Parametric Integer Programming

Part 1

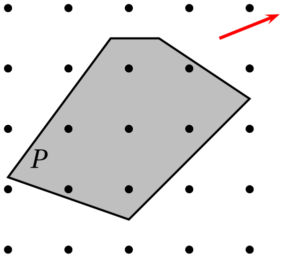
IP in fixed dimension

Integer Programming

IP

Given: Polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ and objective function vector $c \in \mathbb{Z}^n$

Find: Integer point $x \in \mathbb{Z}^n \cap P$ which maximizes or minimizes objective function $c^T x$

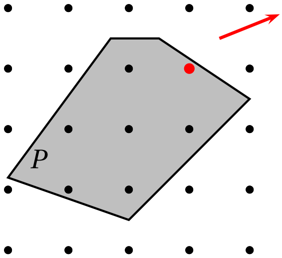


Integer Programming

IP

Given: Polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ and objective function vector $c \in \mathbb{Z}^n$

Find: Integer point $x \in \mathbb{Z}^n \cap P$ which maximizes or minimizes objective function $c^T x$



IP in Fixed Dimension

- Integer programming is NP-complete (Karp 1972, Borosh & Treybig 1976)
- If dimension is fixed, then IP is polynomially solvable (Lenstra 1983)

IP in Fixed Dimension

- Integer programming is NP-complete (Karp 1972, Borosh & Treybig 1976)
- If dimension is fixed, then IP is polynomially solvable (Lenstra 1983)

How does Geometry of Numbers tie in ?

GCDs and IP

Theorem

$\gcd(a,b) = \min\{xa+yb\colon\ x,y\in\mathbb{Z},\ xa+yb\geqslant 1\}$

$$\begin{array}{ll} \textit{minimize} & xa+yb \\ \textit{condition} & xa+yb\geqslant 1 \\ & x,y\in\mathbb{Z}. \end{array}$$

GCDs and IP

Theorem

$\gcd(a,b) = \min\{xa + yb\colon\ x, y \in \mathbb{Z},\ xa + yb \geqslant 1\}$

$$\begin{array}{ll} \textit{minimize} & xa + yb \\ \textit{condition} & xa + yb \geqslant 1 \\ & x, y \in \mathbb{Z}. \end{array}$$

IP with one constraint in dimension 2

Can be solved in time $O(s)$ with Euclidean algorithm

GCDs and IP

Theorem

$\gcd(a,b) = \min\{xa + yb \colon x, y \in \mathbb{Z},\, xa + yb \geqslant 1\}$

$$\begin{array}{ll} \textit{minimize} & xa + yb \\ \textit{condition} & xa + yb \geqslant 1 \\ & x, y \in \mathbb{Z}. \end{array}$$

IP with one constraint in dimension 2

Can be solved in time $O(s)$ with Euclidean algorithm

Two flavors of IP

Combinatorics & Geometry of Numbers

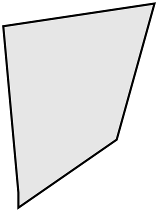
PART 1.1
THE KEY CONCEPT: FLATNESS

Width of a polyhedron P

Width along $d \in \mathbb{R}^n$

Width of $P \subseteq \mathbb{R}^n$ along d

$$w_d(P) = \max\{d^T x \colon x \in P\} - \min\{d^T x \colon x \in P\}$$

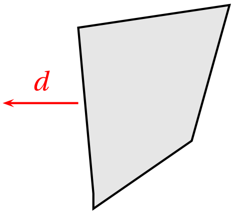


Width of a polyhedron P

Width along $d \in \mathbb{R}^n$

Width of $P \subseteq \mathbb{R}^n$ along d

$$w_d(P) = \max\{d^T x \colon x \in P\} - \min\{d^T x \colon x \in P\}$$

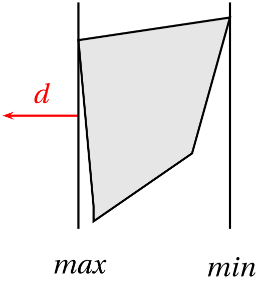


Width of a polyhedron P

Width along $d \in \mathbb{R}^n$

Width of $P \subseteq \mathbb{R}^n$ along d

$$w_d(P) = \max\{d^T x \colon x \in P\} - \min\{d^T x \colon x \in P\}$$



Flatness

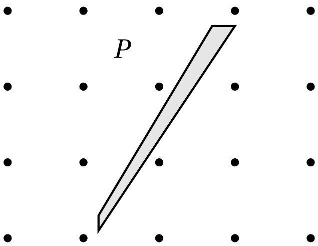
Width of P

$$w(P) = \min_{d \in \mathbb{Z}^n - \{0\}} w_d(P)$$

Theorem (Khinchine's Flatness Theorem)

There exists a constant $\omega(n)$ such that, if $P \cap \mathbb{Z}^n = \emptyset$ then

$$w(P) \leqslant \omega(n).$$



Flatness

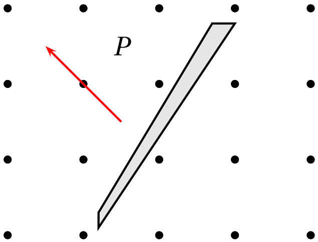
Width of P

$$w(P) = \min_{d \in \mathbb{Z}^n - \{0\}} w_d(P)$$

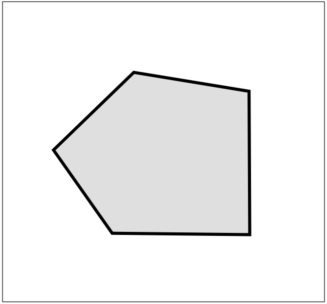
Theorem (Khinchine's Flatness Theorem)

There exists a constant $\omega(n)$ such that, if $P \cap \mathbb{Z}^n = \emptyset$ then

$$w(P) \leqslant \omega(n).$$

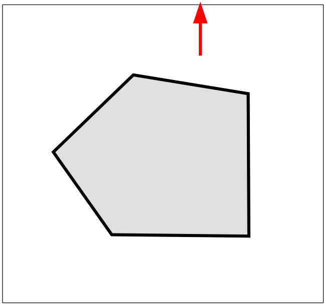


Lenstra's Algorithm



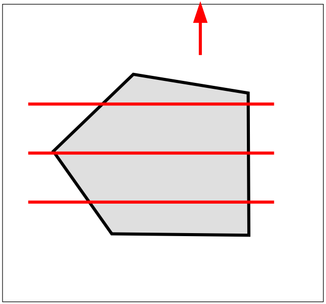
- Algorithm decides $P \cap \mathbb{Z}^n = \emptyset$

Lenstra's Algorithm



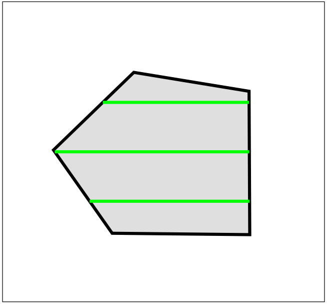
- Algorithm decides $P \cap \mathbb{Z}^n = \emptyset$
- Compute width of P and corresponding integral direction $d \in \mathbb{Z}^d$
- If width too large, then $P \cap \mathbb{Z}^n \neq \emptyset$

Lenstra's Algorithm



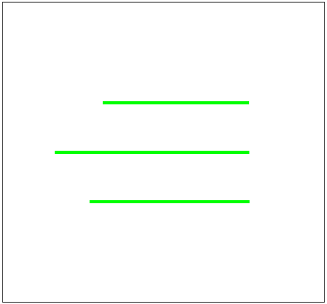
- Algorithm decides $P \cap \mathbb{Z}^n = \emptyset$
- Compute width of P and corresponding integral direction $d \in \mathbb{Z}^d$
- If width too large, then $P \cap \mathbb{Z}^n \neq \emptyset$
- Otherwise search for integer point recursively on one of the hyperplanes $(d^T x = \delta) \cap P$, $\delta \in \mathbb{Z}$

Lenstra's Algorithm



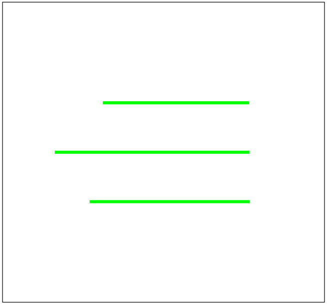
- Algorithm decides $P \cap \mathbb{Z}^n = \emptyset$
- Compute width of P and corresponding integral direction $d \in \mathbb{Z}^d$
- If width too large, then $P \cap \mathbb{Z}^n \neq \emptyset$
- Otherwise search for integer point recursively on one of the hyperplanes $(d^T x = \delta) \cap P$, $\delta \in \mathbb{Z}$

Lenstra's Algorithm



- Algorithm decides $P \cap \mathbb{Z}^n = \emptyset$
- Compute width of P and corresponding integral direction $d \in \mathbb{Z}^d$
- If width too large, then $P \cap \mathbb{Z}^n \neq \emptyset$
- Otherwise search for integer point recursively on one of the hyperplanes $(d^T x = \delta) \cap P$, $\delta \in \mathbb{Z}$

Lenstra's Algorithm



- Algorithm decides $P \cap \mathbb{Z}^n = \emptyset$
- Compute width of P and corresponding integral direction $d \in \mathbb{Z}^d$
- If width too large, then $P \cap \mathbb{Z}^n \neq \emptyset$
- Otherwise search for integer point recursively on one of the hyperplanes $(d^T x = \delta) \cap P$, $\delta \in \mathbb{Z}$

Question

- How to compute a flat direction?

Computing a flat direction of a Simplex

- Simplex $\Sigma = \operatorname{conv}\{0, v_1, \dots, v_n\}$

Computing a flat direction of a Simplex

- Simplex $\Sigma = \text{conv}\{0, v_1, \ldots, v_n\}$
- Width of Σ along d:

$$w_d(\Sigma) = \max\{0, d^T v_1, \ldots, d^T v_n\} - \min\{0, d^T v_1, \ldots, d^T v_n\}$$

Computing a flat direction of a Simplex

- Simplex $\Sigma = \operatorname{conv}\{0, v_1, \ldots, v_n\}$
- Width of Σ along d:

$$w_d(\Sigma) = \max\{0, d^T v_1, \ldots, d^T v_n\} - \min\{0, d^T v_1, \ldots, d^T v_n\}$$

- A matrix with rows $v_1^T, \ldots, v_n^T$ then

$$\|\mathbf{Ad}\|_\infty \leqslant \mathbf{w_d}(\Sigma) \leqslant \mathbf{2}\,\|\mathbf{Ad}\|_\infty$$

Computing a flat direction of a Simplex

- Simplex $\Sigma = \text{conv}\{0, v_1, \ldots, v_n\}$
- Width of Σ along d:

$$w_d(\Sigma) = \max\{0, d^T v_1, \ldots, d^T v_n\} - \min\{0, d^T v_1, \ldots, d^T v_n\}$$

- A matrix with rows $v_1^T, \ldots, v_n^T$ then

$$\|\mathbf{A}\mathbf{d}\|_\infty \leqslant \mathbf{w_d}(\Sigma) \leqslant \mathbf{2}\,\|\mathbf{A}\mathbf{d}\|_\infty$$

- Compute $d \in \mathbb{Z}^n - \{0\}$ s.t. $\|A d\|$ minimal

Computing a flat direction of a Simplex

- Simplex $\Sigma = \operatorname{conv}\{0, v_1, \ldots, v_n\}$
- Width of Σ along d:

$$w_d(\Sigma) = \max\{0, d^T v_1, \ldots, d^T v_n\} - \min\{0, d^T v_1, \ldots, d^T v_n\}$$

- A matrix with rows $v_1^T, \ldots, v_n^T$ then

$$\|\mathbf{A}\mathbf{d}\|_\infty \leqslant \mathbf{w_d}(\Sigma) \leqslant \mathbf{2}\,\|\mathbf{A}\mathbf{d}\|_\infty$$

- Compute $d \in \mathbb{Z}^n - \{0\}$ s.t. $\|Ad\|$ minimal

If d is as above, then there is constant $c_1(n)$ with

$$w(\Sigma) \leqslant w_d(\Sigma) \leqslant c_1(n) \cdot w(\Sigma).$$

Lattices and shortest vectors

$\Lambda(A) = \{Ax\colon\ x \in \mathbb{Z}^n\}$ is lattice generated by $A \in \mathbb{Q}^{n\times n}$

$v \neq 0$ with $\|v\|$ minimal is shortest vector of Λ.

Lattices and shortest vectors

$\Lambda(A) = \{Ax\colon x \in \mathbb{Z}^n\}$ is lattice generated by $A \in \mathbb{Q}^{n \times n}$

$v \neq 0$ with $\|v\|$ minimal is shortest vector of Λ.

With LLL Algorithm (Lenstra, Lenstra & Lovász 1982)

Shortest vector of $\Lambda(A)$

- Can be approximated with factor of $2^{(n-1)/2}$ in polynomial time in varying dimension.
- Can be computed in time $O(s)$ in fixed dimension, where s is binary encoding length of A.

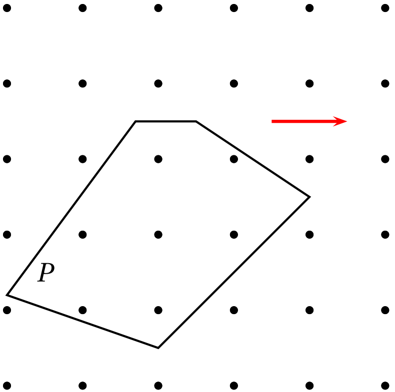
Part 1.2
Vertices of the Integer Hull

Geometric interpretation

- Given a (bounded) Polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$
- Find vertex of the integer hull P_I of P which maximizes objective function $c^T x$

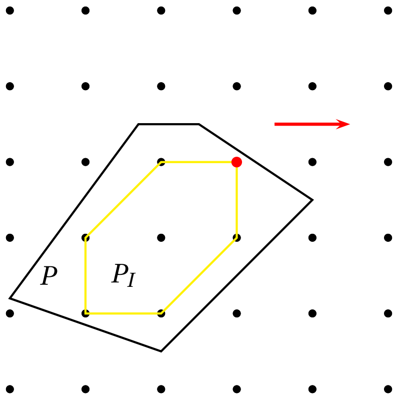
Geometric interpretation

- Given a (bounded) Polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$
- Find vertex of the integer hull P_I of P which maximizes objective function $c^T x$



Geometric interpretation

- Given a (bounded) Polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$
- Find vertex of the integer hull P_I of P which maximizes objective function $c^T x$



Number of vertices

How many extreme points (vertices) can P_I have?

Number of vertices

How many extreme points (vertices) can P_I have?
Consider a Knapsack Polyhedron defined by integral data

$$a(1)x(1) + \cdots + a(n)x(n) \leqslant \beta, \quad x \geqslant 0$$

And two different vertices of P_I

$$(x(1), \ldots, x(n)) \quad \text{and} \quad (y(1), \ldots, y(n))$$

and suppose that $\lfloor \log(x(i)) \rfloor = \lfloor \log(y(i)) \rfloor$ for $i = 1, \ldots, n$.

Number of vertices

How many extreme points (vertices) can P_I have?
Consider a Knapsack Polyhedron defined by integral data

$$a(1)x(1) + \cdots + a(n)x(n) \leqslant \beta, \quad x \geqslant 0$$

And two different vertices of P_I

$$(x(1), \ldots, x(n)) \quad \text{and} \quad (y(1), \ldots, y(n))$$

and suppose that $\lfloor \log(x(i)) \rfloor = \lfloor \log(y(i)) \rfloor$ for $i = 1, \ldots, n$. Then

- $2 \cdot x - y \geqslant 0$ and $2 \cdot y - x \geqslant 0$
- $a^T \left((2 \cdot x - y) + (2 \cdot y - x) \right) = a^T (x + y) \leqslant 2 \cdot \beta$

W.l.o.g. one can assume that $a^T (2 \cdot x - y) \leqslant \beta$.

Number of vertices

How many extreme points (vertices) can P_I have?
Consider a Knapsack Polyhedron defined by integral data

$$a(1)x(1) + \cdots + a(n)x(n) \leqslant \beta, \quad x \geqslant 0$$

And two different vertices of P_I

$$(x(1), \dots, x(n)) \quad \text{and} \quad (y(1), \dots, y(n))$$

and suppose that $\lfloor \log(x(i)) \rfloor = \lfloor \log(y(i)) \rfloor$ for $i = 1, \dots, n$. Then

- $2 \cdot x - y \geqslant 0$ and $2 \cdot y - x \geqslant 0$
- $a^T \big((2 \cdot x - y) + (2 \cdot y - x)\big) = a^T (x + y) \leqslant 2 \cdot \beta$

W.l.o.g. one can assume that $a^T (2 \cdot x - y) \leqslant \beta$.
But then $1/2(2 \cdot x - y) + 1/2 \cdot y = x$ which contradicts that x is a vertex.

The number of vertices is polynomial

- Consider simplex with vertex 0

$$S = \{x \in \mathbb{R}^n \mid Bx \geqslant 0,\ a^T x \leqslant \beta\}$$

  with $B \in \mathbb{Z}^{n\times n}$ invertible.
- $S = \{x \in \mathbb{R}^n \mid Bx \geqslant 0,\ (B^{-1} a)^T (Bx) \leqslant \beta\}$
- $x \in \mathbb{Z}^n$ is vertex of S_I if and only if Bx is vertex of $\mathrm{conv}(K \cap \Lambda(B))$ with

$$K = \{x \in \mathbb{R}^n \mid x \geqslant 0,\ (B^{-1} a)^T x \leqslant \beta\}$$

  and

$$\Lambda(B) = \{Bx \mid x \in \mathbb{Z}^n\}.$$

The number of extreme points is polynomial

By triangulation of P:

Theorem 1.1 (Shevchenko 1981, Hayes & Larman 1983, Schrijver 1986)

Let $Ax \leqslant b$ be an integral system of inequalities, where $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$ and n is fixed. The integer hull P_I of $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ has a polynomial number of extreme points.

polynomial in binary encoding length of A and b

The number of extreme points is polynomial

By triangulation of P:

Theorem 1.1 (Shevchenko 1981, Hayes & Larman 1983, Schrijver 1986)

Let $Ax \leqslant b$ be an integral system of inequalities, where $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$ and n is fixed. The integer hull P_I of $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ has a polynomial number of extreme points. $O(m^n \cdot s^n)$

polynomial in binary encoding length of A and b

Tight bounds for simplices: Bárány, Howe & Lovász 1992

Cook, Hartmann, Kannan & McDiarmid 1992

PART 1.3
COMPLEXITY OF IP

Complexity of IP

Theorem (Lenstra 1983)

An IP can be solved in polynomial time in fixed dimension.

Complexity model:

- Arithmetic model: Count number of arithmetic operations
- Size of numbers in course of algorithm has to remain small
- s: Binary encoding length of largest coefficient

Complexity of IP

Theorem (Lenstra 1983)

An IP can be solved in polynomial time in fixed dimension.

Complexity model:

- Arithmetic model: Count number of arithmetic operations
- Size of numbers in course of algorithm has to remain small
- s: Binary encoding length of largest coefficient

Running time

- $2^{O(n^3)} \cdot poly(s)$ (Lenstra using LLL)
- $2^{O(n \log n)} \cdot poly(s)$ (Kannan 1987)

Complexity of IP in fixed Dimension

m: Number of constraints

s: Largest binary encoding length of number in input

Complexity of IP in fixed Dimension

m: Number of constraints

s: Largest binary encoding length of number in input

- $O(m+s)$ for feasibility
- $O(s\cdot(m+s))$ for optimization (Lenstra 1983)

Complexity of IP in fixed Dimension

m: Number of constraints
s: Largest binary encoding length of number in input

- $O(m+s)$ for feasibility
- $O(s\cdot(m+s))$ for optimization (Lenstra 1983)

Theorem (E. 2003)

IP in fixed dimension can be solved in expected time $O(m+s\cdot\log m)$.

Matches running time of Euclidean algorithm if m is fixed

Complexity of IP

Open Problems

- Is there a deterministic $O(m+s)$ algorithm ?

Complexity of IP

Open Problems

- Is there a deterministic $O(m+s)$ algorithm ?
  Answer is yes in 2-D (E. & Laue 2005)

Complexity of IP

Open Problems

- Is there a deterministic $O(m+s)$ algorithm ?
  Answer is yes in 2-D (E. & Laue 2005)
- Bit complexity: Is $O(ms^2)$ reachable with naive arithmetic ?
  (Nguyen & Stehlé 2005)
- Is there a $2^{O(n)}$-algorithm for IP in varying dimension?
  SV: (Ajtai, Kumar & Sivakumar 2001)

Part 2
Parameterized IP

$\forall\,\exists$-Statements

Frobenius Problem

Given: $a_1, \ldots, a_n \in \mathbb{Z}$ with $\gcd(a_1, \ldots, a_n) = 1$
Compute: Largest $t \in \mathbb{N}$ which cannot be written as

$$x_1 \cdot a_1 + \cdots + x_n \cdot a_n = t, \quad x_1, \ldots, x_n \in \mathbb{N}_0$$

$\forall\,\exists$-Statements

Frobenius Problem

Given: $a_1, \ldots, a_n \in \mathbb{Z}$ with $\gcd(a_1, \ldots, a_n) = 1$
Compute: Smallest N such that the following formula holds

$$\forall y \in \mathbb{Z}, y \geqslant N \quad \exists x_1, \ldots, x_n \in \mathbb{N}_0 \quad : \quad y = x_1 \cdot a_1 + \cdots + x_n a_n$$

$\forall\,\exists$-Statements

Frobenius Problem

Given: $a_1, \ldots, a_n \in \mathbb{Z}$ with $\gcd(a_1, \ldots, a_n) = 1$
Compute: Smallest N such that the following formula holds

$$\forall y \in \mathbb{Z}, y \geq N \quad \exists x_1, \ldots, x_n \in \mathbb{N}_0 \quad : \quad y = x_1 \cdot a_1 + \cdots + x_n a_n$$

$\forall\,\exists$-statements

Given: Polyhedron $Q \subseteq \mathbb{R}^m$, $A \in \mathbb{Z}^{m \times n}$, $t \in \mathbb{N}$
Does the following hold?:

$$\forall b \in \left(Q \cap (\mathbb{R}^{m-t} \times \mathbb{Z}^t)\right) \quad Ax \leq b \text{ is IP-feasible}$$

$\forall\,\exists$-Statements

Frobenius Problem

Given: $a_1, \ldots, a_n \in \mathbb{Z}$ with $\gcd(a_1, \ldots, a_n) = 1$
Compute: Smallest N such that the following formula holds

$$\forall y \in \mathbb{Z}, y \geqslant N \quad \exists x_1, \ldots, x_n \in \mathbb{N}_0 \quad : \quad y = x_1 \cdot a_1 + \cdots + x_n a_n$$

$\forall\,\exists$-statements

Given: Polyhedron $Q \subseteq \mathbb{R}^m$, $A \in \mathbb{Z}^{m \times n}$, $t \in \mathbb{N}$
Does the following hold?:

$$\forall b \in \left(Q \cap (\mathbb{R}^{m-t} \times \mathbb{Z}^t)\right) \quad Ax \leqslant b \text{ is IP-feasible}$$

Theorem (Kannan 1992)

If n, t and $\dim(\mathbf{Q})$ are fixed, then $\forall\,\exists$-statements can be decided in polynomial time.

$\forall\,\exists$-Statements

Frobenius Problem

Given: $a_1, \dots, a_n \in \mathbb{Z}$ with $\gcd(a_1, \dots, a_n) = 1$
Compute: Smallest N such that the following formula holds

$$\forall y \in \mathbb{Z}, y \geqslant N \quad \exists x_1, \dots, x_n \in \mathbb{N}_0 \quad : \quad y = x_1 \cdot a_1 + \cdots + x_n a_n$$

$\forall\,\exists$-statements

Given: Polyhedron $Q \subseteq \mathbb{R}^m$, $A \in \mathbb{Z}^{m \times n}$, $t \in \mathbb{N}$
Does the following hold?:

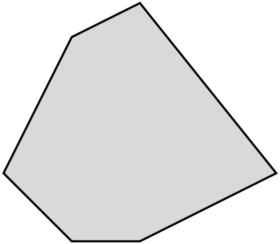
$$\forall b \in \left(Q \cap (\mathbb{R}^{m-t} \times \mathbb{Z}^t)\right) \quad Ax \leqslant b \text{ is IP-feasible}$$

Theorem (E. & Shmonin 2007)

If n, t are fixed, then $\forall\,\exists$-statements can be decided in polynomial time.

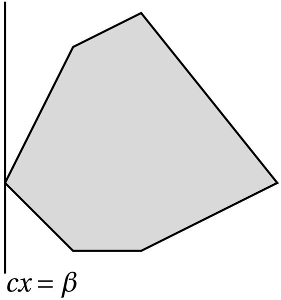
Strengthening of flatness theorem

- Suppose $w(P) = w_c(P)$ with $c \in \mathbb{Z}^n$



Strengthening of flatness theorem

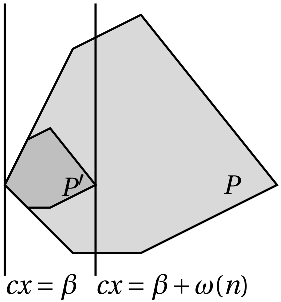
- Suppose $w(P) = w_c(P)$ with $c \in \mathbb{Z}^n$
- Let $\beta = \min\{c^T x : x \in P\}$



Strengthening of flatness theorem

- Suppose $w(P) = w_c(P)$ with $c \in \mathbb{Z}^n$
- Let $\beta = \min\{c^T x \colon x \in P\}$
- If $w(P) > \omega(n)$, then there exists integer point in

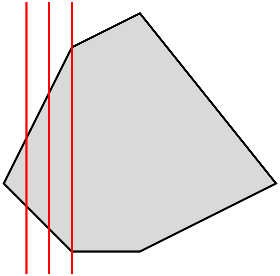
$$P \cap \left(\beta \leqslant c^T x \leqslant \beta + \omega(n)\right)$$



Strengthening of flatness theorem

- Suppose $w(P) = w_c(P)$ with $c \in \mathbb{Z}^n$
- Let $\beta = \min\{c^T x \colon x \in P\}$
- If $w(P) > \omega(n)$, then there exists integer point in

$$P \cap \left(\beta \leqslant c^T x \leqslant \beta + \omega(n)\right)$$



Consequence

P is IP-feasible if and only if at least one of the polyhedra

$$P \cap \left(c^T x = \lceil \beta \rceil + i\right) \quad i = 0, \ldots, \omega(n)$$

IP-feasible.

Simplification

Assumptions

$Q \subseteq \mathbb{R}^m$ polyhedron such that

- $w_{e_1}(P_b) = w(P_b)$ for each $b \in Q$
- $\min\{e_1^T x \colon x \in P_b\} = e_1^T N b$ for some matrix N
- Highest constraint pointing up on line

$$x_1 = \lceil e_1^T N b \rceil + i \quad \text{is} \quad a_{i_j}^T x \leqslant b_{i_j}$$

for $i = 0, \ldots, \omega(2)$

We can write down a fixed number of candidate solutions with mixed integer programs such that, if none of them is feasible, then P_b is IP infeasible.

MIP for i-th candidate

$$
\begin{aligned}
& e_1^T N b \leqslant z < e_1^T N b + 1 \\
& x(1) = z + i \\
& y = \left(b(i_j) - a_{i_j}(1)x(1)\right) / a_{i_j}(2) \\
& y \leqslant x(2) < y + 1 \\
& x(1), x(2), z, y \text{ integral.}
\end{aligned}
$$

Kannan's partitioning algorithm

Partitions the space of right-hand-sides into polynomial number of polyhedra, such that these assumptions can be made.

A key lemma

Lemma (Kannan 1992)

Given: $A \in \mathbb{Z}^{m \times n}$ *and polyhedron* $Q \subseteq \mathbb{R}^m$, *with* n *and* $\dim(Q)$ *fixed*
There exists polynomial algorithm which computes $D \subseteq \mathbb{Z}^n$ *such that for all* $b \in Q$

$$\exists d \in D\colon\ w_d(P_b) \leqslant 2 \cdot w(P_b)$$

A key lemma

Lemma (Kannan 1992)

Given: $A \in \mathbb{Z}^{m \times n}$ *and polyhedron* $Q \subseteq \mathbb{R}^m$, *with* n *and* $\dim(Q)$ *fixed*
There exists polynomial algorithm which computes $D \subseteq \mathbb{Z}^n$ *such that for all* $b \in Q$

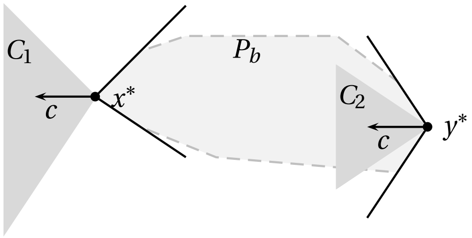
$$\exists d \in D\colon\ w_d(P_b) \leqslant 2 \cdot w(P_b)$$

Lemma (E. & Shmonin 2007)

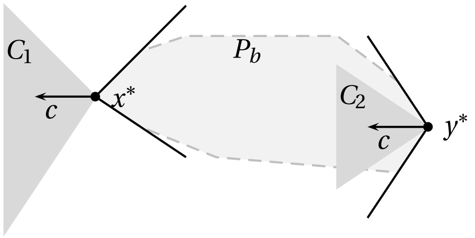
Given: $A \in \mathbb{Z}^{m \times n}$ *with* n *fixed*
There exists polynomial algorithm which computes $D \subseteq \mathbb{Z}^n$ *such that for all* $b \in Q$

$$\exists d \in D\colon\ w_d(P_b) = w(P_b).$$

Sketch of Proof

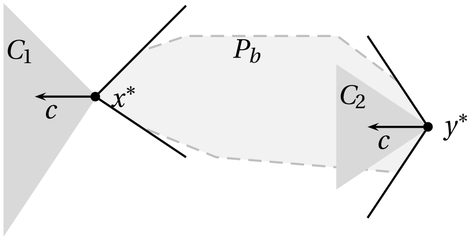


Sketch of Proof



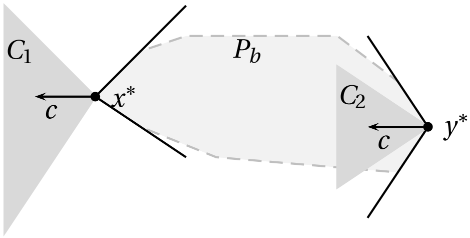
- Width direction c is contained in two cones C_1 and C_2

Sketch of Proof



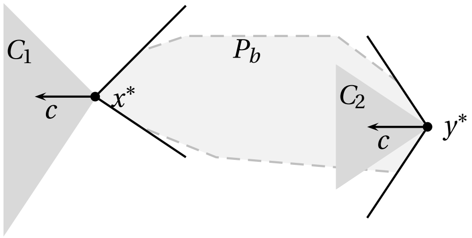
- Width direction c is contained in two cones C_1 and C_2
- c is optimal solution of IP
  $\min\left\{(x^* - y^*)^T d \colon d \in \mathbb{Z}^n \cap C_1 \cap C_2 - \{0\}\right\}$

Sketch of Proof



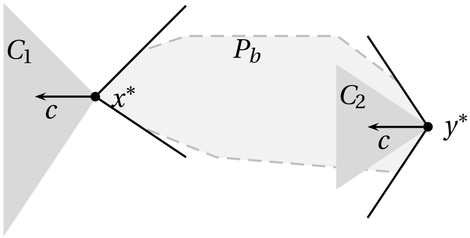
- Width direction c is contained in two cones C_1 and C_2
- c is optimal solution of IP
  $\min\left\{(x^* - y^*)^T d \colon d \in \mathbb{Z}^n \cap C_1 \cap C_2 - \{0\}\right\}$
- c can be replaced by vertex of $\operatorname{conv}\left(\mathbb{Z}^n \cap C_1 \cap C_2 - \{0\}\right)$

Sketch of Proof



- Width direction c is contained in two cones C_1 and C_2
- c is optimal solution of IP
  $\min\left\{(x^* - y^*)^T d \colon d \in \mathbb{Z}^n \cap C_1 \cap C_2 - \{0\}\right\}$
- c can be replaced by vertex of $\operatorname{conv}\left(\mathbb{Z}^n \cap C_1 \cap C_2 - \{0\}\right)$
- Number of vertices is polynomial in fixed dimension
  (Shevchenko 1981, Hayes & Larman 1983, Cook, Hartmann, Kannan, McDiarmid 1992)

Sketch of Proof



- Width direction c is contained in two cones C_1 and C_2
- c is optimal solution of IP
  $\min\left\{(x^* - y^*)^T d\colon\ d \in \mathbb{Z}^n \cap C_1 \cap C_2 - \{0\}\right\}$
- c can be replaced by vertex of $\operatorname{conv}\left(\mathbb{Z}^n \cap C_1 \cap C_2 - \{0\}\right)$
- Number of vertices is polynomial in fixed dimension (Shevchenko 1981, Hayes & Larman 1983, Cook, Hartmann, Kannan, McDiarmid 1992)
- Can be computed in polynomial time

First partitioning step

Width direction is invariant

- Compute polynomial number of triples

$$(d_1, F_1, G_1), \ldots, (d_k, F_k, G_k)$$

  such that for each $b \in \mathbb{R}^m$ there exists index i with
    - $w(P_b) = w_{d_i}(P_b)$
    - $\max\{d_i^T x : x \in P_b\} = d_i^T F_i b$ and $\min\{d_i^T x : x \in P_b\} = d_i^T G_i b$
    - $w(P_b) = d_i^T (F_i - G_i) b$
- The b's corresponding to i are a polyhedron

$$d_i^T (F_i - G_i) b \leqslant d_j^T (F_j - G_j) b \text{ for all } i \neq j.$$

Second partitioning step

Fix the active constraints pointing up

- $\omega(2)$ vertical lines
- For each, we fix the highest constraint pointing up
- $\binom{m}{\omega(2)}$ choices (polynomial)
- Write down linear constraints which partition right-hand-sides

Partitioning Theorem

We sketched the proof of the following theorem for dimension 2.

Theorem 2.1 (E. & Shmonin 2007)

$A \in \mathbb{Z}^{m\times n}$ of full column rank; n fixed.
One can compute in polynomial time a partition of $S_1,\ldots,S_t$ of $\mathbb{R}^m$ together with a fixed number of mixed-integer-programs $A_{ij}b + B_{ij}x + C_{ij}y \leqslant d_{ij}$ for each $i = 1,\ldots,t$ (with a fixed number of integer variables) such that the following holds.

> *For any $b^* \in S_i$, $P_{b^*} \cap \mathbb{Z}^n \neq \emptyset$ if and only if P_{b^*} contains at least one integer vector x determined by an associated Mixed-Integer-Program $A_{ij}b^* + B_{i,j}x + C_{i,j}y \leqslant d_{i,j}$*

Deciding $\forall\,\exists$-statements

$\forall\,\exists$-statements

Given: Polyhedron $Q \subseteq \mathbb{R}^m$, $A \in \mathbb{Z}^{m\times n}$, $t \in \mathbb{N}$
Does the following hold?:

$$\forall b \in \left(Q \cap (\mathbb{R}^{m-t} \times \mathbb{Z}^t)\right) \quad Ax \leqslant b \text{ is IP-feasible}$$

With partitioning theorem

We can assume that there exists a fixed number of mixed integer programs $A_j b + B_j x + C_j y \leqslant d_j$ $j = 1, \ldots, k$ such that solution for b is computed by one of these MIPs.

Deciding ∀ ∃-statements

Searching for a b

- We search a b such that all candidate solutions are infeasible
- To each candidate solution, assign a constraint to be violated; $\binom{m}{k}$ choices (polynomial)
- For each choice, check whether all candidate solutions violate corresponding constraint (MIP in fixed dimension)

Deciding $\forall\,\exists$-statements

Searching for a b

- We search a b such that all candidate solutions are infeasible
- To each candidate solution, assign a constraint to be violated; $\binom{m}{k}$ choices (polynomial)
- For each choice, check whether all candidate solutions violate corresponding constraint (MIP in fixed dimension)

Theorem (E. & Shmonin 2007)

If n, t are fixed, then $\forall\,\exists$-statements can be decided in polynomial time.

Consequences and related Results

Hilbert Bases

- Hilbert-Basis test in fixed dimension is in P (Cook, Lovász & Schrijver 1984)
- If co-dimension is fixed ($d+k$ elements in $\mathbb{R}^d$, where k fixed), HB-test is parametric IP in fixed dimension (Sebő 1999)

Consequences and related Results

Hilbert Bases

- Hilbert-Basis test in fixed dimension is in P (Cook, Lovász & Schrijver 1984)
- If co-dimension is fixed ($d + k$ elements in $\mathbb{R}^d$, where k fixed), HB-test is parametric IP in fixed dimension (Sebő 1999) and thus in P

Consequences and related Results

Hilbert Bases

- Hilbert-Basis test in fixed dimension is in P (Cook, Lovász & Schrijver 1984)
- If co-dimension is fixed ($d+k$ elements in $\mathbb{R}^d$, where k fixed), HB-test is parametric IP in fixed dimension (Sebő 1999) and thus in P

Generating functions

- Rational generating function of integer points in polyhedra can be computed in polynomial time in fixed dimension (Barvinok 1994)
- Köppe & Verdoolaege (2007) compute generating functions of parameterized polyhedra in fixed dimension

Open Problem

Is the following problem in P?
Given $A \in \mathbb{Z}^{m \times n}$ and polyhedron $Q \subseteq \mathbb{R}^m$, where n is fixed, compute $b \in Q$ with number of integer points in $Ax \leqslant b$ is minimal.