New Geometric Algorithms in Discrete Optimization

Jesús A. De Loera, UC Davis

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

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

September 8, 2009

Le Menu

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

Le Menu

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

Le Menu

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

Le Menu

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

Le Menu

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

Challenges in Discrete Optimization why need for new tools

(in particular from computational geometry and algebraic combinatorics).

At the beginning there was...

Linear programs

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

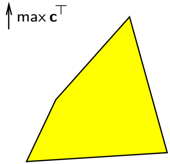
Special integer programs

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

Matrix A is SPECIAL!

Integer programs

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



At the beginning there was...

Linear programs

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

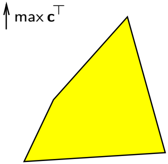
Special integer programs

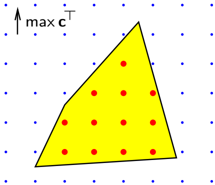
$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$

Matrix A is SPECIAL!

Integer programs

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

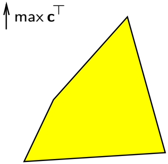




At the beginning there was...

Linear programs

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



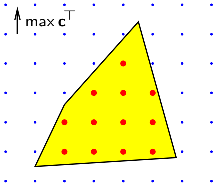
Special integer programs

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

Matrix A is SPECIAL!

Integer programs

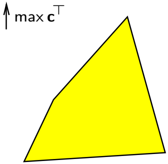
$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$



At the beginning there was...

Linear programs

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



Easy
(polynomial-time solvable)

Special integer programs

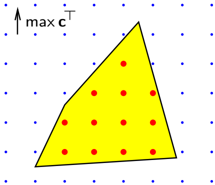
$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$

Matrix A is SPECIAL!

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

Integer programs

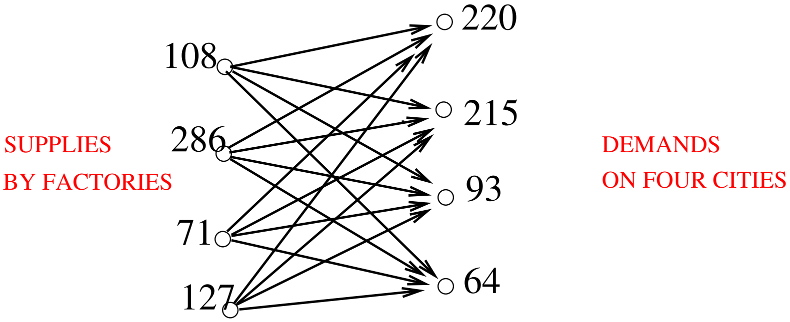
$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \text{all } x_i \text{ integer} \end{aligned}$$



Hard
(NP-hard)

A Useful Example

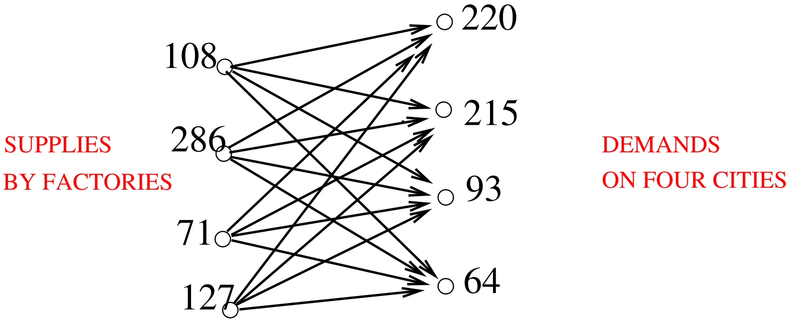
- The Transportation problem: A company builds laptops in four factories, each with certain supply power. Four cities have laptop demands. There is a cost $c_{i,j}$ for transporting a laptop from factory i to city j. What is the best assignment of transport in order to minimize the cost?



- A very special kind of Integer Program. Widely familiar to all, considered easy...

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 very special kind of Integer Program. Widely familiar to all, considered easy...

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 $\sum c_{i,j} x_{i,j}$.

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

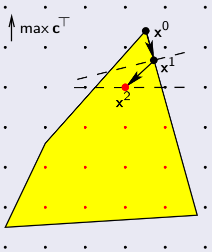
Adhoc methods

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

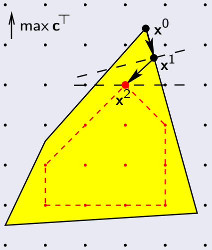
Adhoc methods

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

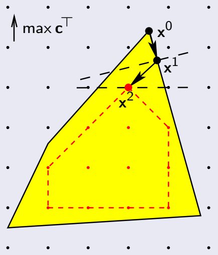
Adhoc methods

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

Mathematical modelling – Strong initial IP formulation

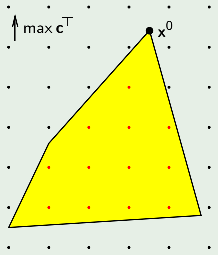
Traditional Algorithms

Dual (polyhedral) techniques



Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

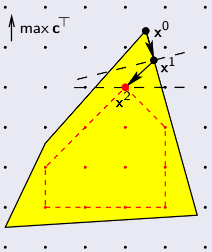
Adhoc methods

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

Mathematical modelling – Strong initial IP formulation

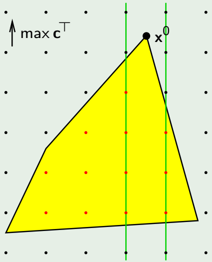
Traditional Algorithms

Dual (polyhedral) techniques



Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

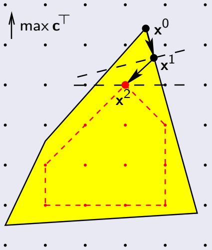
Adhoc methods

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

Mathematical modelling – Strong initial IP formulation

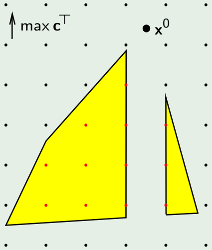
Traditional Algorithms

Dual (polyhedral) techniques



Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

Adhoc methods

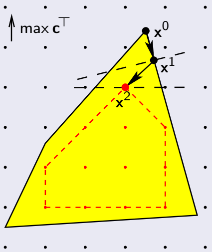
special structure
(e.g. network,
matroids, etc.)

Mathematical modelling – Strong initial IP formulation

Integer Linear Programming: The state of the art

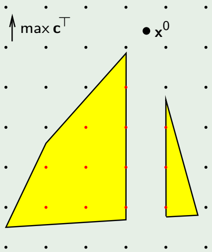
Traditional Algorithms

Dual (polyhedral) techniques



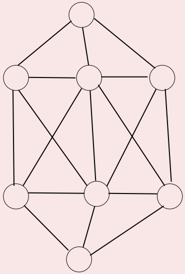
Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

Adhoc methods

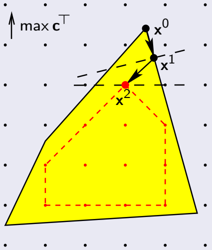


special structure
(e.g. network,
matroids, etc.)

Mathematical modelling – Strong initial IP formulation

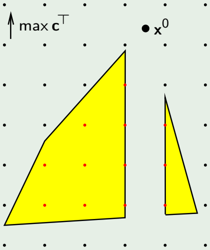
Traditional Algorithms

Dual (polyhedral) techniques



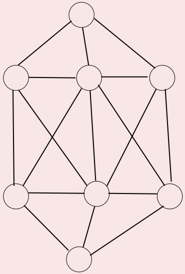
Cutting plane algorithms
– based on polyhedral theory

Enumeration



Branch-and-bound

Adhoc methods



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

Mathematical modelling – Strong initial IP formulation

We wish to handle more complicated Constraints and Objective functions

REASON ONE: Reality is NON-LINEAR!

Non-linear Mixed 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

some x_i integer!

bad news!!

The problem is INCREDIBLY HARD
It is UNDECIDABLE already when f,g_i's are
polynomials and even with number of variables=10.

where

- The constraints f and g_i's can be non-linear functions now!
- Problem has huge modeling power!

REASON ONE: Reality is NON-LINEAR!

Non-linear Mixed 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

some x_i integer!

where

- The constraints f and g_i's can be non-linear functions now!
- Problem has huge modeling power!

bad news!!

The problem is INCREDIBLY HARD
It is UNDECIDABLE already when f,g_i's are polynomials and even with number of variables=10.

How about polyhedral constraints non-linear objective??

Problem type

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

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 GENERAL CONTEXT??

How about polyhedral constraints non-linear objective??

Problem type

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

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 GENERAL CONTEXT??

How about polyhedral constraints non-linear objective??

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 GENERAL CONTEXT??

How about polyhedral constraints non-linear objective??

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 GENERAL CONTEXT??

How about polyhedral constraints non-linear objective??

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 GENERAL CONTEXT??

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 is 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 is 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 is 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.

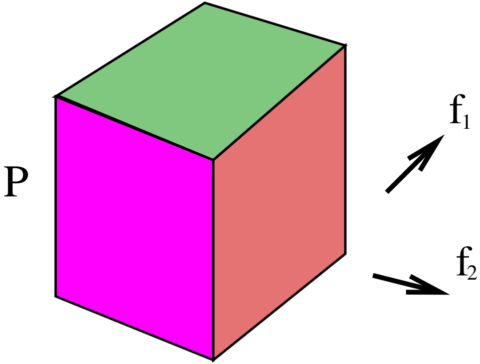
REASON TWO: Multiple criteria optimization!!

Let $A = (a_{ij})$ be an integral $m \times n$-matrix and $\mathbf{b} \in \mathbf{Z}^m$ defining a polytope $P = \{\, \mathbf{u} \in \mathbf{R}^n : A\mathbf{u} \leq \mathbf{b} \,\}$. Given k linear functionals $f_1, f_2, \ldots, f_k \in \mathbf{Z}^n$

$$\begin{array}{rl} \mathit{min} & \left(f_1(\mathbf{u}), \ldots, f_k(\mathbf{u})\right) \\ \text{subject to} & A\mathbf{u} \leq \mathbf{b} \\ & \mathbf{u} \in \mathbf{Z}^n \end{array}$$

where *min* is defined as the problem of finding all Pareto optima and a corresponding Pareto strategy,

First, the Pareto strategies are the lattice points inside P. The Pareto optima are points in the projection, for which no player can decrease a value without increasing one of the criteria

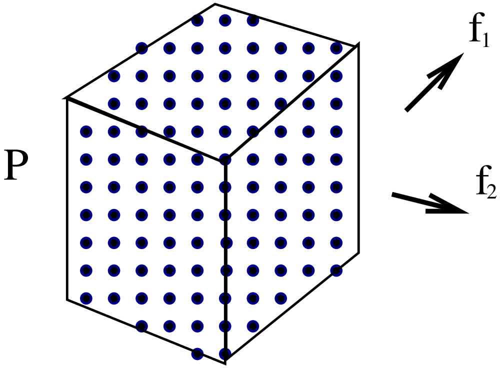


REASON TWO: Multiple criteria optimization!!

Let $A = (a_{ij})$ be an integral $m \times n$-matrix and $\mathbf{b} \in \mathbf{Z}^m$ defining a polytope $P = \{ \mathbf{u} \in \mathbf{R}^n : A\mathbf{u} \leq \mathbf{b} \}$. Given k linear functionals $f_1, f_2, \ldots, f_k \in \mathbf{Z}^n$

$$\begin{aligned} \textit{min} \quad & \left(f_1(\mathbf{u}), \ldots, f_k(\mathbf{u})\right) \\ \text{subject to} \quad & A\mathbf{u} \leq \mathbf{b} \\ & \mathbf{u} \in \mathbf{Z}^n \end{aligned}$$

where *min* is defined as the problem of finding all Pareto optima and a corresponding Pareto strategy, First, the Pareto strategies are the lattice points inside P. The Pareto optima are points in the projection, for which no player can decrease a value without increasing one of the criteria

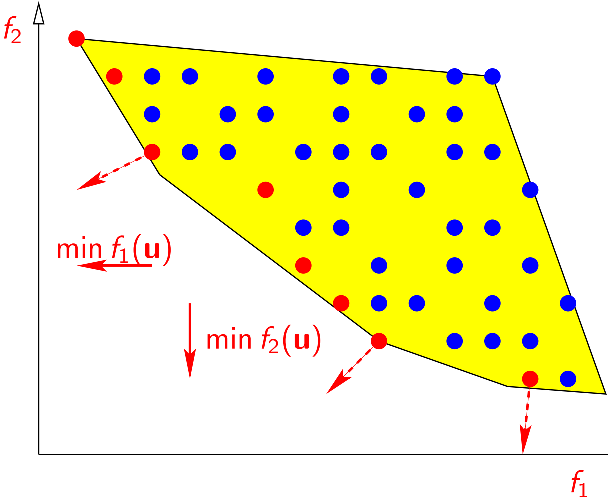


REASON TWO: Multiple criteria optimization!!

Let $A = (a_{ij})$ be an integral $m \times n$-matrix and $\mathbf{b} \in \mathbf{Z}^m$ defining a polytope $P = \{ \mathbf{u} \in \mathbf{R}^n : A\mathbf{u} \leq \mathbf{b} \}$. Given k linear functionals $f_1, f_2, \ldots, f_k \in \mathbf{Z}^n$

$$\begin{array}{rl} \textit{min} & (f_1(\mathbf{u}), \ldots, f_k(\mathbf{u})) \\ \text{subject to} & A\mathbf{u} \leq \mathbf{b} \\ & \mathbf{u} \in \mathbf{Z}^n \end{array}$$

where *min* is defined as the problem of finding all Pareto optima and a corresponding Pareto strategy,
First, the Pareto strategies are the lattice points inside P. The Pareto optima are points in the projection, for which no player can decrease a value without increasing one of the criteria



1. In the traditional transportation problem one cost per edge. Thus we optimize a linear function.
2. but the cost of an edge for the company may not be the same as for an environmentalist. So we get two costs per edge and we are looking to find points where two linear functionals are “minimized”.
3. The two objective functions induce a partial order over the lattice points in the feasible region
4. The multiobjective optimization approach is to find the minimal elements of a partially ordered set.

Example: Multiobjective transportation polytopes

1. In the traditional transportation problem one cost per edge. Thus we optimize a linear function.
2. but the cost of an edge for the company may not be the same as for an environmentalist. So we get two costs per edge and we are looking to find points where two linear functionals are "minimized".
3. The two objective functions induce a partial order over the lattice points in the feasible region
4. The multiobjective optimization approach is to find the minimal elements of a partially ordered set.

Example: Multiobjective transportation polytopes

1. In the traditional transportation problem one cost per edge. Thus we optimize a linear function.
2. but the cost of an edge for the company may not be the same as for an environmentalist. So we get two costs per edge and we are looking to find points where two linear functionals are "minimized".
3. The two objective functions induce a partial order over the lattice points in the feasible region
4. The multiobjective optimization approach is to find the minimal elements of a partially ordered set.

Example: Multiobjective transportation polytopes

1. In the traditional transportation problem one cost per edge. Thus we optimize a linear function.
2. but the cost of an edge for the company may not be the same as for an environmentalist. So we get two costs per edge and we are looking to find points where two linear functionals are "minimized".
3. The two objective functions induce a partial order over the lattice points in the feasible region
4. The multiobjective optimization approach is to find the minimal elements of a partially ordered set.

REASON THREE: Even baby problems unsolvable with traditional techniques!

- Market Share problem (Cornéujols- Dawande, Williams)

minimize $\sum_{i=1}^{m} |s_i|$ subject to the constraints

$$\sum_{j=1}^{n} a_{i,j} x_j + s_i = d_i, \quad i = 1, \ldots, m$$

$$x_j \in \{0, 1\},\ j = 1, \ldots, n,\ \text{and all } s_i \in \text{integer}$$

- Nasty Knapsack problems (Aardal, Bixby et al)

Minimize or maximize $\sum_{i=1}^{10} x_i$, subject to $x_i \geq 0$ and
$3719x_1 + 20289x_2 + 29067x_3 + 60517x_4 + 64354x_5 + 65633x_6 + 76969x_7 + 102024x_8 + 106036x_9 + 119930x_{10} = 13385100$

REASON THREE: Even baby problems unsolvable with traditional techniques!

- Market Share problem (Cornéujols- Dawande, Williams)

minimize $\sum_{i=1}^{m} |s_i|$ subject to the constraints

$$\sum_{j=1}^{n} a_{i,j}x_j + s_i = d_i, \quad i = 1, \ldots, m$$

$$x_j \in \{0,1\},\ j = 1, \ldots, n,\ \text{and all } s_i \in \text{integer}$$

- Nasty Knapsack problems (Aardal, Bixby et al)

Minimize or maximize $\sum_{i=1}^{10} x_i$, subject to $x_i \geq 0$ and
$3719x_1 + 20289x_2 + 29067x_3 + 60517x_4 + 64354x_5 + 65633x_6 + 76969x_7 + 102024x_8 + 106036x_9 + 119930x_{10} = 13385100$

MANY MORE REASONS (not discussed here)

- How to deal with Mixed Integer Variables? Mixed Integer Non-Linear Programming!
- How to deal with Uncertainty, Stochastic? How to deal with error of data? Robustness?
- How to deal with large-scale problems? Heuristics and Approximation?
- In all these topics we also have some results, active ongoing research!! But little time to mention here...
- We will see a rich combination of new tools from discrete mathematics to attack these problems

MANY MORE REASONS (not discussed here)

- How to deal with Mixed Integer Variables? Mixed Integer Non-Linear Programming!
- How to deal with Uncertainty, Stochastic? How to deal with error of data? Robustness?
- How to deal with large-scale problems? Heuristics and Approximation?
- In all these topics we also have some results, active ongoing research!! But little time to mention here...
- We will see a rich combination of new tools from discrete mathematics to attack these problems

MANY MORE REASONS (not discussed here)

- How to deal with Mixed Integer Variables? Mixed Integer Non-Linear Programming!
- How to deal with Uncertainty, Stochastic? How to deal with error of data? Robustness?
- How to deal with large-scale problems? Heuristics and Approximation?
- In all these topics we also have some results, active ongoing research!! But little time to mention here...
- We will see a rich combination of new tools from discrete mathematics to attack these problems

MANY MORE REASONS (not discussed here)

- How to deal with Mixed Integer Variables? Mixed Integer Non-Linear Programming!
- How to deal with Uncertainty, Stochastic? How to deal with error of data? Robustness?
- How to deal with large-scale problems? Heuristics and Approximation?
- In all these topics we also have some results, active ongoing research!! But little time to mention here...
- We will see a rich combination of new tools from discrete mathematics to attack these problems

MANY MORE REASONS (not discussed here)

- How to deal with Mixed Integer Variables? Mixed Integer Non-Linear Programming!
- How to deal with Uncertainty, Stochastic? How to deal with error of data? Robustness?
- How to deal with large-scale problems? Heuristics and Approximation?
- In all these topics we also have some results, active ongoing research!! But little time to mention here...
- We will see a rich combination of new tools from discrete mathematics to attack these problems

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

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!!!**

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).

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).

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

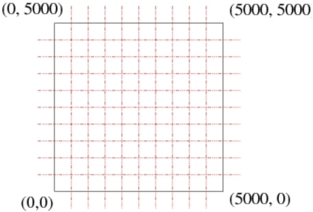
In particular,

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

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

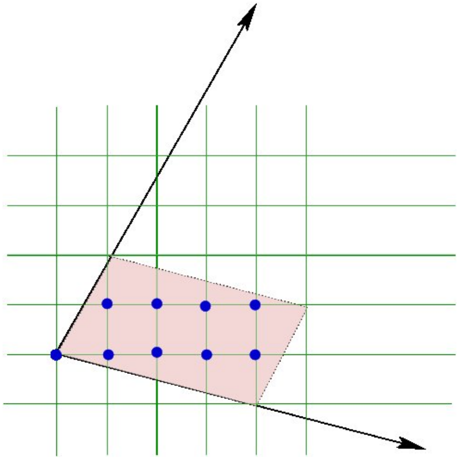
$$\frac{1}{(1-z_1)(1-z_2)} + \frac{{z_1}^{5000}}{(1-{z_1}^{-1})(1-z_2)} + \frac{{z_2}^{5000}}{(1-{z_2}^{-1})(1-z_1)} + \frac{{z_1}^{5000}{z_2}^{5000}}{(1-{z_1}^{-1})(1-{z_2}^{-1})}$$

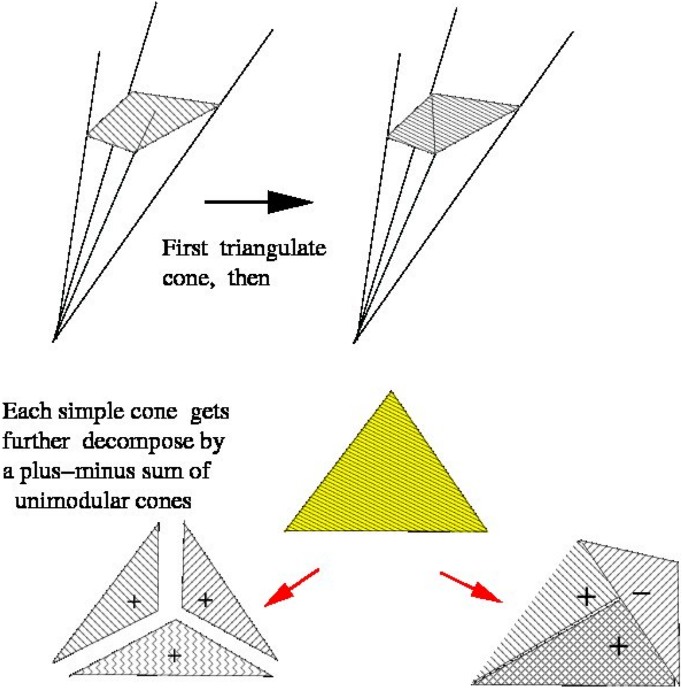
Also, $f(tP, z)$ is

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

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 $f \in \mathbf{Z}[x_1, \ldots, x_d]$ 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 $f \in \mathbf{Z}[x_1, \ldots, x_d]$ 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 $L_k \leq f(\mathbf{x}^{\max}) \leq U_k$ 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
$$\left| f(\mathbf{x}_\epsilon) - f(\mathbf{x}^{\max}) \right| \leq \epsilon f(\mathbf{x}^{\max}).$$

Results on Multiobjective Optimization

Theorem (Counting and enumeration theorem)

Let k and n be fixed integers.

Using the input data $A \in \mathbf{Z}^{m \times n}$, an m-vector $\mathbf{b}$, and linear functions $f_1, \ldots, f_k \in \mathbf{Z}^n$,

(i) *there exists a polynomial-time algorithm to exactly count the Pareto optima;*

(ii) *there exists a polynomial-space polynomial-delay prescribed-order enumeration algorithm to generate the full sequence of Pareto optima ordered lexicographically.*

Theorem (Global-criterion optimization theorem)

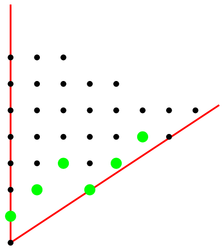
Let the dimension n and the number k of objective functions be fixed.

(i) *There exists a polynomial-time algorithm to find a Pareto optimum $\mathbf{v}$ of (12) that minimizes the distance $\|\mathbf{v} - \hat{\mathbf{v}}\|$ from a prescribed point $\hat{\mathbf{v}} \in \mathbf{Z}^k$ for an arbitrary polyhedral norm.*

(ii) *There exists a fully polynomial-time approximation scheme for the problem of minimizing the Euclidean distance of a Pareto optimum from a prescribed*

A SECOND ALGORITHM: Graver Bases

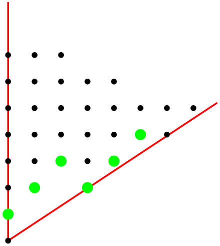
- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b, \ x \geq 0\}$. We will use Computational Geometry and Algebra.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$.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



- Equivalent to the computation of several Hilbert bases computations.

A SECOND ALGORITHM: Graver Bases

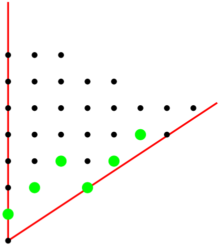
- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b,\ x \geq 0\}$. We will use Computational Geometry and Algebra.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$.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



- Equivalent to the computation of several Hilbert bases computations.

A SECOND ALGORITHM: Graver Bases

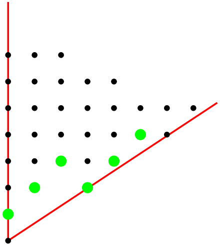
- We are interested on optimization of a convex function over $\{x \in \mathbf{Z}^n : Ax = b, \ x \geq 0\}$. We will use Computational Geometry and Algebra.
- For the lattice $L(A) = \{x \in \mathbf{Z}^n : Ax = 0\}$ introduce a natural partial order on the lattice vectors.
- For $u, v \in \mathbf{Z}^n$.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



- Equivalent to the computation of several Hilbert bases computations.

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 $(-4, -3, 9, 1, -8)$.



- 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

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

.

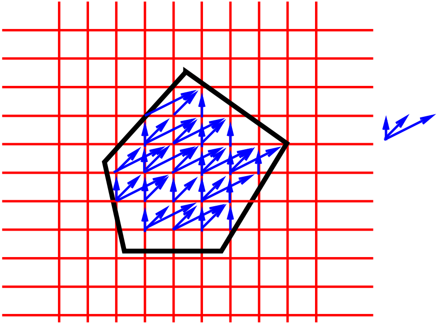
- **Theorem** The Graver basis contains all edges for all integer hulls $conv(\{x|\ Ax = b, x \geq 0,\ x \in \mathbf{Z}^n\})$ as b changes.

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

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

.

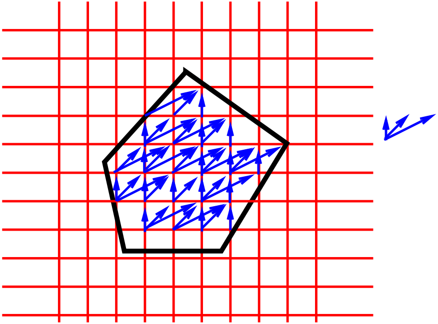
Nodes are lattice points in $L(b)$ and the Graver basis elements give directed edges departing from each lattice point $u \in L(b)$.



- 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

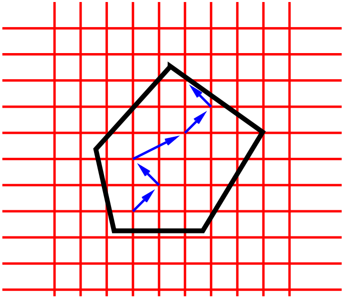
$$L(b) := \{x|\ Ax = b, x \geq 0,\ x \in \mathbf{Z}^n\}$$

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



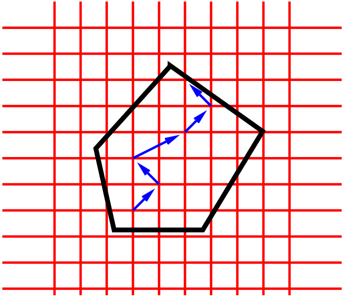
GOOD NEWS: Test Sets and Augmentation Method

- 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.



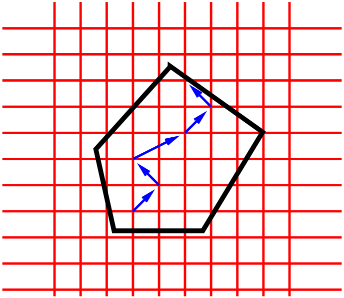
- **Theorem** [J. Graver 1975] Graver bases for A can be used to solve the augmentation problem Given $A \in \mathbf{Z}^{m \times n}$, $x \in \mathbf{N}^n$ and $c \in \mathbf{Z}^n$, either find an improving direction $g \in \mathbf{Z}^n$, namely one with $x - g \in \{y \in \mathbf{N}^n : Ay = Ax\}$ and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

- 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.

BAD NEWS!!

- Graver test sets can be exponentially large even in fixed dimension!
- People typically store a list of the whole test set. Very large indeed. (New ways to store them available, using algebra!).
- Very hard to compute, you don't want to do this too often (NP-hard)
- OUR NEW RESULTS: There are useful cases where Graver bases become very manageable!!!

BAD NEWS!!

- Graver test sets can be exponentially large even in fixed dimension!
- People typically store a list of the whole test set. Very large indeed. (New ways to store them available, using algebra!).
- Very hard to compute, you don't want to do this too often (NP-hard)
- OUR NEW RESULTS: There are useful cases where Graver bases become very manageable!!!

BAD NEWS!!

- Graver test sets can be exponentially large even in fixed dimension!
- People typically store a list of the whole test set. Very large indeed. (New ways to store them available, using algebra!).
- Very hard to compute, you don't want to do this too often (NP-hard)
- OUR NEW RESULTS: There are useful cases where Graver bases become very manageable!!!

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 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.

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

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 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.

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

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 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.

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

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, MULTIOBJECTIVE optimization problems, not just model things linearly anymore.
- 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

- Traditional Methods are not sufficient to solve all current integer optimization models, even the simple linear ones!
- There is demand to solve NON-LINEAR, MULTIOBJECTIVE optimization problems, not just model things linearly anymore.
- 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

- Traditional Methods are not sufficient to solve all current integer optimization models, even the simple linear ones!
- There is demand to solve NON-LINEAR, MULTIOBJECTIVE optimization problems, not just model things linearly anymore.
- 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

- Traditional Methods are not sufficient to solve all current integer optimization models, even the simple linear ones!
- There is demand to solve NON-LINEAR, MULTIOBJECTIVE optimization problems, not just model things linearly anymore.
- 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