Generating Functions for Sets of Lattice Points

Kevin M. Woods

The Frobenius Problem

Let $a_1, \ldots, a_d$ be positive integers such that $\gcd(a_1, \ldots, a_d) = 1$.

Let $S = \{\lambda_1 a_1 + \cdots + \lambda_d a_d \big| \lambda_i \in \mathbb{Z}_{\geq 0}\}$, the semi-group generated by the a_i's.

Example: $a_1 = 3, a_2 = 7$. Then

$$S = \{0, 3, 6, 7, 9, 10, 12, 13, 14, \ldots\}.$$

All sufficiently large integers are in S.

Questions:

1. What is the largest integer not in S?

2. How many positive integers are not in S?

Algorithmic slant: can we answers these questions "quickly"?
(Question 1 previously solved by Kannan)

Our Approach: Generating Functions

Define

$$f(S;x) := \sum_{a \in S} x^a.$$

In our example, $f(S;x) = 1+x^3+x^6+x^7+\cdots$.
Want to find (quickly) a simple formula for $f(S;x)$.

To answer question 2 (how many positive integers are not in S), specialize

$$\frac{1}{1-x} - f(S;x)$$

at $x=1$.

$x=1$ is a pole of the fractions.
Idea: look at points near 1.

In our example, we could write

$$f(S;x) = x^3 + x^6 + x^7 + x^9 + x^{10} + \frac{x^{12}}{1-x},$$

but this is too long, in general.

When $d = 2$, can get

$$f(S;x) = \frac{1 - x^{a_1 a_2}}{(1-x^{a_1})(1-x^{a_2})}.$$

When $d = 3$, can get

$$f(S;x) = \frac{\pm 6 \text{ monomials}}{(1-x^{a_1})(1-x^{a_2})(1-x^{a_3})}. \text{ (Denham)}$$

When $d = 4$, could have

$$f(S;x) = \frac{\sqrt{t} \text{ monomials}}{(1-x^{a_1})(1-x^{a_2})(1-x^{a_3})(1-x^{a_4})},$$

where $t = \min\{a_1, a_2, a_3, a_4\}$. (Székely, Wormald).

$\sqrt{t}$ is too many. We want something like $\log t$ or $(\log t)^{10}$.

“Quick” Algorithms

We want an algorithm that inputs $a_1, a_2, \ldots, a_d$ and outputs $f(S; x)$.

The *input size* is the number of bits needed to encode the input for the algorithm.

Here, input size is approximately

$$(1 + \log_2(a_1)) + \cdots + (1 + \log_2(a_d))$$

$$= d + \sum_{i=1}^{d} \log_2(a_i).$$

An algorithm is *polynomial time* if there is a polynomial p such that the algorithm runs in at most $p(input\ size)$ steps.

General Problem

Fix d.
Let $c_1, \dots, c_n \in \mathbb{Z}^d$ and $b_1, \dots, b_n \in \mathbb{Z}$ be given.
Define a rational polyhedron P by

$$P = \{x \in \mathbb{R}^d \big| \langle c_i, x \rangle \leq b_i, \forall i\}.$$

Input size of P is approximately

$$nd + \sum \log_2 |c_{ij}| + \sum \log_2 |b_i|.$$

Let T be a linear transformation $\mathbb{R}^d \to \mathbb{R}^k$, such that $T(\mathbb{Z}^d) \subset \mathbb{Z}^k$.

Input size of $T = (t_{ij})$ is approximately

$$dk + \sum \log_2 |t_{ij}|.$$

For $S \in \mathbb{Z}^d$ define

$$f(S; \mathbf{x}) = \sum_{s=(s_1,\dots,s_d)\in S} x_1^{s_1} x_2^{s_2} \cdots x_d^{s_d} = \sum_{s\in S} \mathbf{x}^s.$$

Corollary 1 *For fixed d, there is a constant $s = s(d)$ and a polynomial time algorithm which, given $a_1, \ldots, a_d$, writes $f(S; x)$ in the form*

$$f(S;x) = \sum_{i \in I} \alpha_i \frac{x^{p_i}}{(1 - x^{q_{i1}}) \cdots (1 - x^{q_{is}})},$$

where $\alpha_i \in \mathbb{Q}$ and $p_i, q_{ij} \in \mathbb{Z}$.

In particular, the number of terms is bounded by a polynomial in the input size.

We have $s \approx d^d$.

Theorem 1 *(Barvinok) For fixed d, there exists a polynomial time algorithm which, given a rational polyhedron P, computes $f(S;\mathbf{x})$, where $S = P \cap \mathbb{Z}^d$, in the form*

$$\sum_{i \in I} \pm \frac{\mathbf{x}^{p_i}}{(1-\mathbf{x}^{q_{i1}}) \cdots (1-\mathbf{x}^{q_{id}})},$$

where $p_i, q_{ij} \in \mathbb{Z}^d$.

Example: $P = \{x \,\big|\, 0 \le x \le N\}$. Then

$$f(S;x) = 1 + x + \cdots + x^N = \frac{1}{1-x} - \frac{x^{N+1}}{1-x}.$$

Applying to Frobenius Problem

$S = \{\lambda_1 a_1 + \cdots + \lambda_d a_d \big| \lambda_i \in \mathbb{Z}_{\geq 0}\}$.

Let $T : (\lambda_1, \ldots, \lambda_d) \mapsto \lambda_1 a_1 + \cdots + \lambda_d a_d$. Then $T(\mathbb{R}^d_{\geq 0} \cap \mathbb{Z}^d) = S$.

Can't technically apply theorem unless P is bounded. But we can fix this, because only a bounded piece of S is interesting.

Let N be bigger than largest integer not in S (e.g. $N = a_1 a_2 \cdots a_d$). Let

$$P = \{(\lambda_1, \ldots, \lambda_d) \big| \lambda_i \geq 0 \text{ and } \sum_{i=1}^{d} \lambda_i a_i \leq N - 1\}.$$

Then

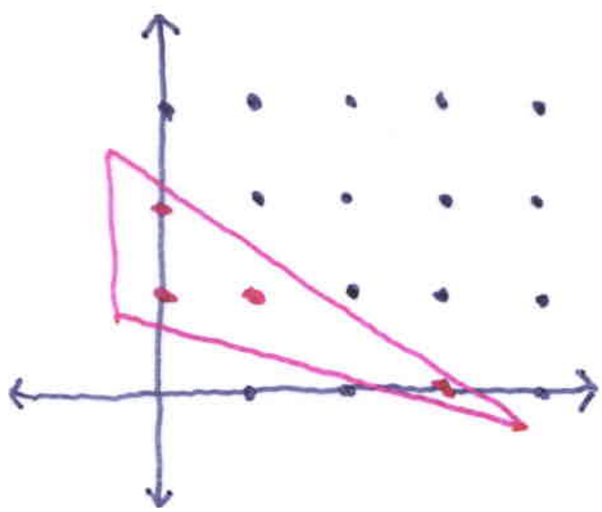
$$S = T(P \cap \mathbb{Z}^d) \dot{\cup} \{N, N+1, \ldots\}.$$

Theorem 2 *(-) For fixed d, there exists a positive integer $s = s(d)$ and a polynomial time algorithm which, given a rational polytope (i.e., bounded polyhedron) P and a linear transformation $T : \mathbb{R}^d \to \mathbb{R}^k$ such that $T(\mathbb{Z}^d) \subset \mathbb{Z}^k$, computes $f(S; \mathbf{x})$, where $S = T(P \cap \mathbb{Z}^d)$, in the form*

$$\sum_{i \in I} \alpha_i \frac{\mathbf{x}^{p_i}}{(1 - \mathbf{x}^{q_{i1}}) \cdots (1 - \mathbf{x}^{q_{is}})},$$

where $\alpha_i \in \mathbb{Q}$ and $p_i, q_{ij} \in \mathbb{Z}^d$.

Usually, T is a projection of some sort.
Example: $T(x, y) = x$.



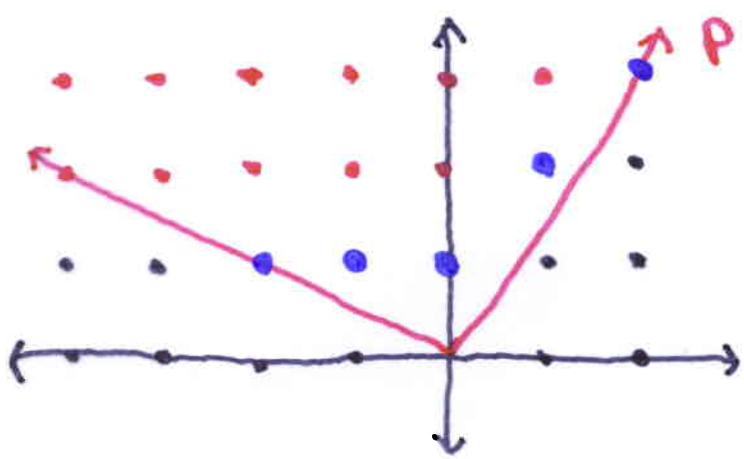
$f(S, x) = 1 + x + x^3$.

Hilbert Bases

Let $a_1, \ldots, a_d \in \mathbb{Z}^d$ be linearly independent vectors.
Let $K = \{\mu_1 a_1 + \cdots + \mu_d a_d \big| \mu_i \in \mathbb{R}_{\geq 0}\}$, the cone generated by $a_1, \ldots, a_d$.

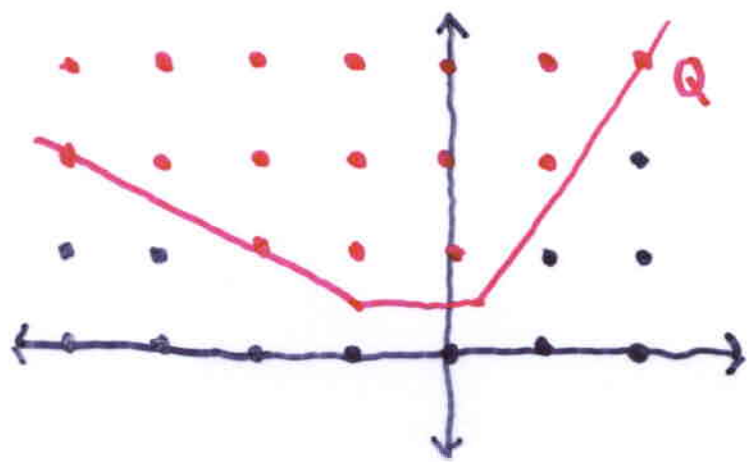
A *Hilbert Basis* is a set $B \subset K \cap \mathbb{Z}^d$ such that every integer vector in K can be written as a nonnegative integer combination of the elements of B.

Example: $d = 2, a_1 = (-2, 1), a_2 = (2, 3)$.



In fact, this is the *Minimal Hilbert Basis* (the set of *indecomposible* integer vectors).

Let Q be a polyhedron such that $Q \cap \mathbb{Z}^d = K \cap \mathbb{Z}^d \setminus 0$.



Let $P = Q \times Q$ and $T : \mathbb{R}^{2d} \to \mathbb{R}^d$ be defined by $T(x, y) = x + y$.

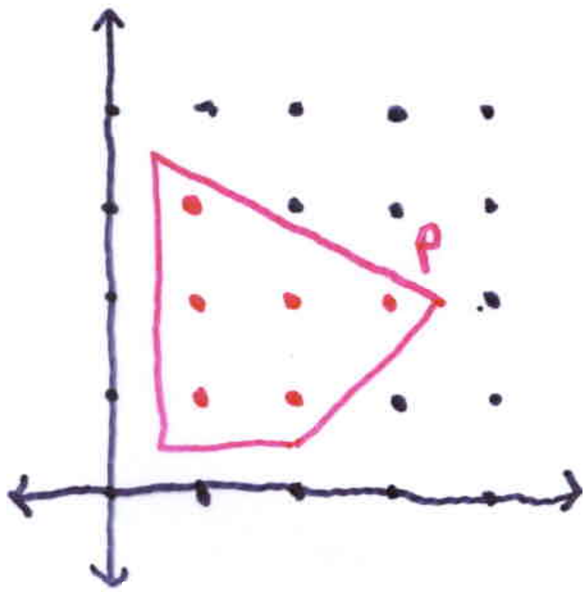
Let $S_1 = T(P \cap \mathbb{Z}^{2d})$, the set of *decomposible* integer vectors, and $S_2 = Q \cap \mathbb{Z}^d$.

Then $MHB = S_2 \setminus S_1$ and

$$f(MHB; \mathbf{x}) = f(S_2; \mathbf{x}) - f(S_1; \mathbf{x}).$$

(Again, technically, we must deal with bounded sets.)

Idea of Proof



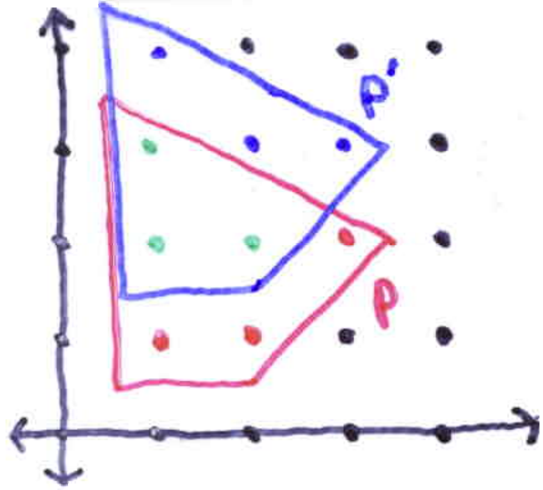
$T(x, y) = x$. Let $S = P \cap \mathbb{Z}^d$ and $S' = T(S)$.

$f(S; x, y) = xy + xy^2 + xy^3 + x^2y + x^2y^2 + x^3y^2$
$f(S'; x) = x + x^2 + x^3$.

$f(S; x, 1) = 3x + 2x^2 + x^3$.

This would work if the projection were 1-1.

"Play" with P so that the projection is 1-1.



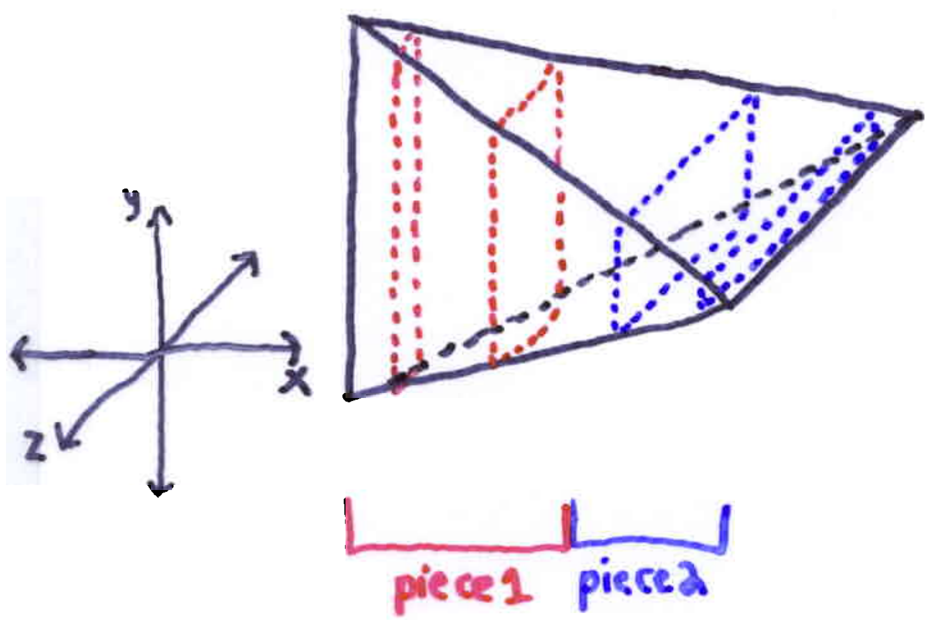
The projection of $P \setminus P'$ is 1-1.

So $f(S'; x) = f(P \setminus P'; x, 1)$.

We can find $f(P \setminus P'; x, y)$ using the following theorem:

Theorem 3 *(Barvinok) For fixed d, if S_1 and S_2 are finite sets and we are given $f(S_1; \mathbf{x})$ and $f(S_2; \mathbf{x})$ in the usual form, we can compute $f(S_1 \cap S_2; \mathbf{x})$, $f(S_1 \cup S_2; \mathbf{x})$, and $f(S_1 \setminus S_2; \mathbf{x})$ in polynomial time.*

For projections with kernal of dim > 1, use following tool (Kannan;Kannan, Lovász, Scarf):



$T(x, y, z) = x$.
$width(B, v) := \max_{x \in B} \langle v, x \rangle - \min_{x \in B} \langle v, x \rangle$.
$width(B) := \min_{v \in \mathbb{Z}^d} width(B, v)$.

Can divide image into pieces such that, in each piece, the fibers are almost the thinnest in a particular direction.

Find $f(S \cap \text{Piece1}; x)$ and $f(S \cap \text{Piece2}; x)$ separately.
$f(S; x)$ is the sum.