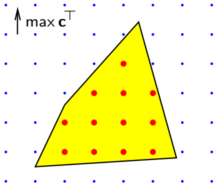
$$\begin{array}{ll} \max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{array}$$

Matrix A is SPECIAL!

Medium
(can be easy or hard)
Network problems
Fixed dimension
knapsacks
0-1 matrices

Integer programs

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



Hard
(NP-hard)

Traditional Algorithms

Dual (polyhedral) techniques

Cutting plane algorithms
– based on polyhedral theory

Enumeration

Branch-and-bound

Adhoc methods

special structure
(e.g. network,
matroids, etc.)

Mathematical modelling – Strong initial IP formulation

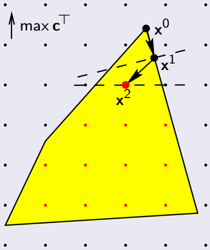
Traditional Algorithms

Dual (polyhedral) techniques	Enumeration	Adhoc methods
Cutting plane algorithms – based on polyhedral theory	Branch-and-bound	special structure (e.g. network, matroids, etc.)

Mathematical modelling – Strong initial IP formulation

Traditional Algorithms

Dual (polyhedral) techniques



Cutting plane algorithms

– based on polyhedral theory

Enumeration

Branch-and-bound

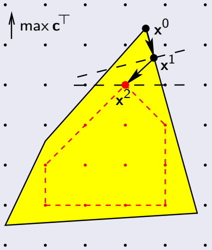
Adhoc methods

special structure (e.g. network, matroids, etc.)

Mathematical modelling – Strong initial IP formulation

Traditional Algorithms

Dual (polyhedral) techniques



Cutting plane algorithms
– based on polyhedral theory

Enumeration

Branch-and-bound

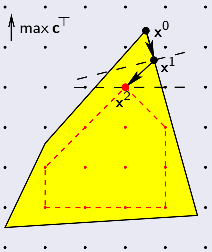
Adhoc methods

special structure
(e.g. network,
matroids, etc.)

Mathematical modelling – Strong initial IP formulation

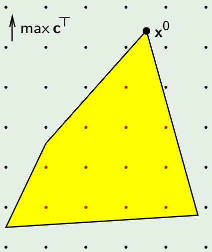
Traditional Algorithms

Dual (polyhedral) techniques



Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

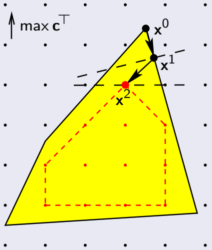
Adhoc methods

special structure (e.g. network, matroids, etc.)

Mathematical modelling – Strong initial IP formulation

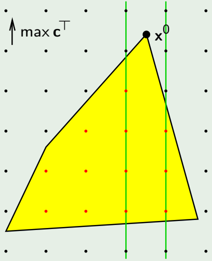
Traditional Algorithms

Dual (polyhedral) techniques



Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

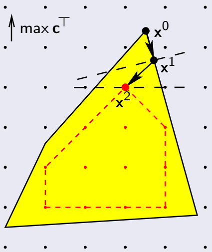
Adhoc methods

special structure
(e.g. network,
matroids, etc.)

Mathematical modelling – Strong initial IP formulation

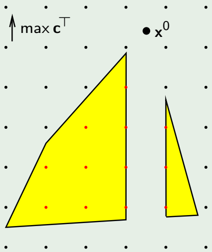
Traditional Algorithms

Dual (polyhedral) techniques



Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

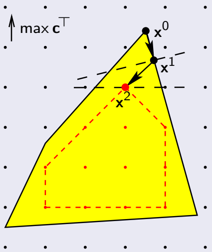
Adhoc methods

special structure
(e.g. network,
matroids, etc.)

Mathematical modelling – Strong initial IP formulation

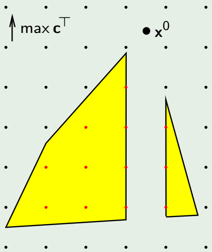
Traditional Algorithms

Dual (polyhedral) techniques



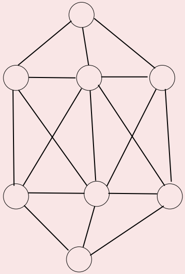
Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

Adhoc methods

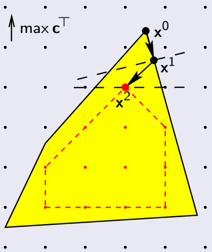


special structure (e.g. network, matroids, etc.)

Mathematical modelling – Strong initial IP formulation

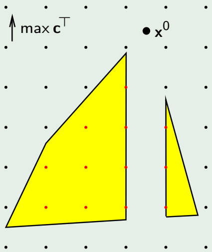
Traditional Algorithms

Dual (polyhedral) techniques



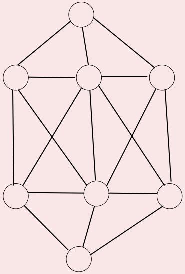
Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

Adhoc methods



special structure
(e.g. network,
matroids, etc.)

Mathematical modelling – Strong initial IP formulation

OUR WISH:

Want to handle more complicated Constraints and Objective functions

Example: Non-linear transportation polytopes

1. In the traditional transportation problem cost at an edge is a constant. Thus we optimize a linear function.
2. but due to congestion or heavy traffic or heavy communication load the transportation cost on an edge could be a non-linear function of the flow at each edge.
3. For example cost at each edge is $f_{ij}(x_{ij}) = c_{ij}|x_{ij}|^{a_{ij}}$ for suitable constant a_{ij}. This results on a non-linear function $\sum f_{ij}$ which is much harder to minimize.

Example: Non-linear transportation polytopes

1. In the traditional transportation problem cost at an edge is a constant. Thus we optimize a linear function.
2. but due to congestion or heavy traffic or heavy communication load the transportation cost on an edge could be a non-linear function of the flow at each edge.
3. For example cost at each edge is $f_{ij}(x_{ij}) = c_{ij}|x_{ij}|^{a_{ij}}$ for suitable constant a_{ij}. This results on a non-linear function $\sum f_{ij}$ which is much harder to minimize.

Example: Non-linear transportation polytopes

1. In the traditional transportation problem cost at an edge is a constant. Thus we optimize a linear function.
2. but due to congestion or heavy traffic or heavy communication load the transportation cost on an edge could be a non-linear function of the flow at each edge.
3. For example cost at each edge is $f_{ij}(x_{ij}) = c_{ij}|x_{ij}|^{a_{ij}}$ for suitable constant a_{ij}. This results on a non-linear function $\sum f_{ij}$ which is much harder to minimize.

Reality is NON-LINEAR and worse!!

Non-linear Discrete 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

with f, g_j Non-Linear

WHAT CAN BE DONE IN THIS GENERAL CONTEXT??

Prove good theorems? Are there efficient algorithms?

- BAD NEWS: The problem is INCREDIBLY HARD

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

- EVEN WORSE

  Theorem: It undecidable even with number of variables=10. (Matiyasevich and Davis 1982).

- THERE IS HOPE with good structure:

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

  (Khachiyan and Porkolab, 2000)

Reality is NON-LINEAR and worse!!

Non-linear Discrete 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

with f, g_j Non-Linear

WHAT CAN BE DONE IN THIS GENERAL CONTEXT??

Prove good theorems? Are there efficient algorithms?

- BAD NEWS: The problem is INCREDIBLY HARD
  Theorem It is UNDECIDABLE already when f,g_i's are arbitrary polynomials (Jeroslow, 1979).
- EVEN WORSE
  Theorem: It undecidable even with number of variables=10. (Matiyasevich and Davis 1982).
- THERE IS HOPE with good structure:
  Theorem: For fixed number of variables AND convex polynomials f, g_i problem can be solved in polynomial time
  (Khachiyan and Porkolab, 2000)

Reality is NON-LINEAR and worse!!

Non-linear Discrete 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

with f, g_j Non-Linear

WHAT CAN BE DONE IN THIS GENERAL CONTEXT??

Prove good theorems? Are there efficient algorithms?

- BAD NEWS: The problem is INCREDIBLY HARD
  Theorem It is UNDECIDABLE already when f,g_i's are arbitrary polynomials (Jeroslow, 1979).
- EVEN WORSE
  Theorem: It undecidable even with number of variables=10. (Matiyasevich and Davis 1982).
- THERE IS HOPE with good structure:
  Theorem: For fixed number of variables AND convex polynomials f, g_i problem can be solved in polynomial time
  (Khachiyan and Porkolab, 2000)

Reality is NON-LINEAR and worse!!

Non-linear Discrete 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

with f, g_j Non-Linear

WHAT CAN BE DONE IN THIS GENERAL CONTEXT??

Prove good theorems? Are there efficient algorithms?

- BAD NEWS: The problem is INCREDIBLY HARD
  Theorem It is UNDECIDABLE already when f,g_i's are arbitrary polynomials (Jeroslow, 1979).
- EVEN WORSE
  Theorem: It undecidable even with number of variables=10. (Matiyasevich and Davis 1982).
- THERE IS HOPE with good structure:
  Theorem: For fixed number of variables AND convex polynomials f, g_i problem can be solved in polynomial time
  (Khachiyan and Porkolab, 2000)

Reality is NON-LINEAR and worse!!

Non-linear Discrete 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

with f, g_j Non-Linear

WHAT CAN BE DONE IN THIS GENERAL CONTEXT??

Prove good theorems? Are there efficient algorithms?

- BAD NEWS: The problem is INCREDIBLY HARD
  Theorem It is UNDECIDABLE already when f,g_i's are arbitrary polynomials (Jeroslow, 1979).
- EVEN WORSE
  Theorem: It undecidable even with number of variables=10. (Matiyasevich and Davis 1982).
- THERE IS HOPE with good structure:
  Theorem: For fixed number of variables AND convex polynomials f, g_i problem can be solved in polynomial time
  (Khachiyan and Porkolab, 2000)

How about polyhedral constraints non-linear objective??

Let f be a multivariate polynomial function,

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

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!

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

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)

How about polyhedral constraints non-linear objective??

Let f be a multivariate polynomial function,

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

Hard
(NP-hard)

Special programs

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

Matrix A is SPECIAL!

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

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)

???

How about polyhedral constraints non-linear objective??

Let f be a multivariate polynomial function,

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

Hard
(NP-hard)

Special programs

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

Matrix A is SPECIAL!

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

Hard
(NP-hard)

???

We study TWO
special cases

Algebraic Geometric Ideas in Optimization

Problem type

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

where

- P is a polytope (bounded polyhedron) given by linear constraints,
- f is a (multivariate) polynomial function non-negative over $P \cap \mathbf{Z}^d$,
- the dimension d is fixed.

Prior Work

- Integer Linear Programming can be solved in polynomial time
  (H. W. Lenstra Jr, 1983)
- Convex polynomials f can be minimized in polynomial time
  (Khachiyan and Porkolab, 2000)
- Optimizing an arbitrary degree-4 polynomial f for $d = 2$ is NP-hard

WHAT CAN BE PROVED IN THIS CASE??

Special Assumption I : FIXED DIMENSION

Problem type

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

where

- P is a polytope (bounded polyhedron) given by linear constraints,
- f is a (multivariate) polynomial function non-negative over $P \cap \mathbf{Z}^d$,
- the dimension d is fixed.

Prior Work

- Integer Linear Programming can be solved in polynomial time (H. W. Lenstra Jr, 1983)
- Convex polynomials f can be minimized in polynomial time (Khachiyan and Porkolab, 2000)
- Optimizing an arbitrary degree-4 polynomial f for $d = 2$ is NP-hard

WHAT CAN BE PROVED IN THIS CASE??

Special Assumption I : FIXED DIMENSION

Problem type

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

where

- P is a polytope (bounded polyhedron) given by linear constraints,
- f is a (multivariate) polynomial function non-negative over $P \cap \mathbf{Z}^d$,
- the dimension d is fixed.

Prior Work

- Integer Linear Programming can be solved in polynomial time (H. W. Lenstra Jr, 1983)
- Convex polynomials f can be minimized in polynomial time (Khachiyan and Porkolab, 2000)
- Optimizing an arbitrary degree-4 polynomial f for $d = 2$ is NP-hard

WHAT CAN BE PROVED IN THIS CASE??

Special Assumption I : FIXED DIMENSION

Problem type

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

where

- P is a polytope (bounded polyhedron) given by linear constraints,
- f is a (multivariate) polynomial function non-negative over $P \cap \mathbf{Z}^d$,
- the dimension d is fixed.

Prior Work

- Integer Linear Programming can be solved in polynomial time (H. W. Lenstra Jr, 1983)
- Convex polynomials f can be minimized in polynomial time (Khachiyan and Porkolab, 2000)
- Optimizing an arbitrary degree-4 polynomial f for $d = 2$ is NP-hard

WHAT CAN BE PROVED IN THIS CASE??

Special Assumption I : FIXED DIMENSION

Problem type

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

where

- P is a polytope (bounded polyhedron) given by linear constraints,
- f is a (multivariate) polynomial function non-negative over $P \cap \mathbf{Z}^d$,
- the dimension d is fixed.

Prior Work

- Integer Linear Programming can be solved in polynomial time (H. W. Lenstra Jr, 1983)
- Convex polynomials f can be minimized in polynomial time (Khachiyan and Porkolab, 2000)
- Optimizing an arbitrary degree-4 polynomial f for $d = 2$ is NP-hard

WHAT CAN BE PROVED IN THIS CASE??

Applications of Barvinok's Algorithms

Idea: New Representation of Lattice Points

- Given $K \subset \mathbf{R}^d$ we define the formal power series

$$f(K) = \sum_{\alpha \in K \cap \mathbf{Z}^d} z_1^{\alpha_1} z_2^{\alpha_2} \dots z_n^{\alpha_n}.$$

Think of the lattice points as monomials!!! EXAMPLE: $(7, 4, -3)$ is $z_1^7 z_2^4 z_3^{-3}$.

- **Theorem** (see R. Stanley EC Vol 1) Given $K = \{x \in \mathbf{R}^n | Ax = b,\ Bx \leq b'\}$ where A, B are integral matrices and b, b' are integral vectors, The generating function $f(K)$ can be encoded as rational function.

- GOOD NEWS: **ALL** the lattice points of the polyhedron K, be encoded in a sum of rational functions **efficiently!!!**

Idea: New Representation of Lattice Points

- Given $K \subset \mathbf{R}^d$ we define the formal power series

$$f(K) = \sum_{\alpha \in K \cap \mathbf{Z}^d} z_1^{\alpha_1} z_2^{\alpha_2} \ldots z_n^{\alpha_n}.$$

  Think of the lattice points as monomials!!! EXAMPLE: $(7, 4, -3)$ is $z_1^7 z_2^4 z_3^{-3}$.

- **Theorem** (see R. Stanley EC Vol 1) Given $K = \{x \in \mathbf{R}^n | Ax = b,\ Bx \leq b'\}$ where A, B are integral matrices and b, b' are integral vectors, The generating function $f(K)$ can be encoded as rational function.

- GOOD NEWS: **ALL** the lattice points of the polyhedron K, be encoded in a sum of rational functions **efficiently!!!**

Idea: New Representation of Lattice Points

- Given $K \subset \mathbf{R}^d$ we define the formal power series

$$f(K) = \sum_{\alpha \in K \cap \mathbf{Z}^d} z_1^{\alpha_1} z_2^{\alpha_2} \dots z_n^{\alpha_n}.$$

  Think of the lattice points as monomials!!! EXAMPLE: $(7, 4, -3)$ is $z_1^7 z_2^4 z_3^{-3}$.

- **Theorem** (see R. Stanley EC Vol 1) Given $K = \{x \in \mathbf{R}^n | Ax = b,\ Bx \leq b'\}$ where A, B are integral matrices and b, b' are integral vectors, The generating function $f(K)$ can be encoded as rational function.

- GOOD NEWS: **ALL** the lattice points of the polyhedron K, be encoded in a sum of rational functions **efficiently!!!**

Barvinok's short rational generating functions

Generating functions

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + \dots z^M$$

Theorem (Alexander Barvinok, 1994)

Let the dimension d be fixed. There is a polynomial-time algorithm for computing a representation of the generating function

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

of the integer points $P \cap \mathbf{Z}^d$ of a polyhedron $P \subset \mathbf{R}^d$ (given by rational inequalities) 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).

Generating functions

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + \dots z^M$$
$$= \frac{1 - z^M}{1 - z} \qquad \text{for } z \neq 1$$

Theorem (Alexander Barvinok, 1994)

Let the dimension d be fixed. There is a polynomial-time algorithm for computing a representation of the generating function

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

of the integer points $P \cap \mathbf{Z}^d$ of a polyhedron $P \subset \mathbf{R}^d$ (given by rational inequalities) 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).

Barvinok's short rational generating functions

Generating functions

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + \dots z^M$$
$$= \frac{1 - z^M}{1 - z} \qquad \text{for } z \neq 1$$

Theorem (Alexander Barvinok, 1994)

Let the dimension d be fixed. There is a polynomial-time algorithm for computing a representation of the generating function

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

of the integer points $P \cap \mathbf{Z}^d$ of a polyhedron $P \subset \mathbf{R}^d$ (given by rational inequalities) 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).

Barvinok's short rational generating functions

Generating functions

$$g_P(z) = z^0 + z^1 + z^2 + z^3 + \dots z^M$$
$$= \frac{1 - z^M}{1 - z} \qquad \text{for } z \neq 1$$

Theorem (Alexander Barvinok, 1994)

Let the dimension d be fixed. There is a polynomial-time algorithm for computing a representation of the generating function

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

of the integer points $P \cap \mathbf{Z}^d$ of a polyhedron $P \subset \mathbf{R}^d$ (given by rational inequalities) in the form of a rational function.

Corollary

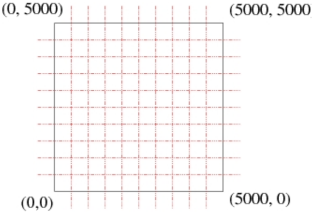
In particular,

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

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

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

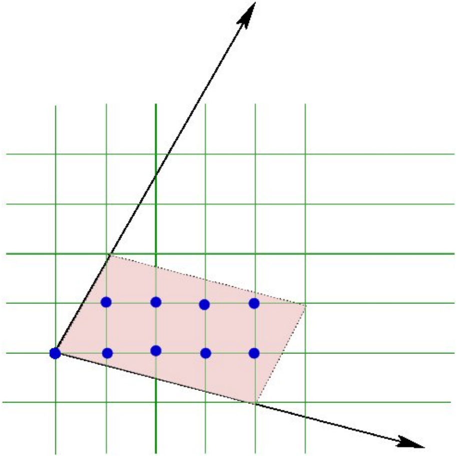
Also, $f(tP, z)$ is

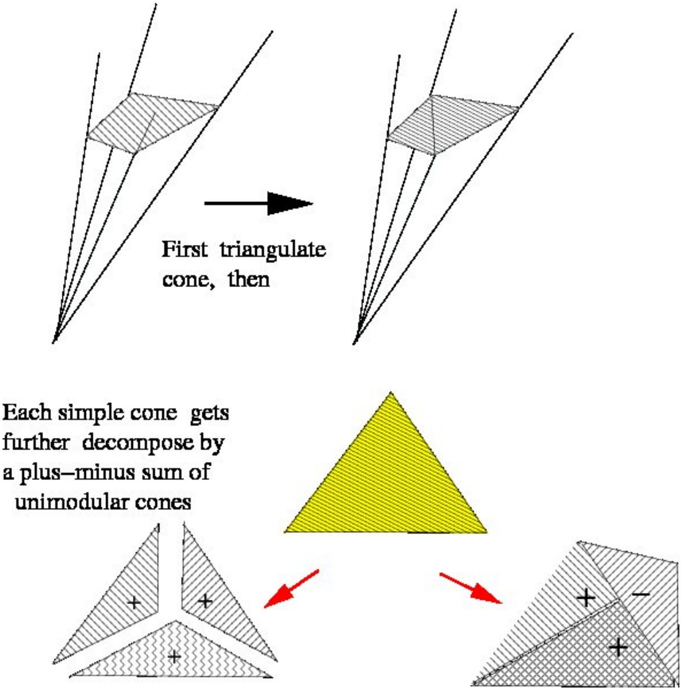
$$\frac{1}{(1-z_1)(1-z_2)} + \frac{{z_1}^{5000\cdot t}}{(1-{z_1}^{-1})(1-z_2)} + \frac{{z_2}^{5000\cdot t}}{(1-{z_2}^{-1})(1-z_1)} + \frac{{z_1}^{5000\cdot t}{z_2}^{5000\cdot t}}{(1-{z_1}^{-1})(1-{z_2}^{-1})}$$

Rational Function of a pointed Cone

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

Theorem (FPTAS for Integer Polynomial Maximization)

Let the dimension d be fixed. There exists an algorithm whose input data are

- a polytope $P \subset \mathbf{R}^d$, given by rational linear inequalities, and
- a polynomial $f \in \mathbf{Z}[x_1, \dots, x_d]$ with integer coefficients and maximum total degree D that is non-negative on $P \cap \mathbf{Z}^d$

with the following properties.

1. For a given k, it computes in running time polynomial in k, the encoding size of P and f, and D lower and upper bounds $L_k \leq f(\mathbf{x}^{\max}) \leq U_k$ satisfying
$$U_k - L_k \leq \left(\sqrt[k]{|P \cap \mathbf{Z}^d|} - 1 \right) \cdot f(\mathbf{x}^{\max}).$$
2. For $k = (1 + 1/\epsilon) \log(|P \cap \mathbf{Z}^d|)$, the bounds satisfy
$$U_k - L_k \leq \epsilon f(\mathbf{x}^{\max}),$$
and they can be computed in time polynomial in the input size, the total degree D, and $1/\epsilon$.
3. By iterated bisection of $P \cap \mathbf{Z}^d$, it constructs a feasible solution $\mathbf{x}_\epsilon \in P \cap \mathbf{Z}^d$ with
$$\left| f(\mathbf{x}_\epsilon) - f(\mathbf{x}^{\max}) \right| \leq \epsilon f(\mathbf{x}^{\max}).$$

Theorem (FPTAS for Integer Polynomial Maximization)

Let the dimension d be fixed. There exists an algorithm whose input data are

- a polytope $P \subset \mathbf{R}^d$, given by rational linear inequalities, and
- a polynomial $f \in \mathbf{Z}[x_1, \ldots, x_d]$ with integer coefficients and maximum total degree D that is non-negative on $P \cap \mathbf{Z}^d$

with the following properties.

1. For a given k, it computes in running time polynomial in k, the encoding size of P and f, and D lower and upper bounds $L_k \leq f(\mathbf{x}^{\max}) \leq U_k$ satisfying
$$U_k - L_k \leq \left(\sqrt[k]{|P \cap \mathbf{Z}^d|} - 1 \right) \cdot f(\mathbf{x}^{\max}).$$
2. For $k = (1 + 1/\epsilon) \log(|P \cap \mathbf{Z}^d|)$, the bounds satisfy
$$U_k - L_k \leq \epsilon f(\mathbf{x}^{\max}),$$
and they can be computed in time polynomial in the input size, the total degree D, and $1/\epsilon$.
3. By iterated bisection of $P \cap \mathbf{Z}^d$, it constructs a feasible solution $\mathbf{x}_\epsilon \in P \cap \mathbf{Z}^d$ with
$$\left| f(\mathbf{x}_\epsilon) - f(\mathbf{x}^{\max}) \right| \leq \epsilon f(\mathbf{x}^{\max}).$$

Theorem (FPTAS for Integer Polynomial Maximization)

Let the dimension d be fixed. There exists an algorithm whose input data are

- a polytope $P \subset \mathbf{R}^d$, given by rational linear inequalities, and
- a polynomial $f \in \mathbf{Z}[x_1, \dots, x_d]$ with integer coefficients and maximum total degree D that is non-negative on $P \cap \mathbf{Z}^d$

with the following properties.

1. For a given k, it computes in running time polynomial in k, the encoding size of P and f, and D lower and upper bounds $L_k \le f(\mathbf{x}^{\max}) \le U_k$ satisfying
$$U_k - L_k \le \left(\sqrt[k]{|P \cap \mathbf{Z}^d|} - 1 \right) \cdot f(\mathbf{x}^{\max}).$$
2. For $k = (1 + 1/\epsilon) \log(|P \cap \mathbf{Z}^d|)$, the bounds satisfy
$$U_k - L_k \le \epsilon\, f(\mathbf{x}^{\max}),$$
and they can be computed in time polynomial in the input size, the total degree D, and $1/\epsilon$.
3. By iterated bisection of $P \cap \mathbf{Z}^d$, it constructs a feasible solution $\mathbf{x}_\epsilon \in P \cap \mathbf{Z}^d$ with
$$\left| f(\mathbf{x}_\epsilon) - f(\mathbf{x}^{\max}) \right| \le \epsilon f(\mathbf{x}^{\max}).$$

Theorem (FPTAS for Integer Polynomial Maximization)

Let the dimension d be fixed. There exists an algorithm whose input data are

- a polytope $P \subset \mathbf{R}^d$, given by rational linear inequalities, and
- a polynomial $f \in \mathbf{Z}[x_1, \dots, x_d]$ with integer coefficients and maximum total degree D that is non-negative on $P \cap \mathbf{Z}^d$

with the following properties.

1. For a given k, it computes in running time polynomial in k, the encoding size of P and f, and D lower and upper bounds $L_k \leq f(\mathbf{x}^{\max}) \leq U_k$ satisfying
$$U_k - L_k \leq \left(\sqrt[k]{|P \cap \mathbf{Z}^d|} - 1\right) \cdot f(\mathbf{x}^{\max}).$$
2. For $k = (1 + 1/\epsilon)\log(|P \cap \mathbf{Z}^d|)$, the bounds satisfy
$$U_k - L_k \leq \epsilon\, f(\mathbf{x}^{\max}),$$
and they can be computed in time polynomial in the input size, the total degree D, and $1/\epsilon$.
3. By iterated bisection of $P \cap \mathbf{Z}^d$, it constructs a feasible solution $\mathbf{x}_\epsilon \in P \cap \mathbf{Z}^d$ with
$$\left|f(\mathbf{x}_\epsilon) - f(\mathbf{x}^{\max})\right| \leq \epsilon f(\mathbf{x}^{\max}).$$

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:

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

Differential operators on generating functions

Lemma

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

can be converted to a differential operator

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

which maps

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

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 polynomial time in D and the size of the input data, a Barvinok rational function representation $g_{P,f}(\mathbf{z})$ for $\sum_{\alpha\in P\cap\mathbf{Z}^d} f(\alpha)\mathbf{z}^\alpha$.

Differential operators on generating functions

Lemma

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

can be converted to a differential operator

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

which maps

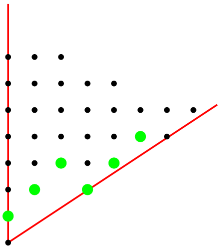
$$g(\mathbf{z}) = \sum_{\alpha \in S} \mathbf{z}^\alpha \quad \longmapsto \quad (D_f g)(\mathbf{z}) = \sum_{\alpha \in S} f(\alpha) \mathbf{z}^\alpha.$$

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 polynomial time in D and the size of the input data, a Barvinok rational function representation $g_{P,f}(\mathbf{z})$ for $\sum_{\alpha \in P \cap \mathbf{Z}^d} f(\alpha) \mathbf{z}^\alpha$.

Graver Bases

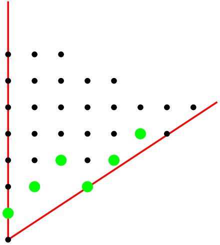
- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b,\ x \geq 0\}$. We will use basic Algebraic Geometry.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$. u is *conformally 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 $(-4, -3, 9, 1, -8)$.



- Equivalent to the computation of several Hilbert bases computations.

Graver Bases Algorithms

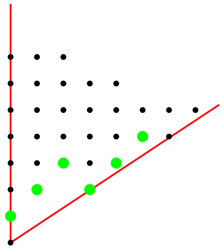
- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b, \; x \geq 0\}$. We will use basic Algebraic Geometry.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$. u is *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 $(-4, -3, 9, 1, -8)$.



- Equivalent to the computation of several Hilbert bases computations.

Graver Bases Algorithms

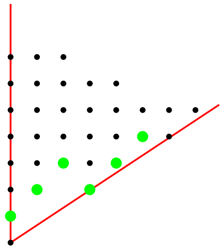
- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b,\ x \geq 0\}$. We will use basic Algebraic Geometry.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$. u is *conformally 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 $(-4, -3, 9, 1, -8)$.



- Equivalent to the computation of several Hilbert bases computations.

Graver Bases Algorithms

- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b,\ x \geq 0\}$. We will use basic Algebraic Geometry.
- 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 $(-4, -3, 9, 1, -8)$.



- Equivalent to the computation of several Hilbert bases computations.

- 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

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

.

- 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

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

.

- 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

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

.

- 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

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

.

- 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

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

.

- 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

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

.

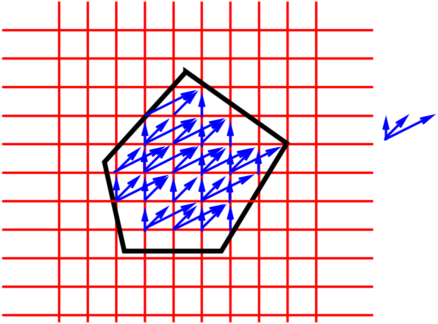
- **Theorem** The Graver basis contains all edges for all integer hulls $conv(\{x \mid Ax = b, x \geq 0,\ x \in \mathbf{Z}^n\})$ as b changes.

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

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

.

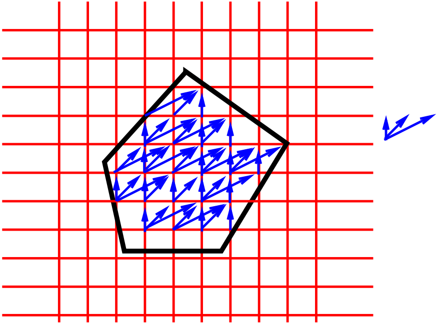
Nodes are lattice points in $L(b)$ and the Graver basis elements give directed edges departing from each lattice point $u \in L(b)$.



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

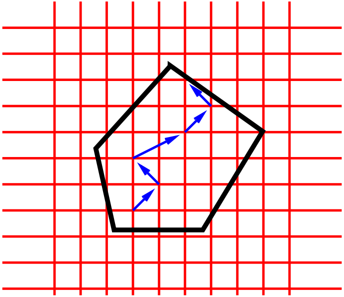
$$L(b) := \{x|\ Ax = b, x \geq 0,\ x \in \mathbf{Z}^n\}$$

.
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

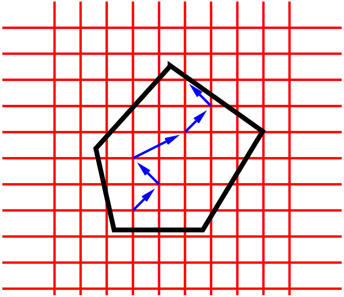
- A TEST SET is a finite collection of integral vectors with the property that every feasible non-optimal solution of an integer program can be improved by adding a vector in the test set.



- **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.

GOOD NEWS: Test Sets and Augmentation Method

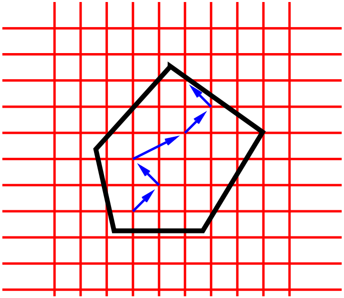
- A TEST SET is a finite collection of integral vectors with the property that every feasible non-optimal solution of an integer program can be improved by adding a vector in the test set.



- **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.

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.

BAD NEWS!!

- Graver bases contain a Gröbner bases, Hilbert bases. Work by Hosten, Graver, Scarf, Sturmfels, Sullivant, Thomas, Weismantel et al. and many others.
- Graver test sets can be exponentially large even in fixed dimension! Very hard to compute, you don't want to do this too often.
- People typically stored as a list of the whole test set and has to search within.
- NP-complete problem to decide whether a list of vectors is a complete Graver bases.
- New Results: There are useful cases where Graver bases become very manageable and efficient.
- BUT WE NEED HIGHLY STRUCTURED MATRICES!!

BAD NEWS!!

- Graver bases contain a Gröbner bases, Hilbert bases. Work by Hosten, Graver, Scarf, Sturmfels, Sullivant, Thomas, Weismantel et al. and many others.
- Graver test sets can be exponentially large even in fixed dimension! Very hard to compute, you don't want to do this too often.
- People typically stored as a list of the whole test set and has to search within.
- NP-complete problem to decide whether a list of vectors is a complete Graver bases.
- New Results: There are useful cases where Graver bases become very manageable and efficient.
- BUT WE NEED HIGHLY STRUCTURED MATRICES!!

BAD NEWS!!

- Graver bases contain a Gröbner bases, Hilbert bases. Work by Hosten, Graver, Scarf, Sturmfels, Sullivant, Thomas, Weismantel et al. and many others.
- Graver test sets can be exponentially large even in fixed dimension! Very hard to compute, you don't want to do this too often.
- People typically stored as a list of the whole test set and has to search within.
- NP-complete problem to decide whether a list of vectors is a complete Graver bases.
- New Results: There are useful cases where Graver bases become very manageable and efficient.
- BUT WE NEED HIGHLY STRUCTURED MATRICES!!

BAD NEWS!!

- Graver bases contain a Gröbner bases, Hilbert bases. Work by Hosten, Graver, Scarf, Sturmfels, Sullivant, Thomas, Weismantel et al. and many others.
- Graver test sets can be exponentially large even in fixed dimension! Very hard to compute, you don't want to do this too often.
- People typically stored as a list of the whole test set and has to search within.
- NP-complete problem to decide whether a list of vectors is a complete Graver bases.
- New Results: There are useful cases where Graver bases become very manageable and efficient.
- BUT WE NEED HIGHLY STRUCTURED MATRICES!!

BAD NEWS!!

- Graver bases contain a Gröbner bases, Hilbert bases. Work by Hosten, Graver, Scarf, Sturmfels, Sullivant, Thomas, Weismantel et al. and many others.
- Graver test sets can be exponentially large even in fixed dimension! Very hard to compute, you don't want to do this too often.
- People typically stored as a list of the whole test set and has to search within.
- NP-complete problem to decide whether a list of vectors is a complete Graver bases.
- New Results: There are useful cases where Graver bases become very manageable and efficient.
- BUT WE NEED HIGHLY STRUCTURED MATRICES!!

BAD NEWS!!

- Graver bases contain a Gröbner bases, Hilbert bases. Work by Hosten, Graver, Scarf, Sturmfels, Sullivant, Thomas, Weismantel et al. and many others.
- Graver test sets can be exponentially large even in fixed dimension! Very hard to compute, you don't want to do this too often.
- People typically stored as a list of the whole test set and has to search within.
- NP-complete problem to decide whether a list of vectors is a complete Graver bases.
- New Results: There are useful cases where Graver bases become very manageable and efficient.
- BUT WE NEED HIGHLY STRUCTURED MATRICES!!

Special Assumption II : Highly structured Matrices

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! Yes, Transportation problems with fixed number of suppliers!

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

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

Special Assumption II : Highly structured Matrices

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! Yes, Transportation problems with fixed number of suppliers!

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

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

Special Assumption II : Highly structured Matrices

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! Yes, Transportation problems with fixed number of suppliers!

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, an integer vectors b, 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 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.
- **Key Idea (from Algebraic Geometry)** [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.

  The smallest constant $g(A, B)$ possible is the Graver complexity of A, B.

- **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.
- **Key Idea (from Algebraic Geometry)** [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.

  The smallest constant $g(A, B)$ possible is the Graver complexity of A, B.

Proof by Example

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

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

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

- LINEAR 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 NON-LINEAR ideas can be applied in classical problems too! (ASK ME about them!):
    - Hilbert's Nullstellensatz Algorithm in Graph Optimization problems
    - Central Paths of Interior point methods as Algebraic Curves
    - Santos' topological thinking for the Hirsch conjecture
- Tools from Algebra, Number Theory, Functional Analysis, Probability, and Convex 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

- LINEAR 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 NON-LINEAR ideas can be applied in classical problems too! (ASK ME about them!):
  - Hilbert's Nullstellensatz Algorithm in Graph Optimization problems
  - Central Paths of Interior point methods as Algebraic Curves
  - Santos' topological thinking for the Hirsch conjecture
- Tools from Algebra, Number Theory, Functional Analysis, Probability, and Convex 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

- LINEAR 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 NON-LINEAR ideas can be applied in classical problems too! (ASK ME about them!):
  - Hilbert's Nullstellensatz Algorithm in Graph Optimization problems
  - Central Paths of Interior point methods as Algebraic Curves
  - Santos' topological thinking for the Hirsch conjecture
- Tools from Algebra, Number Theory, Functional Analysis, Probability, and Convex 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

- LINEAR 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 NON-LINEAR ideas can be applied in classical problems too! (ASK ME about them!):
  - Hilbert's Nullstellensatz Algorithm in Graph Optimization problems
  - Central Paths of Interior point methods as Algebraic Curves
  - Santos' topological thinking for the Hirsch conjecture
- Tools from Algebra, Number Theory, Functional Analysis, Probability, and Convex 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

- LINEAR 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 NON-LINEAR ideas can be applied in classical problems too! (ASK ME about them!):
  - Hilbert's Nullstellensatz Algorithm in Graph Optimization problems
  - Central Paths of Interior point methods as Algebraic Curves
  - Santos' topological thinking for the Hirsch conjecture
- Tools from Algebra, Number Theory, Functional Analysis, Probability, and Convex 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

- LINEAR 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 NON-LINEAR ideas can be applied in classical problems too! (ASK ME about them!):
  - Hilbert's Nullstellensatz Algorithm in Graph Optimization problems
  - Central Paths of Interior point methods as Algebraic Curves
  - Santos' topological thinking for the Hirsch conjecture
- Tools from Algebra, Number Theory, Functional Analysis, Probability, and Convex 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

- LINEAR 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 NON-LINEAR ideas can be applied in classical problems too! (ASK ME about them!):
  - Hilbert's Nullstellensatz Algorithm in Graph Optimization problems
  - Central Paths of Interior point methods as Algebraic Curves
  - Santos' topological thinking for the Hirsch conjecture
- Tools from Algebra, Number Theory, Functional Analysis, Probability, and Convex 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

- LINEAR 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 NON-LINEAR ideas can be applied in classical problems too! (ASK ME about them!):
  - Hilbert's Nullstellensatz Algorithm in Graph Optimization problems
  - Central Paths of Interior point methods as Algebraic Curves
  - Santos' topological thinking for the Hirsch conjecture
- Tools from Algebra, Number Theory, Functional Analysis, Probability, and Convex 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`.

Merci

Thank you

Gracias