MSRI Summer Graduate Workshop: Algebraic, Geometric, and Combinatorial Methods for Optimization

Part IV

Geometry of Numbers and Rational Generating Function Techniques for Integer Programming

Matthias Köppe, UC Davis

September 1, 2010

Contents

Chapter 1

Introduction and Preliminaries

1.1 Integer Optimization Problems and Their Complexity

We study the computational complexity of nonlinear mixed-integer optimization problems, i.e., models of the form

$$\begin{aligned} \max/\min \quad & f(x_1, \ldots, x_n) \\ \text{s.t.} \quad & g_1(x_1, \ldots, x_n) \leq 0 \\ & \vdots \\ & g_m(x_1, \ldots, x_n) \leq 0 \\ & \mathbf{x} \in \mathbf{R}^{n_1} \times \mathbf{Z}^{n_2}, \end{aligned} \tag{1.1}$$

where $n_1 + n_2 = n$ and $f, g_1, \ldots, g_m \colon \mathbf{R}^n \to \mathbf{R}$ are arbitrary nonlinear functions.

This is a very rich topic. From the very beginning, questions such as how to present the problem to an algorithm, and, in view of possible irrational outcomes, what it actually means to solve the problem need to be answered. Fundamental intractability results from number theory and logic on the one hand and from continuous optimization on the other hand come into play. The spectrum of theorems that we present ranges from incomputability results, to hardness and inapproximability theorems, to classes that have efficient approximation schemes, or even polynomial-time or strongly polynomial-time algorithms.

We restrict our attention to deterministic algorithms in the usual bit complexity (Turing) model of computation. For an excellent recent survey focusing on other aspects of the complexity of nonlinear optimization, including the performance of oracle-based models and combinatorial settings such as nonlinear network flows, we refer to Hochbaum (2007). We also do not cover the recent developments by Onn et al. (Berstein and Onn, 2008, Berstein et al., 2008a,b, De Loera and Onn, 2006a,b, De Loera et al., 2008b, Hemmecke et al., 2009, Lee et al., 2008a,b) in the context of discrete convex optimization, for which we refer to the monograph by Onn (2007). Other excellent sources are de Klerk (2008) and Pardalos (1993).

1.1.1 Presentation of the problem

We restrict ourselves to a model where the problem is presented explicitly. In most of this survey, the functions f and g_i will be polynomial functions presented in a sparse encoding, where all coefficients are rational (or integer) and encoded in the binary scheme. It is useful to assume that the exponents of monomials are given in the unary encoding scheme; otherwise already in very simple cases the results of function evaluations will have an encoding length that is exponential in the input size.

In an alternative model, the functions are presented by oracles, such as comparison oracles or evaluation oracles. This model permits to handle more general functions (not just polynomials), and on the other hand it is very useful to obtain hardness results.

1.1.2 Encoding issues for solutions

When we want to study the computational complexity of these optimization problems, we first need to discuss how to encode the input (the data of the optimization problem) and the output (an optimal solution if it exists). In the context of *linear* mixed-integer optimization, this is straightforward: Seldom are we concerned with irrational objective functions or constraints; when we restrict the input to be rational as is usual, then also optimal solutions will be rational.

This is no longer true even in the easiest cases of nonlinear optimization, as can be seen on the following quadratically constrained problem in one continuous variable:

$$\max \quad f(x) = x^4 \quad \text{s.t.} \qquad x^2 \leq 2.$$

Here the unique optimal solution is irrational ($x^* = \sqrt{2}$, with $f(x^*) = 4$), and so it does not have a finite binary encoding. We ignore here the possibilities of using a model of computation and complexity over the real numbers, such as the celebrated Blum–Shub–Smale model (Blum et al., 1989). In the familiar Turing model of computation, we need to resort to approximations.

In the example above it is clear that for every $\epsilon > 0$, there exists a rational x that is a *feasible* solution for the problem and satisfies $|x - x^*| < \epsilon$ (proximity to the optimal solution) or $|f(x) - f(x^*)| < \epsilon$ (proximity to the optimal value). However, in general we cannot expect to find approximations by feasible solutions, as the following example shows.

$$\max \quad f(x) = x \quad \text{s.t.} \qquad x^3 - 2x = 0.$$

(Again, the optimal solution is $x = \sqrt{2}$, but the closest rational feasible solution is $x = 0$.) Thus, in the general situation, we will have to use the following notion of approximation:

Definition 1.1. An algorithm $\mathcal{A}$ is said to *efficiently approximate* an optimization problem if, for every value of the input parameter $\epsilon > 0$, it returns a rational vector $\mathbf{x}$ (not necessarily feasible) with $\|\mathbf{x} - \mathbf{x}^*\| \leq \epsilon$, where $\mathbf{x}^*$ is an optimal solution, and the running time of $\mathcal{A}$ is polynomial in the input encoding of the instance and in $\log 1/\epsilon$.

1.1.3 Approximation algorithms and schemes

The polynomial dependence of the running time in $\log 1/\epsilon$, as defined above, is a very strong requirement. For many problems, efficient approximation algorithms of this type do not exist, unless P = NP. The following, weaker notions of approximation are useful; here it is common to ask for the approximations to be *feasible solutions*, though.

Definition 1.2. (a) An algorithm $\mathcal{A}$ is an *ϵ-approximation algorithm* for a maximization problem with optimal cost $f_{\max}$, if for each instance of the problem of encoding length n, $\mathcal{A}$ runs in polynomial time in n and returns a feasible solution with cost $f_{\mathcal{A}}$, such that

$$f_{\mathcal{A}} \geq (1 - \epsilon) \cdot f_{\max}. \tag{1.2}$$

(b) A family of algorithms $\mathcal{A}_\epsilon$ is a *polynomial time approximation scheme (PTAS)* if for every error parameter $\epsilon > 0$, $\mathcal{A}_\epsilon$ is an ϵ-approximation algorithm and its running time is polynomial in the size of the instance for every fixed ϵ.

(c) A family $\{\mathcal{A}_\epsilon\}_\epsilon$ of ϵ-approximation algorithms is a *fully polynomial time approximation scheme (FPTAS)* if the running time of $\mathcal{A}_\epsilon$ is polynomial in the encoding size of the instance and $1/\epsilon$.

These notions of approximation are the usual ones in the domain of combinatorial optimization. It is clear that they are only useful when the function f (or at least the maximal value $f_{\max}$) are non-negative. For polynomial or general nonlinear optimization problems, various authors (Bellare and Rogaway, 1993, de Klerk et al., 2006, Vavasis, 1993) have proposed to use a different notion of approximation, where we compare the approximation error to the *range* of the objective function on the feasible region,

$$\big|f_{\mathcal{A}} - f_{\max}\big| \leq \epsilon \big|f_{\max} - f_{\min}\big|. \tag{1.3}$$

(Here $f_{\min}$ denotes the minimal value of the function on the feasible region.) It enables us to study objective functions that are not restricted to be non-negative on the feasible region. In addition, this notion of approximation is invariant under shifting of the objective function by a constant, and under exchanging minimization and maximization. On the other hand, it is not useful for optimization problems that have an infinite range. We remark that, when the objective function can take negative values on the feasible region, (4.4) is weaker than (4.3). We will call approximation algorithms and schemes with respect to this notion of approximation *weak*. This

terminology, however, is not consistent in the literature; de Klerk (2008), for instance, uses the notion (4.4) without an additional attribute and instead reserves the word *weak* for approximation algorithms and schemes that give a guarantee on the absolute error:

$$\left|f_{\mathcal{A}} - f_{\max}\right| \leq \epsilon. \tag{1.4}$$

1.1.4 Incomputability

Before we can even discuss the computational complexity of nonlinear mixed-integer optimization, we need to be aware of fundamental incomputability results that preclude the existence of *any* algorithm to solve general integer polynomial optimization problems.

Hilbert's tenth problem asked for an algorithm to decide whether a given multivariate polynomial $p(x_1, \ldots, x_n)$ has an integer root, i.e., whether the Diophantine equation

$$p(x_1, \ldots, x_n) = 0, \quad x_1, \ldots, x_n \in \mathbf{Z} \tag{1.5}$$

is solvable. It was answered in the negative by Matiyasevich (1970), based on earlier work by Davis, Putnam, and Robinson; see also Matiyasevich (1993). A short self-contained proof, using register machines, is presented in Jones and Matiyasevich (1991).

Theorem 1.3. (i) *There does not exist an algorithm that, given polynomials $p_1, \ldots, p_m$, decides whether the system $p_i(x_1, \ldots, x_n) = 0$, $i = 1, \ldots, m$, has a solution over the integers.*

(ii) *There does not exist an algorithm that, given a polynomial p, decides whether $p(x_1, \ldots, x_n) = 0$ has a solution over the integers.*

(iii) *There does not exist an algorithm that, given a polynomial p, decides whether $p(x_1, \ldots, x_n) = 0$ has a solution over the non-negative integers $\mathbf{Z}_+ = \{0, 1, 2, \ldots\}$.*

(iv) *There does not exist an algorithm that, given a polynomial p, decides whether $p(x_1, \ldots, x_n) = 0$ has a solution over the natural numbers $\mathbf{N} = \{1, 2, \ldots\}$.*

These three variants of the statement are easily seen to be equivalent. The solvability of the system $p_i(x_1, \ldots, x_n) = 0$, $i = 1, \ldots, m$, is equivalent to the solvability of $\sum_{i=1}^{m} p_i^2(x_1, \ldots, x_n) = 0$. Also, if $(x_1, \ldots, x_n) \in \mathbf{Z}^n$ is a solution of $p(x_1, \ldots, x_n) = 0$ over the integers, then by splitting variables into their positive and negative parts, $y_i = \max\{0, x_i\}$ and $z_i = \max\{0, -x_i\}$, clearly $(y_1, z_1; \ldots; y_n, z_n)$ is a non-negative integer solution of the polynomial equation $q(y_1, z_1; \ldots; y_n, z_n) := p(y_1 - z_1, \ldots, y_n - z_n) = 0$. (A construction with only one extra variable is also possible: Use the non-negative variables $w = \max\{|x_i| : x_i < 0\}$ and $y_i := x_i + w$.)

In the other direction, using Lagrange's four-square theorem, any non-negative integer x can be represented as the sum $a^2 + b^2 + c^2 + d^2$ with integers a, b, c, d. Thus, if $(x_1, \ldots, x_n) \in \mathbf{Z}_+^n$ is a solution over the non-negative integers, then there exists a solution $(a_1, b_1, c_1, d_1; \ldots; a_n, b_n, c_n, d_n)$ of the polynomial equation

$$r(a_1, b_1, c_1, d_1; \ldots; a_n, b_n, c_n, d_n) := p(a_1^2 + b_1^2 + c_1^2 + d_1^2, \ldots, a_n^2 + b_n^2 + c_n^2 + d_n^2).$$

The equivalence of the statement with non-negative integers and the one with the natural numbers follows from a simple change of variables.

Sharper statements of the above incomputability result can be found in Jones (1982). All incomputability statements appeal to the classic result by (Turing, 1936) on the existence of recursively enumerable (or listable) sets of natural numbers that are not recursive, such as the halting problem of universal Turing machines.

Theorem 1.4. *For the following* universal pairs (ν, δ)

$$(58, 4), \ldots, (38, 8), \ldots, (21, 96), \ldots, (14, 2.0 \times 10^5), \ldots, (9, 1.638 \times 10^{45}),$$

there exists a universal polynomial $U(x; z, u, y; a_1, \ldots, a_\nu)$ *of degree* δ *in* $4 + \nu$ *variables, i.e., for every recursively enumerable (listable) set* X *there exist natural numbers* z, u, y, *such that*

$$x \in X \quad \Longleftrightarrow \quad \exists a_1, \ldots, a_\nu \in \mathbf{N} : U(x; z, u, y; a_1, \ldots, a_\nu) = 0.$$

Jones explicitly constructs these universal polynomials, using and extending techniques by Matiyasevich. Jones also constructs an explicit system of quadratic equations in $4 + 58$ variables that is universal in the same sense. The reduction of the degree, down to 2, works at the expense of introducing additional variables; this technique goes back to Skolem (1938).

In the following, we highlight some of the consequences of these results. Let U be a universal polynomial corresponding to a universal pair (ν, δ), and let X be a recursively enumerable set that is not recursive, i.e., there does not exist any algorithm (Turing machine) to decide whether a given x is in X. By the above theorem, there exist natural numbers z, u, y such that $x \in X$ holds if and only if the polynomial equation $U(x; z, u, y; a_1, \ldots, a_\nu) = 0$ has a solution in natural numbers $a_1, \ldots, a_\nu$ (note that x and z, u, y are fixed parameters here). This implies:

Theorem 1.5. (i) *Let* (ν, δ) *be any of the universal pairs listed above. Then there does not exist any algorithm that, given a polynomial* p *of degree at most* δ *in* ν *variables, decides whether* $p(x_1, \ldots, x_n) = 0$ *has a solution over the non-negative integers.*

(ii) *In particular, there does not exist any algorithm that, given a polynomial* p *in at most* 9 *variables, decides whether* $p(x_1, \ldots, x_n) = 0$ *has a solution over the non-negative integers.*

(iii) *There also does not exist any algorithm that, given a polynomial p in at most* 36 *variables, decides whether $p(x_1, \ldots, x_n) = 0$ has a solution over the integers.*

(iv) *There does not exist any algorithm that, given a polynomial p of degree at most* 4*, decides whether $p(x_1, \ldots, x_n) = 0$ has a solution over the non-negative integers (or over the integers).*

(v) *There does not exist any algorithm that, given a system of quadratic equations in at most* 58 *variables, decides whether it has a solution of the non-negative integers.*

(vi) *There does not exist any algorithm that, given a system of quadratic equations in at most* 232 *variables, decides whether it has a solution of the integers.*

We remark that the bounds of $4 \times 9 = 36$ and $4 \times 58 = 232$ are most probably not sharp; they are obtained by a straightforward application of the reduction using Lagrange's theorem.

For integer polynomial optimization, this has the following fundamental consequences. First of all, Theorem 1.5 can be understood as a statement on the feasibility problem of an integer polynomial optimization problem. Thus, the feasibility of an integer polynomial optimization problem with a single polynomial constraint in 9 non-negative integer variables or 36 free integer variables is undecidable, etc.

If we wish to restrict our attention to *feasible* optimization problems, we can consider the problem of minimizing $p^2(x_1, \ldots, x_n)$ over the integers or non-negative integers and conclude that unconstrained polynomial optimization in 9 non-negative integer or 36 free integer variables is undecidable. We can also follow Jeroslow (1973) and associate with an arbitrary polynomial p in n variables the optimization problem

$$\begin{aligned} \min \quad & u \\ \text{s.t.} \quad & (1-u) \cdot p(x_1, \ldots, x_n) = 0, \\ & u \in \mathbf{Z}_+, \quad \mathbf{x} \in \mathbf{Z}_+^n. \end{aligned}$$

This optimization problem is always feasible and has the optimal solution value 0 if and only if $p(x_1, \ldots, x_n) = 0$ is solvable, and 1 otherwise. Thus, optimizing *linear forms* over one polynomial constraint in 10 non-negative integer variables is incomputable, and similar statements can be derived from the other universal pairs above. Jeroslow (1973) used the above program and a degree reduction (by introducing additional variables) to prove the following.

Theorem 1.6. *The problem of minimizing a linear form over quadratic inequality constraints in integer variables is not computable; this still holds true for the subclass of problems that are feasible, and where the minimum value is either* 0 *or* 1*.*

This statement can be strengthened by giving a bound on the number of integer variables.

1.1.5 Hardness and inapproximability

All incomputability results, of course, no longer apply when finite bounds for all variables are known; in this case, a trivial enumeration approach gives a finite algorithm. This is immediately the case when finite bounds for all variables are given in the problem formulation, such as for 0-1 integer problems.

For other problem classes, even though finite bounds are not given, it is possible to compute such bounds that either hold for all feasible solutions or for an optimal solution (if it exists). This is well-known for the case of linear constraints, where the usual encoding length estimates of basic solutions (Grötschel et al., 1988) are available. Such finite bounds can also be computed for convex and quasi-convex integer optimization problems.

In other cases, algorithms to decide feasibility exist even though no finite bounds for the variables are known. An example is the case of single Diophantine equations of degree 2, which are decidable using an algorithm by Siegel (1972). We discuss the complexity of this case below.

Within any such computable subclass, we can ask the question of the complexity. Below we discuss hardness results that come from the number theoretic side of the problem (section 1.1.5) and those that come from the continuous optimization side (section 1.1.5.

Hardness results from quadratic Diophantine equations in fixed dimension

The computational complexity of single quadratic Diophantine equations in 2 variables is already very interesting and rich in structure; we refer to to the excellent paper by Lagarias (2006). Below we discuss some of these aspects and their implications on optimization.

Testing primality of a number N is equivalent to deciding feasibility of the equation

$$(x+2)(y+2) = N \tag{1.6}$$

over the non-negative integers. Recently, Agrawal et al. (2004) showed that primality can be tested in polynomial time. However, the complexity status of *finding* factors of a composite number, i.e., finding a solution (x, y) of (1.6), is still unclear.

The class also contains subclasses of NP-complete feasibility problems, such as the problem of deciding for given $\alpha, \beta, \gamma \in \mathbf{N}$ whether there exist $x_1, x_2 \in \mathbf{Z}_+$ with $\alpha x_1^2 + \beta x_2 = \gamma$ (Manders and Adleman, 1978). On the other hand, the problem of deciding for given $a, c \in \mathbf{N}$ whether there exist $x_1, x_2 \in \mathbf{Z}$ with $ax_1x_2 + x_2 = c$, lies in NP $\setminus$ coNP unless NP = coNP (Adleman and Manders, 1977).

The feasibility problem of the general class of quadratic Diophantine equations in two (non-negative) variables was shown by Lagarias (2006) to be in NP. This is not straightforward because minimal solutions can have an encoding size that is

exponential in the input size. This can be seen in the case of the so-called *anti-Pellian equation* $x^2 - dy^2 = -1$. Here Lagarias (1980) proved that for all $d = 5^{2n+1}$, there exists a solution, and the solution with minimal binary encoding length has an encoding length of $\Omega(5^n)$ (while the input is of encoding length $\Theta(n)$). (We remark that the special case of the anti-Pellian equation is in coNP, as well.)

Related hardness results include the problem of quadratic congruences with a bound, i.e., deciding for given $a, b, c \in \mathbf{N}$ whether there exists a positive integer $x < c$ with $x^2 \equiv a \pmod{b}$; this is the NP-complete problem AN1 in Garey and Johnson (1979).

From these results, we immediately get the following consequences on optimization.

Theorem 1.7. (i) *The feasibility problem of quadratically constrained problems in $n = 2$ integer variables is NP-complete.*

(ii) *The problems of computing a feasible (or optimal) solution of quadratically constrained problems in $n = 2$ integer variables is not polynomial-time solvable (because the output may require exponential space).*

(iii) *The feasibility problem of quadratically constrained problems in $n > 2$ integer variables is NP-hard (but it is unknown whether it is in NP).*

(iv) *The problem of minimizing a degree-4 polynomial over the lattice points of a convex polygon (dimension $n = 2$) is NP-hard.*

(v) *The problem of finding the minimal value of a degree-4 polynomial over $\mathbf{Z}_+^2$ is NP-hard; writing down an optimal solution cannot be done in polynomial time.*

However, the complexity of minimizing a quadratic form over the integer points in polyhedra of fixed dimension is unclear, even in dimension $n = 2$. Consider the integer convex minimization problem

$$\begin{aligned} \min \quad & \alpha x_1^2 + \beta x_2, \\ \text{s.t.} \quad & x_1, x_2 \in \mathbf{Z}_+ \end{aligned}$$

for $\alpha, \beta \in \mathbf{N}$. Here an optimal solution can be obtained efficiently, as we explain in section **??**; in fact, clearly $x_1 = x_2 = 0$ is the unique optimal solution. On the other hand, the problem whether there exists a point (x_1, x_2) of a prescribed objective value $\gamma = \alpha x_1^2 + \beta x_2$ is NP-complete (see above). For indefinite quadratic forms, even in dimension 2, nothing seems to be known.

In varying dimension, the convex quadratic maximization case, i.e., maximizing positive definite quadratic forms is an NP-hard problem. This is even true in very restricted settings such as the problem to maximize $\sum_i (\mathbf{w}_i^\top \mathbf{x})^2$ over $\mathbf{x} \in \{0, 1\}^n$ (Onn, 2007).

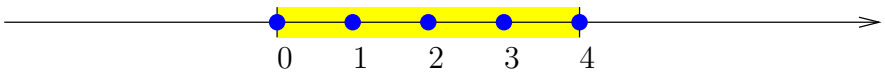


Figure 1.1: A one-dimensional lattice-point set

Inapproximability of nonlinear optimization in varying dimension

Even in the pure continuous case, nonlinear optimization is known to be hard. Bellare and Rogaway (1995, 1993) proved the following inapproximability results using the theory of interactive proof systems.

Theorem 1.8. *Assume that* P $\neq$ NP.

(i) *For any* $\epsilon < \frac{1}{3}$, *there does not exist a polynomial-time weak* ϵ*-approximation algorithm for the problem of (continuous) quadratic programming over polytopes.*

(ii) *There exists a constant* $\delta > 0$ *such that the problem of polynomial programming over polytopes does not have a polynomial-time weak* $(1 - n^{-\delta})$*-approximation algorithm.*

Here the number $1 - n^{-\delta}$ becomes arbitrarily close to 0 for growing n; note that a weak 0-approximation algorithm is one that gives no guarantee other than returning a feasible solution.

Inapproximability still holds for the special case of minimizing a quadratic form over the cube $[-1, 1]^n$ or over the standard simplex. In the case of the cube, inapproximability of the max-cut problem is used. In the case of the standard simplex, it follows via the celebrated Motzkin–Straus theorem (Motzkin and Straus, 1965) from the inapproximability of the maximum stable set problem. These are results by Håstad (1997); see also de Klerk (2008).

1.2 Introduction to generating functions

The main topic of this course is the use of **generating functions** to solve linear and nonlinear optimization over the (mixed) integer points in a polytope, i.e., over linear constraints.

We begin with a simple example to introduce generating functions. Let us consider the set S of integers in the interval $P = [0, \ldots, n]$; see Figure 1.1. We shall associate with the set S the polynomial

$$g(S; z) = z^0 + z^1 + \cdots + z^{n-1} + z^n; \tag{1.7}$$

i.e., every integer $\alpha \in S$ corresponds to a monomial z^α with coefficient 1 in the polynomial $g(S; z)$. This polynomial is called the *generating function* of S (or of P).

From the viewpoint of computational complexity, this generating function is of exponential size (in the encoding length of n), just as an explicit list of all the integers 0, 1, ..., $n-1$, n would be. However, we can observe that (1.7) is a finite geometric series, so there exists a simple summation formula that expresses (1.7) in a much more compact way:

$$g(S;z) = z^0 + z^1 + \cdots + z^{n-1} + z^n = \frac{1-z^{n+1}}{1-z}. \tag{1.8}$$

The "long" polynomial has a "short" representation as a rational function. The encoding length of this new formula is *linear* in the encoding length of n.

Suppose now someone presents to us a finite set S of integers as a generating function $g(S;z)$. Can we decide whether the set is nonempty? In fact, we can do something much stronger even – we can *count* the integers in the set S, simply by evaluating at $g(S;z)$ at $z=1$. On our example we have

$$|S| = g(S;1) = 1^0 + 1^1 + \cdots + 1^{n-1} + 1^n = n+1.$$

We can do the same on the shorter, rational-function formula. We need to be a bit careful, though: The point $z=1$ is a singularity of the formula, but it is removable (in fact, we know that $g(S;z)$ is a polynomial, so it has no poles). We just compute the limit

$$|S| = \lim_{z\to 1} g(S;z) = \lim_{z\to 1} \frac{1-z^{n+1}}{1-z} = \lim_{z\to 1} \frac{-(n+1)z^n}{-1} = n+1$$

using the Bernoulli–l'Hôpital rule. Note that we have avoided to carry out a polynomial division, which would have given us the long polynomial again.

Can these simple observations be generalized and exploited to obtain an algorithmically efficient representation of lattice-point sets in arbitrary polyhedra? It turns out they can – Barvinok (1994b) pioneered a theory of "short" rational generating functions, which gives an efficient calculus for lattice-point sets in polyhedra for every fixed dimension. Before presenting the general theory, though, we continue with our 1-dimensional example (and later with a 2-dimensional example) to investigate some of the features of the approach.

Rational functions and their Laurent expansions. We note that the summation formula (1.8) can also be written in a slightly different way:

$$g(S;z) = \frac{1}{1-z} - \frac{z^{n+1}}{1-z} = \frac{1}{1-z} + \frac{z^n}{1-z^{-1}} \tag{1.9}$$

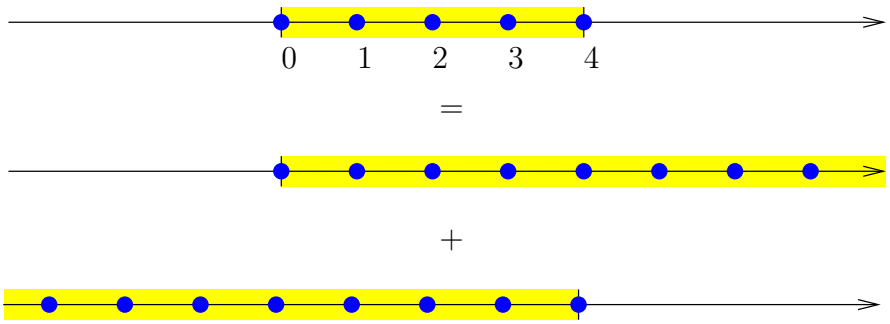


Figure 1.2: One-dimensional Brion theorem

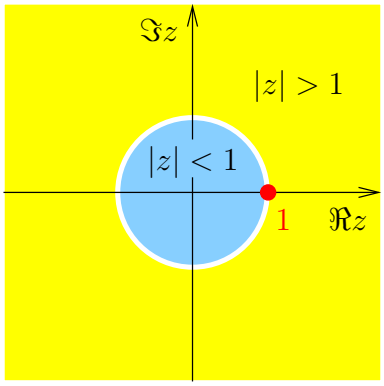


Figure 1.3: The domains of convergence of the Laurent series

Each of the two summands on the right-hand side can be viewed as the summation formula of an infinite geometric series:

$$g_1(z) = \frac{1}{1-z} = z^0 + z^1 + z^2 + \dots, \tag{1.10a}$$

$$g_2(z) = \frac{z^n}{1-z^{-1}} = z^n + z^{n-1} + z^{n-2} + \dots. \tag{1.10b}$$

The two summands have a geometrical interpretation. If we view each geometric series as the generating function of an (infinite) lattice point set, we arrive at the picture shown in Figure 1.2. *Something in this calculation seems wrong – all integer points in the interval* $[0, n]$ *are covered twice, and also all integer points outside the interval are covered once.* Where is the mistake?

We have observed a phenomenon that is due to the *one-to-many correspondence* of rational functions to their Laurent series. When we consider Laurent series of the function $g_1(z)$ about $z = 0$, the pole $z = 1$ splits the complex plane into two domains

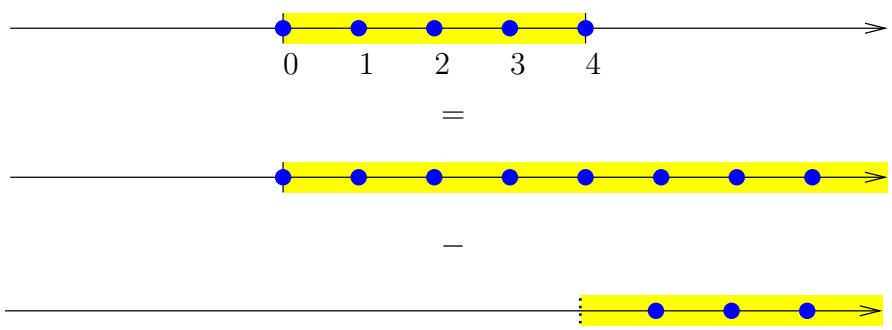


Figure 1.4: Another one-dimensional identity

of convergence (Figure 1.3): For $|z| < 1$, the power series

$$z^0 + z^1 + z^2 + \dots \tag{1.11}$$

converges to $g_1(z)$. As a matter of fact, it converges absolutely and uniformly on every compact subset of the open circle $\{\, z \in \mathbf{C} : |z| < 1 \,\}$. For $|z| > 1$, however, the series (1.11) diverges. On the other hand, the Laurent series

$$-z^{-1} - z^{-2} - z^{-3} - \dots \tag{1.12}$$

converges (absolutely and compact-uniformly) on the open circular ring $\{\, z \in \mathbf{C} : |z| > 1 \,\}$ to the function $g_1(z)$, whereas it diverges for $|z| < 1$. The same holds for the second summand $g_2(z)$. Altogether we have:

$$g_1(z) = \begin{cases} z^0 + z^1 + z^2 + \dots & \text{for } |z| < 1 \\ -z^{-1} - z^{-2} - z^{-3} - \dots & \text{for } |z| > 1 \end{cases} \tag{1.13}$$

$$g_2(z) = \begin{cases} -z^{n+1} - z^{n+2} - z^{n+3} - \dots & \text{for } |z| < 1 \\ z^n + z^{n-1} + z^{n-2} + \dots & \text{for } |z| > 1 \end{cases} \tag{1.14}$$

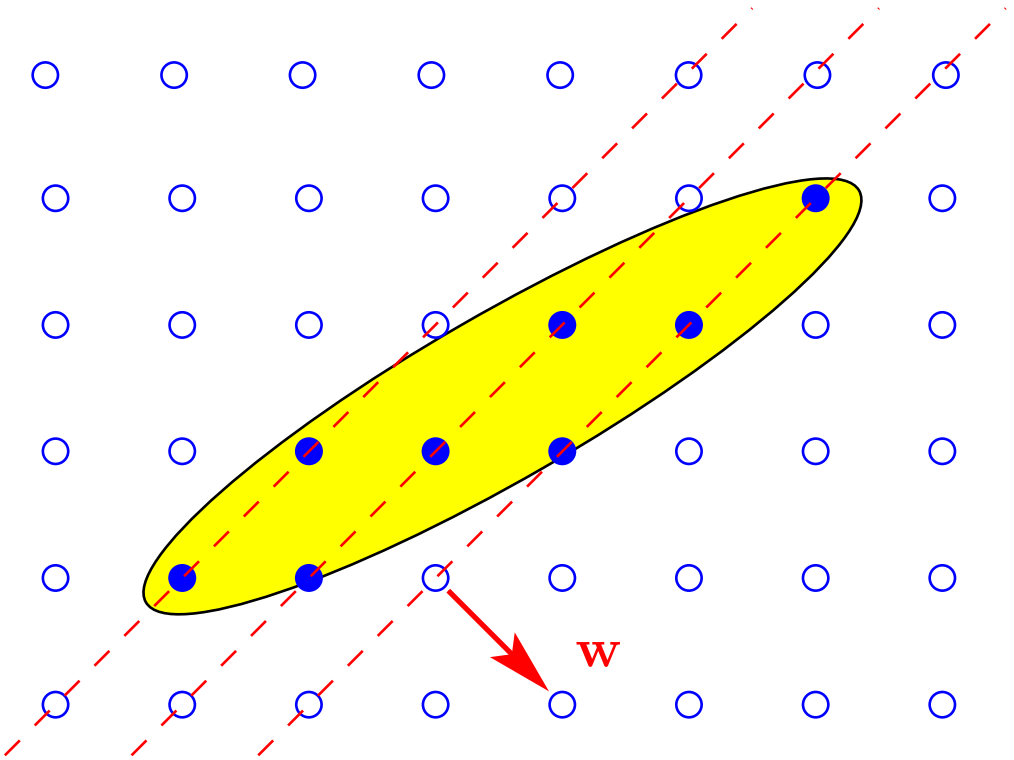
We can now see that the "mistake" we observed in formula (1.10) and Figure 1.2 is due to the fact that we had picked two Laurent series for the summands $g_1(z)$ and $g_2(z)$ that do not have a common domain of convergence. If for each summand we choose the series that converge for $|z| < 1$, we obtain the more intuitive formula

$$g(S; z) = \big(z^0 + z^1 + z^2 + \dots\big) - \big(z^{n+1} + z^{n+2} + z^{n+3} + \dots\big)\,; \tag{1.15}$$

see also Figure 1.4. Nevertheless, it turns out that using Laurent series with disjoint domains of convergence is a powerful technique; we will meet the situation of formula (1.10) and Figure 1.2 again in the multidimensional case as *Brion's Theorem*.

Chapter 2

Tools from the Geometry of Numbers



We mostly follow Barvinok, "Course in Convexity".

Definition 2.1. A *lattice* Λ of $\mathbf{R}^n$ is a discrete additive subgroup of $\mathbf{R}^n$ whose linear span is $\mathbf{R}^n$.

Equivalently, a lattice is a set of the form $B\mathbf{Z}^n$ with B a regular $n \times n$ matrix. The columns of B form a *basis* of the lattice. Bases are not unique; multiplication from the right by a unimodular matrix gives another basis (these are all bases).

Definition 2.2. *Dual (polar, reciprocal) lattice*

$$\Lambda^* = \{\, \mathbf{x} \in \mathbf{R}^d : \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbf{Z} \text{ for all } \mathbf{y} \in \Lambda \,\}$$

.

2.1 Minkowski's 1st theorem

Theorem 2.3 (Blichfeldt)**.** *Let $\Lambda \subset \mathbf{R}^d$ lattice, X measurable,* $\operatorname{vol} X > \det \Lambda$*. Then there exist $\mathbf{x} \neq \mathbf{y}$ in X with $\mathbf{x} - \mathbf{y} \in \Lambda$.*

Proof. Let $\Lambda = B\mathbf{Z}^d$, denote (half-open) fundamental parallelepiped $\Pi = B[0,1)$. From tiling by copies of fundamental parallelepipeds,

$$\mathbf{R}^d = \coprod_{\mathbf{u} \in \Lambda} (\Pi + \mathbf{u}),$$

get

$$\begin{aligned} X &= \coprod_{\mathbf{u} \in \Lambda} [(\Pi + \mathbf{u}) \cap X] \\ &= \coprod_{\mathbf{u} \in \Lambda} [\underbrace{(\Pi + \mathbf{u}) \cap X - \mathbf{u}}_{=: X_{\mathbf{u}}} + \mathbf{u}] \end{aligned}$$

(so $X_{\mathbf{u}} = \Pi \cap (X - \mathbf{u})$ is the set of all points in Π that, translated by $\mathbf{u}$, fall into X.)

Volume argument:

$$\sum_{\mathbf{u}} \operatorname{vol} X_{\mathbf{u}} = \operatorname{vol} X > \operatorname{vol} \Pi = \det \Lambda$$

but each $X_{\mathbf{u}} \subset \Pi$, so the $X_{\mathbf{u}}$ cannot be pairwise disjoint.

Take an intersecting pair: $X_{\mathbf{u}} \cap X_{\mathbf{v}} \neq \emptyset$, let $\mathbf{a} \in X_{\mathbf{u}} \cap X_{\mathbf{v}}$.

Then: $\mathbf{a} \in X_{\mathbf{u}}$ implies $\mathbf{x} := \mathbf{a} + \mathbf{u} \in X$

and: $\mathbf{a} \in X_{\mathbf{v}}$ implies $\mathbf{y} := \mathbf{a} + \mathbf{v} \in X$

so $\mathbf{x} - \mathbf{y} = \mathbf{u} - \mathbf{v} \in \Lambda$. □

Theorem 2.4 (Minkowski's 1st convex body theorem)**.** *Let $\Lambda \subset \mathbf{R}^d$ lattice, $A \subseteq \mathbf{R}^d$ convex, symmetric, and either*

- $\operatorname{vol} A > 2^d \det \Lambda$ *or*
- $\operatorname{vol} A \geq 2^d \det \Lambda$ *and A compact,*

Then there exist $\mathbf{0} \neq \mathbf{u} \in A \cap \Lambda$.

Proof. (Case of a strict inequality.) Let $X = \frac{1}{2}A$, so $\operatorname{vol} X > \det \Lambda$.
By Blichfeldt, there exist $\mathbf{x} \neq \mathbf{y}$ in X with $\mathbf{x} - \mathbf{y} \in \Lambda$.
So $2\mathbf{x}$, $2\mathbf{y} \in A$,
by symmetry $-2y \in A$,
by convexity $\mathbf{0} \neq \mathbf{u} = \frac{1}{2}(2x - 2y) = \mathbf{x} - \mathbf{y} \in A$.
(Compact case.) Use ρA for $\rho > 1$, get from theorem sequence $\{\mathbf{u}_\rho\}_\rho \subset A$, compactness argument: limit point $\mathbf{u}$, discreteness: $\mathbf{0} \neq \mathbf{u} \in A \cap \Lambda$. □

Note: Without symmetry, volumes of lattice-point-free bodies can be arbitrarily large. But we get "flatness" (later); some preparations first.

2.2 Packing, covering, shortest vectors

Definition 2.5. The *packing radius* of Λ:

$$\rho(\Lambda) = \sup\big\{\, \rho : B(\mathbf{x}, \rho) \cap B(\mathbf{y}, \rho) = \emptyset \text{ for } \mathbf{x} \neq \mathbf{y} \text{ in } \Lambda \,\big\}$$

Note: The packing radius is $\frac{1}{2}$ of the length of a *shortest nonzero lattice vector* (in ℓ_2).

Minkowski's 1st gives nice bounds for shortest nonzero lattice vectors:

Lemma 2.6. *Shortest vector in ℓ_∞:*

$$\|\mathbf{x}^*_\infty\|_\infty \leq (\det \Lambda)^{1/d}.$$

Proof. Take the cube $\square = \{\mathbf{x} : \|\mathbf{x}\|_\infty \leq (\det \Lambda)^{1/d}\}$. It has volume $2^d \det \Lambda$. By Minkowski's 1st, there exists $\mathbf{0} \neq \mathbf{u} \in \square \cap \Lambda$. □

Lemma 2.7. *Packing radius ($\frac{1}{2}$ shortest vector in ℓ_2):*

$$\rho(\Lambda) \leq \frac{1}{2}\sqrt{d}\,(\det \Lambda)^{1/d}.$$

Proof. Let $\mathbf{x}^*$ shortest vector (ℓ_2), $\mathbf{x}^*_\infty$ any shortest vector (ℓ_∞). Then

$$\|\mathbf{x}^*\| \leq \|\mathbf{x}^*_\infty\| \leq \sqrt{d}\,\|\mathbf{x}^*_\infty\|_\infty \leq \sqrt{d}\,(\det \Lambda)^{1/d}.$$

□

Sharper estimates using the volume of the d-dimensional ball:

$$\sigma_d = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)},$$

which gives

$$\rho(\Lambda) \le \sqrt{\frac{d}{2\pi e}}\,\big(1+O(\tfrac{1}{d})\big)(\det\Lambda)^{1/d}.$$

From above lemma we get a nice relation for the packing radii of a lattice and its dual.

Lemma 2.8.

$$\rho(\Lambda)\cdot\rho(\Lambda^*) \le \frac{d}{4}.$$

Proof. By lemma,

$$\rho(\Lambda) \le \frac{1}{2}\sqrt{d}\,(\det\Lambda)^{1/d}$$
$$\rho(\Lambda^*) \le \frac{1}{2}\sqrt{d}\,(\det\Lambda^*)^{1/d}$$

and $\det\Lambda\cdot\det\Lambda^* = 1$. □

"Dual" to packing is covering.

Definition 2.9. The *covering radius* of Λ is

$$\mu(\Lambda) = \inf\{\,\mu>0 : \Lambda + \bar{B}(\mathbf{0},\mu) = \mathbf{R}^d\,\}.$$

We get an important relation between the covering radius of a lattice and the packing radius of its dual.

Theorem 2.10.

$$\frac{1}{4} \le \mu(\Lambda)\cdot\rho(\Lambda^*) \le \frac{1}{4}\sqrt{\sum_{k=1}^{d} k^2} \le \frac{1}{4}d^{3/2}.$$

It is important that the bound only depends on the dimension. It can be reduced to $O(d)$ with a different proof technique. The above bound is the best one that can be proved by elementary means.

Proof. We only prove (and use) the upper bound. The proof is by induction. For $d=1$, have $\Lambda = \alpha\mathbf{Z}$ (some α) and $\Lambda^* = \alpha^{-1}\mathbf{Z}$, so covering radius $\mu(\Lambda) = \alpha/2$ and packing radius $\rho(\Lambda^*)$, so $\mu\cdot\lambda = 1/4$.

For $d > 1$, let $\mathbf{u}$ shortest of Λ, so $\frac{1}{2}\|\mathbf{u}\| = \rho(\Lambda)$. By orthogonal change of coordinates, can assume $\mathbf{u} = 2\rho \cdot \mathbf{e}_d$. We use the canonical projection $\pi_{d-1}\colon \mathbf{R}^d \to \mathbf{R}^{d-1} \hookrightarrow \mathbf{R}^d$ onto the first $d-1$ coordinates. $\Lambda_{d-1} := \pi_{d-1}(\Lambda)$ is a lattice of $\mathbf{R}^{d-1}$.

We first prove $\rho(\Lambda^*_{d-1}) \geq \rho(\Lambda^*)$. Let $\mathbf{a}^* \in \Lambda^*_{d-1}$ and $\mathbf{a} \in \Lambda$; then $\pi_{d-1}(\mathbf{a}) \in \Lambda_{d-1}$, so

$$\mathbf{Z} \ni \langle \mathbf{a}^*, \pi_{d-1}(\mathbf{a}) \rangle = \langle \mathbf{a}^*, \mathbf{a} \rangle,$$

so $\mathbf{a}^* \in \Lambda^*$. So we have the inclusion $\Lambda^*_{d-1} \subseteq \Lambda^*$.

Next let $\mathbf{x} \in \mathbf{R}^d$ and $\mathbf{y} = \pi_{d-1}(\mathbf{x})$. Let $\mathbf{v}$ closest to $\mathbf{y}$ in Λ_{d-1}, so (covering)

$$\|\mathbf{v} - \mathbf{y}\| \leq \mu(\Lambda_{d-1})$$

Since $\mathbf{v}$ is a point of the projected lattice Λ_{d-1}, there is a lattice vector $\mathbf{w}^0 \in \Lambda$ that is a preimage. All points in $\mathbf{w}^0 + \mathbf{u}\mathbf{Z}$ lie in Λ. Let $\mathbf{w}$ be a point that is closest to $\mathbf{x}$; then

$$|w_d - x_d| \leq \frac{1}{2}\|\mathbf{u}\| = \rho(\Lambda).$$

Thus

$$\|\mathbf{w} - \mathbf{x}\|^2 \leq \mu^2(\Lambda_{d-1}) + \rho^2(\Lambda).$$

This proves $\mu^2(\Lambda) \leq \mu^2(\Lambda_{d-1}) + \rho^2(\Lambda)$.

Now we use the induction:

$$\mu^2(\Lambda) \cdot \rho^2(\Lambda^*) \leq \mu^2(\Lambda_{d-1}) \underbrace{\rho^2(\Lambda^*)}_{\leq \rho^2(\Lambda^*_{d-1})} + \underbrace{\rho^2(\Lambda)\rho^2(\Lambda^*)}_{\leq \frac{1}{16}d^2 \quad \text{(Lemma)}}$$

$$\leq \frac{1}{16}\left(\sum_{k=1}^{d-1} k^2\right) + \frac{1}{16}d^2,$$

which completes the proof. □

2.3 Flatness for ellipsoids

We first prove the flatness theorem for balls.

Theorem 2.11. *Let $\Lambda \subset \mathbf{R}^d$ lattice, $B = \bar{B}(\mathbf{a}, \beta)$ a ball, $B \cap \Lambda = \emptyset$.*

Then there exists $\mathbf{v} \in \Lambda^*$ with

$$\mathrm{width}_{\mathbf{v}}(B) := \max\{\, \langle \mathbf{v}, \mathbf{x} \rangle : \mathbf{x} \in B \,\} - \min\{\, \langle \mathbf{v}, \mathbf{x} \rangle : \mathbf{x} \in B \,\} \leq d^{3/2}.$$

Proof. For any $\mathbf{v}$, we have

$$\mathrm{width}_{\mathbf{v}}(B) = (\langle \mathbf{v}, \mathbf{a} \rangle + \beta\|\mathbf{v}\|) - (\langle \mathbf{v}, \mathbf{a} \rangle - \beta\|\mathbf{v}\|) = 2\beta \cdot \|\mathbf{v}\|.$$

Since $B \cap \Lambda = \emptyset$, certainly $\beta \leq \mu(\Lambda)$ (covering).

We pick $\mathbf{v}$ to be shortest in Λ^*, so $\|\mathbf{v}\| = 2\rho(\Lambda^*)$.

From the previous theorem, $\mu(\Lambda) \cdot \rho(\Lambda^*) \leq d^{3/2}/4$.

This gives $\mathrm{width}_{\mathbf{v}}(B) \leq 4\mu(\Lambda) \cdot \rho(\Lambda^*) \leq d^{3/2}$ as desired. □

Next, for ellipsoids.

Theorem 2.12. *Let $\Lambda \subset \mathbf{R}^d$ lattice, E ellipsoid, $E \cap \Lambda = \emptyset$. Then there exists $\mathbf{v} \in \Lambda^*$ with*

$$\mathrm{width}_{\mathbf{v}}(E) \leq d^{3/2}.$$

Proof. The ellipsoid E is a regular linear image of a ball, $E = TB$. Instead of E, Λ, and Λ^* we consider the ball $B = T^{-1}E$, the lattice $T^{-1}\Lambda$, and its dual $(T^{-1}\Lambda)^* = T^\top \Lambda^*$, for which the statement already holds by the above theorem.

Now for any $\mathbf{v} \in \Lambda^*$ and $\mathbf{x} \in E$ (thus $\mathbf{x} = T\mathbf{y}$ for some $\mathbf{y} \in B$) we have

$$\langle \mathbf{v}, \mathbf{x} \rangle = \langle \mathbf{v}, T\mathbf{y} \rangle = \langle T^\top \mathbf{v}, \mathbf{y} \rangle,$$

where $T^\top \mathbf{v} \in (T^{-1}\Lambda)^*$. Thus the estimate carries over. □

2.4 Approximation of convex bodies by ellipsoids

Theorem 2.13. *Let $K \subseteq \mathbf{R}^d$ be a convex body. There exists a unique ellipsoid E ("Löwner–John ellipsoid") of maximum volume contained in K. The concentric d-dilation of E contains K.*

Algorithmically, approximations by ellipsoids can be obtained by various means, including the shallow-cut ellipsoid method.

2.5 Flatness of convex bodies

The approximation of arbitrary convex bodies by ellipsoids then implies:

Theorem 2.14. *Let $\Lambda \subset \mathbf{R}^d$ lattice, K a convex body, $K \cap \Lambda = \emptyset$. Then there exists $\mathbf{v} \in \Lambda^*$ with*

$$\mathrm{width}_{\mathbf{v}}(K) \leq d^{5/2}.$$

The estimate can be reduced to $O(d^{3/2})$.

2.6 Algorithms

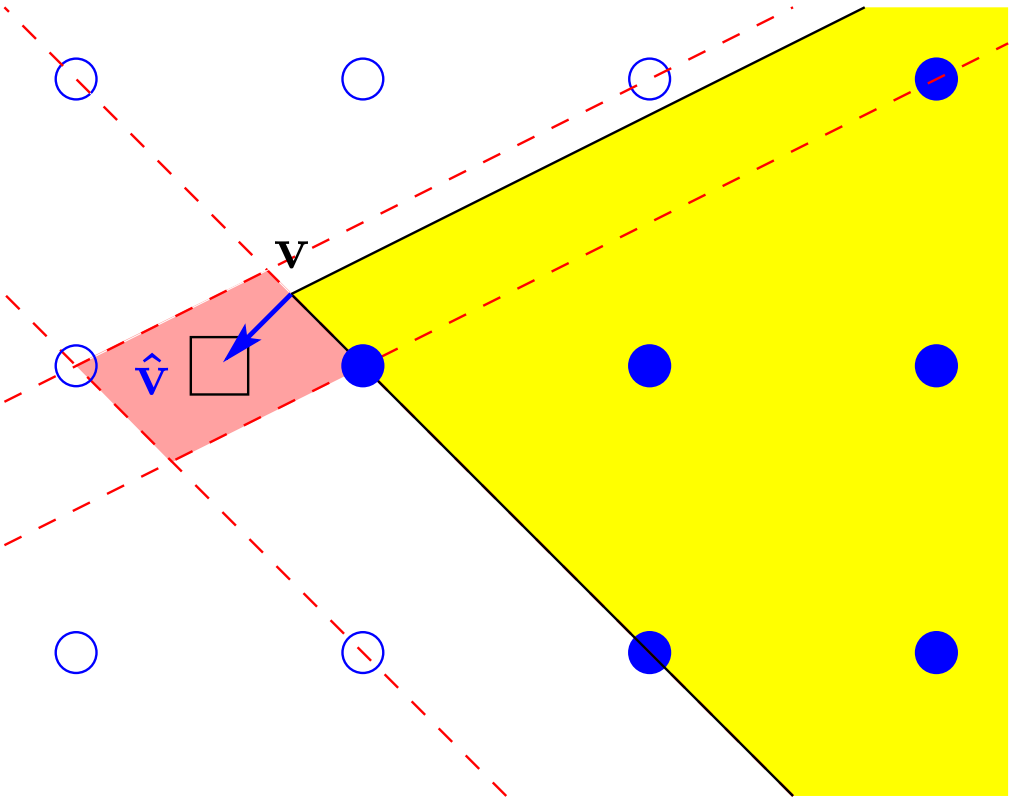
The Lenstra–Lenstra–Lovász (LLL) algorithm computes in polynomial time an "almost orthogonal" lattice basis. All known efficient deterministic shortest vector algorithms first compute an LLL basis. In addition, probabilistic algorithms based on sieving are known.

Computing a flatness direction of a convex body can be achieved by first computing an ellipsoidal approximation, and then using a shortest vector computation to find a flatness direction of the ellipsoid.

Lenstra's algorithm (Lenstra, 1983) was the first algorithm to establish the fact that integer linear optimization problems in fixed dimension can be solved in polynomial time. A modern description of Lenstra-type algorithms can be found in Eisenbrand (2010); see also Hildebrand and Köppe (2010). In a nutshell, the algorithm computes flatness directions and performs *branching on hyperplanes* orthogonal to the flatness direction. Because the flatness constant only depends on the dimension, this creates only constantly many branches, as opposed to potentially exponentially many branches that could appear in single-variable branching. Then this is applied recursively. In constant dimension, only a fixed number of subproblems is created.

Chapter 3

Barvinok's short rational generating functions



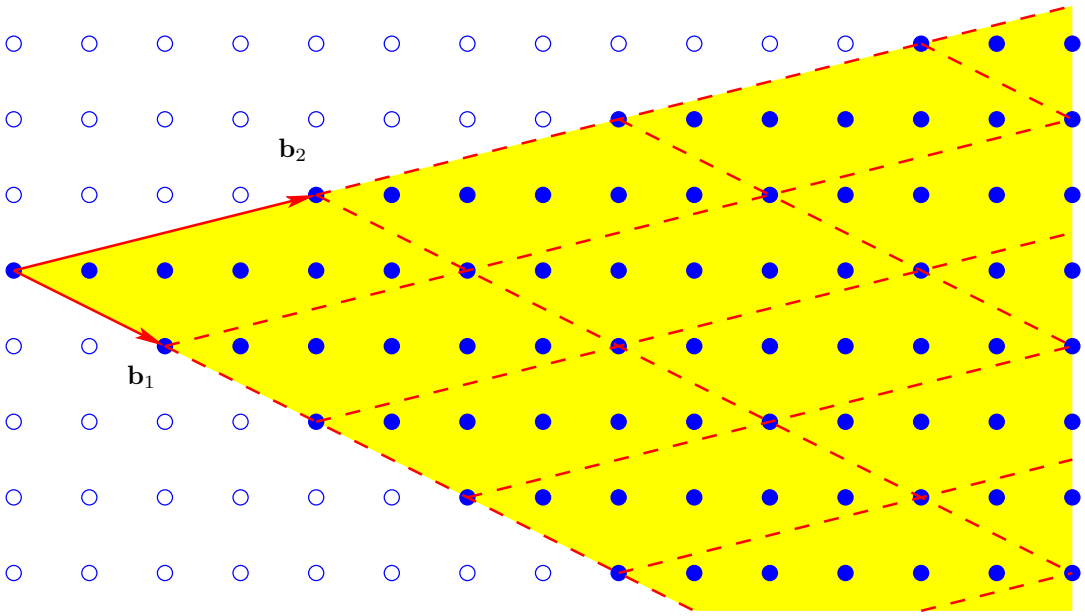


Figure 3.1: Tiling a rational two-dimensional cone with copies of the fundamental parallelepiped

3.1 Dimension two

Let us consider a cone C spanned by the vectors $\mathbf{b}_1 = (\alpha, -1)$ and $\mathbf{b}_2 = (\beta, 1)$; see Figure 3.1 for an example with $\alpha = 2$ and $\beta = 4$. We would like to write down a generating function for the integer points in this cone. We apparently need a generalization of the geometric series, of which we made use in the one-dimensional case. The key observation now is that using copies of the half-open *fundamental parallelepiped*,

$$\Pi = \{\, \lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 : \lambda_1 \in [0,1), \lambda_2 \in [0,1) \,\}, \tag{3.1}$$

the cone can be *tiled*:

$$C = \bigcup_{\mathbf{s} \in S} (\mathbf{s} + \Pi) \quad \text{where} \quad S = \{\, \mu_1 \mathbf{b}_1 + \mu_2 \mathbf{b}_2 : (\mu_1, \mu_2) \in \mathbf{Z}_+^2 \,\} \tag{3.2}$$

(a disjoint union). Moreover, because we have chosen *integral* generators $\mathbf{b}_1, \mathbf{b}_2$ for our cone, the integer points are "the same" in each copy of the fundamental parallelepiped. Therefore, also the integer points of C can be tiled by copies of the integer points of Π:

$$C \cap \mathbf{Z}^2 = \bigcup_{\mathbf{s} \in S} \big(\mathbf{s} + (\Pi \cap \mathbf{Z}^2)\big) \tag{3.3}$$

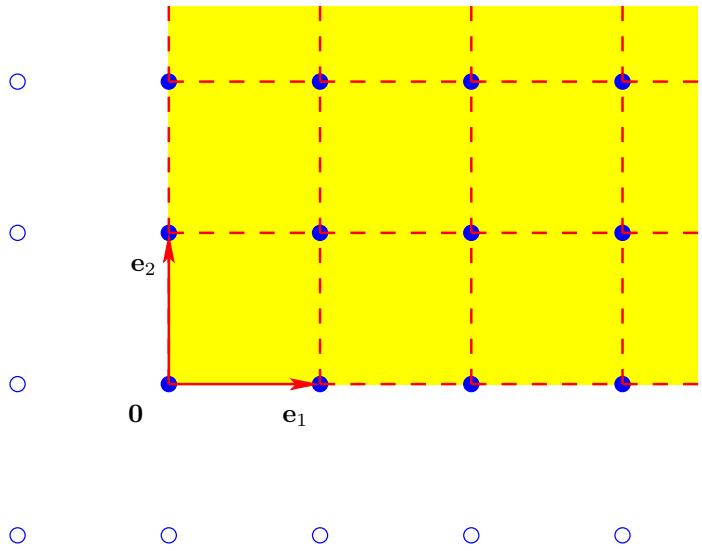


Figure 3.2: The semigroup $S \subseteq \mathbf{Z}^2$ generated by $\mathbf{b}_1$ and $\mathbf{b}_2$ is a linear image of $\mathbf{Z}_+^2$

We can also see $C \cap \mathbf{Z}^2$ as a finite disjoint union of copies of the set S, shifted by the integer points of the fundamental parallelepiped:

$$C \cap \mathbf{Z}^2 = \bigcup_{\mathbf{x} \in \Pi \cap \mathbf{Z}^2} (\mathbf{x} + S). \tag{3.4}$$

The benefit of this representation is that the set S is just the image of $\mathbf{Z}_+^2$ under the matrix $(\mathbf{b}_1, \mathbf{b}_2) \in \mathbf{Z}^{2\times 2}$; cf. Figure 3.1 and Figure 3.2. Now $\mathbf{Z}_+^2$ is the direct product of $\mathbf{Z}_+$ with itself, whose generating function we already know – it is given by the geometric series,

$$g(\mathbf{Z}_+; z) = z^0 + z^1 + z^2 + z^3 + \cdots = \frac{1}{1-z}.$$

We thus obtain the generating function as a product,

$$g(\mathbf{Z}_+^2; z_1, z_2) = (z_1^0 + z_1^1 + z_1^2 + z_1^3 + \dots)(z_2^0 + z_2^1 + z_2^2 + z_2^3 + \dots) = \frac{1}{1-z_1} \cdot \frac{1}{1-z_2}.$$

Applying the linear transformation $(\mathbf{b}_1, \mathbf{b}_2) = \left(\begin{smallmatrix} \alpha & \beta \\ -1 & 1 \end{smallmatrix}\right)$, we obtain the generating function

$$g(S; z_1, z_2) = \frac{1}{(1 - z_1^{\alpha} z_2^{-1})(1 - z_1^{\beta} z_2^{1})}.$$

From the representation (3.4) it is now clear that

$$g(C; z_1, z_2) = \sum_{\mathbf{x} \in \Pi \cap \mathbf{Z}^2} z_1^{x_1} z_2^{x_2} g(S; z_1, z_2);$$

the multiplication with the monomial $z_1^{x_1} z_2^{x_2}$ corresponds to the shifting of the set S by the vector (x_1, x_2). In our example, it is easy to see that

$$\Pi \cap \mathbf{Z}^2 = \{\, (i, 0) : i = 0, \ldots, \alpha + \beta - 1 \,\}.$$

We thus obtain the generating function

$$g(C; z_1, z_2) = \frac{z_1^0 + z_1^1 + \cdots + z_1^{\alpha+\beta-2} + z_1^{\alpha+\beta-1}}{(1 - z_1^\alpha z_2^{-1})(1 - z_1^\beta z_2^1)}.$$

Unfortunately, this formula has an exponential size as the numerator contains $\alpha + \beta$ summands. In our example, the numerator again is a finite geometric series, so we could use a short summation formula. However, this technique does not seem to be helpful in general because the structure of the integer points in the fundamental parallelepiped is usually more complicated than in our example.

Triangulations. A different idea to make the formula shorter is to break the cone into "smaller" cones, each of which have a shorter formula. We have observed that the length of the formula is essentially determined by the number of summands in the numerator – which correspond to the integer points in the fundamental parallelepiped. Thus the right measure of size of a cone seems to be the number of integer points in the fundamental parallelepiped; this is called the *index* of the cone.

We show on our example that triangulations can be used to reduce the index of cones. Indeed, by using an interior vector $\mathbf{w} = (1, 0)$, we can triangulate the cone into the cones $C_1 = \text{cone}\{\mathbf{b}_1, \mathbf{w}\}$ and $C_2 = \text{cone}\{\mathbf{w}, \mathbf{b}_2\}$. For each of the cones, the fundamental parallelepiped contains a single integer point – the origin; see Figure 3.4. (Such cones are called *unimodular* cones.) Thus we can write down their generating functions,

$$g(C_1; z_1, z_2) = \frac{1}{(1 - z_1^\alpha z_2^{-1})(1 - z_1)}$$
$$g(C_2; z_1, z_2) = \frac{1}{(1 - z_1^\beta z_2^1)(1 - z_1)}.$$

Note however that if we just added these two functions, all the integer points in the intersection $C_1 \cap C_2$ would be counted *twice*. However, the intersection $C_1 \cap C_2$ is just a one-dimensional cone, whose generating function we can easily write as

$$g(C_1 \cap C_2; z_1, z_2) = \frac{1}{1 - z_1}.$$

Thus we can fix the overcounting using the inclusion–exclusion principle, writing

$$\begin{aligned} g(C; z_1, z_2) &= g(C_1; z_1, z_2) + g(C_2; z_1, z_2) - g(C_1 \cap C_2; z_1, z_2) \\ &= \frac{1}{(1 - z_1^\alpha z_2^{-1})(1 - z_1)} + \frac{1}{(1 - z_1^\beta z_2^1)(1 - z_1)} - \frac{1}{1 - z_1}. \end{aligned} \tag{3.5}$$

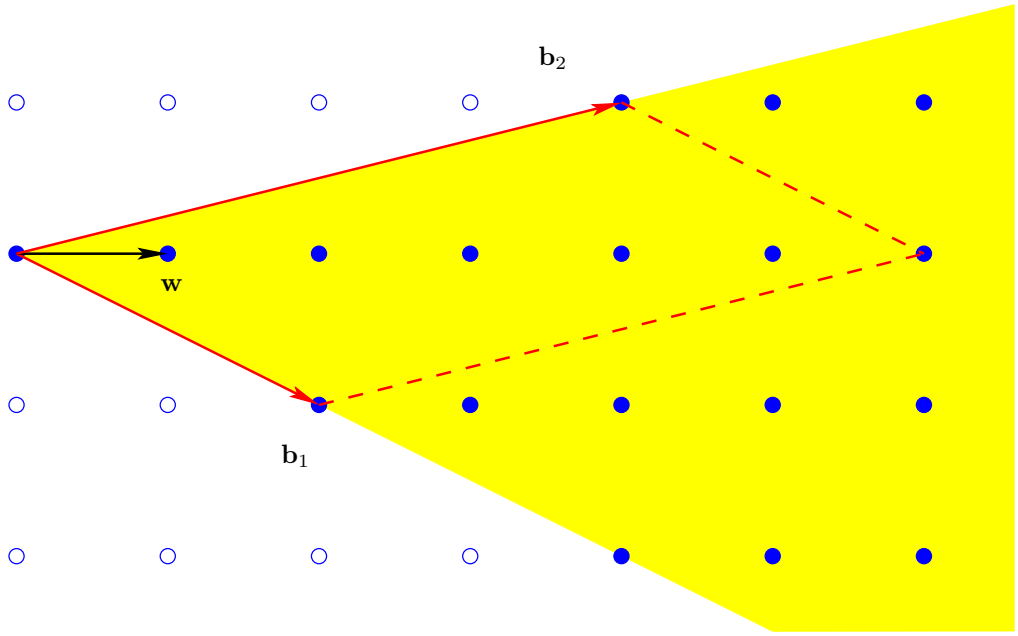


Figure 3.3: A two-dimensional cone of index 6 with its fundamental parallelepiped. Using the interior vector $\mathbf{w}$, a triangulation can be constructed.

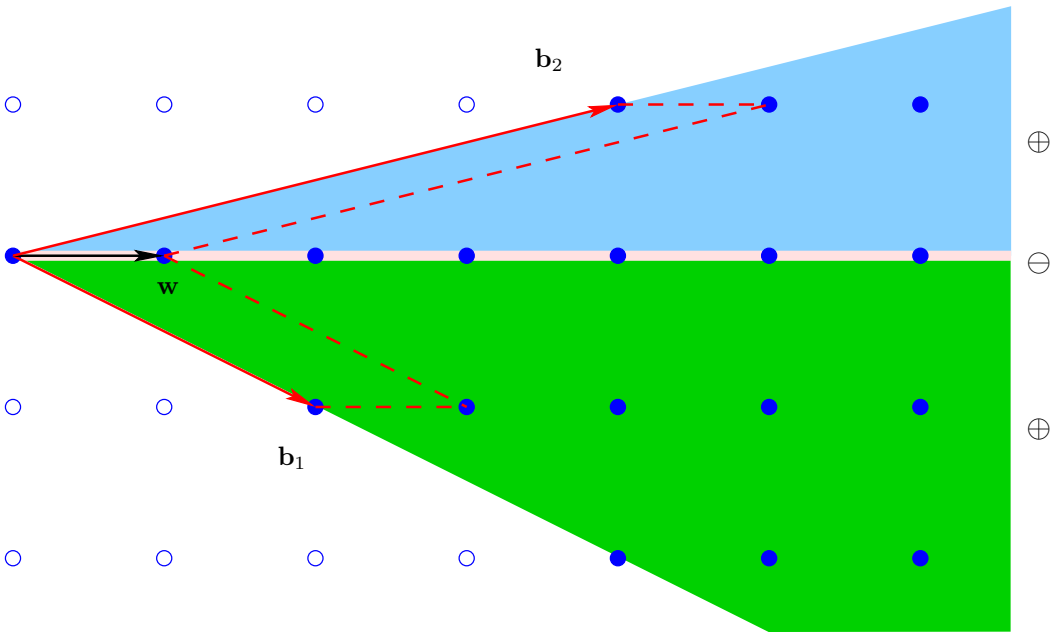


Figure 3.4: A two-dimensional cone of index 6, triangulated into two unimodular cones. The integer points in the one-dimensional intersection would be counted twice, so we subtract them once (inclusion–exclusion principle).

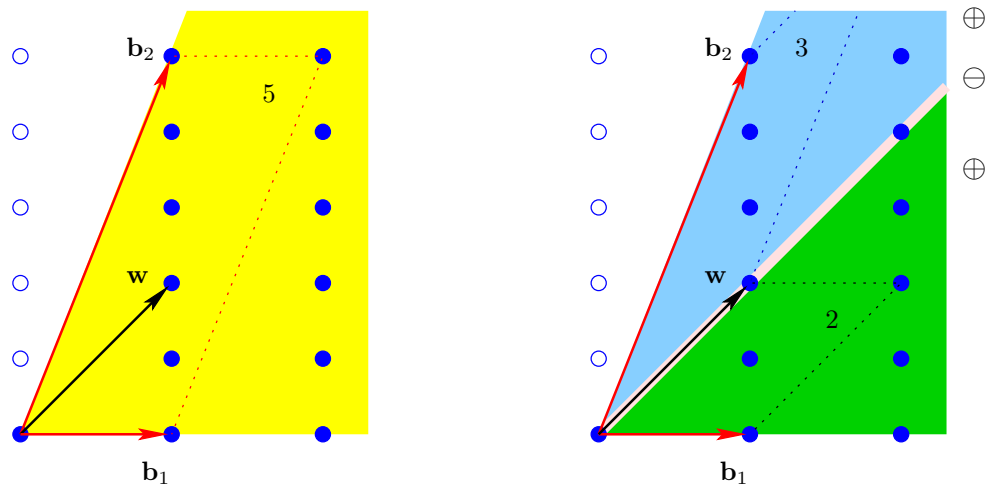


Figure 3.5: A triangulation of the cone of index 5 generated by $\mathbf{b}^1$ and $\mathbf{b}^2$ into the two cones spanned by $\{\mathbf{b}^1, \mathbf{w}\}$ and $\{\mathbf{b}^2, \mathbf{w}\}$, having an index of 2 and 3, respectively. We have the inclusion-exclusion formula $g(\text{cone}\{\mathbf{b}_1, \mathbf{b}_2\}; \mathbf{z}) = g(\text{cone}\{\mathbf{b}_1, \mathbf{w}\}; \mathbf{z}) + g(\text{cone}\{\mathbf{b}_2, \mathbf{w}\}; \mathbf{z}) - g(\text{cone}\{\mathbf{w}\}; \mathbf{z})$; here the one-dimensional cone spanned by $\mathbf{w}$ needed to be subtracted.

We have thus obtained a short formula.

The bad news, however, is that triangulations don't always work. Let us consider another two-dimensional example, a cone C' generated by $\mathbf{b}_1 = (1, 0)$ and $\mathbf{b}_2 = (1, \alpha)$. An example for $\alpha = 5$ is shown in Figure 3.5. The integer points in the fundamental parallelepiped are

$$\Pi' \cap \mathbf{Z}^2 = \{(0, 0)\} \cup \{\, (1, i) : i = 1, \ldots, \alpha - 1 \,\},$$

so again the rational generating function would have α summands in the numerator, and thus have exponential size. Unfortunately, every attempt to use triangulations to reduce the size of the formula fails in this example. The choice of an interior vector $\mathbf{w}$ in Figure 3.5, for instance, splits the cone of index 5 into two cones of index 2 and 3, respectively – and also a one-dimensional cone. Indeed, every possible triangulation of C' into unimodular cones contains at least α two-dimensional cones!

Signed decompositions. The important new idea by Barvinok (1994b) was to use so-called *signed decompositions* in addition to triangulations in order to reduce the index of a cone. In our example, we can choose the vector $\mathbf{w} = (0, 1)$ *from the outside of the cone* to define cones $C_1 = \text{cone}\{\mathbf{b}_1, \mathbf{w}\}$ and $C_2 = \text{cone}\{\mathbf{w}, \mathbf{b}_2\}$; see Figure 3.6. Using these cones, we have the inclusion-exclusion formula

$$g(C'; z_1, z_2) = g(C_1; z_1, z_2) - g(C_2; z_1, z_2) + g(C_1 \cap C_2; z_1, z_2)$$

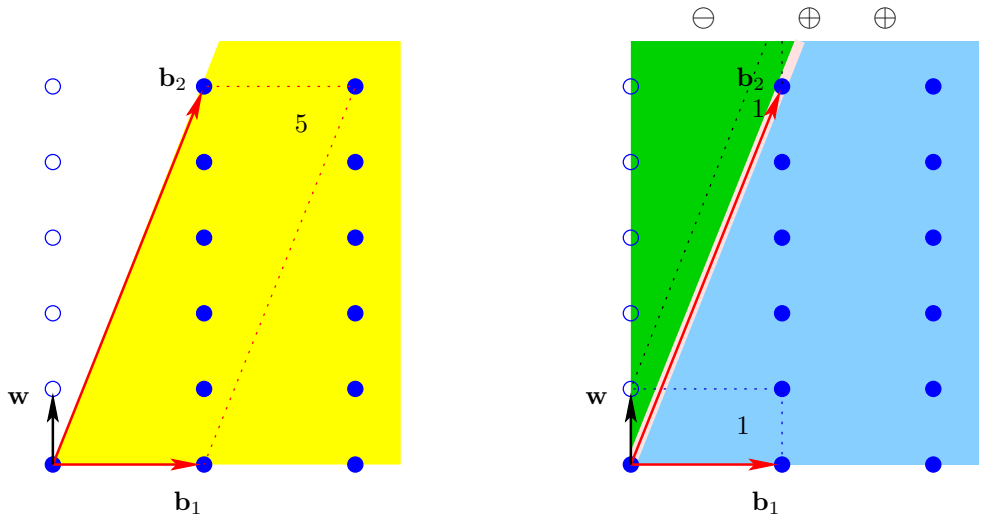


Figure 3.6: A signed decomposition of the cone of index 5 generated by $\mathbf{b}^1$ and $\mathbf{b}^2$ into the two unimodular cones spanned by $\{\mathbf{b}^1, \mathbf{w}\}$ and $\{\mathbf{b}^2, \mathbf{w}\}$. We have the inclusion-exclusion formula $g(\text{cone}\{\mathbf{b}_1, \mathbf{b}_2\}; \mathbf{z}) = g(\text{cone}\{\mathbf{b}_1, \mathbf{w}\}; \mathbf{z}) - g(\text{cone}\{\mathbf{b}_2, \mathbf{w}\}; \mathbf{z}) + g(\text{cone}\{\mathbf{w}\}; \mathbf{z})$.

It turns out that both cones C_1 and C_2 are unimodular, the only integer point in the fundamental parallelepiped being the origin. We obtain the rational generating functions

$$\begin{aligned} g(C_1; z_1, z_2) &= \frac{1}{(1-z_1)(1-z_2)}, \\ g(C_2; z_1, z_2) &= \frac{1}{(1-z_1 z_2^\alpha)(1-z_2)}, \\ g(C_1 \cap C_2; z_1, z_2) &= \frac{1}{1-z_1 z_2^\alpha}, \end{aligned}$$

hence the short formula

$$g(C'; z_1, z_2) = \frac{1}{(1-z_1)(1-z_2)} - \frac{1}{(1-z_1^1 z_2^\alpha)(1-z_2)} + \frac{1}{1-z_1 z_2^\alpha}.$$

In general, as we will see in the next section, we will not be able to obtain a decomposition into unimodular cones in one simple step; however, we will show that a signed decomposition can be constructed that provably reduces the index of cones very quickly.

We continue in the next section with the general multidimensional theory.

3.2 Preparation for n dimensions: Decompositions of polyhedra and cones

3.2.1 Indicator functions and inclusion–exclusion

Definition 3.1. The *indicator function* of a set $A \subseteq \mathbf{R}^d$ will be denoted by $[A]$; so $[A]\colon \mathbf{R}^d \to \mathbf{R}$ with $[A](\mathbf{x}) = 1$ if $\mathbf{x} \in A$ and 0 otherwise.

The indicator functions of sets of $\mathbf{R}^d$ span a vector space by pointwise addition (and scalar multiplication), which is also an *algebra* with respect to pointwise multiplication of functions.

These operations are convenient because they represent the disjoint union and intersection of sets:

$$[A] \cdot [B] = [A \cap B], \qquad [A] + [B] = [A \sqcup B].$$

Calculations with indicator functions often give short and elegant proofs. We illustrate this with the inclusion–exclusion identity that generalizes the formula

$$[A_1 \cup A_2] = [A_1] + [A_2] - [A_1 \cap A_2]$$

to several sets. We will use this formula shortly.

Lemma 3.2 (Inclusion–exclusion)**.** *Let $A_1, \dots, A_m \subseteq \mathbf{R}^n$. Then*

$$[A_1 \cup \cdots \cup A_m] = \sum_{\emptyset \neq I \subseteq [m]} (-1)^{|I|-1} \left[\bigcap_{i \in I} A_i \right]$$

(Here $[m]$ denotes $\{1, \dots, m\}$.)

Proof. We write the "de Morgan formula"

$$\mathbf{R}^n \setminus (A_1 \cup \cdots \cup A_m) = (\mathbf{R}^n \setminus A_1) \cap \cdots \cap (\mathbf{R}^n \setminus A_m)$$

as

$$1 - [A_1 \cup \cdots \cup A_m] = (1 - [A_1]) \cdots (1 - [A_m]).$$

Multiplying out gives:

$$1 - [A_1 \cup \cdots \cup A_m] = 1 + \sum_{\emptyset \neq I \subseteq [m]} (-1)^{|I|} \prod_{i \in I} [A_i],$$

which proves the result. □

3.2.2 Gram–Brianchon and Brion

We now apply this identity to obtain an interesting formula for the indicator function of a simplex in terms of the tangent cones of its faces.

Definition 3.3. Let $P \subseteq \mathbf{R}^d$ be a polyhedron. Let $\mathbf{x} \in P$. Then the *tangent cone* (*supporting cone*) of P at $\mathbf{x}$ is the shifted ("affine") polyhedral cone defined by

$$\operatorname{tcone}(P, \mathbf{x}) = \mathbf{x} + \operatorname{cone}(P - \mathbf{x}).$$

The tangent cone $\operatorname{tcone}(P, \mathbf{x})$ is the same as the cone of feasible directions, with apex shifted to $\mathbf{x}$. The inequality description of the tangent cones consists of the inequalities of P that are *active* (tight, satisfied with equality) at $\mathbf{x}$.

We will need the tangent cones of vertices and also, more generally, that of faces.

Definition 3.4. Let $F \subseteq P$ be a face. Then the *tangent cone* of F is defined as

$$\operatorname{tcone}(P, F) = \operatorname{tcone}(P, \mathbf{x}_F),$$

where $\mathbf{x}_F$ is any point in the relative interior of F.

Note that the tangent cone of a face F always contains the affine hull of the face, and so the tangent cones of vertices are the only pointed ones.

Theorem 3.5 (Gram–Brianchon)**.** *Let $P \subseteq \mathbf{R}^d$ be any polyhedron; then*

$$[P] = \sum_{\substack{\emptyset \neq F \\ \text{face of } P}} (-1)^{\dim F} [\operatorname{tcone}(P, F)],$$

where the summation includes the face $F = P$.

We prove it for the case of the standard simplex $\Delta = \operatorname{conv}\{\, \mathbf{e}_i : i = 1, \ldots, n \,\}$ only. The theorem holds, however, for arbitrary (also unbounded) polyhedra.

Proof. The simplex has the inequality description

$$\Delta = \big\{\, \mathbf{x} \in \mathbf{R}^n : \langle \mathbf{1}, \mathbf{x} \rangle = 1,\ x_i \geq 0 \text{ for } i = 1, \ldots, n \,\big\}.$$

Its affine hull is the hyperplane

$$A = \big\{\, \mathbf{x} \in \mathbf{R}^n : \langle \mathbf{1}, \mathbf{x} \rangle = 1 \,\big\}.$$

At a vertex $\mathbf{e}_i$, all inequalities "$x_j \geq 0$" for $j \neq i$ are tight, so the corresponding tangent cone has the description

$$\operatorname{tcone}(\Delta, \mathbf{e}_i) = \big\{\, \mathbf{x} \in A : x_j \geq 0 \text{ for } j \neq i \,\big\}.$$

More generally, the simplex has precisely 2^n faces (including the empty set and Δ itself), which are indexed by the subsets $I \subseteq \{1, \dots, n\}$,

$$\begin{aligned} F_I &= \{\, \mathbf{x} \in \Delta : x_j = 0 \text{ for } j \in I \,\} \\ &= \operatorname{conv}\{\, \mathbf{e}_i : i \notin I \,\}, \qquad\qquad \text{so } \dim F_I = n - |I| - 1, \end{aligned}$$

with

$$\operatorname{tcone}(\Delta, F_I) = \{\, \mathbf{x} \in A : x_j \geq 0 \text{ for } j \in I \,\}.$$

The facets (largest proper faces) of Δ are indexed by singletons $\{j\}$, and their tangent cones are affine halfspaces of A,

$$\operatorname{tcone}(\Delta, F_{\{j\}}) = \{\, \mathbf{x} \in A : x_j \geq 0 \,\} =: H_j.$$

Thus, in general, we can write each tangent cone as an intersection of these halfspaces.

$$\operatorname{tcone}(\Delta, F_I) = \bigcap_{j \in I} H_j,$$

and of course

$$[\Delta] = \bigcap_{j=1}^{n} H_j.$$

On the other hand, the *union* of the H_j is the entire affine space A (because if the sum of the coordinates, $\langle \mathbf{1}, \mathbf{x} \rangle$ is positive, at least one coordinate is positive.)

From our inclusion–exclusion lemma, we get

$$[A] = [H_1 \cup \cdots \cup H_n] = \sum_{\emptyset \neq I \subseteq [n]} (-1)^{|I|-1} \left[\bigcap_{i \in I} H_i \right].$$

Since $A = \operatorname{tcone}(\Delta, \Delta)$, we get

$$[\operatorname{tcone}(\Delta, \Delta)] = \sum_{\emptyset \neq I \subsetneq [n]} (-1)^{|I|-1} \big[\operatorname{tcone}(\Delta, F_I)\big] + (-1)^{|n|-1} [\Delta],$$

and, by rearranging, the desired identity. □

When we read this identity "modulo contributions of non-pointed polyhedra", it simplifies considerably because the only pointed tangent cones belong to the vertices; see Figure 3.7.

Corollary 3.6 (Brion's theorem: decomposition modulo non-pointed polyhedra)**.** *Let $P \subseteq \mathbf{R}^d$ be a polyhedron. Then*

$$[P] \equiv \sum_{\mathbf{v} \text{ vertex of } P} [\operatorname{tcone}(P, \mathbf{v})] \pmod{\textit{indicator functions of non-pointed polyhedra.}}$$

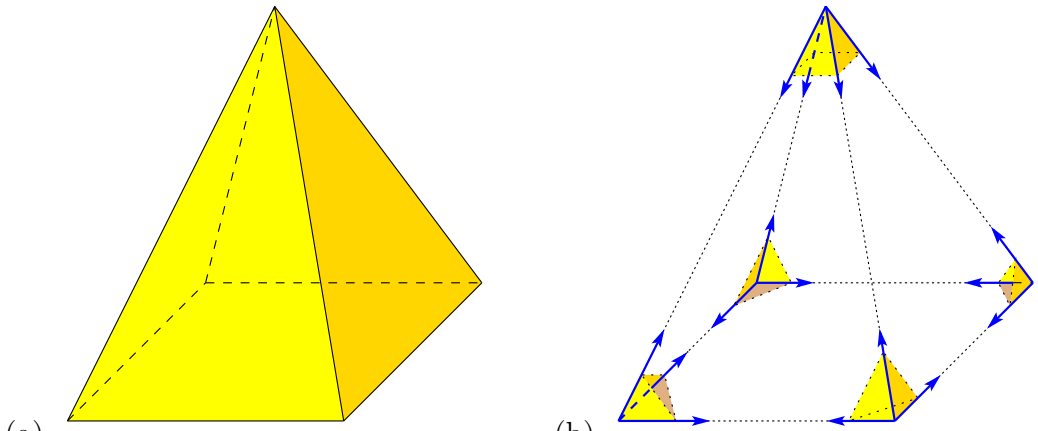


Figure 3.7: Brion's theorem

3.2.3 Avoiding inclusion–exclusion with half-open decompositions

In this section, based on Köppe and Verdoolaege (2008), we show that identities of indicator functions of full-dimensional polyhedra modulo lower-dimensional polyhedra can be translated to *exact* identities of indicator functions of full-dimensional half-open polyhedra. Thus, in triangulations etc., it is possible to avoid any hint of inclusion–exclusion. This improves the computational complexity of the methods.

Theorem 3.7. *Let*

$$\sum_{i\in I_1} \epsilon_i [P_i] + \sum_{i\in I_2} \epsilon_i [P_i] = 0 \tag{3.6}$$

be a (finite) linear identity of indicator functions of closed polyhedra $P_i \subseteq \mathbf{R}^d$, where the polyhedra P_i are full-dimensional for $i \in I_1$ and lower-dimensional for $i \in I_2$, and where $\epsilon_i \in \mathbf{Q}$. Let each closed polyhedron be given as

$$P_i = \left\{ \mathbf{x} : \langle \mathbf{b}^*_{i,j}, \mathbf{x} \rangle \le \beta_{i,j} \text{ for } j \in J_i \right\}. \tag{3.7}$$

Let $\mathbf{y} \in \mathbf{R}^d$ be a vector such that $\langle \mathbf{b}^*_{i,j}, \mathbf{y} \rangle \neq 0$ for all $i \in I_1 \cup I_2$, $j \in J_i$. For $i \in I_1$, we define the half-open polyhedron

$$\begin{aligned} \tilde{P}_i = \Big\{ \mathbf{x} \in \mathbf{R}^d : \langle \mathbf{b}^*_{i,j}, \mathbf{x} \rangle \le \beta_{i,j} \text{ for } j \in J_i \text{ with } \langle \mathbf{b}^*_{i,j}, \mathbf{y} \rangle < 0, \\ \langle \mathbf{b}^*_{i,j}, \mathbf{x} \rangle < \beta_{i,j} \text{ for } j \in J_i \text{ with } \langle \mathbf{b}^*_{i,j}, \mathbf{y} \rangle > 0 \Big\}. \end{aligned} \tag{3.8}$$

Then

$$\sum_{i\in I_1} \epsilon_i [\tilde{P}_i] = 0. \tag{3.9}$$

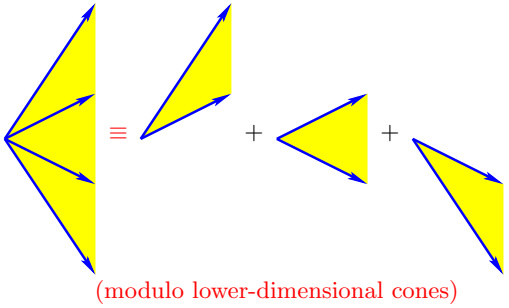


Figure 3.8: An identity, valid modulo lower-dimensional cones, corresponding to a polyhedral subdivision of a cone

The geometry of Theorem 3.7 is illustrated in Figure 3.8 and Figure 3.10.

Proof. We will show that (3.9) holds for an arbitrary $\bar{\mathbf{x}} \in \mathbf{R}^d$. To this end, fix an arbitrary $\bar{\mathbf{x}} \in \mathbf{R}^d$. We define

$$\mathbf{x}_\lambda = \bar{\mathbf{x}} + \lambda \mathbf{y} \quad \text{for } \lambda \in [0, +\infty).$$

Consider the function

$$f\colon [0, +\infty) \ni \lambda \mapsto \left(\sum_{i \in I_1} \epsilon_i [\tilde{P}_i] \right) (\mathbf{x}_\lambda).$$

We need to show that $f(0) = 0$. To this end, we first show that f is constant in a neighborhood of 0.

First, let $i \in I_1$ such that $\bar{\mathbf{x}} \in \tilde{P}_i$. For $j \in J_i$ with $\langle \mathbf{b}_{i,j}^*, \mathbf{y} \rangle < 0$, we have $\langle \mathbf{b}_{i,j}^*, \bar{\mathbf{x}} \rangle \leq \beta_{i,j}$, thus $\langle \mathbf{b}_{i,j}^*, \mathbf{x}_\lambda \rangle \leq \beta_{i,j}$. For $j \in J_i$ with $\langle \mathbf{b}_{i,j}^*, \mathbf{y} \rangle > 0$, we have $\langle \mathbf{b}_{i,j}^*, \bar{\mathbf{x}} \rangle < \beta_{i,j}$, thus $\langle \mathbf{b}_{i,j}^*, \mathbf{x}_\lambda \rangle < \beta_{i,j}$ for $\lambda > 0$ small enough. Hence, $\mathbf{x}_\lambda \in \tilde{P}_i$ for $\lambda > 0$ small enough.

Second, let $i \in I_1$ such that $\bar{\mathbf{x}} \notin \tilde{P}_i$. Then either there exists a $j \in J_i$ with $\langle \mathbf{b}_{i,j}^*, \mathbf{y} \rangle < 0$ and $\langle \mathbf{b}_{i,j}^*, \bar{\mathbf{x}} \rangle > \beta_{i,j}$. Then $\langle \mathbf{b}_{i,j}^*, \mathbf{x}_\lambda \rangle > \beta_{i,j}$ for $\lambda > 0$ small enough. Otherwise, there exists a $j \in J_i$ with $\langle \mathbf{b}_{i,j}^*, \mathbf{y} \rangle > 0$ and $\langle \mathbf{b}_{i,j}^*, \bar{\mathbf{x}} \rangle \geq \beta_{i,j}$. Then $\langle \mathbf{b}_{i,j}^*, \mathbf{x}_\lambda \rangle \geq \beta_{i,j}$. Hence, in either case, $\mathbf{x}_\lambda \notin \tilde{P}_i$ for $\lambda > 0$ small enough.

Next we show that f vanishes on some interval $(0, \lambda_0)$. We consider the function

$$g\colon [0, +\infty) \ni \lambda \mapsto \left(\sum_{i \in I_1} \epsilon_i [P_i] + \sum_{i \in I_2} \epsilon_i [P_i] \right) (\mathbf{x}_\lambda)$$

which is constantly zero by (3.6). Since $[P_i](\mathbf{x}_\lambda)$ for $i \in I_2$ vanishes on all but finitely many λ, we have

$$g(\lambda) = \left(\sum_{i \in I_1} \epsilon_i [P_i] \right) (\mathbf{x}_\lambda)$$

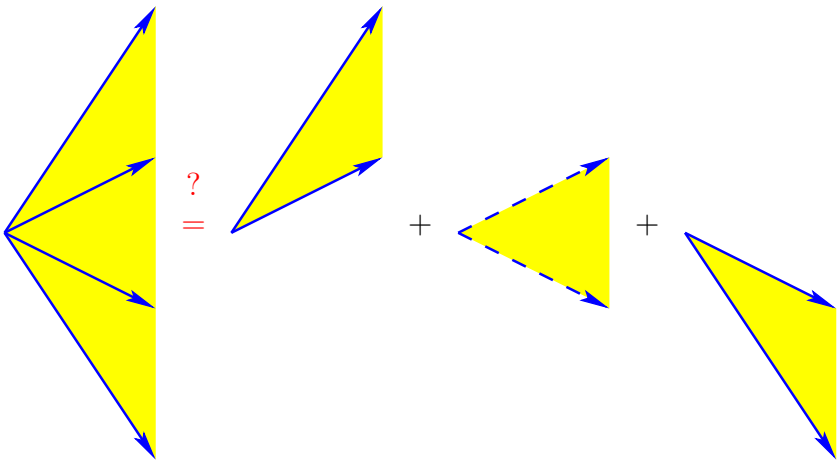


Figure 3.9: The technique of half-open exact decomposition. The above ad-hoc choice of strict inequalities (broken lines) and weak inequalities (solid lines) appears to give an exact identity on first look. However, the apex of the cone is still counted twice.

for λ from some interval $(0, \lambda_1)$. Also, $[P_i](\mathbf{x}_\lambda) = [\tilde{P}_i](\mathbf{x}_\lambda)$ for some interval $(0, \lambda_2)$. Hence $f(\lambda) = g(\lambda) = 0$ for some interval $(0, \lambda_0)$.

Hence, since f is constant in a neighborhood of 0, it is also zero at $\lambda = 0$. Thus the identity (3.9) holds for $\bar{\mathbf{x}}$. □

Remark 3.8. Theorem 3.7 can be easily generalized to a situation where the weights ϵ_i are not constants but continuous real-valued functions. In the proof, rather than showing that f is constant in a neighborhood of 0, one shows that f is continuous at 0.

The exact polyhedral subdivision of a closed polyhedral cone

For obtaining an exact polyhedral subdivision of a full-dimensional closed polyhedral cone $C = \text{cone}\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$,

$$[C] = \sum_{i \in I_1} [\tilde{C}_i],$$

we apply the above theorem using an arbitrary vector $\mathbf{y} \in \text{int}\, C$ that avoids all facets of the cones C_i, for instance

$$\mathbf{y} = \sum_{i=1}^{n} (1 + \gamma^i)\mathbf{b}_i$$

for a suitable $\gamma > 0$.

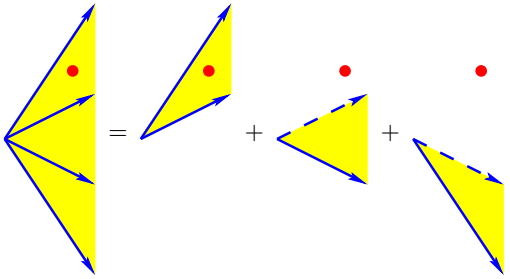


Figure 3.10: The technique of half-open exact decomposition. The relative location of the vector $\mathbf{y}$ (represented by a dot) determines which defining inequalities are strict (broken lines) and which are weak (solid lines).

The exact signed decomposition of half-open simplicial cones

Let $\tilde{C} \subseteq \mathbf{R}^d$ be a half-open simplicial full-dimensional cone with the double description

$$\tilde{C} = \left\{ \mathbf{x} \in \mathbf{R}^d : \langle \mathbf{b}_j^*, \mathbf{x} \rangle \leq 0 \text{ for } j \in J_\leq \text{ and } \langle \mathbf{b}_j^*, \mathbf{x} \rangle < 0 \text{ for } j \in J_< \right\} \tag{3.10}$$

$$\tilde{C} = \left\{ \sum_{j=1}^d \lambda_j \mathbf{b}_j : \lambda_j \geq 0 \text{ for } j \in J_\leq \text{ and } \lambda_j > 0 \text{ for } j \in J_< \right\} \tag{3.11}$$

where $J_< \cup J_\leq = \{1, \ldots, d\}$, with the *biorthogonality property* for the outer normal vectors $\mathbf{b}_j^*$ and the ray vectors $\mathbf{b}_i$,

$$\langle \mathbf{b}_j^*, \mathbf{b}_i \rangle = -\delta_{i,j} = \begin{cases} -1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

In the following we introduce a generalization of Barvinok's *signed decomposition* (Barvinok, 1994a) to half-open simplicial cones C_i, which will give an exact identity of half-open cones. To this end, we first compute the usual signed decomposition of the closed cone $C = \operatorname{cl} \tilde{C}$,

$$[C] \equiv \sum_i \epsilon_i [C_i] \quad \text{(mod lower-dimensional cones)} \tag{3.13}$$

using an extra ray $\mathbf{w}$, which has the representation

$$\mathbf{w} = \sum_{i=1}^d \alpha_i \mathbf{b}_i \quad \text{where } \alpha_i = -\langle \mathbf{b}_i^*, \mathbf{w} \rangle. \tag{3.14}$$

Each of the cones C_i is spanned by d vectors from the set $\{\mathbf{b}_1, \ldots, \mathbf{b}_d, \mathbf{w}\}$. The signs $\epsilon_i \in \{\pm 1\}$ are determined according to the location of $\mathbf{w}$, see (Barvinok, 1994a).

An exact identity

$$[\tilde{C}] = \sum_i \epsilon_i [\tilde{C}_i] \quad \text{with } \epsilon \in \{\pm 1\},$$

can now be obtained from (3.13) as follows. We define cones $\tilde{C}_i$ that are half-open counterparts of C_i. We only need to determine which of the defining inequalities of the cones $\tilde{C}_i$ should be strict. To this end, we first show how to construct a vector $\mathbf{y}$ that characterizes which defining inequalities of $\tilde{C}$ are strict by the means of (3.8).

Lemma 3.9. *Let*

$$\sigma_i = \begin{cases} 1 & \text{for } i \in J_{\leq}, \\ -1 & \text{for } i \in J_{<}, \end{cases} \tag{3.15}$$

and let $\mathbf{y} \in R = \operatorname{int} \operatorname{cone}\{\, \sigma_1 \mathbf{b}_i, \ldots, \sigma_d \mathbf{b}_d \}$ *be arbitrary. Then*

$$J_{\leq} = \big\{\, j \in \{1, \ldots, d\} : \langle \mathbf{b}_j^*, \mathbf{y} \rangle < 0 \,\big\},$$
$$J_{<} = \big\{\, j \in \{1, \ldots, d\} : \langle \mathbf{b}_j^*, \mathbf{y} \rangle > 0 \,\big\}.$$

We remark that the construction of such a vector $\mathbf{y}$ is not possible for a half-open non-simplicial cone in general.

Proof of Lemma 3.9. Such a $\mathbf{y}$ has the representation

$$\mathbf{y} = \sum_{i \in J_{\leq}} \lambda_i \mathbf{b}_i - \sum_{i \in J_{<}} \lambda_i \mathbf{b}_i \quad \text{with } \lambda_i > 0.$$

Thus

$$\langle \mathbf{b}_j^*, \mathbf{y} \rangle = \begin{cases} -\lambda_j & \text{for } j \in J_{\leq}, \\ +\lambda_j & \text{for } j \in J_{<}, \end{cases}$$

which proves the claim. □

Now let $\mathbf{y} \in R$ be an arbitrary vector that is not orthogonal to any of the facets of the cones $\tilde{C}_i$. Then such a vector $\mathbf{y}$ can determine which of the defining inequalities of the cones $\tilde{C}_i$ are strict.

In the following, we give a specific construction of such a vector $\mathbf{y}$. To this end, let $\mathbf{b}_m$ be the unique ray of $\tilde{C}$ that is not a ray of $\tilde{C}_i$. Then we denote by $\tilde{\mathbf{b}}_{0,m}^*$ the outer normal vector of the unique facet of $\tilde{C}_i$ not incident to $\mathbf{w}$. Now consider any facet F of a cone $\tilde{C}_i$ that is incident to $\mathbf{w}$. Since $\tilde{C}_i$ is simplicial, there is exactly one ray of $\tilde{C}_i$, say $\mathbf{b}_l$, not incident to F. The outer normal vector of the facet is therefore characterized up to scale by the indices l and m; thus we denote it by $\tilde{\mathbf{b}}_{l,m}^*$.

Let $\mathbf{b}_0 = \mathbf{w}$. Then, for every outer normal vector $\tilde{\mathbf{b}}_{l,m}^*$ and every ray $\mathbf{b}_i$, $i = 0, \ldots, d$, we have

$$\beta_{i;l,m} := -\langle \tilde{\mathbf{b}}_{l,m}^*, \mathbf{b}_i \rangle \begin{cases} > 0 & \text{for } i = l, \\ = 0 & \text{for } i \neq l, m, \\ \in \mathbf{R} & \text{for } i = m. \end{cases} \tag{3.16}$$

Now the outer normal vector has the representation

$$\tilde{\mathbf{b}}^*_{l,m} = \sum_{i=1}^{d} \beta_{i;l,m} \mathbf{b}^*_i.$$

The conditions of (3.16) determine the outer normal vector $\tilde{\mathbf{b}}^*_{l,m}$ up to scale. For the normals $\tilde{\mathbf{b}}^*_{0,m}$, we can choose

$$\tilde{\mathbf{b}}^*_{0,m} = \alpha_m \mathbf{b}^*_m. \tag{3.17}$$

For the other facets $\tilde{\mathbf{b}}^*_{l,m}$, we can choose

$$\tilde{\mathbf{b}}^*_{l,m} = |\alpha_m| \, \mathbf{b}^*_l - \operatorname{sign} \alpha_m \cdot \alpha_l \mathbf{b}^*_m. \tag{3.18}$$

Now consider

$$\mathbf{y} = \sum_{i=1}^{d} \sigma_i (|\alpha_i| + \gamma^i) \mathbf{b}_i, \tag{3.19}$$

which lies in the cone R for every $\gamma > 0$. We obtain

$$\langle \tilde{\mathbf{b}}^*_{0,m}, \mathbf{y} \rangle = -\sigma_m \alpha_m (|\alpha_m| + \gamma^m) \tag{3.20}$$

and

$$\begin{aligned} \langle \tilde{\mathbf{b}}^*_{l,m}, \mathbf{y} \rangle &= |\alpha_m| \, \langle \mathbf{b}^*_l, \mathbf{y} \rangle - \operatorname{sign} \alpha_m \cdot \alpha_l \langle \mathbf{b}^*_m, \mathbf{y} \rangle \\ &= -|\alpha_m| \, \sigma_l (|\alpha_l| + \gamma^l) + \operatorname{sign} \alpha_m \cdot \alpha_l \sigma_m (|\alpha_m| + \gamma^m) \\ &= (\operatorname{sign}(\alpha_l \alpha_m) \sigma_m - \sigma_l) \, |\alpha_l| \, |\alpha_m| \\ &\quad - \sigma_l \, |\alpha_m| \, \gamma^l + \operatorname{sign}(\alpha_l \alpha_m) \sigma_m \, |\alpha_l| \, \gamma^m, \end{aligned} \tag{3.21}$$

for $l \neq 0$. The right-hand side of (3.21), as a polynomial in γ, only has finitely many roots. Thus there are only finitely many values of γ for which a scalar product $\langle \tilde{\mathbf{b}}^*_{l,m}, \mathbf{y} \rangle$ can vanish for any of the finitely many facet normals $\tilde{\mathbf{b}}^*_{l,m}$. Let $\gamma > 0$ be an arbitrary number for which none of the scalar products vanishes. Then the vector $\mathbf{y}$ defined by (3.19) determines which of the defining inequalities of the cones $\tilde{C}_i$ should be strict.

Remark 3.10. It is possible to construct an a-priori vector $\mathbf{y}$ that is suitable to determine which defining inequalities are strict for all the cones that arise in the hierarchy of triangulations and signed decompositions of a cone $C = \operatorname{cone}\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ in Barvinok's algorithm. The construction uses the methods from Köppe (2007) and can be found in Köppe and Verdoolaege (2008).

Remark 3.11. For performing the exact signed decomposition in a software implementation, it is not actually necessary to construct the vector $\mathbf{y}$ and to evaluate scalar products. In the following, we show that we can devise simple, "combinatorial" rules to decide which defining inequalities should be strict. To this end, let $\gamma > 0$ in (3.19) be small enough that none of the signs

$$\sigma_{l,m} = -\operatorname{sign}\langle \tilde{\mathbf{b}}^*_{l,m}, \mathbf{y}\rangle$$

given by (3.21) change if γ is decreased even more. We can now determine $\sigma_{l,m}$ for all possible cases.

Case 0: $\alpha_m = 0$. The cone would be lower-dimensional in this case, since $\mathbf{w}$ lies in the space spanned by the ray vectors except $\mathbf{b}_m$, and is hence discarded.

Case 1: $l = 0$. From (3.20), we have

$$\sigma_{0,m} = \operatorname{sign}(\alpha_m)\sigma_m.$$

Case 2: $l \neq 0$, $\alpha_l = 0$, $\alpha_m \neq 0$. Here we have $\langle \tilde{\mathbf{b}}^*_{l,m}, \mathbf{y}\rangle = -\sigma_l \left|\alpha_m\right| \gamma^l$, thus

$$\sigma_{l,m} = \sigma_l.$$

Case 3: $l \neq 0$, $\alpha_l\alpha_m > 0$. In this case (3.21) simplifies to

$$\langle \tilde{\mathbf{b}}^*_{l,m}, \mathbf{y}\rangle = (\sigma_m - \sigma_l)\left|\alpha_l\right|\left|\alpha_m\right| - \sigma_l\left|\alpha_m\right|\gamma^l + \sigma_m\left|\alpha_l\right|\gamma^m. \tag{3.22}$$

Case 3 a: $\sigma_l = \sigma_m$. Here the first term of (3.22) cancels, so

$$\sigma_{l,m} = -\operatorname{sign}\langle \tilde{\mathbf{b}}^*_{l,m}, \mathbf{y}\rangle = \begin{cases} 1 & \text{if } l < m, \\ -1 & \text{if } l > m. \end{cases}$$

Case 3 b: $\sigma_l \neq \sigma_m$. Here the first term of (3.22) dominates, so

$$\sigma_{l,m} = -\operatorname{sign}\langle \tilde{\mathbf{b}}^*_{l,m}, \mathbf{y}\rangle = \sigma_l.$$

Case 4: $l \neq 0$, $\alpha_l\alpha_m < 0$. In this case (3.21) simplifies to

$$\langle \tilde{\mathbf{b}}^*_{l,m}, \mathbf{y}\rangle = -(\sigma_m + \sigma_l)\left|\alpha_l\right|\left|\alpha_m\right| - \sigma_l\left|\alpha_m\right|\gamma^l - \sigma_m\left|\alpha_l\right|\gamma^m. \tag{3.23}$$

Case 4 a: $\sigma_l = \sigma_m$. Here the first term of (3.23) dominates, so

$$\sigma_{l,m} = \sigma_l = \sigma_m.$$

Case 4 b: $\sigma_l \neq \sigma_m$. Here the first term of (3.23) cancels, so

$$\sigma_{l,m} = -\operatorname{sign}\langle \tilde{\mathbf{b}}^*_{l,m}, \mathbf{y}\rangle = \begin{cases} \sigma_l & \text{if } l < m, \\ \sigma_m & \text{if } l > m. \end{cases}$$

Further details and examples can be found in Köppe and Verdoolaege (2008).

3.3 Generating functions and the algorithm of Barvinok

Generating functions as formal Laurent series. Let $P \subseteq \mathbf{R}^d$ be a rational polyhedron. We first define its generating function as a formal series, that is, without any consideration of convergence properties. Since we do not wish to confine our polyhedron to the non-negative orthant, the generating function will have monomials with negative exponents. Therefore a formal *power series* is not sufficient; we need formal *Laurent series*.

We note that usually formal Laurent series are defined as series of the form

$$\sum_{(-M,\ldots,-M)\leq \boldsymbol{\alpha}\in\mathbf{Z}^d} c_{\boldsymbol{\alpha}}\mathbf{z}^{\boldsymbol{\alpha}}$$

for some integer M. Here (and in the following) we are using the *multi-exponent* notation $\mathbf{z}^{\boldsymbol{\alpha}} = \prod_{i=1}^{d} z_i^{\alpha_i}$. Since there is only a finite number of monomials with negative exponents, the multiplication of series is well-defined (because each coefficient of the product series is just a *finite* sum). Hence these "one-sided" infinite series form a ring, which is usually denoted by $\mathbf{Q}((z_1,\ldots,z_d))$.

However, in order to deal with arbitrary polyhedra, we will be in need of "two-sided" formal Laurent series, in which an infinite number of monomials with negative exponents can appear. Note that the multiplication of such series is not defined in general, so the series only form a module $\mathbf{Z}[[z_1\ldots,z_d,z_1^{-1},\ldots,z_d^{-1}]]$ (over the ring of integers $\mathbf{Z}$ or of Laurent polynomials $\mathbf{Z}[z_1,\ldots,z_d,z_1^{-1},\ldots,z_d^{-1}]$), but not a ring.

Definition 3.12. The *generating function* of $P \cap \mathbf{Z}^d$ is defined as the formal (two-sided) Laurent series

$$\tilde{g}(P;\mathbf{z}) = \sum_{\boldsymbol{\alpha}\in P\cap\mathbf{Z}^d} \mathbf{z}^{\boldsymbol{\alpha}} \in \mathbf{Z}[[z_1,\ldots,z_d,z_1^{-1},\ldots,z_d^{-1}]].$$

As we remarked in the introduction, the encoding of the set of lattice points of a polyhedron as a formal Laurent series does not give an immediate benefit in terms of complexity. We will get short formulas only when we can identify the Laurent series with certain rational functions.

The map from formal Laurent series to rational functions. If P is a bounded polyhedron (a polytope) or if P is unbounded, but does not contain any lattice point, then $\tilde{g}(P;\mathbf{z})$ is a *Laurent polynomial* (i.e., a *finite* sum of monomials with arbitrary – positive or negative – integer exponents). Clearly every such Laurent polynomial can be naturally identified with a rational function $g(P;\mathbf{z})$,

$$\begin{aligned}\mathbf{Z}[z_1,\ldots,z_d,z_1^{-1},\ldots,z_d^{-1}] &\hookrightarrow \mathbf{Q}(z_1,\ldots,z_d),\\ \mathbf{z}^{\boldsymbol{\alpha}} &\mapsto \mathbf{z}^{\boldsymbol{\alpha}}.\end{aligned}$$

Convergence comes into play whenever P is not bounded, since then $\tilde{g}(P;\mathbf{z})$ can be an infinite formal sum. We first consider the case of a *pointed* polyhedron P, i.e., P does not contain a straight line.

Theorem 3.13. *Let $P \subseteq \mathbf{R}^d$ be a pointed rational polyhedron. Then there exists a non-empty open subset $U \subseteq \mathbf{C}^d$ such that the series $\tilde{g}(P;\mathbf{z})$ converges absolutely and uniformly on every compact subset of U to a rational function $g(P;\mathbf{z}) \in \mathbf{Q}(z_1,\dots,z_d)$. The rational function $g(P;\mathbf{z})$ is independent from the choice of U.*

Remark 3.14. We remark that an arbitrary formal Laurent series $\tilde{g}(\mathbf{z}) \in \mathbf{Z}[[z_1,\dots,z_d, z_1^{-1},\dots,z_d^{-1}]]$, when it converges absolutely and uniformly on every compact subset of some non-empty open set U, usually defines a *meromorphic function* $g(\mathbf{z}) \in \mathbf{C}((z_1,\dots,z_d,z_1^{-1},\dots,z_d^{-1}))$ on U. This is a much larger class of functions than rational functions. This already happens when we allow irrational polyhedra.

Proof of Theorem 3.13 (sketch). First consider the case of a *simplicial rational cone*, i.e., $K = \mathbf{v} + B\mathbf{R}_+^d$ with linearly independent *basis vectors* $\mathbf{b}_1,\dots,\mathbf{b}_d$ (i.e., representatives of its extreme rays) given by the columns of some matrix $B \in \mathbf{Z}^{d\times d}$. We assume that the basis vectors are primitive vectors of the standard lattice $\mathbf{Z}^d$. Then the *index* of K is defined to be $\operatorname{ind} K = |\det B|$; it can also be interpreted as the cardinality of $\Pi \cap \mathbf{Z}^d$, where Π is the *fundamental parallelepiped* of K, i.e., the half-open parallelepiped

$$\Pi = \mathbf{v} + \left\{ \textstyle\sum_{i=1}^d \lambda_i \mathbf{b}_i : 0 \le \lambda_i < 1 \right\}.$$

We remark that the set $\Pi \cap \mathbf{Z}^d$ can also be seen as a set of representatives of the cosets of the lattice $B\mathbf{Z}^d$ in the standard lattice $\mathbf{Z}^d$; we shall make use of this interpretation in **??**. Then the generating function is, as illustrated in the two-dimensional case (section 3.1), given by a geometric series. The region U of convergence is related to the dual cone of K, and is always full-dimensional because K is pointed.

For the case of a *pointed rational cone* K, we first compute a triangulation into simplicial cones. Using the technique of subsection 3.2.3, we construct a set-theoretic partition of K into half-open simplicial cones. The corresponding series have domains of convergence that overlap in a full-dimensional set related to the dual cone of K.

Finally, for the case of a *pointed rational polyhedron* P, we cone over the polyhedron, i.e., we consider the cone $K := \{ (\mathbf{x}, \xi\mathbf{x}) : \mathbf{x} \in P,\ \xi \ge 0 \} \subset \mathbf{R}^{d+1}$. This is a pointed rational cone, to which we can associate the rational function $g(K;\mathbf{z},\zeta)$, where ζ corresponds to the extra dimension. Then

$$\frac{\partial}{\partial\zeta} g(K;\mathbf{z},\zeta)\Big|_{\zeta=0}$$

is the desired rational function for P. □

When P contains an integer point and also a straight line, there does not exist any point $\mathbf{z} \in \mathbf{C}^d$ where the series $\tilde{g}(P;\mathbf{z})$ converges absolutely.

Example 3.15. We consider the univariate two-sided infinite series

$$\tilde{g}(\mathbf{R}; z) = \sum_{k=-\infty}^{+\infty} z^k \in \mathbf{Z}[[z, z^{-1}]], \tag{3.24}$$

which is the generating function of $\mathbf{Z}$. It is clear that this series does not converge absolutely for any $z \in \mathbf{C}$, since for each z, the positive or the negative half-series of magnitudes diverges:

$$\sum_{k=0}^{+\infty} |z|^k = +\infty \qquad \text{for } |z| \geq 1,$$
$$\sum_{k=-\infty}^{0} |z|^k = +\infty \qquad \text{for } |z| \leq 1.$$

Now let us consider the general case. Let $\mathbf{x} \in P \cap \mathbf{Z}^d$ and $\mathbf{t} \in \mathbf{Z}^d \setminus \{\mathbf{0}\}$ such that $\mathbf{x} + \mathbf{t}\mathbf{R} \subseteq P$. Then $\tilde{g}(P; \mathbf{z})$ contains the subseries

$$\sum_{k=-\infty}^{+\infty} \mathbf{z}^{\mathbf{x}+k\mathbf{t}},$$

which is equivalent (by a monomial substitution) to series (3.24) from the example. Thus there does not exist any point where the series converges absolutely, so we cannot use convergence to define a rational function $g(P; \mathbf{z})$.

Lawrence (1991) and, independently, Pukhlikov and Khovanskii (1993) showed how to assign a rational function to arbitrary polyhedra in a "consistent" (valuative) way.

Theorem 3.16 (Lawrence–Khovanskii–Pukhlikov)**.** *There exists a* linear *map* F (valuation) *from the vector space spanned by the indicator functions of rational polyhedra in* $\mathbf{R}^d$ *to the space* $\mathbf{Q}(z_1, \ldots, z_d)$ *of rational functions such that:*

1. *For a pointed rational polyhedron* P*, the function* $F([P])(\mathbf{z})$ *equals the function* $g(P; \mathbf{z})$ *defined by the above convergent series.*

2. *For any integer vector* $\mathbf{t} \in \mathbf{Z}^d$*, we have* $F([\mathbf{t} + P]) = \mathbf{z}^{\mathbf{t}} F([P])$.

3. *For any non-pointed rational polyhedron* P*, we have* $F([P]) = 0$.

Proof (sketch). For pointed rational polyhedra, define F using g.

Show linearity of the map F for pointed rational polyhedra. Given a linear equation $\sum_i \alpha_i [P_i] = 0$, where the P_i are pointed, one needs to show that $\sum_i \alpha_i F[P_i] = 0$ holds. Note that the series associated with the P_i do not necessarily have a common

domain of convergence; but using inclusion–exclusion one can break the formula down to obtain a common domain of convergence, so the definition of $g(P_i, \mathbf{z})$ using convergent series can be used.

Thus, F can be extended by linearity to all rational polyhedra. The translation property holds for pointed polyhedra and clearly extends by linearity. Finally, if P is non-pointed, it contains a rational line, so $[P] = [\mathbf{t} + P]$ for some $\mathbf{0} \neq \mathbf{t} \in \mathbf{Z}^d$, and thus $F([P]) = \mathbf{z}^{\mathbf{t}} F([P])$, and so $F([P]) = 0$ because $\mathbf{z}^{\mathbf{t}}$ is not a zero-divisor. □

Taking all together, we define:

Definition 3.17. The rational function $g(P; \mathbf{z}) = F([P])(\mathbf{z}) \in \mathbf{Q}(z_1, \ldots, z_d)$ defined as above is called the *rational generating function* of $P \cap \mathbf{Z}^d$.

The theorem of Brion. From the decomposition of polyhedra, modulo non-pointed polyhedra, into vertex cones (Corollary 3.6), the following theorem follows.

Theorem 3.18 (Brion, 1988)**.** *Let P be a rational polyhedron and $V(P)$ be the set of vertices of P. Then,*

$$g_P(\mathbf{z}) = \sum_{\mathbf{v} \in V(P)} g_{C_P(\mathbf{v})}(\mathbf{z}),$$

where $C_P(\mathbf{v})$ is the tangent cone of $\mathbf{v}$.

It needs to be remarked that Brion obtained this theorem with different techniques and earlier than the results of Lawrence (1991) and Pukhlikov and Khovanskii (1993); see also Barvinok and Pommersheim (1999a). See also Beck et al. (2006) for an interesting discussion of this and related theorems.

We remark that in the case of a non-pointed polyhedron P, i.e., a polyhedron that has no vertices because it contains a straight line, both sides of the equation are zero.

Construction of Barvinok's signed decomposition. The missing link to an efficient algorithm is a procedure to compute a *signed decomposition* of the simplicial cone K (spanned by linearly independent integer vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$) to produce other simplicial cones with smaller index. Let $\mathbf{w}$ be any nonzero vector of $\mathbf{Z}^d$. Let K_i denote the cone spanned by the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{w}, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_d$. Then there exist $\epsilon_i \in \pm 1$ such that

$$[K] \equiv \sum_{i=1}^{d} \epsilon_i [K_i] \quad \text{(mod indicator functions of non-pointed polyhedra)}.$$

In general, these cones form a signed decomposition of K (see Figure 3.6); if $\mathbf{w}$ lies inside K, then $\epsilon_i = 1$, and the cones form a triangulation of K (see Figure 3.5). The goal is to simultaneously reduce the index of the cones K_i by a specific choice of $\mathbf{w}$.

Let $B^* = (B^{-1})^\top$, with column vectors $\mathbf{b}_1^*, \dots, \mathbf{b}_d^*$, be the biorthonormal basis, i.e., $\langle \mathbf{b}_i^*, \mathbf{b}_j \rangle = \delta_{ij}$. If we write $\mathbf{w} = \sum_{i=1}^d \alpha_i \mathbf{b}^i$, then $\alpha_i = \langle \mathbf{b}_i^*, \mathbf{w} \rangle$.

Collecting the determinants of the cones K_i into a vector, we obtain

$$\begin{aligned}\begin{pmatrix} \det(\mathbf{w}, \mathbf{b}_2, \dots, \mathbf{b}_{d-1}, \mathbf{b}_d) \\ \vdots \\ \det(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{d-1}, \mathbf{w}) \end{pmatrix} &= \begin{pmatrix} \det(\alpha_1 \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{d-1}, \mathbf{b}_d) \\ \vdots \\ \det(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{d-1}, \alpha_d \mathbf{b}_d) \end{pmatrix} \\ &= \det B \cdot \begin{pmatrix} \langle \mathbf{b}_1^*, \mathbf{w} \rangle \\ \vdots \\ \langle \mathbf{b}_d^*, \mathbf{w} \rangle \end{pmatrix} \\ &= \det B \cdot B^* \mathbf{w}.\end{aligned}$$

Since $\mathbf{w} \in \mathbf{Z}^n$ is arbitrary nonzero, simultaneously reducing the index means to solve the shortest vector problem, with respect to the ℓ_∞-norm, in the lattice generated by $\det B \cdot B^*$. (This construction is due to Dyer and Kannan (1997).) Using the bound from Lemma 2.6, we obtain

$$\|\mathbf{w}^*\|_\infty \le \left((\det B)^d (\det B)^{-1} \right)^{1/d} = (\det B)^{(d-1)/d}.$$

Thus the cones in the decomposition have

$$\log \operatorname{ind} K_i \le \tfrac{d-1}{d} \log \operatorname{ind} K;$$

that is, the logarithm of the index decreases geometrically in this construction.

The resulting cones and their intersecting proper faces (arising in an inclusion-exclusion formula) are recursively processed, until *unimodular* cones, i.e., cones of index 1, or cones of index smaller than some chosen constant are obtained. For such low-index cones $\mathbf{v} + B\mathbf{R}_+^d$, the rational generating function can be easily written down as

$$\frac{\sum_{\mathbf{a} \in \Pi \cap \mathbf{Z}^d} \mathbf{z}^{\mathbf{a}}}{\prod_{j=1}^d (1 - \mathbf{z}^{\mathbf{b}_j})}, \tag{3.25}$$

where Π is the fundamental parallelepiped of the cone.

In practical implementations of Barvinok's algorithm, one observes that in the hierarchy of cone decompositions, the index of the decomposed cones quickly descends from large numbers to fairly low numbers. The "last mile," i.e., decomposing many cones with fairly low index, creates a huge number of unimodular cones and thus is the bottleneck of the whole computation in many instances. In practice, one therefore stops decomposing cones that have an index smaller than about 1000.

The recursive decomposition of cones defines a *decomposition tree*. Due to the descent of the indices in the signed decomposition procedure, the following estimate holds for its depth:

Lemma 3.19 (Barvinok, 1994b)**.** *Let $B\mathbf{R}^d_+$ be a simplicial full-dimensional cone, whose basis is given by the columns of the matrix $B \in \mathbf{Z}^{d\times d}$. Let $D = |\det B|$. Then the depth of the decomposition tree is at most*

$$k(D) = \left\lfloor 1 + \frac{\log_2 \log_2 D}{\log_2 \frac{d}{d-1}} \right\rfloor . \tag{3.26}$$

Because at each decomposition step at most d cones are created and the depth of the tree is doubly logarithmic in the index of the input cone, one obtains a polynomiality result *in fixed dimension*, which was first proved by Barvinok (1994b):

Theorem 3.20 (Barvinok, 1994b)**.** *Let d be fixed. There exists a polynomial-time algorithm for computing the rational generating function of a polyhedron $P \subseteq \mathbf{R}^d$ given by rational inequalities.*

The above discussion contains algorithmic refinements upon Barvinok's original algorithm, which improve the theoretical and practical efficiency of the algorithm.

The computation of the integer points in the fundamental parallelepiped. To enumerate all points in $\Pi \cap \mathbf{Z}^d$ and compute the numerator of (3.25), we follow the technique of (Barvinok, 1993, Lemma 5.1), which we adapt for the case of half-open cones.

Lemma 3.21. *Let B be the matrix with the $\mathbf{b}_j$ as columns and let S be the Smith normal form of B, i.e., $BV = WS$, with V and W unimodular matrices and S a diagonal matrix $S = \operatorname{diag} \mathbf{s}$. Then, if Π is the fundamental parallelepiped of the half-open cone (section 3.2.3), then*

$$\Pi \cap \mathbf{Z}^d = \{\, \boldsymbol{\alpha}(\mathbf{k}) : k_j \in \mathbf{Z}, 0 \le k_j < s_j \,\},$$

with

$$\begin{aligned}\boldsymbol{\alpha}(\mathbf{k}) &= \mathbf{v} + \sum_{j\in J_\le} \operatorname{frac}\left\langle \mathbf{b}_j^*, \mathbf{v} - W\mathbf{k}\right\rangle \mathbf{b}_j + \sum_{j\in J_<} \left\{\!\left\{ \left\langle \mathbf{b}_j^*, \mathbf{v} - W\mathbf{k}\right\rangle \right\}\!\right\} \mathbf{b}_j \\ &= W\mathbf{k} - \sum_{j\in J_\le} \left\lfloor \left\langle \mathbf{b}_j^*, \mathbf{v} - W\mathbf{k}\right\rangle \right\rfloor \mathbf{b}_j - \sum_{j\in J_<} \left\lceil \left\langle \mathbf{b}_j^*, \mathbf{v} - W\mathbf{k}\right\rangle - 1 \right\rceil \mathbf{b}_j,\end{aligned}$$

with $\operatorname{frac}\cdot$ *the (lower) fractional part* $\operatorname{frac} x = x - \lfloor x \rfloor$ *and* $\{\!\{\cdot\}\!\}$ *the (upper) fractional part* $\{\!\{x\}\!\} = x - \lceil x - 1\rceil = 1 - \operatorname{frac} -x$.

Proof. It is clear that each $\boldsymbol{\alpha}(\mathbf{k}) \in \Pi \cap \mathbf{Z}^d$. To see that all integer points in Π are exhausted, note that $\det B = \det S$ and that all $\boldsymbol{\alpha}(\mathbf{k})$ are distinct. The latter follows from the fact that $\boldsymbol{\alpha}(\mathbf{k})$ can be written as $\boldsymbol{\alpha}(\mathbf{k}) = W\mathbf{k} + B\boldsymbol{\gamma} = W\mathbf{k} + WSV^{-1}\boldsymbol{\gamma}$ for some $\boldsymbol{\gamma} \in \mathbf{Z}^d$. If $\boldsymbol{\alpha}(\mathbf{k}_1) = \boldsymbol{\alpha}(\mathbf{k}_2)$, we must therefore have $\mathbf{k}_1 \equiv \mathbf{k}_2 \pmod{\mathbf{s}}$, i.e., $\mathbf{k}_1 = \mathbf{k}_2$. □

The overall Barvinok algorithm. We summarize Barvinok's algorithm below.

Algorithm 3.22 (Barvinok's algorithm, primal half-open variant)**.**
Input: A polyhedron $P \subset \mathbf{R}^d$ given by rational inequalities.
Output: The rational generating function for $P \cap \mathbf{Z}^d$ in the form

$$g_P(\mathbf{z}) = \sum_{i \in I} \epsilon_i \frac{\sum_{\mathbf{a} \in A_i} \mathbf{z}^{\mathbf{a}}}{\prod_{j=1}^{d} (1 - \mathbf{z}^{\mathbf{b}_{ij}})} \tag{3.27}$$

where $\epsilon_i \in \{\pm 1\}$, $\mathbf{a}_i \in \mathbf{Z}^d$, and $\mathbf{b}_{ij} \in \mathbf{Z}^d$.

1. Compute all vertices $\mathbf{v}_i$ and corresponding supporting cones C_i of P.
2. Triangulate C_i into simplicial cones C_{ij}
3. Replace cones by half-open variants.
4. Apply signed half-open decomposition to the cones $\mathbf{v}_i + C_{ij}$ to obtain unimodular (or low-index) cones $\mathbf{v}_i + C_{ijl}$.
5. Enumerate the integer points in the fundamental parallelepipeds of all resulting cones $\mathbf{v}_i + C_{ijl}$ to obtain the sets A_i.
6. Write down the formula (3.27).

Variants and implementations of the algorithm. The original algorithm by Barvinok (1994b) used decompositions into closed cones, inclusion–exclusion to handle the intersection of proper faces, and Lenstra's algorithm to obtain the decomposition vector for constructing the signed decomposition. The use of a shortest vector algorithm for this purpose appeared in Dyer and Kannan (1997). A dual variant of the algorithm appeared in Barvinok and Pommersheim (1999a); the "duality trick" (decomposing the polars of cones) allowed to remove the use of inclusion–exclusion to handle the intersection of proper faces. The practical benefit of stopping the decomposition before reaching unimodular cones was explored in Köppe (2007). A primal algorithm that avoids inclusion–exclusion using "irrational" (Beck and Sottile, 2007) decompositions appeared in Köppe (2007). The variant using half-open decompositions is a refinement of this technique, which appeared first in Köppe and Verdoolaege (2008). Since stopped decomposition works significantly better with a primal algorithm than a dual algorithm, as observed in Köppe (2007), the current method of choice is to use the primal half-open algorithm with stopped decomposition, which was presented in this section.

Many of these variants are implemented in the software packages `LattE macchiato` (Köppe, 2008) and `barvinok` (Verdoolaege, 2007). Instead of using the vertex cones of a polytope via Brion's theorem, one can also do the computations using the cone

over the polytope as in the proof of the Lawrence–Khovanskii–Pukhlikov theorem; this is known as the "homogenized" variant. For some polytopes, it is more efficient to compute triangulations in the dual space. Using dual triangulations and primal signed decompositions is known as the "primal" variant, whereas using primal triangulations and primal signed decompositions is known as the "all-primal" algorithm.

3.4 Evaluation (specialization)

We now compute the number of integer points $\#(P \cap \mathbf{Z}^n)$ from the multivariate rational generating function $g_P(\mathbf{z})$. This amounts to the problem of evaluating or *specializing* a rational generating function $g_P(\mathbf{z})$ at the point $\mathbf{z} = \mathbf{1}$. This is a pole of each of its summands but a regular point (removable singularity) of the function itself. From now on we call this the *specialization problem*.

To this end, let the generating function of a polytope $P \subseteq \mathbf{R}^n$ be given in the form

$$g_P(\mathbf{z}) = \sum_{i \in I} \epsilon_i \frac{\mathbf{z}^{\mathbf{a}_i}}{\prod_{j=1}^{s_i} (1 - \mathbf{z}^{\mathbf{b}_{ij}})} \tag{3.28}$$

where $\epsilon_i \in \{\pm 1\}$, $\mathbf{a}_i \in \mathbf{Z}^n$, and $\mathbf{b}_{ij} \in \mathbf{Z}^n \setminus \{\mathbf{0}\}$. Let $s = \max_{i \in I} s_i$ be the maximum number of binomials in the denominators. In general, if s is allowed to grow, more poles need to be considered for each summand, so the evaluation will need more computational effort.

Remark 3.23. In the literature, the specialization problem has been considered in various degrees of generality. In the original paper by Barvinok (1994b, Lemma 4.3), the dimension n is fixed, and each summand has exactly $s_i = n$ binomials in the denominator. The same restriction can be found in the survey by Barvinok and Pommersheim (1999b). In the more general algorithmic theory of monomial substitutions developed by Barvinok and Woods (2003), Woods (2004), there is no assumption on the dimension n, but the number s of binomials in the denominators is fixed. The same restriction appears in the paper by Verdoolaege and Woods (2008, Lemma 2.15). In a recent paper, Barvinok (2006a, section 5) gives a polynomial-time algorithm for the specialization problem for rational functions of the form

$$g(\mathbf{z}) = \sum_{i \in I} \epsilon_i \frac{\mathbf{z}^{\mathbf{a}_i}}{\prod_{j=1}^{s} (1 - \mathbf{z}^{\mathbf{b}_{ij}})^{\gamma_{ij}}} \tag{3.29}$$

where the dimension n is fixed, the number s of different binomials in each denominator equals n, but the multiplicity γ_{ij} is varying. Here we show that the technique from Barvinok (2006a, section 5) can be implemented in a way such that we obtain a polynomial-time algorithm even for the case of a general formula (3.27), when the dimension and the number of binomials are allowed to grow. The present section is based on De Loera et al. (2009a).

Theorem 3.24 (Polynomial-time specialization)**.** (a) *There exists an algorithm for computing the specialization of a rational function of the form*

$$g_P(\mathbf{z}) = \sum_{i \in I} \epsilon_i \frac{\mathbf{z}^{\mathbf{a}_i}}{\prod_{j=1}^{s_i} (1 - \mathbf{z}^{\mathbf{b}_{ij}})} \tag{3.30}$$

at its removable singularity $\mathbf{z} = \mathbf{1}$*, which runs in time polynomial in the encoding size of its data* $\epsilon_i \in \mathbf{Q}$*,* $\mathbf{a}_i \in \mathbf{Z}^n$ *for* $i \in I$ *and* $\mathbf{b}_{ij} \in \mathbf{Z}^n$ *for* $i \in I$*,* $j = 1, \ldots, s_i$*, even when the dimension* n *and the numbers* s_i *of terms in the denominators are not fixed.*

(b) *In particular, there exists a polynomial-time algorithm that, given data* $\epsilon_i \in \mathbf{Q}$*,* $\mathbf{a}_i \in \mathbf{Z}^n$ *for* $i \in I$ *and* $\mathbf{b}_{ij} \in \mathbf{Z}^n$ *for* $i \in I$*,* $j = 1, \ldots, s_i$ *describing a rational function in the form* (3.30)*, computes a vector* $\boldsymbol{\lambda} \in \mathbf{Q}^n$ *with* $\langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle \neq 0$ *for all* i, j *and rational weights* $w_{i,l}$ *for* $i \in I$ *and* $l = 0, \ldots, s_i$*. Then the number of integer points is given by*

$$\#(P \cap \mathbf{Z}^n) = \sum_{i \in I} \epsilon_i \sum_{l=0}^{s_i} w_{i,l} \langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle^l . \tag{3.31}$$

The remainder of this section contains the proof of Theorem 3.24. We follow Barvinok and Pommersheim (1999b) and recall the definition of Todd polynomials. We will prove that they can be efficiently evaluated in rational arithmetic.

Definition 3.25. We consider the function

$$H(x, \xi_1, \ldots, \xi_s) = \prod_{i=1}^{s} \frac{x\xi_i}{1 - \exp\{-x\xi_i\}},$$

a function that is analytic in a neighborhood of $\mathbf{0}$. The m-th (s-variate) *Todd polynomial* is the coefficient of x^m in the Taylor expansion

$$H(x, \xi_1, \ldots, \xi_s) = \sum_{m=0}^{\infty} \mathrm{td}_m(\xi_1, \ldots, \xi_s) x^m.$$

We remark that, when the numbers s and m are allowed to vary, the Todd polynomials have an exponential number of monomials.

Theorem 3.26. *The Todd polynomial* $\mathrm{td}_m(\xi_1, \ldots, \xi_s)$ *can be evaluated for given rational data* $\xi_1, \ldots, \xi_s$ *in time polynomial in* s*,* m*, and the encoding length of* $\xi_1, \ldots, \xi_s$*.*

The proof makes use of the following lemma.

Lemma 3.27. *The function $h(x) = x/(1 - \mathrm{e}^{-x})$ is a function that is analytic in a neighborhood of 0. Its Taylor series about $x = 0$ is of the form*

$$h(x) = \sum_{n=0}^{\infty} b_n x^n \quad \text{where} \quad b_n = \frac{1}{n!\,(n+1)!} c_n \tag{3.32}$$

with integer coefficients c_n that have a binary encoding length of $\mathrm{O}(n^2 \log n)$. The coefficients c_n can be computed from the recursion

$$\begin{aligned} c_0 &= 1 \\ c_n &= \sum_{j=1}^{n} (-1)^{j+1} \binom{n+1}{j+1} \frac{n!}{(n-j+1)!} c_{n-j} \quad \text{for } n = 1, 2, \ldots. \end{aligned} \tag{3.33}$$

Proof. The reciprocal function $h^{-1}(x) = (1 - \mathrm{e}^{-x})/x$ has the Taylor series

$$h^{-1}(x) = \sum_{i=0}^{\infty} a_n x^n \quad \text{with} \quad a_n = \frac{(-1)^n}{(n+1)!}.$$

Using the identity $h^{-1}(x)h(x) = \left(\sum_{n=0}^{\infty} a_n x^n\right)\left(\sum_{n=0}^{\infty} b_n x^n\right) = 1$, we obtain the recursion

$$\begin{aligned} b_0 &= \tfrac{1}{a_0} = 1 \\ b_n &= -(a_1 b_{n-1} + a_2 b_{n-2} + \cdots + a_n b_0) \quad \text{for } n = 1, 2, \ldots. \end{aligned} \tag{3.34}$$

We prove (3.32) inductively. Clearly $b_0 = c_0 = 1$. For $n = 1, 2, \ldots$, we have

$$\begin{aligned} c_n &= n!\,(n+1)!\,b_n \\ &= -n!\,(n+1)!\,(a_1 b_{n-1} + a_2 b_{n-2} + \cdots + a_n b_0) \\ &= n!\,(n+1)! \sum_{j=1}^{n} \frac{(-1)^{j+1}}{(j+1)!} \cdot \frac{1}{(n-j)!\,(n-j+1)!} c_{n-j} \\ &= \sum_{j=1}^{n} (-1)^{j+1} \frac{(n+1)!}{(j+1)!\,(n-j)!} \cdot \frac{n!}{(n-j+1)!} c_{n-j}. \end{aligned}$$

Thus we obtain the recursion formula (3.33), which also shows that all c_n are integers. A rough estimate shows that

$$|c_n| \le n(n+1)!\,n!\;|c_{n-1}| \le \left((n+1)!\right)^2 |c_{n-1}|\,,$$

thus $|c_n| \le \left((n+1)!\right)^{2n}$, so c_n has a binary encoding length of $\mathrm{O}(n^2 \log n)$. □

Proof of Theorem 3.26. By definition, we have

$$H(x, \xi_1, \ldots, \xi_s) = \sum_{m=0}^{\infty} \mathrm{td}_m(\xi_1, \ldots, \xi_s) x^m = \prod_{j=1}^{s} h(x\xi_j).$$

From Lemma 3.27 we have

$$h(x\xi_j) = \sum_{n=0}^{m} \beta_{j,n} x^n + o(x^m) \quad \text{where} \quad \beta_{j,n} = \frac{\xi_j^n}{n!\,(n+1)!} c_n \tag{3.35}$$

with integers c_n given by the recursion (3.33). Thus we can evaluate $\mathrm{td}_m(\xi_1, \ldots, \xi_s)$ by summing over all the possible compositions $n_1 + \cdots + n_s = m$ of the order m from the orders n_j of the factors:

$$\mathrm{td}_m(\xi_1, \ldots, \xi_s) = \sum_{\substack{(n_1,\ldots,n_s)\in\mathbf{Z}_+^s \\ n_1+\cdots+n_s=m}} \beta_{1,n_1} \ldots \beta_{s,n_s} \tag{3.36}$$

We remark that the length of the above sum is equal to the number of compositions of m into s non-negative parts,

$$\begin{aligned} \mathrm{C}'_s(m) &= \binom{m+s-1}{s-1} \\ &= \frac{(m+s-1)(m+s-2)\ldots(m+s-(s-1))}{(s-1)(s-2)\ldots 2\cdot 1} \\ &= \Omega\left(\left(1+\tfrac{m}{s-1}\right)^s\right), \end{aligned}$$

which is *exponential* in s (whenever $m \geq s$). Thus we cannot evaluate the formula (3.36) efficiently when s is allowed to grow.

However, we show that we can evaluate $\mathrm{td}_m(\xi_1, \ldots, \xi_s)$ more efficiently. To this end, we multiply up the s truncated Taylor series (3.35), one factor at a time, truncating after order m. Let us denote

$$\begin{aligned} H_1(x) &= h(x\xi_1) \\ H_2(x) &= H_1(x) \cdot h(x\xi_2) \\ &\vdots \\ H_s(x) &= H_{s-1}(x) \cdot h(x\xi_s) = H(x, \xi_1, \ldots, \xi_s). \end{aligned}$$

Each multiplication can be implemented in $\mathrm{O}(m^2)$ elementary rational operations. We finally show that all numbers appearing in the calculations have polynomial encoding size. Let Ξ be the largest binary encoding size of any of the rational numbers $\xi_1, \ldots, \xi_s$. Then every $\beta_{j,n}$ given by (3.35) has a binary encoding size $\mathrm{O}(\Xi n^5 \log^3 n)$. Let $H_j(x)$ have the truncated Taylor series $\sum_{n=0}^{m} \alpha_{j,n} x^n + o(x^m)$ and let A_j denote the largest binary encoding length of any $\alpha_{j,n}$ for $n \leq m$. Then

$$H_{j+1}(x) = \sum_{n=0}^{m} \alpha_{j+1,n} x^n + o(x^m) \quad \text{with} \quad \alpha_{j+1,n} = \sum_{l=0}^{n} \alpha_{j,l} \beta_{j,n-l}.$$

Thus the binary encoding size of $\alpha_{j+1,n}$ (for $n \le m$) is bounded by $A_j + \mathrm{O}(\Xi m^5 \log^3 m)$. Thus, after s multiplication steps, the encoding size of the coefficients is bounded by $\mathrm{O}(s\Xi m^5 \log^3 m)$, a polynomial quantity. □

Proof of Theorem 3.24. Parts (a) and (b). We recall the technique of Barvinok (1994b, Lemma 4.3), refined by Barvinok (2006a, section 5).

We first construct a rational vector $\boldsymbol{\lambda} \in \mathbf{Z}^n$ such that $\langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle \neq 0$ for all i, j. One such construction is to consider the *moment curve* $\boldsymbol{\lambda}(\xi) = (1, \xi, \xi^2, \dots, \xi^{n-1}) \in \mathbf{R}^n$. For each exponent vector $\mathbf{b}_{ij}$ occuring in a denominator of (3.27), the function $f_{ij} \colon \xi \mapsto \langle \boldsymbol{\lambda}(\xi), \mathbf{b}_{ij} \rangle$ is a polynomial function of degree at most $n-1$. Since $\mathbf{b}_{ij} \neq \mathbf{0}$, the function f_{ij} is not identically zero. Hence f_{ij} has at most $n-1$ zeros. By evaluating all functions f_{ij} for $i \in I$ and $j = 1, \dots, s_i$ at $M = (n-1)s|I|+1$ different values for ξ, for instance at the integers $\xi = 0, \dots, M$, we can find one $\xi = \bar{\xi}$ that is not a zero of any f_{ij}. Clearly this search can be implemented in polynomial time, even when the dimension n and the number s of terms in the denominators are not fixed. We set $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\bar{\xi})$.

For $\tau > 0$, let us consider the points $\mathbf{z}_\tau = \mathbf{e}^{\tau\boldsymbol{\lambda}} = (\exp\{\tau\lambda_1\}, \dots, \exp\{\tau\lambda_n\})$. We have

$$\mathbf{z}_\tau^{\mathbf{b}_{ij}} = \prod_{l=1}^{n} \exp\{\tau \lambda_l b_{ijl}\} = \exp\{\tau \langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle\};$$

since $\langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle \neq 0$ for all i, j, all the denominators $1 - \mathbf{z}_\tau^{\mathbf{b}_{ij}}$ are nonzero. Hence for every $\tau > 0$, the point $\mathbf{z}_\tau$ is a regular point not only of $g(\mathbf{z})$ but also of the individual summands of (3.27). We have

$$\begin{aligned}
g(\mathbf{1}) &= \lim_{\tau \to 0^+} \sum_{i \in I} \epsilon_i \frac{\mathbf{z}_\tau^{\mathbf{a}_i}}{\prod_{j=1}^{s_i} (1 - \mathbf{z}_\tau^{\mathbf{b}_{ij}})} \\
&= \lim_{\tau \to 0^+} \sum_{i \in I} \epsilon_i \frac{\exp\{\tau \langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle\}}{\prod_{j=1}^{s_i} (1 - \exp\{\tau \langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle\})} \\
&= \lim_{\tau \to 0^+} \sum_{i \in I} \epsilon_i \, \tau^{-s_i} \exp\{\tau \langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle\} \prod_{j=1}^{s_i} \frac{\tau}{1 - \exp\{\tau \langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle\}} \\
&= \lim_{\tau \to 0^+} \sum_{i \in I} \epsilon_i \, \tau^{-s_i} \exp\{\tau \langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle\} \prod_{j=1}^{s_i} \frac{-1}{\langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle} h(-\tau \langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle) \\
&= \lim_{\tau \to 0^+} \sum_{i \in I} \epsilon_i \frac{(-1)^{s_i}}{\prod_{j=1}^{s_i} \langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle} \tau^{-s_i} \exp\{\tau \langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle\} H(\tau, -\langle \boldsymbol{\lambda}, \mathbf{b}_{i1} \rangle, \dots, -\langle \boldsymbol{\lambda}, \mathbf{b}_{is_i} \rangle)
\end{aligned}$$

where $H(x, \xi_1, \dots, \xi_{s_i})$ is the function from Definition 3.25. We will compute the limit by computing the constant term of the Laurent expansion of each summand about $\tau = 0$. Now the function $\tau \mapsto \exp\{\tau \langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle\}$ is holomorphic and has the Taylor

series

$$\exp\{\tau \langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle\} = \sum_{l=0}^{s_i} \alpha_{i,l} \tau^l + o(\tau^{s_i}) \quad \text{where} \quad \alpha_{i,l} = \frac{\langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle^l}{l!}, \tag{3.37}$$

and $H(\tau, \xi_1, \ldots, \xi_{s_i})$ has the Taylor series

$$H(\tau, \xi_1, \ldots, \xi_s) = \sum_{m=0}^{s_i} \mathrm{td}_m(\xi_1, \ldots, \xi_s) \tau^m + o(\tau^{s_i}).$$

Because of the factor τ^{-s_i}, which gives rise to a pole of order s_i in the summand, we can compute the constant term of the Laurent expansion by summing over all the possible compositions $s_i = l + (s_i - l)$ of the order s_i:

$$g(\mathbf{1}) = \sum_{i \in I} \epsilon_i \frac{(-1)^{s_i}}{\prod_{j=1}^{s_i} \langle \boldsymbol{\lambda}, \mathbf{b}_{ij} \rangle} \sum_{l=0}^{s_i} \frac{\langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle^l}{l!} \mathrm{td}_{s_i - l}(-\langle \boldsymbol{\lambda}, \mathbf{b}_{i1} \rangle, \ldots, -\langle \boldsymbol{\lambda}, \mathbf{b}_{is_i} \rangle). \tag{3.38}$$

We use the notation

$$w_{i,l} = (-1)^{s_i} \frac{\mathrm{td}_{s_i - l}(-\langle \boldsymbol{\lambda}, \mathbf{b}_{i,1} \rangle, \ldots, -\langle \boldsymbol{\lambda}, \mathbf{b}_{i,s_i} \rangle)}{l! \cdot \langle \boldsymbol{\lambda}, \mathbf{b}_{i,1} \rangle \cdots \langle \boldsymbol{\lambda}, \mathbf{b}_{i,s_i} \rangle} \quad \text{for } i \in I \text{ and } l = 0, \ldots, s_i;$$

these rational numbers can be computed in polynomial time using Theorem 3.26. We now obtain the formula of the claim,

$$g(\mathbf{1}) = \sum_{i \in I} \epsilon_i \sum_{l=0}^{s_i} w_{i,l} \langle \boldsymbol{\lambda}, \mathbf{a}_i \rangle^l .$$

□

A general theorem on monomial specialization appears in Barvinok and Woods (2003).

3.5 Boolean operations and projections

Barvinok and Woods (2003) further developed a set of powerful manipulation rules for using these short rational functions in Boolean constructions on various sets of lattice points.

Here we assume that the polyhedron $P = \{ \mathbf{u} \in \mathbf{R}^n : A\mathbf{u} \leq \mathbf{b} \}$ is bounded.

Theorem 3.28 (Intersection Lemma; Theorem 3.6 in Barvinok and Woods (2003))**.** *Let ℓ be a fixed integer. Let S_1, S_2 be finite subsets of $\mathbf{Z}^n$. Let $g(S_1; \mathbf{x})$ and $g(S_2; \mathbf{x})$ be their generating functions, given as short rational functions with at most ℓ binomials in each denominator. Then there exists a polynomial time algorithm, which computes*

$$g(S_1 \cap S_2; \mathbf{x}) = \sum_{i \in I} \gamma_i \frac{\mathbf{x}^{\mathbf{c}_i}}{(1 - \mathbf{x}^{\mathbf{d}_{i1}}) \ldots (1 - \mathbf{x}^{\mathbf{d}_{is}})}$$

with $s \leq 2\ell$, where the γ_i are rational numbers, $\mathbf{c}_i, \mathbf{d}_{ij}$ are nonzero integer vectors, and I is a polynomial-size index set.

The following theorem was proved by Barvinok and Woods using Theorem 3.28:

Theorem 3.29 (Boolean Operations Lemma; Corollary 3.7 in Barvinok and Woods (2003))**.** *Let m and ℓ be fixed integers. Let $S_1, S_2, \ldots, S_m$ be finite subsets of $\mathbf{Z}^n$. Let $g(S_i; \mathbf{x})$ for $i = 1, \ldots, m$ be their generating functions, given as short rational functions with at most ℓ binomials in each denominator. Let a set $S \subseteq \mathbf{Z}^n$ be defined as a Boolean combination of $S_1, \ldots, S_m$ (i.e., using any of the operations $\cup$, $\cap$, $\setminus$). Then there exists a polynomial time algorithm, which computes*

$$g(S; \mathbf{x}) = \sum_{i \in I} \gamma_i \frac{\mathbf{x}^{\mathbf{c}_i}}{(1 - \mathbf{x}^{\mathbf{d}_{i1}}) \ldots (1 - \mathbf{x}^{\mathbf{d}_{is}})}$$

where $s = s(\ell, m)$ is a constant, the γ_i are rational numbers, $\mathbf{c}_i, \mathbf{d}_{ij}$ are nonzero integer vectors, and I is a polynomial-size index set.

We will use the *Intersection Lemma* and the *Boolean Operations Lemma* to extract special monomials present in the expansion of a generating function. The essential step in the intersection algorithm is the use of the *Hadamard product* (Barvinok and Woods, 2003, Definition 3.2) and a special monomial substitution. The Hadamard product is a bilinear operation on rational functions (we denote it by $*$). The computation is carried out for pairs of basic rational summands. Note that the Hadamard product $m_1 * m_2$ of two monomials m_1, m_2 is zero unless $m_1 = m_2$.

Another key subroutine introduced by Barvinok and Woods is the following *Projection Theorem*.

Theorem 3.30 (Projection Theorem; Theorem 1.7 in Barvinok and Woods (2003))**.** *Assume the dimension n is a fixed constant. Consider a rational polytope $P \subset \mathbf{R}^n$ and a linear map $T\colon \mathbf{Z}^n \to \mathbf{Z}^k$. There is a polynomial time algorithm which computes a short representation of the generating function $f\big(T(P \cap \mathbf{Z}^n); \mathbf{x}\big)$.*

This results again uses algorithmic geometry of numbers techniques, such as the computation of a *Kannan partition* of the parameter space of parametric polyhedra into cells where the flatness direction is constant. Details, using a strengthened construction of a Kannan partition due to Eisenbrand and Shmonin (2008) can be found in Köppe et al. (2008b).

Chapter 4

Mixed-integer polynomial optimization via the summation method

$$\max\{\, f(\mathbf{x}) : \mathbf{x} \in P \cap \mathbf{Z}^d \,\}$$
$$= \lim_{k \to \infty} \left\{ f^k \left(z_1 \frac{\partial}{\partial z_1}, \dots \right) g(P; \mathbf{z}) \Big|_{\mathbf{z}=\mathbf{1}} \right\}^{1/k}$$

Here we consider the problem

$$\begin{array}{rl} \max/\min & f(x_1,\dots,x_n) \\ \text{subject to} & A\mathbf{x} \le \mathbf{b} \\ & \mathbf{x} \in \mathbf{R}^{n_1} \times \mathbf{Z}^{n_2}, \end{array} \tag{4.1}$$

where A is a rational matrix and $\mathbf{b}$ is a rational vector, and where f is a polynomial function of maximum total degree D with rational coefficients. We are interested in general polynomial objective functions f *without any convexity assumptions.*

As we pointed out in the introduction (Theorem 1.7), optimizing degree-4 polynomials over problems with two integer variables ($n_1 = 0$, $n_2 = 2$) is already a hard problem. Thus, even when we fix the dimension, we cannot get a polynomial-time algorithm for solving the optimization problem. The best we can hope for, even when the number of both the continuous and the integer variables is fixed, is an approximation result.

We present here the FPTAS obtained in De Loera et al. (2006a,b, 2008a), which uses the "summation method" and short rational generating functions.

4.1 The summation method

The summation method for optimization is the idea to use of elementary relation

$$\max\{s_1,\dots,s_N\} = \lim_{k\to\infty} \sqrt[k]{s_1^k + \cdots + s_N^k}, \tag{4.2}$$

which holds for any finite set $S = \{s_1,\dots,s_N\}$ of non-negative real numbers. This relation can be viewed as an approximation result for ℓ_k-norms. Now if P is a polytope and f is an objective function non-negative on $P \cap \mathbf{Z}^d$, let $\mathbf{x}^1,\dots,\mathbf{x}^N$ denote all the feasible integer solutions in $P \cap \mathbf{Z}^d$ and collect their objective function values $s_i = f(\mathbf{x}^i)$ in a vector $\mathbf{s} \in \mathbf{Q}^N$. Then, comparing the unit balls of the ℓ_k-norm and the ℓ_∞-norm (Figure 4.1), we get the relation

$$L_k := N^{-1/k}\|\mathbf{s}\|_k \le \|\mathbf{s}\|_\infty \le \|\mathbf{s}\|_k =: U_k.$$

These estimates are independent of the function f. (Different estimates that make use of the properties of f, and that are suitable also for the continuous case, can be obtained from the Hölder inequality; see for instance Baldoni et al. (2010).)

Thus, for obtaining a good approximation of the maximum, it suffices to solve a summation problem of the polynomial function $h = f^k$ on $P \cap \mathbf{Z}^d$ for a value of k that is large enough. Indeed, for $k = \lceil (1 + 1/\epsilon)\log N \rceil$, we obtain $U_k - L_k \le \epsilon f(\mathbf{x}^{\max})$. On the other hand, this choice of k is polynomial in the input size (because $1/\epsilon$ is encoded in unary in the input, and $\log N$ is bounded by a polynomial in the binary encoding size of the polytope P). Hence, when the dimension d is fixed, we can expand the polynomial function f^k as a list of monomials in polynomial time.

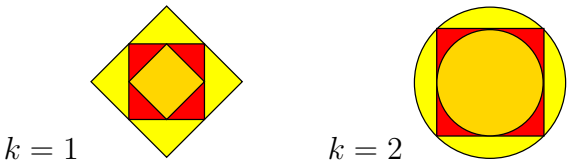


Figure 4.1: Approximation properties of ℓ_k-norms

Below we show how to solve the summation problem for the pure integer case. This lead to the FPTAS for the pure integer, non-negative case in section 4.2. Then we show an extension to mixed-integer optimization by discretization in section 4.3, i.e., the following result:

Theorem 4.1 (Fully polynomial-time approximation scheme)**.** *Let the dimension $d = d_1 + d_2$ be fixed. There exists a fully polynomial time approximation scheme (FPTAS) for the maximization problem (4.1) for all polynomial functions $f(x_1, \dots, x_{d_1}, z_1, \dots, z_{d_2})$ with rational coefficients that are non-negative on the feasible region.*

Finally, in section 4.4, we study the mixed-integer case with a different notion of approximation that is suitable for objective functions that can take negative values on the feasible region. The usual definition of an FPTAS uses the notion of ϵ-approximation that is common when considering combinatorial optimization problems, where the approximation error is compared to the optimal solution value,

$$\big|f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) - f(\mathbf{x}_{\max}, \mathbf{z}_{\max})\big| \leq \epsilon f(\mathbf{x}_{\max}, \mathbf{z}_{\max}), \tag{4.3}$$

where $(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)$ denotes an approximate solution and $(\mathbf{x}_{\max}, \mathbf{z}_{\max})$ denotes a maximizer of the objective function. In section 4.4, we now compare the approximation error to the *range* of the objective function on the feasible region,

$$\big|f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) - f(\mathbf{x}_{\max}, \mathbf{z}_{\max})\big| \leq \epsilon \big|f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\big|, \tag{4.4}$$

where additionally $(\mathbf{x}_{\min}, \mathbf{z}_{\min})$ denotes a *minimizer* of the objective function on the feasible region. This notion of approximation was proposed by various authors (Bellare and Rogaway, 1993, de Klerk et al., 2006, Vavasis, 1993). It enables us to study objective functions that are not restricted to be non-negative on the feasible region. Indeed we prove:

Theorem 4.2 (Fully polynomial-time weak-approximation scheme)**.** *Let the dimension $d = d_1 + d_2$ be fixed. Let f be an arbitrary polynomial function with rational coefficients and maximum total degree D, and let $P \subset \mathbf{R}^d$ be a rational convex polytope.*

(a) *In time polynomial in the input size and D, it is possible to decide whether f is constant on $P \cap \big(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\big)$.*

(b) *In time polynomial in the input size, D, and $\frac{1}{\epsilon}$ it is possible to compute a solution* $(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) \in P \cap \left(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\right)$ *with*

$$\left|f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) - f(\mathbf{x}_{\max}, \mathbf{z}_{\max})\right| \leq \epsilon \left|f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\right|.$$

4.2 FPTAS for optimizing non-negative polynomials over integer points of polytopes

Here we prove the following theorem (FPTAS) for the pure integer case.

Theorem 4.3 (FPTAS for maximizing non-negative polynomials over finite lattice point sets)**.** *For all fixed integers k (dimension) and s (maximum number of binomials in the denominator), there exists an algorithm with running time polynomial in the encoding size of the problem and $\frac{1}{\epsilon}$ for the following problem.*

Input: *Let $V \subseteq \mathbf{Z}^k$ be a finite set, given by a rational generating function in the form*

$$g(V; \mathbf{x}) = \sum_{i \in I} \gamma_i \frac{\mathbf{x}^{\mathbf{c}_i}}{(1 - \mathbf{x}^{\mathbf{d}_{i1}}) \dots (1 - \mathbf{x}^{\mathbf{d}_{is_i}})}$$

where the the numbers s_i of binomials in the denominators are at most s. Furthermore, let two vectors $\mathbf{v}_\mathrm{L}$, $\mathbf{v}_\mathrm{U} \in \mathbf{Z}^k$ be given such that V is contained in the box $\{\, \mathbf{v} : \mathbf{v}_\mathrm{L} \leq \mathbf{v} \leq \mathbf{v}_\mathrm{U} \,\}$.

Let $f \in \mathbf{Q}[v_1, \dots, v_k]$ be a polynomial with rational coefficients that is non-negative on V, given by a list of its monomials, whose coefficients are encoded in binary and whose exponents are encoded in unary.

Finally, let $\epsilon \in \mathbf{Q}$.

Output: *Compute a point $\mathbf{v}_\epsilon \in V$ that satisfies*

$$f(\mathbf{v}_\epsilon) \geq (1 - \epsilon) f^* \quad \textit{where} \quad f^* = \max_{\mathbf{v} \in V} f(\mathbf{v}).$$

Remark 4.4. A version of this result first appeared in De Loera et al. (2006b). There the result was stated and proved only for sets V that consist of the lattice points of a rational polytope; however, the same proof yields the result above.

The proof uses the summation method. To solve the summation problem, we use the technique of short rational generating functions. We start with a simple, one-dimensional example. Once more, we consider the generating function of the integer points of the interval $P = [0, 4]$,

$$g(P; z) = z^0 + z^1 + z^2 + z^3 + z^4 = \frac{1}{1 - z} - \frac{z^5}{1 - z}.$$

We now apply the differential operator $z\frac{\mathrm{d}}{\mathrm{d}z}$ and obtain

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

Applying the same differential operator again, we obtain

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

We have thus evaluated the monomial function $h(\alpha) = \alpha^2$ for $\alpha = 0, \ldots, 4$; the results appear as the coefficients of the respective monomials. Substituting $z = 1$ yields the desired sum

$$\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right)\left(z\frac{\mathrm{d}}{\mathrm{d}z}\right) g(P;z)\Big|_{z=1} = 1+4+9+16 = 30$$

The idea now is to evaluate this sum instead by computing the limit of the rational function for $z \to 1$,

$$\sum_{\alpha=0}^{4} \alpha^2 = \lim_{z\to 1}\left[\frac{z+z^2}{(1-z)^3} - \frac{25z^5-39z^6+16z^7}{(1-z)^3}\right];$$

again this evaluation problem can be solved using residue techniques.

Such differential operators can be constructed and efficiently applied in general.

Lemma 4.5 (Barvinok, 2006b)**.** *Let $g_P(\mathbf{z})$ be the rational generating function of the lattice points of P. Let f be a polynomial in $\mathbf{Z}[x_1, \ldots, x_d]$ of maximum total degree D. We can compute, in time polynomial on D and the size of the input data, a rational generating function $g_{P,f}(z)$ that represents $\sum_{\boldsymbol{\alpha}\in P\cap\mathbf{Z}^d} f(\boldsymbol{\alpha})\mathbf{z}^{\boldsymbol{\alpha}}$.*

Proof (sketch). In general, we apply the differential operator

$$D_f = f\big(z_1\frac{\partial}{\partial z_1}, \ldots, z_n\frac{\partial}{\partial z_n}\big).$$

Consider first the case $f(\mathbf{z}) = z_r$. Consider the action of the differential operator $z_r\frac{\partial}{\partial z_r}$ in the rational generating function $g_P(\mathbf{z})$. On one hand, for the generating function

$$z_r\frac{\partial}{\partial z_r} g_P(\mathbf{z}) = \sum_{\boldsymbol{\alpha}\in P\cap\mathbf{Z}^d} z_r\frac{\partial}{\partial z_r}\mathbf{z}^{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\alpha}\in P\cap\mathbf{Z}^d} \alpha_r\mathbf{z}^{\boldsymbol{\alpha}}.$$

On the other hand, by linearity of the operator, we have that in terms of rational functions

$$z_r\frac{\partial}{\partial z_r} g_P(z) = \sum_{i\in I}\epsilon_i z_r\frac{\partial}{\partial z_r}\ \frac{\mathbf{z}^{\mathbf{u}_i}}{\prod_{j=1}^{d}(1-\mathbf{z}^{\mathbf{v}_{ij}})}\ .$$

Thus it is enough to prove that the summands of the expression above can be written in terms of rational functions computable in polynomial time. The quotient rule for derivatives says that

$$\frac{\partial}{\partial z_r}\,\frac{\mathbf{z}^{\mathbf{u}_i}}{\prod_{j=1}^{d}(1-z^{v_{ij}})} = \frac{(\frac{\partial \mathbf{z}^{\mathbf{u}_i}}{\partial z_r})\prod_{j=1}^{d}(1-\mathbf{z}^{\mathbf{v}_{ij}}) - z^{u_i}(\frac{\partial}{\partial z_r}\prod_{j=1}^{d}(1-\mathbf{z}^{\mathbf{v}_{ij}}))}{\prod_{j=1}^{d}(1-\mathbf{z}^{\mathbf{v}_{ij}})^2}.$$

We can expand the numerator as a sum of no more than 2^d monomials. This is a constant number because d, the number of variables, is assumed to be a constant.

For the case of f being any monomial, repeat this construction, remembering to cancel powers in the denominator after the repeated application of the quotient rule (this is important). The general case of a polynomial f of many monomial terms follows by linearity. □

Now we present the algorithm to obtain bounds U_k, L_k that reach the optimum. Step 1 of preprocessing is necessary because we rely on the elementary fact that, for a collection $S = \{s_1, \ldots, s_r\}$ of non-negative real numbers, maximum$\{s_i | s_i \in S\}$ equals $\lim_{k\to\infty} \sqrt[k]{\sum_{j=1}^{r} s_j^k}$.

Algorithm

Input: A rational convex polytope $P \subset \mathbf{R}^d$, a polynomial objective $f \in \mathbf{Z}[x_1, \ldots, x_d]$ of maximum total degree D.

Output: An increasing sequence of lower bounds L_k, and a decreasing sequence of upper bounds U_k reaching the maximal function value f^* of f over all lattice points of P.

Step 1. If f is known to be non-negative in all points of P, then go directly to Step 2. Else, solving $2d$ linear programs over P, we find lower and upper integer bounds for each of the variables $x_1, \ldots, x_d$. Let M be the maximum of the absolute values of these $2d$ numbers. Thus $|x_i| \leq M$ for all i. Let C be the maximum of the absolute values of all coefficients, and r be the number of monomials of $f(x)$. Then

$$L := -rCM^D \leq f(x) \leq rCM^D =: U,$$

as we can bound the absolute value of each monomial of $f(x)$ by CM^D. Replace f by $\bar{f}(x) = f(x) - L \leq U - L$, a non-negative polynomial over P. Go to Steps 2, 3, etc. and return the optimal value of $\bar{f}$. Trivially, if we find the optimal value of $\bar{f}$ over P we can extract the optimal value for f.

Step 2. Via Barvinok's algorithm (see Barvinok (1994b, 2006b), Barvinok and Pommersheim (1999a)), compute a short rational function expression for the generating function $g_P(z) = \sum_{\alpha \in P \cap \mathbf{Z}^d} z^\alpha$. From $g_P(z)$ compute the number $|P \cap \mathbf{Z}^d| = g_P(1)$ of lattice points in P in polynomial time.

Step 3. From the rational function representation $g_P(z)$ of the generating function $\sum_{\alpha \in P \cap \mathbf{Z}^d} z^\alpha$ compute the rational function representation of $g_{P,f^k}(z)$ of $\sum_{\alpha \in P \cap \mathbf{Z}^d} f^k(\alpha) z^\alpha$ in polynomial time by application of Lemma 4.5. We define

$$L_k := \sqrt[k]{g_{P,f^k}(1)/g_{P,f^0}(1)} \quad \text{and} \quad U_k := \sqrt[k]{g_{P,f^k}(1)}.$$

When $\lfloor U_k \rfloor - \lceil L_k \rceil < 1$ stop and return $\lceil L_k \rceil = \lfloor U_k \rfloor$ as the optimal value.

Lemma 4.6. *The algorithm is correct.*

Proof. Using the fact that the arithmetic mean of a finite set of nonnegative values is at most as big as the maximum value, which in turn is at most as big as the sum of all values, we obtain the sequences of lower and upper bounds, L_k and U_k, for the maximum:

$$L_k = \sqrt[k]{\frac{\sum_{\alpha \in P \cap \mathbf{Z}^d} f(\alpha)^k}{|P \cap \mathbf{Z}^d|}} \leq \max\{f(\alpha) : \alpha \in P \cap \mathbf{Z}^d\} \leq \sqrt[k]{\sum_{\alpha \in P \cap \mathbf{Z}^d} f(\alpha)^k} = U_k.$$

Note that as $s \to \infty$, L_k and U_k approach this maximum value monotonously (from below and above, respectively). Trivially, if the difference between (rounded) upper and lower bounds becomes strictly less than 1, we have determined the value $\max\{f(x) : x \in P \cap \mathbf{Z}^d\} = \lceil L_k \rceil$. Thus the algorithm terminates with the correct answer. □

The main theorem will follow from the next lemma:

Lemma 4.7. *Let f be a polynomial with integer coefficients and maximum total degree D. When the dimension d is fixed,*

1. *the bounds L_k, U_k can be computed in time polynomial in k, the input size of P and f, and the total degree D. The bounds satisfy the following inequality:*

$$U_k - L_k \leq f^* \cdot \left(\sqrt[k]{|P \cap \mathbf{Z}^d|} - 1 \right).$$

2. *In addition, when f is non-negative over P (i.e. $f(x) \geq 0$ for all $x \in P$), for $k = (1+1/\epsilon)\log(|P \cap \mathbf{Z}^d|)$, L_k is a $(1-\epsilon)$-approximation to the optimal value f^* and it can be computed in time polynomial in the input size, the total degree D, and $1/\epsilon$. Similarly, U_k gives a $(1+\epsilon)$-approximation to f^*. Moreover, with the same complexity, one can also find a feasible lattice point that approximates an optimal solution with similar quality.*

Proof. Part (i). From Lemma 4.5 on fixed dimension d, we can compute $g_{P,f} = \sum_{\alpha\in P\cap\mathbf{Z}^d} f(\alpha)z^\alpha$ as a rational function in time polynomial in D, the total degree of f, and the input size of P. Thus, because f^k has total degree of Dk and the encoding length for the coefficients of f^k is bounded by $k\log(kC)$ (with C the largest coefficient in f), we can also compute $g_{P,f^k} = \sum_{\alpha\in P\cap\mathbf{Z}^d} f^k(\alpha)z^\alpha$ in time polynomial in k, the total degree D, and the input size of P. Note that using residue techniques Barvinok (2006b), we can evaluate $g_{P,f^k}(1)$ in polynomial time. Finally observe

$$\begin{aligned} U_k - L_k &= \sqrt[k]{\sum_{\alpha\in P\cap\mathbf{Z}^d} f^k(\alpha)} - \sqrt[k]{\frac{\sum_{\alpha\in P\cap\mathbf{Z}^d} f^k(\alpha)}{|P\cap\mathbf{Z}^d|}} = \sqrt[k]{\frac{\sum_{\alpha\in P\cap\mathbf{Z}^d} f^k(\alpha)}{|P\cap\mathbf{Z}^d|}}\left(\sqrt[k]{|P\cap\mathbf{Z}^d|}-1\right) \\ &= L_k\left(\sqrt[k]{|P\cap\mathbf{Z}^d|}-1\right) \le f^*\left(\sqrt[k]{|P\cap\mathbf{Z}^d|}-1\right). \end{aligned}$$

Part (ii). Note that if $\left(\sqrt[k]{|P\cap\mathbf{Z}^d|}-1\right) \le \epsilon$ then L_k is indeed a $(1-\epsilon)$-approximation because

$$f^* \le U_k = L_k + (U_k - L_k) \le L_k + f^*\left(\sqrt[k]{|P\cap\mathbf{Z}^d|}-1\right) \le L_k + f^*\epsilon.$$

Observe that $\phi(\epsilon) := (1+1/\epsilon)/(1/\log(1+\epsilon))$ is an increasing function for $\epsilon < 1$ and $\lim_{\epsilon\to 0}\phi(\epsilon) = 1$, thus $\phi(\epsilon) \ge 1$ for $0 < \epsilon \le 1$. Hence, for all $k \ge \log(|P\cap\mathbf{Z}^d|) + \log(|P\cap\mathbf{Z}^d|)/\epsilon \ge \log(|P\cap\mathbf{Z}^d|)/\log(1+\epsilon)$, we have indeed $\left(\sqrt[k]{|P\cap\mathbf{Z}^d|}-1\right) \le \epsilon$. Finally, from Lemma 4.5, the calculation of L_k for $k = \log(|P\cap\mathbf{Z}^d|)+\log(|P\cap\mathbf{Z}^d|)/\epsilon$ would require a number of steps polynomial in the input size and $1/\epsilon$. A very similar argument can be written for U_k but we omit it here.

To complete the proof of part (ii) it remains to show that not only we approximate the optimal value f^* but we can also efficiently find a lattice point α with $f(\alpha)$ giving that quality approximation of f^*. Let $k = (1+1/\epsilon)\log(|P\cap\mathbf{Z}^d|)$, thus, by the above discussion, L_k is an $(1-\epsilon)$-approximation to f^*. Let $Q_0 := [-M, M]^d$ denote the box computed in Step 1 of the algorithm such that $P \subseteq Q_0$. By bisecting Q_0, we obtain two boxes Q_1' and Q_1''. By applying the algorithm separately to the polyhedra $P\cap Q_1'$ and $P\cap Q_1''$, we compute lower bounds L_k' and L_k'' for the optimization problems restricted to Q_1' and Q_1'', respectively. Because L_k^k is the arithmetic mean of $f^k(\alpha)$ for $\alpha \in P\cap\mathbf{Z}^d$, clearly

$$\min\{L_k', L_k''\} \le L_k \le \max\{L_k', L_k''\}.$$

Without loss of generality, let $L_k' \ge L_k''$. We now apply the bisection procedure iteratively on Q_k'. After $d\log M$ bisection steps, we obtain a box Q_k' that contains a single lattice point $\alpha \in P\cap Q_k'\cap Z^d$, which has an objective value $f(\alpha) = L_k' \ge L_k \ge (1-\epsilon)f^*$. □

We remark that if we need to apply the construction of Step 1 of the algorithm because f takes negative values on P, then we can only obtain an $(1-\epsilon)$-approximation (and $(1+\epsilon)$-approximation, respectively) for the modified function $\bar{f}$ in polynomial time, but not the original function f. We also emphasize that, although our algorithm requires the computation of $\sum_{\alpha\in P} f^q(\alpha)$ for different powers of f, these numbers are obtained without explicitly listing all lattice points (a hard task), nor we assume any knowledge of the individual values $f(\alpha)$. We can access the power means $\sum_{\alpha\in P} f^q(\alpha)$ indirectly via rational functions. Here are two small examples:

Example 1, monomial optimization over a quadrilateral: The problem we consider is that of maximizing the value of the monomial x^3y over the lattice points of the quadrilateral

$$\{(x,y)|3991 \le 3996\,x - 4\,y \le 3993,\ 1/2 \le x \le 5/2\}.$$

It contains only 2 lattice points. The sum of rational functions encoding the lattice points is

$$\frac{x^2y^{1000}}{(1-(xy^{999})^{-1})\,(1-y^{-1})} + \frac{xy}{(1-xy^{999})\,(1-y^{-1})} + \frac{xy}{(1-xy^{999})\,(1-y)} + \frac{x^2y^{1000}}{(1-(xy^{999})^{-1})\,(1-y)}.$$

In the first iteration $L_1 = 4000.50$ while $U_1 = 8001$. After thirty iterations, we see $L_{30} = 7817.279750$ while $U_{30} = 8000$, the true optimal value.

Example 2, nvs04 from MINLPLIB: A somewhat more complicated example, from a well-known library of test examples (see http://www.gamsworld.org/minlp/), is the problem given by

$$\begin{aligned} \min\quad & 100\left(\frac{1}{2}+i_2-\left(\frac{3}{5}+i_1\right)^2\right)^2+\left(\frac{2}{5}-i_1\right)^2 \\ \text{s.t.}\quad & i_1, i_2 \in [0,200]\cap\mathbf{Z}. \end{aligned} \tag{4.5}$$

Its optimal solution as given in MINLPLIB is $i_1 = 1$, $i_2 = 2$ with an objective value of 0.72. Clearly, to apply our algorithm from page 62 literally, the objective function needs to be multiplied by a factor of 100 to obtain an integer valued polynomial.

Using the bounds on i_1 and i_2 we obtain an upper bound of $165\cdot 10^9$ for the objective function, which allows us to convert the problem into an equivalent maximization problem, where all feasible points have a non-negative objective value. The new

optimal objective value is 164999999999.28. Expanding the new objective function and translating it into a differential operator yields

$$\frac{4124999999947}{25}\mathrm{Id} - 28z_2\frac{\partial}{\partial z_2} + \frac{172}{5}z_1\frac{\partial}{\partial z_1} - 117\left(z_1\frac{\partial}{\partial z_1}\right)^{(2)} - 100\left(z_2\frac{\partial}{\partial z_2}\right)^{(2)}$$
$$+ 240\left(z_2\frac{\partial}{\partial z_2}\right)\left(z_1\frac{\partial}{\partial z_1}\right) + 200\left(z_2\frac{\partial}{\partial z_2}\right)\left(z_1\frac{\partial}{\partial z_1}\right)^{(2)} - 240\left(z_1\frac{\partial}{\partial z_1}\right)^{(3)} - 100\left(z_1\frac{\partial}{\partial z_1}\right)^{(4)}.$$

The short generating function can be written as $g(z_1, z_2) = \left(\frac{1}{1-z_1} - \frac{z_1^{201}}{1-z_1}\right)\left(\frac{1}{1-z_2} - \frac{z_2^{201}}{1-z_2}\right)$.

In this example, the number of lattice points is $|P \cap \mathbf{Z}^2| = 40401$. The first bounds are $L_1 = 139463892042.292155534$, $U_1 = 28032242300500.723262442$. After 30 iterations the bounds become $L_{30} = 164999998845.993553019$ and $U_{30} = 165000000475.892451381$.

4.3 Extension to mixed-integer optimization via discretization

Our main approach is to use grid refinement in order to approximate the mixed-integer optimal value via auxiliary pure integer problems. One of the difficulties on constructing approximations is the fact that not every sequence of grids whose widths converge to zero leads to a convergent sequence of optimal solutions of grid optimization problems. This difficulty is addressed in subsection 4.3.1. In subsection 4.3.2 we develop techniques for bounding differences of polynomial function values. Section 4.3.3 contains the proof of the main theorem.

4.3.1 Grid approximation results

An important step in the development of an FPTAS for the mixed-integer optimization problem is the reduction of the mixed-integer problem (**??**) to an auxiliary optimization problem over a lattice $\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}$. To this end, we consider the *grid problem* with grid size m,

$$\begin{aligned} \max \quad & f(x_1, \ldots, x_{d_1}, z_1, \ldots, z_{d_2}) \\ \text{s.t.} \quad & A\mathbf{x} + B\mathbf{z} \leq \mathbf{b} \\ & x_i \in \tfrac{1}{m}\mathbf{Z} \qquad \text{for } i = 1, \ldots, d_1, \\ & z_i \in \mathbf{Z} \qquad \text{for } i = 1, \ldots, d_2. \end{aligned} \tag{4.6}$$

We can solve this problem approximately using the integer FPTAS (Lemma 4.7):

Corollary 4.8. *For fixed dimension $d = d_1 + d_2$ there exists an algorithm with running time polynomial in $\log m$, the encoding length of f and of P, the maximum*

total degree D of f, and $\frac{1}{\epsilon}$ for computing a feasible solution $(\mathbf{x}_\epsilon^m, \mathbf{z}_\epsilon^m) \in P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$ to the grid problem (4.6) with an objective function f that is non-negative on the feasible region, with

$$f(\mathbf{x}_\epsilon^m, \mathbf{z}_\epsilon^m) \geq (1-\epsilon) f(\mathbf{x}^m, \mathbf{z}^m), \tag{4.7}$$

where $(\mathbf{x}^m, \mathbf{z}^m) \in P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$ is an optimal solution to (4.6).

Proof. We apply Lemma 4.7 to the pure integer optimization problem:

$$\begin{array}{lll} \max & \tilde{f}(\tilde{\mathbf{x}}, \mathbf{z}) & \\ \text{s.t.} & A\tilde{\mathbf{x}} + mB\mathbf{z} \leq m\mathbf{b} & \\ & \tilde{x}_i \in \mathbf{Z} & \text{for } i = 1, \ldots, d_1, \\ & z_i \in \mathbf{Z} & \text{for } i = 1, \ldots, d_2, \end{array} \tag{4.8}$$

where $\tilde{f}(\tilde{\mathbf{x}}, \mathbf{z}) := m^D f(\frac{1}{m}\tilde{\mathbf{x}}, \mathbf{z})$ is a polynomial function with integer coefficients. Clearly the binary encoding length of the coefficients of $\tilde{f}$ increases by at most $\lceil D \log m \rceil$, compared to the coefficients of f. Likewise, the encoding length of the coefficients of mB and $m\mathbf{b}$ increases by at most $\lceil \log m \rceil$. By Theorem 1.1 of De Loera et al. (2006b), there exists an algorithm with running time polynomial in the encoding length of $\tilde{f}$ and of $A\mathbf{x} + mB\mathbf{z} \leq m\mathbf{b}$, the maximum total degree D, and $\frac{1}{\epsilon}$ for computing a feasible solution $(\mathbf{x}_\epsilon^m, \mathbf{z}_\epsilon^m) \in P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$ such that $\tilde{f}(\mathbf{x}_\epsilon^m, \mathbf{z}_\epsilon^m) \geq (1-\epsilon)\tilde{f}(\mathbf{x}^m, \mathbf{z}^m)$, which implies the estimate (4.7). □ □

One might be tempted to think that for large-enough choice of m, we immediately obtain an approximation to the mixed-integer optimum with arbitrary precision. However, this is not true, as the following example demonstrates.

Example 4.9. Consider the mixed-integer optimization problem

$$\begin{array}{ll} \max & 2z - x \\ \text{s.t.} & z \leq 2x \\ & z \leq 2(1-x) \\ & x \in \mathbf{R}_{\geq 0},\ z \in \{0, 1\}, \end{array} \tag{4.9}$$

whose feasible region consists of the point $(\frac{1}{2}, 1)$ and the segment $\{(x, 0) : x \in [0, 1]\}$. The unique optimal solution to (4.9) is $x = \frac{1}{2}$, $z = 1$. Now consider the sequence of grid approximations of (4.9) where $x \in \frac{1}{m}\mathbf{Z}_{\geq 0}$. For even m, the unique optimal solution to the grid approximation is $x = \frac{1}{2}$, $z = 1$. However, for odd m, the unique optimal solution is $x = 0$, $z = 0$. Thus the full sequence of the optimal solutions to the grid approximations does not converge since it has two limit points; see Figure 4.2.

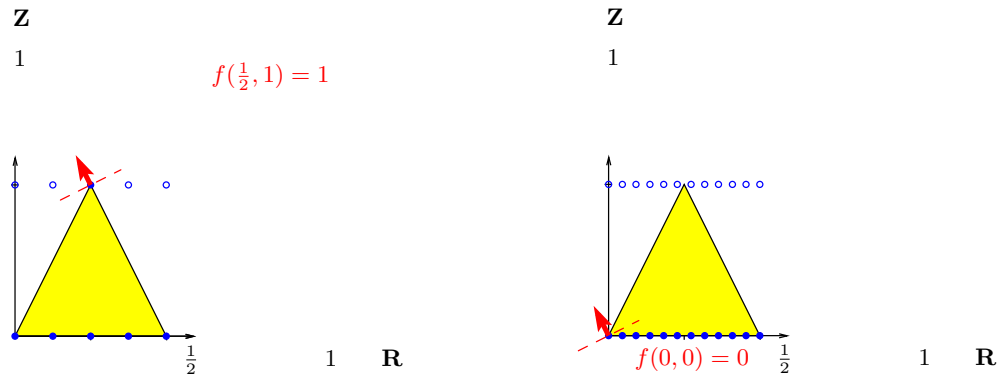


Figure 4.2: A sequence of optimal solutions to grid problems with two limit points, for even m and for odd m

Even though taking the limit does not work, taking the upper limit does. More strongly, we can prove that it is possible to construct, in polynomial time, a subsequence of finer and finer grids that contain a lattice point $(\lfloor \mathbf{x}^* \rceil_\delta, \mathbf{z}^*)$ that is arbitrarily close to the mixed-integer optimum $(\mathbf{x}^*, \mathbf{z}^*)$. This is the central statement of this section and a basic building block of the approximation result.

Theorem 4.10 (Grid Approximation)**.** *Let d_1 be fixed. Let $P = \{ (\mathbf{x}, \mathbf{z}) \in \mathbf{R}^{d_1+d_2} : A\mathbf{x} + B\mathbf{z} \leq \mathbf{b} \}$, where $A \in \mathbf{Z}^{p \times d_1}$, $B \in \mathbf{Z}^{p \times d_2}$. Let $M \in \mathbf{R}$ be given such that $P \subseteq \{ (\mathbf{x}, \mathbf{z}) \in \mathbf{R}^{d_1+d_2} : |x_i| \leq M \text{ for } i = 1, \ldots, d_1 \}$. There exists a polynomial-time algorithm to compute a number Δ such that for every $(\mathbf{x}^*, \mathbf{z}^*) \in P \cap (\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2})$ and $\delta > 0$ the following property holds:*

> *Every lattice $\frac{1}{m}\mathbf{Z}^{d_1}$ for $m = k\Delta$ and $k \geq \frac{2}{\delta} d_1 M$ contains a lattice point $\lfloor \mathbf{x}^* \rceil_\delta$ such that $(\lfloor \mathbf{x}^* \rceil_\delta, \mathbf{z}^*) \in P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$ and $\left\| \lfloor \mathbf{x}^* \rceil_\delta - \mathbf{x}^* \right\|_\infty \leq \delta$.*

The geometry of Theorem 4.10 is illustrated in Figure 4.3. The notation $\lfloor \mathbf{x}^* \rceil_\delta$ has been chosen to suggest that the coordinates of $\mathbf{x}^*$ have been "rounded" to obtain a nearby lattice point. The rounding method is provided by the next two lemmas; Theorem 4.10 follows directly from them.

Lemma 4.11 (Integral Scaling Lemma)**.** *Let $P = \{ (\mathbf{x}, \mathbf{z}) \in \mathbf{R}^{d_1+d_2} : A\mathbf{x} + B\mathbf{z} \leq \mathbf{b} \}$, where $A \in \mathbf{Z}^{p \times d_1}$, $B \in \mathbf{Z}^{p \times d_2}$. For fixed d_1, there exists a polynomial time algorithm to compute a number $\Delta \in \mathbf{Z}_{>0}$ such that for every $\mathbf{z} \in \mathbf{Z}^{d_2}$ the polytope*

$$\Delta P_{\mathbf{z}} = \{ \Delta \mathbf{x} : (\mathbf{x}, \mathbf{z}) \in P \}$$

is integral, i.e., all vertices have integer coordinates. In particular, the number Δ has an encoding length that is bounded by a polynomial in the encoding length of P.

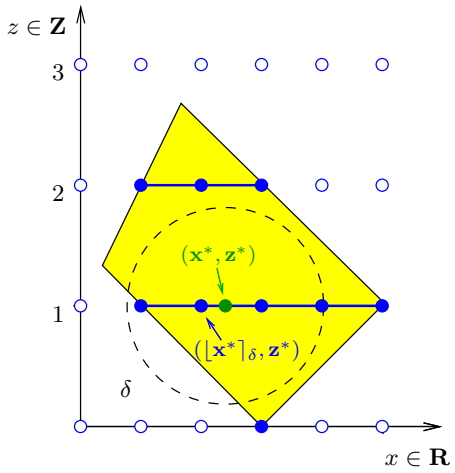


Figure 4.3: The principle of grid approximation. Since we can refine the grid only in the direction of the continuous variables, we need to construct an approximating grid point $(\mathbf{x}, \mathbf{z}^*)$ in the same integral slice as the target point $(\mathbf{x}^*, \mathbf{z}^*)$.

Proof. Because the dimension d_1 is fixed, there exist only polynomially many simplex bases of the inequality system $A\mathbf{x} \leq \mathbf{b} - B\mathbf{z}$, and they can be enumerated in polynomial time. The determinant of each simplex basis can be computed in polynomial time. Then Δ can be chosen as the least common multiple of all these determinants. □

Lemma 4.12. *Let $Q \subset \mathbf{R}^d$ be an integral polytope. Let $M \in \mathbf{R}$ be such that $Q \subseteq \{\, \mathbf{x} \in \mathbf{R}^d : |x_i| \leq M \text{ for } i = 1, \ldots, d \,\}$. Let $\mathbf{x}^* \in Q$ and let $\delta > 0$. Then every lattice $\frac{1}{k}\mathbf{Z}^d$ for $k \geq \frac{2}{\delta} dM$ contains a lattice point $\mathbf{x} \in Q \cap \frac{1}{k}\mathbf{Z}^d$ with $\|\mathbf{x} - \mathbf{x}^*\|_\infty \leq \delta$.*

Proof. By Carathéodory's Theorem, there exist $d+1$ vertices $\mathbf{x}^0, \ldots, \mathbf{x}^d \in \mathbf{Z}^d$ of Q and convex multipliers $\lambda_0, \ldots, \lambda_d$ such that $\mathbf{x}^* = \sum_{i=0}^d \lambda_i \mathbf{x}^i$. Let $\lambda'_i := \frac{1}{k}\lfloor k\lambda_i \rfloor \geq 0$ for $i = 1, \ldots, d$ and $\lambda'_0 := 1 - \sum_{i=1}^d \lambda'_i \geq 0$. Moreover, we conclude $\lambda_i - \lambda'_i \leq \frac{1}{k}$ for $i = 1, \ldots, d$ and $\lambda'_0 - \lambda_0 = \sum_{i=1}^d (\lambda_i - \lambda'_i) \leq d\frac{1}{k}$. Then $\mathbf{x} := \sum_{i=0}^d \lambda'_i \mathbf{x}^i \in Q \cap \frac{1}{k}\mathbf{Z}^d$, and we have

$$\|\mathbf{x} - \mathbf{x}^*\|_\infty \leq \sum_{i=0}^{d} |\lambda'_i - \lambda_i| \|\mathbf{x}^i\|_\infty \leq 2d\frac{1}{k}M \leq \delta,$$

which proves the lemma. □

4.3.2 Bounding techniques

Using the results of subsection 4.3.1 we are now able to approximate the mixed-integer optimal point by a point of a suitably fine lattice. The question arises how

we can use the geometric distance of these two points to estimate the difference in objective function values. We prove Lemma 4.13 that provides us with a local Lipschitz constant for the polynomial to be maximized.

Lemma 4.13 (Local Lipschitz constant)**.** *Let f be a polynomial in d variables with maximum total degree D. Let C denote the largest absolute value of a coefficient of f. Then there exists a Lipschitz constant L such that $|f(\mathbf{x}) - f(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|_\infty$ for all $|x_i|, |y_i| \leq M$. The constant L is $O(D^{d+1}CM^D)$.*

Proof. Let $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{D}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}$, where $\mathcal{D} \subseteq \mathbf{Z}^d_{\geq 0}$ is the set of exponent vectors of monomials appearing in f. Let $r = |\mathcal{D}|$ be the number of monomials of f. Then we have

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq \sum_{\boldsymbol{\alpha}\neq\mathbf{0}} |c_{\boldsymbol{\alpha}}|\,|\mathbf{x}^{\boldsymbol{\alpha}} - \mathbf{y}^{\boldsymbol{\alpha}}|.$$

We estimate all summands separately. Let $\boldsymbol{\alpha} \neq \mathbf{0}$ be an exponent vector with $n := \sum_{i=1}^d \alpha_i \leq D$. Let

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}^0 \geq \boldsymbol{\alpha}^1 \geq \cdots \geq \boldsymbol{\alpha}^n = \mathbf{0}$$

be a decreasing chain of exponent vectors with $\boldsymbol{\alpha}^{i-1} - \boldsymbol{\alpha}^i = \mathbf{e}^{j_i}$ for $i = 1, \ldots, n$. Let $\boldsymbol{\beta}^i := \boldsymbol{\alpha} - \boldsymbol{\alpha}^i$ for $i = 0, \ldots, n$. Then $\mathbf{x}^{\boldsymbol{\alpha}} - \mathbf{y}^{\boldsymbol{\alpha}}$ can be expressed as the "telescope sum"

$$\begin{aligned}
\mathbf{x}^{\boldsymbol{\alpha}} - \mathbf{y}^{\boldsymbol{\alpha}} &= \mathbf{x}^{\boldsymbol{\alpha}^0}\mathbf{y}^{\boldsymbol{\beta}^0} - \mathbf{x}^{\boldsymbol{\alpha}^1}\mathbf{y}^{\boldsymbol{\beta}^1} + \mathbf{x}^{\boldsymbol{\alpha}^1}\mathbf{y}^{\boldsymbol{\beta}^1} - \mathbf{x}^{\boldsymbol{\alpha}^2}\mathbf{y}^{\boldsymbol{\beta}^2} + - \cdots - \mathbf{x}^{\boldsymbol{\alpha}^n}\mathbf{y}^{\boldsymbol{\beta}^n} \\
&= \sum_{i=1}^n \left(\mathbf{x}^{\boldsymbol{\alpha}^{i-1}}\mathbf{y}^{\boldsymbol{\beta}^{i-1}} - \mathbf{x}^{\boldsymbol{\alpha}^i}\mathbf{y}^{\boldsymbol{\beta}^i}\right) \\
&= \sum_{i=1}^n \left((x_{j_i} - y_{j_i})\mathbf{x}^{\boldsymbol{\alpha}^i}\mathbf{y}^{\boldsymbol{\beta}^{i-1}}\right).
\end{aligned}$$

Since $\left|\mathbf{x}^{\boldsymbol{\alpha}^i}\mathbf{y}^{\boldsymbol{\beta}^{i-1}}\right| \leq M^{n-1}$ and $n \leq D$, we obtain

$$|\mathbf{x}^{\boldsymbol{\alpha}} - \mathbf{y}^{\boldsymbol{\alpha}}| \leq D \cdot \|\mathbf{x} - \mathbf{y}\|_\infty \cdot M^{n-1},$$

thus

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq CrDM^{D-1}\|\mathbf{x} - \mathbf{y}\|_\infty.$$

Let $L := CrDM^{D-1}$. Now, since $r = O(D^d)$, we have $L = O(D^{d+1}CM^D)$. □

Moreover, in order to obtain an FPTAS, we need to put differences of function values in relation to the maximum function value. To do this, we need to deal with the special case of polynomials that are constant on the feasible region; here trivially every feasible solution is optimal. For non-constant polynomials, we can prove a lower bound on the maximum function value. The technique is to bound the difference of

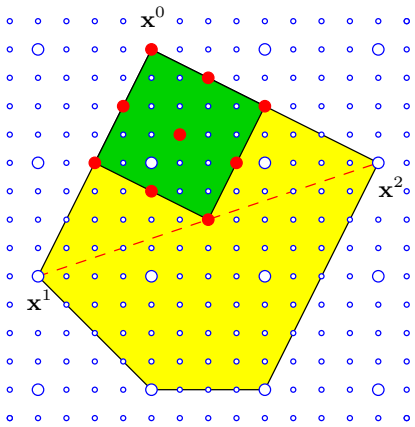


Figure 4.4: The geometry of Lemma 4.15. For a polynomial with maximum total degree of 2, we construct a refinement $\frac{1}{k}\mathbf{Z}^d$ (small circles) of the standard lattice (large circles) such that $P \cap \frac{1}{k}\mathbf{Z}^d$ contains an affine image of the set $\{0,1,2\}^d$ (large dots).

the minimum and the maximum function value on the mixed-integer set from below; if the polynomial is non-constant, this implies, for a non-negative polynomial, a lower bound on the maximum function value. We will need a simple fact about the roots of multivariate polynomials.

Lemma 4.14. *Let $f \in \mathbf{Q}[x_1, \ldots, x_d]$ be a polynomial and let D be the largest power of any variable that appears in f. Then $f = 0$ if and only if f vanishes on the set $\{0, \ldots, D\}^d$.*

Proof. This is a simple consequence of the Fundamental Theorem of Algebra. See, for instance, (Cox et al., 1992, Chapter 1, §1, Exercise 6 b). □

Lemma 4.15. *Let $f \in \mathbf{Q}[x_1, \ldots, x_d]$ be a polynomial with maximum total degree D. Let $Q \subset \mathbf{R}^d$ be an integral polytope of dimension $d' \leq d$. Let $k \geq D\,d'$. Then f is constant on Q if and only if f is constant on $Q \cap \frac{1}{k}\mathbf{Z}^d$.*

Proof. Let $\mathbf{x}^0 \in Q \cap \mathbf{Z}^d$ be an arbitrary vertex of Q. There exist vertices $\mathbf{x}^1, \ldots, \mathbf{x}^{d'} \in Q \cap \mathbf{Z}^d$ such that the vectors $\mathbf{x}^1 - \mathbf{x}^0, \ldots, \mathbf{x}^{d'} - \mathbf{x}^0 \in \mathbf{Z}^d$ are linearly independent. By convexity, Q contains the parallelepiped

$$S := \left\{ \mathbf{x}^0 + \textstyle\sum_{i=1}^{d'} \lambda_i (\mathbf{x}^i - \mathbf{x}^0) : \lambda_i \in [0, \frac{1}{d'}] \text{ for } i = 1, \ldots, d' \right\}.$$

We consider the set

$$S_k = \tfrac{1}{k}\mathbf{Z}^d \cap S \supseteq \left\{ \mathbf{x}^0 + \textstyle\sum_{i=1}^{d'} \frac{n_i}{k} (\mathbf{x}^i - \mathbf{x}^0) : n_i \in \{0, 1, \ldots, D\} \text{ for } i = 1, \ldots, d' \right\};$$

see Figure 4.4. Now if there exists a $c \in \mathbf{R}$ with $f(\mathbf{x}) = c$ for all $\mathbf{x} \in Q \cap \frac{1}{k}\mathbf{Z}^d$, then all the points in S_k are roots of the polynomial $f - c$, which has only maximum total degree D. By Lemma 4.14 (after an affine transformation), $f - c$ is zero on the affine hull of S_k; hence f is constant on the polytope Q. □

Theorem 4.16. *Let $f \in \mathbf{Z}[x_1, \dots, x_{d_1}, z_1, \dots, z_{d_2}]$. Let P be a rational convex polytope, and let Δ be the number from Lemma 4.11. Let $m = k\Delta$ with $k \geq D\, d_1$, $k \in \mathbf{Z}$. Then f is constant on the feasible region $P \cap \left(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\right)$ if and only if f is constant on $P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$. If f is not constant, then*

$$\left|f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\right| \geq m^{-D}, \tag{4.10}$$

where $(\mathbf{x}_{\max}, \mathbf{z}_{\max})$ is an optimal solution to the maximization problem over the feasible region $P \cap \left(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\right)$ and $(\mathbf{x}_{\min}, \mathbf{z}_{\min})$ is an optimal solution to the minimization problem.

Proof. Let f be constant on $P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$. For fixed integer part $\mathbf{z} \in \mathbf{Z}^{d_2}$, we consider the polytope $\Delta P_{\mathbf{z}} = \left\{ \Delta\mathbf{x} : (\mathbf{x}, \mathbf{z}) \in P \right\}$, which is a slice of P scaled to become an integral polytope. By applying Lemma 4.15 with $k = (D+1)d$ on every polytope $\Delta P_{\mathbf{z}}$, we obtain that f is constant on every slice $P_{\mathbf{z}}$. Because f is also constant on the set $P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$, which contains a point of every non-empty slice $P_{\mathbf{z}}$, it follows that f is constant on P.

If f is not constant, there exist $(\mathbf{x}^1, \mathbf{z}^1), (\mathbf{x}^2, \mathbf{z}^2) \in P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$ with $f(\mathbf{x}^1, \mathbf{z}^1) \neq f(\mathbf{x}^2, \mathbf{z}^2)$. By the integrality of all coefficients of f, we obtain the estimate

$$|f(\mathbf{x}^1, \mathbf{z}^1) - f(\mathbf{x}^2, \mathbf{z}^2)| \geq m^{-D}.$$

Because $(\mathbf{x}^1, \mathbf{z}^1)$, $(\mathbf{x}^2, \mathbf{z}^2)$ are both feasible solutions to the maximization problem and the minimization problem, this implies (4.10). □

4.3.3 Proof of Theorem 4.1

Now we are in the position to prove the main result.

Proof of Theorem 4.1. Part (a). Let $(\mathbf{x}^*, \mathbf{z}^*)$ denote an optimal solution to the mixed-integer problem.Let $\epsilon > 0$. We show that, in time polynomial in the input length, the maximum total degree, and $\frac{1}{\epsilon}$, we can compute a point $(\mathbf{x}, \mathbf{z})$ that satisfies the constraints such that

$$|f(\mathbf{x}, \mathbf{z}) - f(\mathbf{x}^*, \mathbf{z}^*)| \leq \epsilon f(\mathbf{x}^*, \mathbf{z}^*). \tag{4.11}$$

We prove this by establishing several estimates, which are illustrated in Figure 4.5.

First we note that we can restrict ourselves to the case of polynomials with integer coefficients, simply by multiplying f with the least common multiple of all denominators of the coefficients. We next establish a lower bound on $f(\mathbf{x}^*, \mathbf{z}^*)$. To this end,

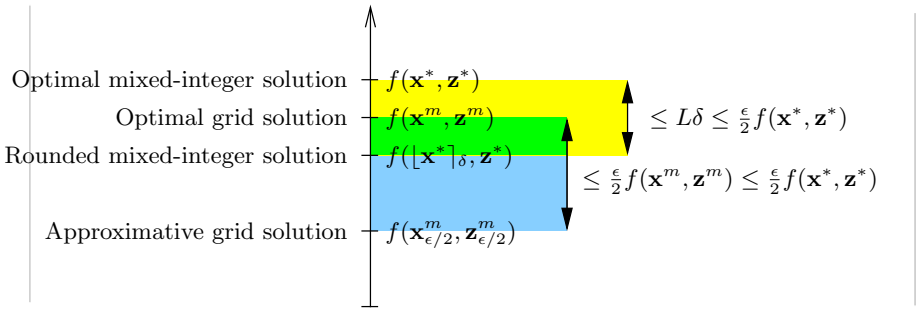


Figure 4.5: Estimates in the proof of Theorem 4.1 (a)

let Δ be the integer from Lemma 4.11, which can be computed in polynomial time. By Theorem 4.16 with $m = D\, d_1 \Delta$, either f is constant on the feasible region, or

$$f(\mathbf{x}^*, \mathbf{z}^*) \geq (D\, d_1 \Delta)^{-D}, \tag{4.12}$$

where D is the maximum total degree of f. Now let

$$\delta := \frac{\epsilon}{2(D d_1 \Delta)^D L(C, D, M)} \tag{4.13}$$

and let us choose the grid size

$$m := \Delta \left\lceil \frac{4}{\epsilon} (D d_1 \Delta)^D L(C, D, M) d_1 M \right\rceil, \tag{4.14}$$

where $L(C, D, M)$ is the Lipschitz constant from Lemma 4.13. Then we have $m \geq \Delta \frac{2}{\delta} d_1 M$, so by Theorem 4.10, there is a point $(\lfloor \mathbf{x}^* \rceil_\delta, \mathbf{z}^*) \in P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$ with $\left\| \lfloor \mathbf{x}^* \rceil_\delta - \mathbf{x}^* \right\|_\infty \leq \delta$. Let $(\mathbf{x}^m, \mathbf{z}^m)$ denote an optimal solution to the grid problem (4.6). Because $(\lfloor \mathbf{x}^* \rceil_\delta, \mathbf{z}^*)$ is a feasible solution to the grid problem (4.6), we have

$$f(\lfloor \mathbf{x}^* \rceil_\delta, \mathbf{z}^*) \leq f(\mathbf{x}^m, \mathbf{z}^m) \leq f(\mathbf{x}^*, \mathbf{z}^*). \tag{4.15}$$

Now we can estimate

$$\begin{aligned} \left| f(\mathbf{x}^*, \mathbf{z}^*) - f(\mathbf{x}^m, \mathbf{z}^m) \right| &\leq \left| f(\mathbf{x}^*, \mathbf{z}^*) - f(\lfloor \mathbf{x}^* \rceil_\delta, \mathbf{z}^*) \right| \\ &\leq L(C, D, M) \left\| \mathbf{x}^* - \lfloor \mathbf{x}^* \rceil_\delta \right\|_\infty \\ &\leq L(C, D, M)\, \delta \\ &= \frac{\epsilon}{2} (D\, d_1 \Delta)^{-D} \\ &\leq \frac{\epsilon}{2} f(\mathbf{x}^*, \mathbf{z}^*), \end{aligned} \tag{4.16}$$

where the last estimate is given by (4.12) in the case that f is not constant on the feasible region. On the other hand, if f is constant, the estimate (4.16) holds trivially.

By Corollary 4.8 we can compute a point $(\mathbf{x}^m_{\epsilon/2}, \mathbf{z}^m_{\epsilon/2}) \in P \cap \left(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\right)$ such that

$$(1 - \tfrac{\epsilon}{2}) f(\mathbf{x}^m, \mathbf{z}^m) \leq f(\mathbf{x}^m_{\epsilon/2}, \mathbf{z}^m_{\epsilon/2}) \leq f(\mathbf{x}^m, \mathbf{z}^m) \tag{4.17}$$

in time polynomial in $\log m$, the encoding length of f and P, the maximum total degree D, and $1/\epsilon$. Here $\log m$ is bounded by a polynomial in $\log M$, D and $\log C$, so we can compute $(\mathbf{x}^m_{\epsilon/2}, \mathbf{z}^m_{\epsilon/2})$ in time polynomial in the input size, the maximum total degree D, and $1/\epsilon$. Now, using (4.17) and (4.16), we can estimate

$$\begin{aligned}
&f(\mathbf{x}^*, \mathbf{z}^*) - f(\mathbf{x}^m_{\epsilon/2}, \mathbf{z}^m_{\epsilon/2}) \\
&\leq f(\mathbf{x}^*, \mathbf{z}^*) - (1 - \tfrac{\epsilon}{2}) f(\mathbf{x}^m, \mathbf{z}^m) \\
&= \tfrac{\epsilon}{2} f(\mathbf{x}^*, \mathbf{z}^*) + (1 - \tfrac{\epsilon}{2}) \big(f(\mathbf{x}^*, \mathbf{z}^*) - f(\mathbf{x}^m, \mathbf{z}^m)\big) \\
&\leq \tfrac{\epsilon}{2} f(\mathbf{x}^*, \mathbf{z}^*) + \tfrac{\epsilon}{2} f(\mathbf{x}^*, \mathbf{z}^*) \\
&= \epsilon f(\mathbf{x}^*, \mathbf{z}^*).
\end{aligned}$$

Hence $f(\mathbf{x}^m_{\epsilon/2}, \mathbf{z}^m_{\epsilon/2}) \geq (1 - \epsilon) f(\mathbf{x}^*, \mathbf{z}^*)$.

□

4.4 Extension to polynomials of arbitrary range

In this section we drop the requirement of the polynomial being positive over the feasible region. We will show an approximation result like the one in de Klerk et al. (2006), i.e., we compute a solution $(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)$ such that

$$\big|f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) - f(\mathbf{x}_{\max}, \mathbf{z}_{\max})\big| \leq \epsilon \big|f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\big|, \tag{4.18}$$

where $(\mathbf{x}_{\max}, \mathbf{z}_{\max})$ is an optimal solution to the maximization problem over the feasible region and $(\mathbf{x}_{\min}, \mathbf{z}_{\min})$ is an optimal solution to the minimization problem. Our algorithm has a running time that is polynomial in the input size, the maximum total degree of f, and $\frac{1}{\epsilon}$. This means that while the result of de Klerk et al. (2006) was a weak version of a PTAS (for fixed degree), our result is a weak version of an FPTAS (for fixed dimension).

The approximation algorithms for the integer case (Lemma 4.7) and the mixed-integer case (Theorem 4.1) only work for polynomial objective functions that are non-negative on the feasible region. In order to apply them to an arbitrary polynomial objective function f, we need to add a constant term to f that is large enough. As proposed in De Loera et al. (2006b), we can use linear programming techniques to obtain a bound M on the variables and then estimate

$$f(\mathbf{x}) \geq -rCM^D =: L_0,$$

where C is the largest absolute value of a coefficient, r is the number of monomials of f, and D is the maximum total degree. However, the range $\big|f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) -$

$f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\big|$ can be exponentially small compared to L_0, so in order to obtain an approximation $(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)$ satisfying (4.18), we would need an $(1-\epsilon')$-approximation to the problem of maximizing $g(\mathbf{x}, \mathbf{z}) := f(\mathbf{x}, \mathbf{z}) - L_0$ with an exponentially small value of ϵ'.

To address this difficulty, we will first apply an algorithm which will compute an approximation $[L_i, U_i]$ of the range $[f(\mathbf{x}_{\min}, \mathbf{z}_{\min}), f(\mathbf{x}_{\max}, \mathbf{z}_{\max})]$ with constant quality. To this end, we first prove a simple corollary of Theorem 4.1.

Corollary 4.17 (Computation of upper bounds for mixed-integer problems)**.** *Let the dimension $d = d_1 + d_2$ be fixed. Let $P \subseteq \mathbf{R}^d$ be a rational convex polytope. Let $f \in \mathbf{Z}[x_1, \dots, x_{d_1}, z_1, \dots, z_{d_2}]$ be a polynomial function with integer coefficientsand maximum total degree D that is non-negative on $P \cap \left(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\right)$. Let $\delta > 0$. There exists an algorithm with running time polynomial in the input size, D, and $\frac{1}{\delta}$ for computing an upper bound u such that*

$$f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) \leq u \leq (1+\delta) f(\mathbf{x}_{\max}, \mathbf{z}_{\max}), \tag{4.19}$$

where $(\mathbf{x}_{\max}, \mathbf{z}_{\max})$ is an optimal solution to the maximization problem of f over $P \cap \left(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\right)$.

Proof. Let $\epsilon = \frac{\delta}{1+\delta}$. By Theorem 4.1, we can, in time polynomial in the input size, D, and $\frac{1}{\epsilon} = 1 + \frac{1}{\delta}$, compute a solution $(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)$ with

$$\left|f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)\right| \leq \epsilon f(\mathbf{x}_{\max}, \mathbf{z}_{\max}). \tag{4.20}$$

Let $u := \frac{1}{1-\epsilon} f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) = (1+\delta) f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)$. Then

$$f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) \leq \frac{1}{1-\epsilon} f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) = u \tag{4.21}$$

and

$$\begin{aligned}(1+\delta) f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) &\geq (1+\delta) f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) \\ &= (1+\delta)(1-\epsilon) u \\ &= (1+\delta)\left(1 - \frac{\delta}{1+\delta}\right) u = u.\end{aligned} \tag{4.22}$$

This proves the estimate (4.19). □

Algorithm 4.18 (Range approximation)**.**

Input: Mixed-integer polynomial optimization problem (**??**), a number $0 < \delta < 1$.
Output: Sequences $\{L_i\}$, $\{U_i\}$ of lower and upper bounds of f over the feasible region $P \cap \left(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\right)$ such that

$$L_i \leq f(\mathbf{x}_{\min}, \mathbf{z}_{\min}) \leq f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) \leq U_i \tag{4.23}$$

and

$$\lim_{i\to\infty} |U_i - L_i| = c\big(f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\big), \tag{4.24}$$

where c depends only on the choice of δ.

1. By solving $2d$ linear programs over P, we find lower and upper integer bounds for each of the variables $x_1, \dots, x_{d_1}, z_1, \dots, z_{d_2}$. Let M be the maximum of the absolute values of these $2d$ numbers. Thus $|x_i|, |z_i| \leq M$ for all i. Let C be the maximum of the absolute values of all coefficients, and r be the number of monomials of $f(x)$. Then

   $$L_0 := -rCM^D \leq f(\mathbf{x}, \mathbf{z}) \leq rCM^D =: U_0,$$

   as we can bound the absolute value of each monomial of $f(x)$ by CM^D.

2. Let $i := 0$.

3. Using the algorithm of Corollary 4.17, compute an upper bound u for the problem

   $$\begin{aligned} \max \quad & g(\mathbf{x}, \mathbf{z}) := f(\mathbf{x}, \mathbf{z}) - L_i \\ \text{s.t.} \quad & (\mathbf{x}, \mathbf{z}) \in P \cap \big(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\big) \end{aligned}$$

   that gives a $(1+\delta)$-approximation to the optimal value. Let $U_{i+1} := L_i + u$.

4. Likewise, compute an upper bound u for the problem

   $$\begin{aligned} \max \quad & h(\mathbf{x}, \mathbf{z}) := U_i - f(\mathbf{x}, \mathbf{z}) \\ \text{s.t.} \quad & (\mathbf{x}, \mathbf{z}) \in P \cap \big(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\big) \end{aligned}$$

   that gives a $(1+\delta)$-approximation to the optimal value. Let $L_{i+1} := U_i - u$.

5. $i := i + 1$.

6. Go to 3.

Lemma 4.19. *Algorithm 4.18 is correct. For fixed $0 < \delta < 1$, it computes the bounds L_n, U_n satisfying (4.23) and (4.24) in time polynomial in the input size and n.*

Proof. We have

$$U_i - L_{i+1} \leq (1+\delta)\big(U_i - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\big) \tag{4.25}$$

and

$$U_{i+1} - L_i \leq (1+\delta)\big(f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - L_i\big). \tag{4.26}$$

This implies

$$U_{i+1} - L_{i+1} \leq \delta(U_i - L_i) + (1+\delta)\big(f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\big).$$

Therefore

$$\begin{aligned} U_n - L_n &\leq \delta^n(U_0 - L_0) + (1+\delta)\left(\sum_{i=0}^{n-2} \delta^i\right)\big(f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\big) \\ &= \delta^n(U_0 - L_0) + (1+\delta)\frac{1-\delta^{n-1}}{1-\delta}\big(f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\big) \\ &\to \frac{1+\delta}{1-\delta}\big(f(\mathbf{x}_{\max}, \mathbf{z}_{\max}) - f(\mathbf{x}_{\min}, \mathbf{z}_{\min})\big) \quad (n \to \infty). \end{aligned}$$

The bound on the running time requires a careful analysis. Because in each step the result u (a rational number) of the bounding procedure (Corollary 4.17) becomes part of the input in the next iteration, the encoding length of the input could grow exponentially after only polynomially many steps. However, we will show that the encoding length only grows very slowly.

First we need to remark that the auxiliary objective functions g and h have integer coefficients except for the constant term, which may be rational. It turns out that the estimates in the proof of Theorem 4.1 (in particular, the local Lipschitz constant L and the lower bound on the optimal value) are independent from the constant term of the objective function. Therefore, the *same* approximating grid $\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}$ can be chosen in all iterations of Algorithm 4.18; the number m only depends on δ, the polytope P, the maximum total degree D, and the coefficients of f with the exception of the constant term.

The construction in the proof of Corollary 4.17 obtains the upper bound u by multiplying the approximation $f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)$ by $(1+\delta)$. Therefore we have

$$\begin{aligned} U_{i+1} &= L_i + u \\ &= L_i + (1+\delta)\big(f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) - L_i\big) \\ &= -\delta L_i + (1+\delta) f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon). \end{aligned} \tag{4.27}$$

Because the solution $(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)$ lies in the grid $\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}$, the value $f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)$ is an integer multiple of m^{-D}. This implies that, because $L_0 \leq f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon) \leq U_0$, the encoding length of the rational number $f(\mathbf{x}_\epsilon, \mathbf{z}_\epsilon)$ is bounded by a polynomial in the input size of f and P. Therefore the encoding length U_{i+1} (and likewise L_{i+1}) only increases by an additive term that is bounded by a polynomial in the input size of f and P. □

We are now in the position to prove Theorem 4.2.

Proof of Theorem 4.2. Clearly we can restrict ourselves to polynomials with integer coefficients. Let $m = (D+1)d_1\Delta$, where Δ is the number from Theorem 4.10. We

apply Algorithm 4.18 using $0 < \delta < 1$ arbitrary to compute bounds U_n and L_n for

$$n = \lceil -\log_\delta\big(2m^D(U_0 - L_0)\big)\rceil.$$

Because n is bounded by a polynomial in the input size and the maximum total degree D, this can be done in polynomial time. Now, by the proof of Lemma 4.19, we have

$$\begin{aligned} U_n - L_n &\le \delta^n(U_0 - L_0) + (1+\delta)\frac{1-\delta^{n-1}}{1-\delta}\big(f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - f(\mathbf{x}_{\min},\mathbf{z}_{\min})\big) \\ &\le \frac{1}{2}m^{-D} + \frac{1+\delta}{1-\delta}\big(f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - f(\mathbf{x}_{\min},\mathbf{z}_{\min})\big). \end{aligned} \tag{4.28}$$

If f is constant on $P \cap \big(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\big)$, it is constant on $P \cap \big(\frac{1}{m}\mathbf{Z}^{d_1} \times \mathbf{Z}^{d_2}\big)$, then $U_n - L_n \le \frac{1}{2}m^{-D}$. Otherwise, by Theorem 4.16, we have $U_n - L_n \ge f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - f(\mathbf{x}_{\min},\mathbf{z}_{\min}) \ge m^{-D}$. This settles part (a).

For part (b), if f is constant on $P \cap \big(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\big)$, we return an arbitrary solution as an optimal solution. Otherwise, we can estimate further:

$$U_n - L_n \le \left(\frac{1}{2} + \frac{1+\delta}{1-\delta}\right)\big(f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - f(\mathbf{x}_{\min},\mathbf{z}_{\min})\big). \tag{4.29}$$

Now we apply the algorithm of Theorem 4.1 to the maximization problem of the polynomial function $f' := f - L_n$, which is non-negative over the feasible region $P \cap \big(\mathbf{R}^{d_1} \times \mathbf{Z}^{d_2}\big)$. We compute a point $(\mathbf{x}_{\epsilon'}, \mathbf{z}_{\epsilon'})$ where $\epsilon' = \epsilon\big(\frac{1}{2} + \frac{1+\delta}{1-\delta}\big)^{-1}$ such that

$$\big|f'(\mathbf{x}_{\epsilon'},\mathbf{z}_{\epsilon'}) - f'(\mathbf{x}_{\max},\mathbf{z}_{\max})\big| \le \epsilon' f'(\mathbf{x}_{\max},\mathbf{z}_{\max}).$$

Then we obtain the estimate

$$\begin{aligned} \big|f(\mathbf{x}_{\epsilon'},\mathbf{z}_{\epsilon'}) - f(\mathbf{x}_{\max},\mathbf{z}_{\max})\big| &\le \epsilon'\big(f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - L_n\big) \\ &\le \epsilon'\big(U_n - L_n\big) \\ &\le \epsilon'\left(\frac{1}{2} + \frac{1+\delta}{1-\delta}\right)\big(f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - f(\mathbf{x}_{\min},\mathbf{z}_{\min})\big) \\ &= \epsilon\big(f(\mathbf{x}_{\max},\mathbf{z}_{\max}) - f(\mathbf{x}_{\min},\mathbf{z}_{\max})\big), \end{aligned}$$

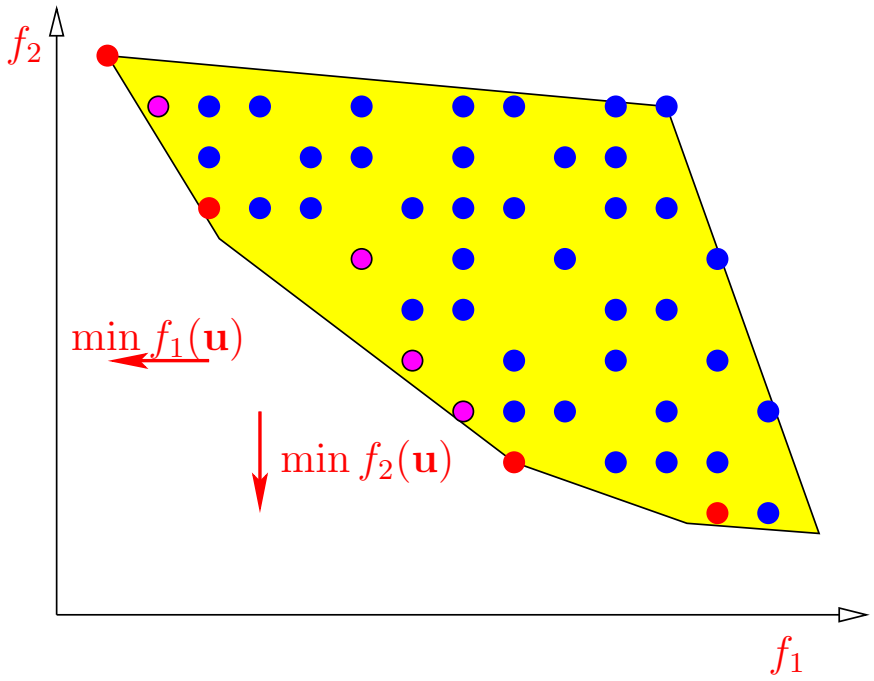
which proves part (b). □

Notes and sources

The pure integer case (section 4.2) appeared in De Loera et al. (2006b). The mixed-integer case (section 4.3) was first published in De Loera et al. (2006c). The extension to polynomials of arbitrary range appeared in De Loera et al. (2008a).

Chapter 5

Multicriteria mixed-integer optimization



We settle the computational complexity of fundamental questions related to multicriteria integer linear programs, when the dimensions of the strategy space and of the outcome space are considered fixed constants. In particular we construct:

1. polynomial-time algorithms to exactly determine the number of Pareto optima and Pareto strategies;

2. a polynomial-space polynomial-delay prescribed-order enumeration algorithm for arbitrary projections of the Pareto set;

3. an algorithm to minimize the distance of a Pareto optimum from a prescribed comparison point with respect to arbitrary polyhedral norms;

4. a fully polynomial-time approximation scheme for the problem of minimizing the distance of a Pareto optimum from a prescribed comparison point with respect to the Euclidean norm.

5.1 Introduction

Let $A = (a_{ij})$ be an integral $m \times n$-matrix and $\mathbf{b} \in \mathbf{Z}^m$ such that the convex polyhedron $P = \{\, \mathbf{u} \in \mathbf{R}^n : A\mathbf{u} \leq \mathbf{b} \,\}$ is bounded. Given k linear functionals $f_1, f_2, \ldots, f_k \in \mathbf{Z}^n$, we consider the *multicriterion integer linear programming problem*

$$\begin{aligned} \text{vmin} \quad & \big(f_1(\mathbf{u}), f_2(\mathbf{u}), \ldots, f_k(\mathbf{u})\big) \\ \text{subject to} \quad & A\mathbf{u} \leq \mathbf{b} \\ & \mathbf{u} \in \mathbf{Z}^n \end{aligned} \tag{5.1}$$

where vmin is defined as the problem of finding all Pareto optima and a corresponding Pareto strategy; see Figure 5.1.

For a lattice point $\mathbf{u}$ the vector $\mathbf{f}(\mathbf{u}) = \big(f_1(\mathbf{u}), \ldots, f_k(\mathbf{u})\big)$ is called an *outcome vector*. Such an outcome vector is a *Pareto optimum* for the above problem if and only if there is no other point $\tilde{\mathbf{u}}$ in the feasible set such that $f_i(\tilde{\mathbf{u}}) \leq f_i(\mathbf{u})$ for all i and $f_j(\tilde{\mathbf{u}}) < f_j(\mathbf{u})$ for at least one index j; see Figure 5.2. The corresponding feasible point $\mathbf{u}$ is called a *Pareto strategy*. Thus a feasible vector is a Pareto strategy if no feasible vector can decrease some criterion without causing a simultaneous increase in at least one other criterion; see Figure 5.3. For general information about the multicriteria problems see, e.g., Figueira et al. (2005), Sawaragi et al. (1985).

In general multiobjective problems the number of Pareto optimal solutions may be infinite, but in our situation the number of Pareto optima and strategies is finite. There are several well-known techniques to generate Pareto optima. Some popular methods used to solve such problems include, e.g., weighting the objectives or using a so-called global criterion approach (see Ehrgott and Gandibleux (2000)). In abnormally nice situations, such as multicriteria *linear* programs (Isermann, 1974), one

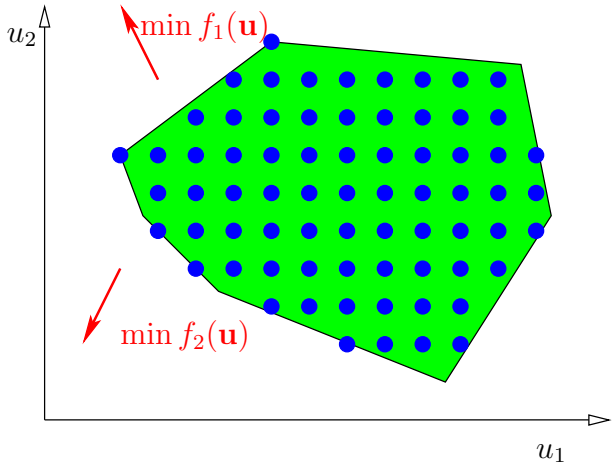


Figure 5.1: Strategy space

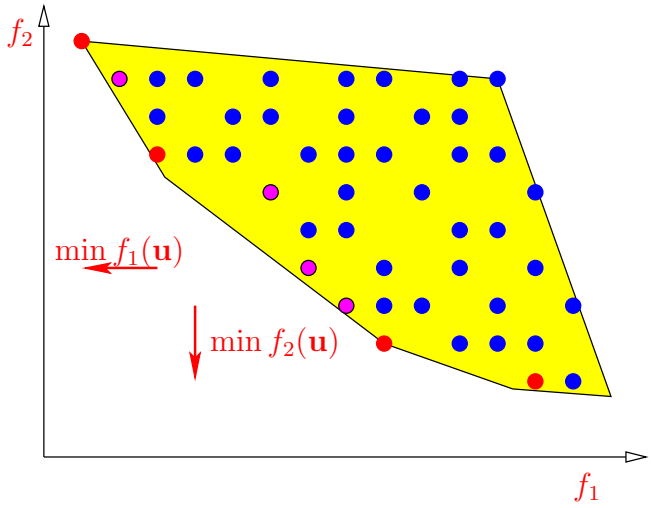


Figure 5.2: Outcome space

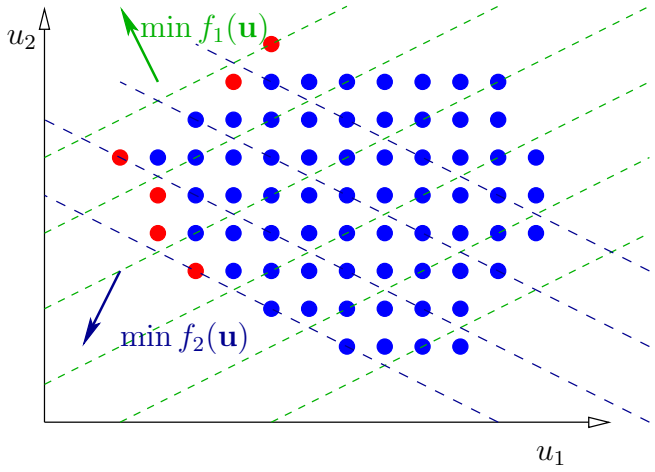


Figure 5.3: Pareto strategies

knows a way to generate all Pareto optima, but most techniques reach only some of the Pareto optima.

The purpose of this article is to study the sets of *all* Pareto optima and strategies of a multicriterion integer linear program using the algebraic structures of generating functions. The set of Pareto points can be described as the formal sum of monomials

$$\sum \{\, \mathbf{z}^{\mathbf{v}} : \ \mathbf{u} \in P \cap \mathbf{Z}^n \text{ and } \mathbf{v} = \mathbf{f}(\mathbf{u}) \in \mathbf{Z}^k \text{ is a Pareto optimum} \,\}. \tag{5.2}$$

Our main theoretical result states that, under the assumption that the number of variables is fixed, we can compute in polynomial time a compact expression for the huge polynomial above, thus *all* its Pareto optima can in fact be counted exactly. The same can be done for the corresponding Pareto strategies when written in the form

$$\sum \{\, \mathbf{x}^{\mathbf{u}} : \ \mathbf{u} \in P \cap \mathbf{Z}^n \text{ and } \mathbf{f}(\mathbf{u}) \text{ is a Pareto optimum} \,\}. \tag{5.3}$$

Theorem 5.1. *Let $A \in \mathbf{Z}^{m\times n}$, an m-vector $\mathbf{b}$, and linear functions $f_1, \ldots, f_k \in \mathbf{Z}^n$ be given. There are algorithms to perform the following tasks:*

(i) *Compute the generating function (5.2) of all the Pareto optima as a sum of rational functions. In particular we can count how many Pareto optima are there. If we assume k and n are fixed, the algorithm runs in time polynomial in the size of the input data.*

(ii) *Compute the generating function (5.3) of all the Pareto strategies as a sum of rational functions. In particular we can count how many Pareto strategies are there in P. If we assume k and n are fixed, the algorithm runs in time polynomial in the size of the input data.*

(iii) *Generate the full sequence of Pareto optima ordered lexicographically or by any other term ordering. If we assume k and n are fixed, the algorithm runs in polynomial time on the input size and the number of Pareto optima. (More strongly, there exists a* polynomial-space polynomial-delay prescribed-order enumeration algorithm*.)*

In contrast it is known that for non-fixed dimension it is #P-hard to enumerate Pareto optima and NP-hard to find them Emelichev and Perepelitsa (1992), Sergienko and Perepelitsa (1991). The proof of Theorem 5.1 parts (i) and (ii) will be given in section 5.2. Again it is based on the theory of rational generating functions. Part (iii) of Theorem 5.1 will be proved in section 5.3.

For a user that knows some or all of the Pareto optima or strategies, a goal is to select the "best" member of the family. One is interested in selecting one Pareto optimum that realizes the "best" compromise between the individual objective functions. The quality of the compromise is often measured by the distance of a Pareto optimum $\mathbf{v}$ from a user-defined comparison point $\hat{\mathbf{v}}$. For example, often users take as a good comparison point the so-called *ideal point* $\mathbf{v}^{\text{ideal}} \in \mathbf{Z}^k$ of the multicriterion problem, which is defined as

$$v_i^{\text{ideal}} = \min\{\, f_i(\mathbf{u}) : \mathbf{u} \in P \cap \mathbf{Z}^n \,\}.$$

The criteria of comparison with the point $\hat{\mathbf{v}}$ are quite diverse, but some popular ones include computing the minimum over the possible sums of absolute differences of the individual objective functions, evaluated at the different Pareto strategies, from the comparison point $\hat{\mathbf{v}}$, i.e.,

$$f(\mathbf{u}) = |f_1(\mathbf{u}) - \hat{v}_1| + \cdots + |f_k(\mathbf{u}) - \hat{v}_k|, \tag{5.4a}$$

or the maximum of the absolute differences,

$$f(\mathbf{u}) = \max\bigl\{|f_1(\mathbf{u}) - \hat{v}_1|, \ldots, |f_k(\mathbf{u}) - \hat{v}_k|\bigr\}, \tag{5.4b}$$

over all Pareto optima $(f_1(\mathbf{u}), \ldots, f_k(\mathbf{u}))$. Another popular criterion, sometimes called the *global criterion*, is to minimize the sum of relative distances of the individual objectives from their known minimal values, i.e.,

$$f(\mathbf{u}) = \frac{f_1(\mathbf{u}) - v_1^{\text{ideal}}}{|v_1^{\text{ideal}}|} + \cdots + \frac{f_k(\mathbf{u}) - v_k^{\text{ideal}}}{|v_k^{\text{ideal}}|}. \tag{5.4c}$$

We stress that if we take any one of these functions as an objective function of an integer program, the optimal solution will be a non-Pareto solution of the multicriterion problem (5.1) in general; see Figure 5.4. In contrast, we show here that by encoding Pareto optima and strategies as a rational function we avoid this

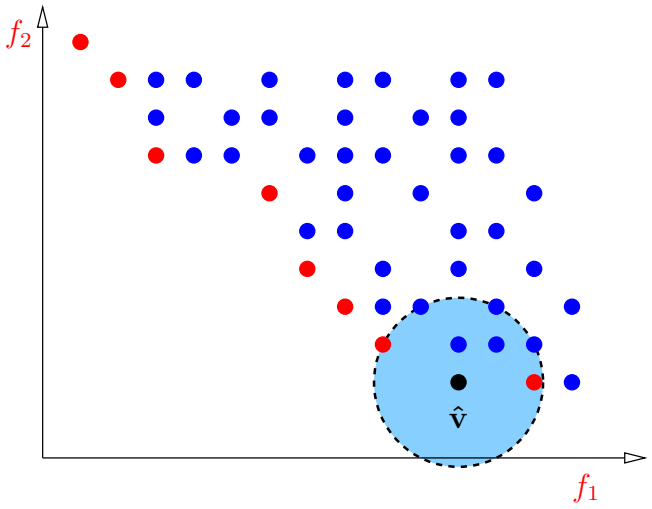


Figure 5.4: Global criteria

problem, since we evaluate the objective functions directly on the space of Pareto optima.

All of the above criteria (5.4) measure the distance from a prescribed point with respect to a *polyhedral norm*. In section 5.4, we prove:

Theorem 5.2. *Let the dimension n and the number k of objective functions be fixed. Let a multicriterion integer linear program (5.1) be given. Let a polyhedral norm $\|\cdot\|_Q$ be given by the vertex or inequality description of its unit ball $Q \subseteq \mathbf{R}^k$. Finally, let a prescribed point $\hat{\mathbf{v}} \in \mathbf{Z}^k$ be given.*

(i) *There exists a polynomial-time algorithm to find a Pareto optimum $\mathbf{v}$ of (5.1) that minimizes the distance $\|\mathbf{v} - \hat{\mathbf{v}}\|_Q$ from the prescribed point.*

(ii) *There exists a polynomial-space polynomial-delay enumeration algorithm for enumerating the Pareto optima of (5.1) in the order of increasing distances from the prescribed point $\hat{\mathbf{v}}$.*

Often users are actually interested in finding a Pareto optimum that minimizes the *Euclidean* distance from a prescribed comparison point $\hat{\mathbf{v}}$,

$$f(\mathbf{u}) = \sqrt{|f_1(\mathbf{u}) - \hat{v}_1|^2 + \cdots + |f_k(\mathbf{u}) - \hat{v}_k|^2}, \tag{5.5}$$

but to our knowledge no method of the literature gives a satisfactory solution to that problem. In section 5.4, however, we prove the following theorem, which gives a very strong approximation result.

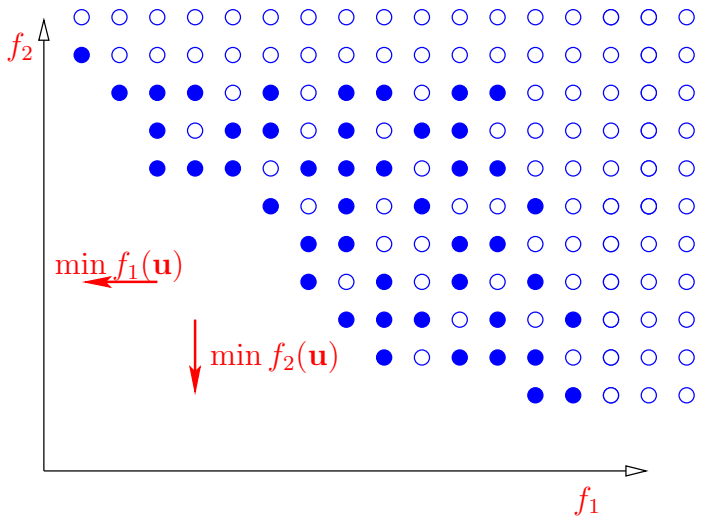


Figure 5.5: Outcome space

Theorem 5.3. *Let the dimension n and the number k of objective functions be fixed. There exists a fully polynomial-time approximation scheme for the problem of minimizing the Euclidean distance of a Pareto optimum of (5.1) from a prescribed comparison point $\hat{\mathbf{v}} \in \mathbf{Z}^k$.*

We actually prove this theorem in a somewhat more general setting, using an arbitrary norm whose unit ball is representable by a homogeneous polynomial inequality.

5.2 The rational function encoding of all Pareto optima

One has to be careful when using the Barvinok–Woods theory (especially the Projection Theorem) that the sets in question are finite. The proof of Theorem 5.1 will require us to project and intersect sets of lattice points represented by rational functions. We cannot, in principle, do those operations for *infinite* sets of lattice points. Fortunately, in our setting it is possible to restrict our attention to finite sets.

Proof of Theorem 5.1, part (i) and (ii). The proof of part (i) has three steps:

Step 1. For $i = 1, \ldots, k$ let $\bar{v}_i \in \mathbf{Z}$ be an upper bound of polynomial encoding size for the value of f_i over P. Such a bound exists because of the boundedness of P, and it can be computed in polynomial time by linear programming. We will denote the vector of upper bounds by $\bar{\mathbf{v}} \in \mathbf{Z}^k$. We consider the *truncated multi-epigraph* of the

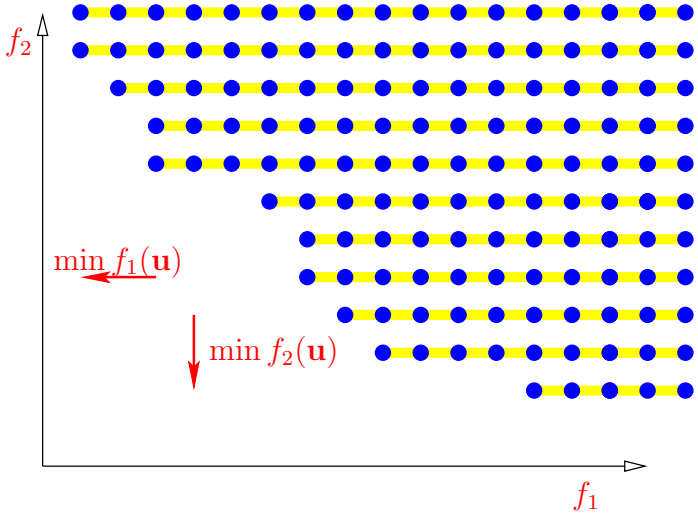


Figure 5.6: Outcome space, epigraph

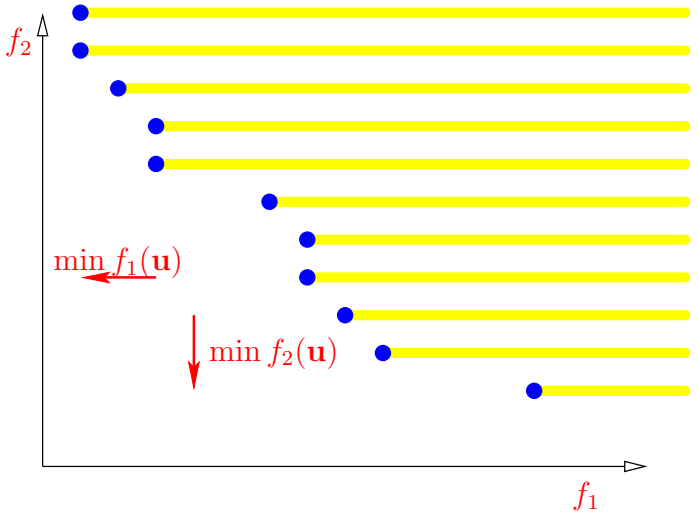


Figure 5.7: Outcome space, after erasing horizontally dominated solutions

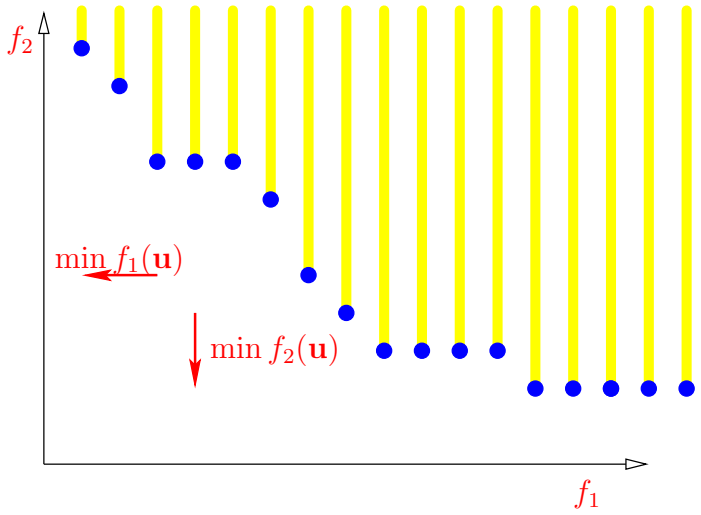


Figure 5.8: Outcome space, after erasing vertically dominated solutions

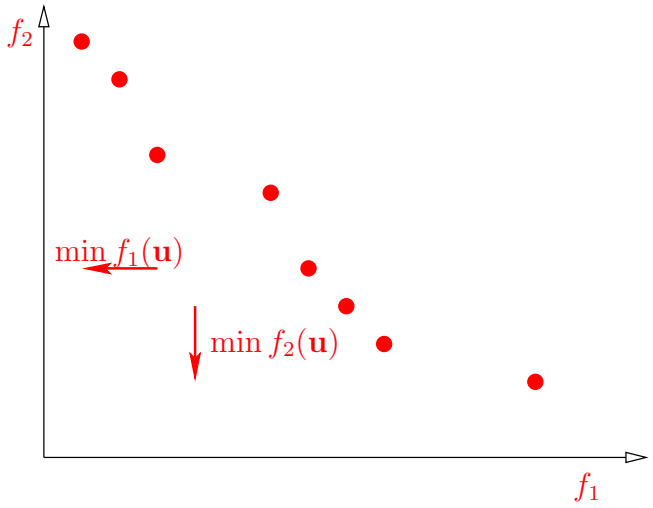


Figure 5.9: Outcome space, after erasing all dominated solutions

objective functions $f_1, \ldots, f_k$ over the linear relaxation of the feasible region P,

$$P^{\geq}_{f_1,\ldots,f_k} = \big\{\, (\mathbf{u},\mathbf{v}) \in \mathbf{R}^n \times \mathbf{R}^k : \mathbf{u} \in P, \\ \bar{v}_i \geq v_i \geq f_i(\mathbf{u}) \text{ for } i = 1,\ldots,k \,\big\}, \tag{5.6}$$

which is a rational convex polytope in $\mathbf{R}^n \times \mathbf{R}^k$. Let $V^{\geq} \subseteq \mathbf{Z}^k$ denote the integer projection of $P^{\geq}_{f_1,\ldots,f_k}$ on the $\mathbf{v}$ variables, i.e., the set

$$V^{\geq} = \big\{\, \mathbf{v} \in \mathbf{Z}^k : \exists \mathbf{u} \in \mathbf{Z}^n \text{ with } (\mathbf{u},\mathbf{v}) \in P^{\geq}_{f_1,\ldots,f_k} \cap (\mathbf{Z}^n \times \mathbf{Z}^k) \,\big\}. \tag{5.7}$$

Clearly, the vectors in $V^{\geq}$ are all integer vectors in the outcome space which are weakly dominated by some outcome vector $\big(f_1(\mathbf{u}), f_2(\mathbf{u}), \ldots, f_k(\mathbf{u})\big)$ for a feasible solution $\mathbf{u}$ in $P \cap \mathbf{Z}^n$; however, we have truncated away all outcome vectors which weakly dominate the computed bound $\bar{\mathbf{v}}$. Let us consider the generating function of $V^{\geq}$, the multivariate polynomial

$$g(V^{\geq};\mathbf{z}) \;= \sum \big\{\, \mathbf{z}^{\mathbf{v}} : \mathbf{v} \in V^{\geq} \,\big\}.$$

In the terminology of polynomial ideals, the monomials in $g(V^{\geq};\mathbf{z})$ form a truncated ideal generated by the Pareto optima. By the Projection Theorem (our Theorem 3.30), we can compute $g(V^{\geq};\mathbf{z})$ in the form of a polynomial-size rational function in polynomial time.

Step 2. Let $V^{\text{Pareto}} \subseteq \mathbf{Z}^k$ denote the set of Pareto optima. Clearly we have

$$V^{\text{Pareto}} = \big(V^{\geq} \setminus (\mathbf{e}_1 + V^{\geq})\big) \cap \cdots \cap \big(V^{\geq} \setminus (\mathbf{e}_k + V^{\geq})\big)\,,$$

where $\mathbf{e}_i \in \mathbf{Z}^k$ denotes the i-th unit vector and

$$\mathbf{e}_i + V^{\geq} = \{\, \mathbf{e}_i + \mathbf{v} : \mathbf{v} \in V^{\geq} \,\}.$$

The generating function $g(V^{\text{Pareto}};\mathbf{z})$ can be computed by the Boolean Operations Lemma (Theorem 3.29) in polynomial time from $g(V^{\geq};\mathbf{z})$ as

$$g(V^{\text{Pareto}};\mathbf{z}) = \big(g(V^{\geq};\mathbf{z}) - g(V^{\geq};\mathbf{z}) * z_1 g(V^{\geq};\mathbf{z})\big) \\ * \cdots * \big(g(V^{\geq};\mathbf{z}) - g(V^{\geq};\mathbf{z}) * z_k g(V^{\geq};\mathbf{z})\big), \tag{5.8}$$

where $*$ denotes taking the Hadamard product of the rational functions.

Step 3. To obtain the number of Pareto optima, we compute the specialization $g(V^{\text{Pareto}};\mathbf{z}=\mathbf{1})$. This is possible in polynomial time using residue techniques as outlined before the beginning of the proof.

Proof of part (ii). Now we recover the Pareto *strategies* that gave rise to the Pareto optima, i.e., we compute a generating function for the set

$$U^{\text{Pareto}} = \big\{\, \mathbf{u} \in \mathbf{Z}^n : \mathbf{u} \in P \cap \mathbf{Z}^n \text{ and } \mathbf{f}(\mathbf{u}) \text{ is a Pareto optimum} \,\big\}.$$

To this end, we first compute the generating function for the set

$$S^{\text{Pareto}} = \big\{\, (\mathbf{u}, \mathbf{v}) \in \mathbf{Z}^n \times \mathbf{Z}^k : \mathbf{v} \text{ is a Pareto point with Pareto strategy } \mathbf{u} \,\big\}.$$

For this purpose, we consider the multi-graph of the objective functions $f_1, \ldots, f_k$ over P,

$$\begin{aligned} P^{=}_{f_1,\ldots,f_k} = \big\{\, (\mathbf{u}, \mathbf{v}) \in \mathbf{R}^n \times \mathbf{R}^k : \mathbf{u} \in P, \\ v_i = f_i(\mathbf{u}) \text{ for } i = 1, \ldots, k \,\big\}. \end{aligned} \tag{5.9}$$

Using Barvinok's theorem, we can compute in polynomial time the generating function for the integer points in P,

$$g(P; \mathbf{x}) = \sum \big\{\, \mathbf{x}^{\mathbf{u}} : \mathbf{u} \in P \cap \mathbf{Z}^n \,\big\},$$

and also, using the monomial substitution $x_j \to x_j z_1^{f_1(\mathbf{e}_j)} \cdots z_k^{f_k(\mathbf{e}_j)}$ for all j, the generating function is transformed into

$$g(P^{=}_{f_1,\ldots,f_k}; \mathbf{x}, \mathbf{z}) = \sum \big\{\, \mathbf{x}^{\mathbf{u}} \mathbf{z}^{\mathbf{v}} : (\mathbf{u}, \mathbf{v}) \in P^{=}_{f_1,\ldots,f_k} \cap (\mathbf{Z}^n \times \mathbf{Z}^k) \,\big\},$$

where the variables $\mathbf{x}$ carry on the monomial exponents the information of the $\mathbf{u}$-coordinates of $P^{=}_{f_1,\ldots,f_k}$ and the $\mathbf{z}$ variables of the generating function carry the $\mathbf{v}$-coordinates of lattice points in $P^{=}_{f_1,\ldots,f_k}$. Now

$$g(S^{\text{Pareto}}; \mathbf{x}, \mathbf{z}) = \big(g(P; \mathbf{x})\, g(V^{\text{Pareto}}; \mathbf{z})\big) * g(P^{=}_{f_1,\ldots,f_k}; \mathbf{x}, \mathbf{z}), \tag{5.10}$$

which can be computed in polynomial time for fixed dimension by the theorems outlined early on this section. Finally, to obtain the generating function $g(U^{\text{Pareto}}; \mathbf{x})$ of the Pareto strategies, we need to compute the projection of S^{Pareto} into the space of the strategy variables $\mathbf{u}$. Since the projection is one-to-one, it suffices to compute the specialization

$$g(U^{\text{Pareto}}; \mathbf{x}) = g(S^{\text{Pareto}}; \mathbf{x}, \mathbf{z} = \mathbf{1}),$$

which can be done in polynomial time. □

A simplified construction of the rational generating function was proposed by V. Blanco (2007, personal communication); this also removes the need to fix the number of criteria in advance.

5.3 Efficiently listing all Pareto optima

The Pareto optimum that corresponds to the "best" compromise between the individual objective functions is often chosen in an *interactive mode*, where a visualization of the Pareto optima is presented to the user, who then chooses a Pareto optimum. Since the outcome space frequently is of a too large dimension for visualization, an important task is to list (explicitly enumerate) the elements of the *projection* of the Pareto set into some lower-dimensional linear space.

It is clear that the set of Pareto optima (and thus also any projection) is of exponential size in general, ruling out the existence of a polynomial-time enumeration algorithm. In order to analyze the running time of an enumeration algorithm, we must turn to *output-sensitive complexity analysis*.

Various notions of output-sensitive efficiency have appeared in the literature; we follow the discussion of Johnson et al. (1988). Let $W \subseteq \mathbf{Z}^p$ be a finite set to be enumerated. An enumeration algorithm is said to run in *polynomial total time* if its running time is bounded by a polynomial in the encoding size of the input and the output. A stronger notion is that of *incremental polynomial time*: Such an algorithm receives a list of solutions $\mathbf{w}_1, \ldots, \mathbf{w}_N \in W$ as an additional input. In polynomial time, it outputs one solution $\mathbf{w} \in W \setminus \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ or asserts that there are no more solutions. An even stronger notion is that of a *polynomial-delay* algorithm, which takes only polynomial time (in the encoding size of the input) before the first solution is output, between successive outputs of solutions, and after the last solution is output to the termination of the algorithm. Since the algorithm could take exponential time to output all solutions, it could also build exponential-size data structures in the course of the enumeration. This observation gives rise to an even stronger notion of efficiency, a *polynomial-space polynomial-delay* enumeration algorithm.

We also wish to prescribe an *order*, like the lexicographic order, in which the elements are to be enumerated. We consider term orders $\prec_R$ on monomials $\mathbf{y}^{\mathbf{w}}$ that are defined as in Mora and Robbiano (1988) by a non-negative integral $p \times p$-matrix R of full rank. Two monomials satisfy $\mathbf{y}^{\mathbf{w}_1} \prec_R \mathbf{y}^{\mathbf{w}_2}$ if and only if $R\mathbf{w}_1$ is lexicographically smaller than $R\mathbf{w}_2$. In other words, if $\mathbf{r}_1, \ldots, \mathbf{r}_n$ denote the rows of R, there is some $j \in \{1, \ldots, n\}$ such that $\langle \mathbf{r}_i, \mathbf{w}_1 \rangle = \langle \mathbf{r}_i, \mathbf{w}_2 \rangle$ for $i < j$, and $\langle \mathbf{r}_j, \mathbf{w}_1 \rangle < \langle \mathbf{r}_j, \mathbf{w}_2 \rangle$. For example, the unit matrix $R = I_n$ describes the lexicographic term ordering.

We prove the existence of a polynomial-space polynomial-delay prescribed-order enumeration algorithm in a general setting, where the set W to be enumerated is given as the projection of a set presented by a rational generating function.

Theorem 5.4. *Let the dimension k and the maximum number ℓ of binomials in the denominator be fixed.*

Let $V \subseteq \mathbf{Z}^k$ be a bounded set of lattice points with $V \subseteq [-M, M]^k$, given only by the

bound $M \in \mathbf{Z}_+$ *and its rational generating function encoding* $g(V;\mathbf{z})$ *with at most* ℓ *binomials in each denominator. Let*

$$W = \{\, \mathbf{w} \in \mathbf{Z}^p : \exists \mathbf{t} \in \mathbf{Z}^{k-p} \text{ such that } (\mathbf{t},\mathbf{w}) \in V \,\}$$

denote the projection of V *onto the last* p *components. Let* $\prec_R$ *be the term order on monomials in* $y_1,\dots,y_p$ *induced by a given matrix* $R \in \mathbf{N}^{p\times p}$.

There exists a polynomial-space polynomial-delay enumeration algorithm for the points in the projection W*, which outputs the points of* W *in the order given by* $\prec_R$*. The algorithm can be implemented without using the Projection Lemma.*

We remark that Theorem 5.4 is a stronger result than what can be obtained by the repeated application of the monomial-extraction technique of Lemma 7 from De Loera et al. (2004), which would only give an incremental polynomial time enumeration algorithm.

Proof. We give a simple recursive algorithm that is based on the iterative bisection of intervals.

> *Input:* Lower and upper bound vectors $\mathbf{l},\mathbf{u} \in \mathbf{Z}^p$.
>
> *Output:* All vectors $\mathbf{w}$ in W with $\mathbf{l} \le R\mathbf{w} \le \mathbf{u}$, sorted in the order $\preceq_R$.
>
> 1. If the set $W \cap \{\, \mathbf{w} : \mathbf{l} \le R\mathbf{w} \le \mathbf{u} \,\}$ is empty, do nothing.
> 2. Otherwise, if $\mathbf{l} = \mathbf{u}$, compute the unique point $\mathbf{w} \in \mathbf{Z}^k$ with $R\mathbf{w} = \mathbf{l} = \mathbf{u}$ and output $\mathbf{w}$.
> 3. Otherwise, let j be the smallest index with $l_j \ne u_j$. We bisect the integer interval $\{l_j,\dots,u_j\}$ evenly into $\{l_j,\dots,m_j\}$ and $\{m_j+1,\dots,u_j\}$, where $m_j = \left\lfloor \frac{l_j+u_j}{2} \right\rfloor$. We invoke the algorithm recursively on the first part, then on the second part, using the corresponding lower and upper bound vectors.

We first need to compute appropriate lower and upper bound vectors $\mathbf{l},\mathbf{u}$ to start the algorithm. To this end, let N be the largest number in the matrix R and let $\mathbf{l} = -pMN\mathbf{1}$ and $\mathbf{u} = pMN\mathbf{1}$. Then $\mathbf{l} \le R\mathbf{w} \le \mathbf{u}$ holds for all $\mathbf{w} \in W$. Clearly the encoding length of $\mathbf{l}$ and $\mathbf{u}$ is bounded polynomially in the input data.

In step 1 of the algorithm, to determine whether

$$W \cap \{\, \mathbf{w} : \mathbf{l} \le R\mathbf{w} \le \mathbf{u} \,\} = \emptyset, \tag{5.11}$$

we consider the polytope

$$Q_{\mathbf{l},\mathbf{u}} = [-M,M]^{k-p} \times \{\, \mathbf{w} \in \mathbf{R}^p : \mathbf{l} \le R\mathbf{w} \le \mathbf{u} \,\} \subseteq \mathbf{R}^k, \tag{5.12}$$

a parallelelepiped in $\mathbf{R}^k$. Since W is the projection of V and since $V \subseteq [-M,M]^k$, we have (5.11) if and only if $V \cap Q_{\mathbf{l},\mathbf{u}} = \emptyset$. The rational generating function $g(Q_{\mathbf{l},\mathbf{u}};\mathbf{z})$ can

be computed in polynomial time. By using the Intersection Lemma, we can compute the rational generating function $g(V \cap Q_{\mathbf{l},\mathbf{u}}; \mathbf{z})$ in polynomial time. The specialization $g(V \cap Q_{\mathbf{l},\mathbf{u}}; \mathbf{z} = \mathbf{1})$ can also be computed in polynomial time. It gives the number of lattice points in $V \cap Q_{\mathbf{l},\mathbf{u}}$; in particular, we can decide whether $V \cap Q_{\mathbf{l},\mathbf{u}} = \emptyset$.

It is clear that the algorithm outputs the elements of W in the order given by $\prec_R$. We next show that the algorithm is a polynomial-space polynomial-delay enumeration algorithm. The subproblem in step 1 only depends on the input data as stated in the theorem and on the vectors $\mathbf{l}$ and $\mathbf{u}$, whose encoding length only decreases in recursive invocations. Therefore each of the subproblems can be solved in polynomial time (thus also in polynomial space).

The recursion of the algorithm corresponds to a binary tree whose nodes are labeled by the bound vectors $\mathbf{l}$ and $\mathbf{u}$. There are two types of leaves in the tree, one corresponding to the "empty-box" situation (5.11) in step 1, and one corresponding to the "solution-output" situation in step 2. Inner nodes of the tree correspond to the recursive invocation of the algorithm in step 3. It is clear that the depth of the recursion is $\mathrm{O}(p \log(pMN))$, because the integer intervals are bisected evenly. Thus the stack space of the algorithm is polynomially bounded. Since the algorithm does not maintain any global data structures, the whole algorithm uses polynomial space only.

Let $\mathbf{w}_i \in W$ be an arbitrary solution and let $\mathbf{w}_{i+1}$ be its direct successor in the order $\prec_R$. We shall show that the algorithm only spends polynomial time between the output of $\mathbf{w}_i$ and the output of $\mathbf{w}_{i+1}$. The key property of the recursion tree of the algorithm is the following:

> Every inner node is the root of a subtree that contains at least one solution-output leaf. (5.13)

The reason for that property is the test for situation (5.11) in step 1 of the algorithm. Therefore, the algorithm can visit only $\mathrm{O}(p \log(pMN))$ inner nodes and empty-box leaves between the solution-output leaves for $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$. For the same reason, also the time before the first solution is output and the time after the last solution is output are polynomially bounded. □

The following corollary, which is a stronger formulation of Theorem 5.1 (iii), is immediate.

Corollary 5.5. *Let n and k be fixed integers. There exist polynomial-space polynomial-delay enumeration algorithms to enumerate the set of Pareto optima of the multicriterion integer linear program (5.1), the set of Pareto strategies, or arbitrary projections thereof in lexicographic order (or an arbitrary term order).*

Remark 5.6. We remark that Theorem 5.4 is of general interest. For instance, it also implies the existence of a polynomial-space polynomial-delay prescribed-order

enumeration algorithm for Hilbert bases of rational polyhedral cones in fixed dimension.

Indeed, fix the dimension d and let $C = \operatorname{cone}\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subseteq \mathbf{R}^d$ be a pointed rational polyhedral cone. The *Hilbert basis* of C is defined as the inclusion-minimal set $H \subseteq C \cap \mathbf{Z}^d$ which generates $C \cap \mathbf{Z}^d$ as a monoid. For *simplicial* cones C (where $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent), Barvinok and Woods (2003) proved that one can compute the rational generating function $g(H; \mathbf{z})$ (having a constant number of binomials in the denominators) of the Hilbert basis of $C \cap \mathbf{Z}^d$ using the Projection Theorem. The same technique works for non-simplicial pointed cones. Now Theorem 5.4 gives a polynomial-space polynomial-delay prescribed-order enumeration algorithm.

5.4 Selecting a Pareto optimum using polyhedral global criteria

Now that we know that all Pareto optima of a multicriteria integer linear programs can be encoded in a rational generating function, and that they can be listed efficiently on the output size, we can aim to apply selection criteria stated by a user. The advantage of our setup is that when we optimize a global objective function it guarantees to return a Pareto optimum, because we evaluate the global criterion only on the Pareto optima. Let us start with the simplest global criterion which generalizes the use of the ℓ_1 norm distance function:

Theorem 5.7. *Let the dimension k and the maximum number ℓ of binomials in the denominator be fixed.*

Let $V \subseteq \mathbf{Z}^k$ be a bounded set of lattice points with $V \subseteq [-M, M]^{n+k}$, given only by the bound $M \in \mathbf{Z}_+$ and its rational generating function encoding $g(V; \mathbf{z})$ with at most ℓ binomials in the denominators.

Let $Q \subseteq \mathbf{R}^k$ be a rational convex central-symmetric polytope with $\mathbf{0} \in \operatorname{int} Q$, given by its vertex or inequality description. Let the polyhedral norm $\|\cdot\|_Q$ be defined using the Minkowski functional

$$\|\mathbf{y}\|_Q = \inf\{\, \lambda \geq 0 : \mathbf{y} \in \lambda Q \,\}. \tag{5.14}$$

Finally, let a prescribed point $\hat{\mathbf{v}} \in \mathbf{Z}^k$ be given.

(i) *There exists a polynomial-time algorithm to find a point $\mathbf{v} \in V$ that minimizes the distance $d_Q(\mathbf{v}, \hat{\mathbf{v}}) = \|\mathbf{v} - \hat{\mathbf{v}}\|_Q$ from the prescribed point.*

(ii) *There exists a polynomial-space polynomial-delay enumeration algorithm for enumerating the points of V in the order of increasing distances d_Q from the prescribed point $\hat{\mathbf{v}}$, refined by an arbitrary term order $\prec_R$ given by a matrix $R \in \mathbf{N}^{k \times k}$.*

Theorem 5.2, as stated in the introduction, is an immediate corollary of this theorem.

Proof. Since the dimension k is fixed, we can compute an inequality description

$$Q = \{\, \mathbf{y} \in \mathbf{R}^k : A\mathbf{y} \leq \mathbf{b} \,\}$$

of Q with $A \in \mathbf{Z}^{m\times k}$ and $\mathbf{b} \in \mathbf{Z}^k$ in polynomial time, if Q is not already given by an inequality description. Let $\mathbf{v} \in V$ be arbitrary; then

$$\begin{aligned} d_Q(\hat{\mathbf{v}}, \mathbf{v}) &= \|\mathbf{v} - \hat{\mathbf{v}}\|_Q \\ &= \inf\{\, \lambda \geq 0 : \mathbf{v} - \hat{\mathbf{v}} \in \lambda Q \,\} \\ &= \min\{\, \lambda \geq 0 : \lambda\mathbf{b} \geq A(\mathbf{v} - \hat{\mathbf{v}}) \,\}. \end{aligned}$$

Thus there exists an index $i \in \{1, \ldots, m\}$ such that

$$d_Q(\hat{\mathbf{v}}, \mathbf{v}) = \frac{(A\mathbf{v})_i - (A\hat{\mathbf{v}})_i}{b_i};$$

so $d_Q(\hat{\mathbf{v}}, \mathbf{v})$ is an integer multiple of $1/b_i$. Hence for every $\mathbf{v} \in V$, we have that

$$d_Q(\hat{\mathbf{v}}, \mathbf{v}) \in \frac{1}{\operatorname{lcm}(b_1, \ldots, b_m)}\mathbf{Z}_+, \tag{5.15}$$

where $\operatorname{lcm}(b_1, \ldots, b_m)$ clearly is a number of polynomial encoding size. On the other hand, every $\mathbf{v} \in V$ certainly satisfies

$$d_Q(\hat{\mathbf{v}}, \mathbf{v}) \leq ka\big(M + \max\{|\hat{v}_1|, \ldots, |\hat{v}_d|\}\big) \tag{5.16}$$

where a is the largest number in A, which is also a bound of polynomial encoding size.

Using Barvinok's algorithm, we can compute the rational generating function $g(\hat{\mathbf{v}} + \lambda Q; \mathbf{z})$ for any rational λ of polynomial enoding size in polynomial time. We can also compute the rational generating function $g(V \cap (\hat{\mathbf{v}} + \lambda Q); \mathbf{z})$ using the Intersection Lemma. By computing the specialization $g(V \cap (\hat{\mathbf{v}} + \lambda Q); \mathbf{z} = \mathbf{1})$, we can compute the number of points in $V \cap (\hat{\mathbf{v}} + \lambda Q)$, thus we can decide whether this set is empty or not.

Hence we can employ binary search for the smallest $\lambda \geq 0$ such that $V \cap (\hat{\mathbf{v}} + \lambda Q)$ is nonempty. Because of (5.15) and (5.16), it runs in polynomial time. By using the recursive bisection algorithm of Theorem 5.4, it is then possible to construct one Pareto optimum in $V \cap (\hat{\mathbf{v}} + \lambda Q)$ for part (i), or to construct a sequence of Pareto optima in the desired order for part (ii). □

5.5 Selecting a Pareto optimum using non-polyhedral global criteria

Now we consider a global criterion using a distance function corresponding to a non-polyhedral norm like the Euclidean norm $\|\cdot\|_2$ (or any other ℓ_p-norm for $1 < p < \infty$). We are able to prove a very strong type of approximation result, a so-called fully polynomial-time approximation scheme (FPTAS), in a somewhat more general setting.

Definition 5.8 (FPTAS)**.** Consider the optimization problems

$$\max\{\, f(\mathbf{v}) : \mathbf{v} \in V \,\}, \tag{5.17a}$$
$$\min\{\, f(\mathbf{v}) : \mathbf{v} \in V \,\}. \tag{5.17b}$$

A *fully polynomial-time approximation scheme (FPTAS)* for the maximization problem (5.17a) or the minimization problem (5.17b), respectively, is a family $\{\, \mathcal{A}_\epsilon : \epsilon \in \mathbf{Q},\ \epsilon > 0 \,\}$ of approximation algorithms $\mathcal{A}_\epsilon$, each of which returns an *ϵ-approximation*, i.e., a solution $\mathbf{v}_\epsilon \in V$ with

$$f(\mathbf{v}_\epsilon) \geq (1-\epsilon) f^* \quad \text{where} \quad f^* = \max_{\mathbf{v} \in V} f(\mathbf{v}), \tag{5.18a}$$

or, respectively,

$$f(\mathbf{v}_\epsilon) \leq (1+\epsilon) f^* \quad \text{where} \quad f^* = \min_{\mathbf{v} \in V} f(\mathbf{v}), \tag{5.18b}$$

such that the algorithms $\mathcal{A}_\epsilon$ run in time polynomial in the input size and $\frac{1}{\epsilon}$.

Remark 5.9. An FPTAS is based on the notion of ϵ-approximation (5.18), which gives an approximation guarantee relative to the value f^* of an optimal solution. It is clear that this notion is most useful for objective functions f that are non-negative on the feasible region V. Since the approximation quality of a solution changes when the objective function is changed by an additive constant, it is non-trivial to convert an FPTAS for a maximization problem to an FPTAS for a minimization problem.

We shall present an FPTAS for the problem of minimizing the distance of a Pareto optimum from a prescribed outcome vector $\hat{\mathbf{v}} \in \mathbf{Z}^k$. We consider distances $d(\hat{\mathbf{v}}, \cdot)$ induced by a pseudo-norm $\|\cdot\|_Q$ via

$$d(\hat{\mathbf{v}}, \mathbf{v}) = \|\mathbf{v} - \hat{\mathbf{v}}\|_Q \tag{5.19a}$$

To this end, let $Q \subseteq \mathbf{R}^k$ be a compact basic semialgebraic set with $\mathbf{0} \in \operatorname{int} Q$, which is described by one polynomial inequality,

$$Q = \left\{\, \mathbf{y} \in \mathbf{R}^k : q(\mathbf{y}) \leq 1 \,\right\}, \tag{5.19b}$$

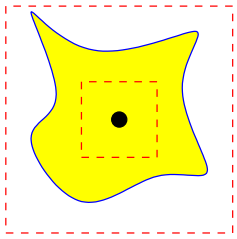


Figure 5.10: A set defining a pseudo-norm with the inscribed and circumscribed cubes αB_∞ and βB_∞ (dashed).

where $q \in \mathbf{Q}[y_1, \ldots, y_k]$ is a homogeneous polynomial of (even) degree D. The pseudo-norm $\|\cdot\|_Q$ is now defined using the Minkowski functional

$$\|\mathbf{y}\|_Q = \inf\{\, \lambda \geq 0 : \mathbf{y} \in \lambda Q \,\} \tag{5.19c}$$

Note that we do not make any assumptions of convexity of Q, which would make $\|\cdot\|_Q$ a norm. Since Q is compact and $\mathbf{0} \in \operatorname{int} Q$, there exist positive rational numbers (norm equivalence constants) α, β with

$$\alpha B_\infty \subseteq Q \subseteq \beta B_\infty \quad \text{where} \quad B_\infty = \{\, \mathbf{y} \in \mathbf{R}^k : \|\mathbf{y}\|_\infty \leq 1 \,\}; \tag{5.20}$$

see Figure 5.10.

Now we can formulate our main theorem, which has Theorem 5.3, which we stated in the introduction, as an immediate corollary.

Theorem 5.10. *Let the dimension n and the number k of objective functions be fixed. Moreover, let a degree D and two rational numbers $0 < \alpha \leq \beta$ be fixed. Then there exists a fully polynomial-time approximation scheme for the problem of minimizing the distance $d_Q(\hat{\mathbf{v}}, \mathbf{v})$, defined via (5.19) by a homogeneous polynomial $q \in \mathbf{Q}[y_1, \ldots, y_k]$ of degree D satisfying (5.20), whose coefficients are encoded in binary and whose exponent vectors are encoded in unary, of a Pareto optimum of (5.1) from a prescribed outcome vector $\hat{\mathbf{v}} \in \mathbf{Z}^k$.*

The proof is based on the FPTAS for polynomial optimization over lattice point sets (Theorem 4.3).

Proof of Theorem 5.10. Using Theorem 5.1, we first compute the rational generating function $g(V^{\text{Pareto}}; \mathbf{z})$ of the Pareto optima. With binary search using the Intersection Lemma with generating functions of cubes as in section 5.3, we can find the smallest non-negative integer γ such that

$$(\hat{\mathbf{v}} + \gamma B_\infty) \cap V^{\text{Pareto}} \neq \emptyset. \tag{5.21}$$

If $\gamma = 0$, then the prescribed outcome vector $\hat{\mathbf{v}}$ itself is a Pareto optimum, so it is the optimal solution to the problem.

Otherwise, let $\mathbf{v}_0$ be an arbitrary outcome vector in $(\hat{\mathbf{v}} + \gamma B_\infty) \cap V^{\text{Pareto}}$. Then

$$\begin{aligned}\gamma \geq \|\mathbf{v}_0 - \hat{\mathbf{v}}\|_\infty &= \inf\{\, \lambda : \mathbf{v}_0 - \hat{\mathbf{v}} \in \lambda B_\infty \,\} \\ &\geq \inf\{\, \lambda : \mathbf{v}_0 - \hat{\mathbf{v}} \in \lambda \tfrac{1}{\alpha} Q \,\} = \alpha \|\mathbf{v}_0 - \hat{\mathbf{v}}\|_Q ,\end{aligned}$$

thus $\|\mathbf{v}_0 - \hat{\mathbf{v}}\|_Q \leq \gamma/\alpha$. Let $\delta = \beta\gamma/\alpha$. Then, for every $\mathbf{v}_1 \in \mathbf{R}^k$ with $\|\mathbf{v}_1 - \hat{\mathbf{v}}\|_\infty \geq \delta$ we have

$$\begin{aligned}\delta \leq \|\mathbf{v}_1 - \hat{\mathbf{v}}\|_\infty &= \inf\{\, \lambda : \mathbf{v}_1 - \hat{\mathbf{v}} \in \lambda B_\infty \,\} \\ &\leq \inf\{\, \lambda : \mathbf{v}_1 - \hat{\mathbf{v}} \in \lambda \tfrac{1}{\beta} Q \,\} = \beta \|\mathbf{v}_1 - \hat{\mathbf{v}}\|_Q ,\end{aligned}$$

thus

$$\|\mathbf{v}_1 - \hat{\mathbf{v}}\|_Q \geq \delta/\beta = \gamma/\alpha \geq \|\mathbf{v}_0 - \hat{\mathbf{v}}\|_Q .$$

Therefore, a Pareto optimum $\mathbf{v}^* \in V^{\text{Pareto}}$ minimizing the distance d_Q from the prescribed outcome vector $\hat{\mathbf{v}}$ is contained in the cube $\hat{\mathbf{v}} + \delta B_\infty$. Moreover, for all points $\mathbf{v} \in \hat{\mathbf{v}} + \delta B_\infty$ we have

$$\|\mathbf{v}_0 - \hat{\mathbf{v}}\|_Q \leq \delta/\alpha = \beta\gamma/\alpha^2.$$

We define a function f by

$$f(\mathbf{v}) = \left(\beta\gamma/\alpha^2\right)^D - \|\mathbf{v} - \hat{\mathbf{v}}\|_Q^D , \tag{5.22}$$

which is non-negative over the cube $\hat{\mathbf{v}} + \delta B_\infty$. Since q is a homogeneous polynomial of degree D, we obtain

$$f(\mathbf{v}) = \left(\beta\gamma/\alpha^2\right)^D - q(\mathbf{v} - \hat{\mathbf{v}}) \tag{5.23}$$

so f is a polynomial.

We next compute the rational generating function

$$g(V^{\text{Pareto}} \cap (\hat{\mathbf{v}} + \delta B_\infty); \mathbf{z})$$

from $g(V^{\text{Pareto}}; \mathbf{z})$ using the Intersection Lemma. Let $\epsilon' > 0$ be a rational number, which we will determine later. By Theorem 4.3, we compute a solution $\mathbf{v}_{\epsilon'} \in V^{\text{Pareto}}$ with

$$f(\mathbf{v}_{\epsilon'}) \geq (1 - \epsilon') f(\mathbf{v}^*),$$

or, equivalently,

$$f(\mathbf{v}^*) - f(\mathbf{v}_{\epsilon'}) \leq \epsilon' f(\mathbf{v}^*).$$

Thus,

$$\begin{aligned}[d_Q(\hat{\mathbf{v}}, \mathbf{v}_{\epsilon'})]^D - [d_Q(\hat{\mathbf{v}}, \mathbf{v}^*)]^D &= \|\mathbf{v}_{\epsilon'} - \hat{\mathbf{v}}\|_Q^D - \|\mathbf{v}^* - \hat{\mathbf{v}}\|_Q^D \\ &= f(\mathbf{v}^*) - f(\mathbf{v}_{\epsilon'}) \\ &\leq \epsilon' f(\mathbf{v}^*) \\ &= \epsilon' \left(\left(\beta\gamma/\alpha^2\right)^D - \|\mathbf{v}^* - \hat{\mathbf{v}}\|_Q^D \right).\end{aligned}$$

Since γ is the smallest integer with (5.21) and also $\|\mathbf{v}^* - \hat{\mathbf{v}}\|_\infty$ is an integer, we have

$$\gamma \leq \|\mathbf{v}^* - \hat{\mathbf{v}}\|_\infty \leq \beta \|\mathbf{v}^* - \hat{\mathbf{v}}\|_Q .$$

Thus,

$$[d_Q(\hat{\mathbf{v}}, \mathbf{v}_{\epsilon'})]^D - [d_Q(\hat{\mathbf{v}}, \mathbf{v}^*)]^D \leq \epsilon' \left[\left(\frac{\beta}{\alpha} \right)^{2D} - 1 \right] \|\mathbf{v}^* - \hat{\mathbf{v}}\|_Q^D .$$

An elementary calculation yields

$$d_Q(\hat{\mathbf{v}}, \mathbf{v}_{\epsilon'}) - d_Q(\hat{\mathbf{v}}, \mathbf{v}^*) \leq \frac{\epsilon'}{D} \left[\left(\frac{\beta}{\alpha} \right)^{2D} - 1 \right] d_Q(\hat{\mathbf{v}}, \mathbf{v}^*).$$

Thus we can choose

$$\epsilon' = \epsilon \, D \left[\left(\frac{\beta}{\alpha} \right)^{2D} - 1 \right]^{-1} \tag{5.24}$$

to get the desired estimate. Since α, β and D are fixed constants, we have $\epsilon' = \Theta(\epsilon)$. Thus the computation of $\mathbf{v}_{\epsilon'} \in V^{\text{Pareto}}$ by Theorem 4.3 runs in time polynomial in the input encoding size and $\frac{1}{\epsilon}$. □

Remark 5.11. It is straightforward to extend this result to also include the ℓ_p norms for *odd* integers p, by solving the approximation problem separately for all of the $2^k = \mathrm{O}(1)$ shifted orthants $\hat{\mathbf{v}} + O_{\boldsymbol{\sigma}} = \{ \mathbf{v} : \sigma_i(v_i - \hat{v}_i) \geq 0 \}$, where $\boldsymbol{\sigma} \in \{\pm 1\}^k$. On each of the orthants, the ℓ_p-norm has a representation by a polynomial as required by Theorem 5.10.

Notes and sources

This chapter is based on De Loera et al. (2009b).

Chapter 6

Further applications

An application of rational generating functions to the computation of pure Nash equilibria in integer programming games appears in Köppe et al. (2008a). A continuous version of the summation method is explored in Baldoni et al. (2010).

Bibliography

L. Adleman and K. Manders. Reducibility, randomness and intractability. In *Proc. 9th Annual ACM Symposium on Theory of Computing*, pages 151–163, 1977.

M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annals of Math.*, 160: 781–793, 2004.

V. Baldoni, N. Berline, J. A. De Loera, M. Köppe, and M. Vergne. How to integrate a polynomial over a simplex. *Mathematics of Computation*, posted online July 14, 2010.

A. I. Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research*, 19:769–779, 1994a.

A. I. Barvinok. Polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research*, 19:769–779, 1994b.

A. I. Barvinok. Computing the volume, counting integral points, and exponential sums. *Discrete Comput. Geom.*, 10(2):123–141, 1993.

A. I. Barvinok. Computing the Ehrhart quasi-polynomial of a rational simplex. *Math. Comp.*, 75(255):1449–1466 (electronic), 2006a.

A. I. Barvinok. Computing the Ehrhart quasi-polynomial of a rational simplex. *Math. Comp.*, 75(255):1449–1466, 2006b.

A. I. Barvinok and J. E. Pommersheim. An algorithmic theory of lattice points in polyhedra. In L. J. Billera, A. Björner, C. Greene, R. E. Simion, and R. P. Stanley, editors, *New Perspectives in Algebraic Combinatorics*, volume 38 of *Math. Sci. Res. Inst. Publ.*, pages 91–147. Cambridge Univ. Press, Cambridge, 1999a.

A. I. Barvinok and J. E. Pommersheim. An algorithmic theory of lattice points in polyhedra. In L. J. Billera, A. Björner, C. Greene, R. E. Simion, and R. P. Stanley, editors, *New Perspectives in Algebraic Combinatorics*, volume 38 of *Math. Sci. Res. Inst. Publ.*, pages 91–147. Cambridge Univ. Press, Cambridge, 1999b.

A. I. Barvinok and K. Woods. Short rational generating functions for lattice point problems. *Journal of the AMS*, 16(4):957–979, 2003.

M. Beck and F. Sottile. Irrational proofs for three theorems of Stanley. *European Journal of Combinatorics*, 28(1):403–409, 2007.

M. Beck, C. Haase, and F. Sottile. Formulas of Brion, Lawrence, and Varchenko on rational generating functions for cones. eprint arXiv:math.CO/0506466, 2006.

M. Bellare and P. Rogaway. The complexity of approximating a nonlinear program. *Mathematical Programming*, 69(1):429–441, Jul 1995. doi: 10.1007/BF01585569. URL http://dx.doi.org/10.1007/BF01585569.

M. Bellare and P. Rogaway. The complexity of aproximating a nonlinear program. In Pardalos (1993).

Y. Berstein and S. Onn. Nonlinear bipartite matching. *Discrete Optimization*, 5: 53–65, 2008.

Y. Berstein, J. Lee, H. Maruri-Aguilar, S. Onn, E. Riccomagno, R. Weismantel, and H. Wynn. Nonlinear matroid optimization and experimental design. *SIAM Journal on Discrete Mathematics*, 22(3):901–919, 2008a.

Y. Berstein, J. Lee, S. Onn, and R. Weismantel. Nonlinear optimization for matroid intersection and extensions. *IBM Research Report RC24610*, 2008b.

L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bull. Am. Math. Soc.*, 21:1–46, 1989.

M. Brion. Points entiers dans les polyédres convexes. *Ann. Sci. École Norm. Sup.*, 21(4):653–663, 1988.

D. A. Cox, J. B. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, Berlin, Germany, 1992.

E. de Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research*, 16(2):111–125, Jun 2008. doi: 10.1007/s10100-007-0052-9. URL http://dx.doi.org/10.1007/s10100-007-0052-9.

E. de Klerk, M. Laurent, and P. A. Parrilo. A PTAS for the minimization of polynomials of fixed degree over the simplex. *Theoretical Computer Science*, 361:210–225, 2006.

J. A. De Loera and S. Onn. All linear and integer programs are slim 3-way transportation programs. *SIAM Journal of Optimization*, 17:806–821, 2006a.

J. A. De Loera and S. Onn. Markov bases of three-way tables are arbitrarily complicated. *Journal of Symbolic Computation*, 41:173–181, 2006b.

J. A. De Loera, D. Haws, R. Hemmecke, P. Huggins, B. Sturmfels, and R. Yoshida. Short rational functions for toric algebra and applications. *Journal of Symbolic Computation*, 38(2):959–973, 2004.

J. A. De Loera, R. Hemmecke, M. Köppe, and R. Weismantel. FPTAS for mixed-integer polynomial optimization with a fixed number of variables. In *17th ACM-SIAM Symposium on Discrete Algorithms*, pages 743–748, 2006a.

J. A. De Loera, R. Hemmecke, M. Köppe, and R. Weismantel. Integer polynomial optimization in fixed dimension. *Mathematics of Operations Research*, 31(1):147–153, 2006b.

J. A. De Loera, R. Hemmecke, M. Köppe, and R. Weismantel. FPTAS for mixed-integer polynomial optimization with a fixed number of variables. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, Miami, FL, January 22–24, 2006*, pages 743–748, 2006c.

J. A. De Loera, R. Hemmecke, M. Köppe, and R. Weismantel. FPTAS for optimizing polynomials over the mixed-integer points of polytopes in fixed dimension. *Mathematical Programming, Series A*, 118:273–290, 2008a. doi: 10.1007/s10107-007-0175-8.

J. A. De Loera, R. Hemmecke, S. Onn, and R. Weismantel. N-fold integer programming. *Disc. Optim., to appear*, 2008b.

J. A. De Loera, D. C. Haws, and M. Köppe. Ehrhart polynomials of matroid polytopes and polymatroids. *Discrete Comput. Geom.*, 42(4):670–702, 2009a. doi: 10.1007/s00454-008-9080-z.

J. A. De Loera, R. Hemmecke, and M. Köppe. Pareto optima of multicriteria integer linear programs. *INFORMS Journal on Computing*, 21(1):39–48, Winter 2009b. doi: 10.1287/ijoc.1080.0277.

M. Dyer and R. Kannan. On Barvinok's algorithm for counting lattice points in fixed dimension. *Mathematics of Operations Research*, 22:545–549, 1997.

M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, 22:425–460, 2000.

F. Eisenbrand. Integer programming and algorithmic geometry of numbers. In M. Jünger, T. Liebling, D. Naddef, W. Pulleyblank, G. Reinelt, G. Rinaldi, and L. Wolsey, editors, *50 Years of Integer Programming 1958–2008*. Springer-Verlag, 2010.

F. Eisenbrand and G. Shmonin. Parametric integer programming in fixed dimension, 2008. http://arXiv.org/abs/0801.4336.

V. A. Emelichev and V. A. Perepelitsa. On the cardinality of the set of alternatives in discrete many-criterion problems. *Discrete Mathematics and Applications*, 2: 461–471, 1992.

J. Figueira, S. Greco, and M. Ehrgott, editors. *Multiple Criteria Decision Analysis. State of the Art Surveys*. Springer, 2005.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, New York, NY, 1979.

M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, Germany, 1988.

J. Håstad. Some optimal inapproximability results. In *Proceedings of the 29th Symposium on the Theory of Computing (STOC)*, pages 1–10. ACM, 1997.

R. Hemmecke, S. Onn, and R. Weismantel. A polynomial oracle-time algorithm for convex integer minimization. *Mathematical Programming, Series A*, 2009. doi: 10.1007/s10107-009-0276-7. Published online 06 March 2009.

R. Hildebrand and M. Köppe. A faster algorithm for quasi-convex integer polynomial optimization. eprint arXiv:1006.4661 [math.OC], 2010.

D. Hochbaum. Complexity and algorithms for nonlinear optimization problems. *Annals of Operations Research*, 153(1):257–296, Sep 2007. doi: 10.1007/s10479-007-0172-6. URL http://dx.doi.org/10.1007/s10479-007-0172-6.

H. Isermann. Proper efficiency and the linear vector maximum problem. *Operations Research*, 22:189–191, 1974.

R. G. Jeroslow. There cannot be any algorithm for integer programming with quadratic constraints. *Operations Research*, 21(1):221–224, 1973.

D. S. Johnson, M. Yannakakis, and Ch. H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27:119–123, 1988.

J. P. Jones. Universal diophantine equation. *Journal of Symbolic Logic*, 47(3):403–410, 1982.

J. P. Jones and Yu. V. Matiyasevich. Proof of recursive unsolvability of Hilbert's tenth problem. *The American Mathematical Monthly*, 98(8):689–709, Oct. 1991.

M. Köppe. A primal Barvinok algorithm based on irrational decompositions. *SIAM Journal on Discrete Mathematics*, 21(1):220–236, 2007. doi: 10.1137/060664768.

M. Köppe. LattE macchiato, version 1.2-mk-0.9.3, an improved version of De Loera et al.'s LattE program for counting integer points in polyhedra with variants of Barvinok's algorithm. Available from URL http://www.math.ucdavis.edu/~mkoeppe/latte/, 2008.

M. Köppe and S. Verdoolaege. Computing parametric rational generating functions with a primal Barvinok algorithm. *The Electronic Journal of Combinatorics*, 15: 1–19, 2008. #R16.

M. Köppe, C. T. Ryan, and M. Queyranne. Rational generating functions and integer programming games. eprint arXiv:0809.0689v1 [cs.GT], 2008a.

M. Köppe, S. Verdoolaege, and K. M. Woods. An implementation of the Barvinok–Woods integer projection algorithm. Information Theory and Statistical Learning (ITSL 2008), Las Vegas, Proceedings, 2008b.

J. C. Lagarias. On the computational complexity of determining the solvability or unsolvability of the equation $x^2 - dy^2 = -1$. *Transactions of the American Mathematical Society*, 260(2):485–508, 1980. ISSN 00029947. URL http://www.jstor.org/stable/1998017.

J. C. Lagarias. Succinct certificates for the solvability of binary quadratic diophantine equations. e-print arXiv:math/0611209v1, 2006. Extended and updated version of a 1979 FOCS paper.

J. Lawrence. Rational-function-valued valuations on polyhedra. In *Discrete and Computational Geometry (New Brunswick, NJ, 1989/1990)*, volume 6 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 199–208. Amer. Math. Soc., Providence, RI, 1991.

J. Lee, S. Onn, and R. Weismantel. On test sets for nonlinear integer maximization. *Operations Research Letters*, 36:439–443, 2008a.

J. Lee, S. Onn, and R. Weismantel. Nonlinear optimization over a weighted independence system. *IBM Research Report RC24513*, 2008b.

H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

K. Manders and L. Adleman. NP-complete decision problems for binary quadratics. *J. Comp. Sys. Sci.*, 16:168–184, 1978.

Yu. V. Matiyasevich. Enumerable sets are diophantine. *Doklady Akademii Nauk SSSR*, 191:279–282, 1970. (Russian); English translation, Soviet Mathematics Doklady, vol. 11 (1970), pp. 354–357.

Yu. V. Matiyasevich. *Hilbert's tenth problem*. The MIT Press, Cambridge, MA, USA, 1993.

T. Mora and L. Robbiano. The Gröbner fan of an ideal. *Journal of Symbolic Computation*, 6(2–3):183–208, 1988.

T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of Turán. *Canadian Journal of Mathematics*, 17:533–540, 1965.

S. Onn. Convex discrete optimization. eprint arXiv:math/0703575, 2007.

P. M. Pardalos, editor. *Complexity in Numerical Optimization*. World Scientific, 1993.

A. V. Pukhlikov and A. G. Khovanskii. A riemann-roch theorem for integrals and sums of quasi-polynomials over virtual polytopes. *St. Petersburg Math. J.*,, 4(4): 789–812, 1993.

Y. Sawaragi, H. Nakayama, and T. Tanino, editors. *Theory of Multiobjective Optimization*. Academic Press, 1985.

I. V. Sergienko and V. A. Perepelitsa. Finding the set of alternatives in discrete multi-criterion problems. *Cybernetics*, 3:673–683, 1991.

C. L. Siegel. Zur Theorie der quadratischen Formen. *Nachrichten der Akademie der Wissenschaften in Göttingen, II, Mathematisch-Physikalische Klasse*, 3:21–46, 1972.

T. Skolem. *Diophantische Gleichungen*, volume 5 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. 1938.

A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, Series 2*, 42: 230–265, 1936. Errata in ibidem, 43 (1937):544–546.

S. A. Vavasis. Polynomial time weak approximation algorithms for quadratic programming. In Pardalos (1993).

S. Verdoolaege. `barvinok`. Available from URL `http://freshmeat.net/projects/barvinok/`, 2007.

S. Verdoolaege and K. M. Woods. Counting with rational generating functions. *J. Symb. Comput.*, 43(2):75–91, 2008. ISSN 0747-7171. doi: http://dx.doi.org/10.1016/j.jsc.2007.07.007.

K. Woods. *Rational Generating Functions and Lattice Point Sets*. PhD thesis, University of Michigan, 2004.