

Nonlinear Integer Programming *

Raymond Hemmecke, Matthias Köppe, Jon Lee and Robert Weismantel

Abstract. Research efforts of the past fifty years have led to a development of linear integer programming as a mature discipline of mathematical optimization. Such a level of maturity has not been reached when one considers nonlinear systems subject to integrality requirements for the variables. This chapter is dedicated to this topic.

The primary goal is a study of a simple version of general nonlinear integer problems, where all constraints are still linear. Our focus is on the computational complexity of the problem, which varies significantly with the type of nonlinear objective function in combination with the underlying combinatorial structure. Numerous boundary cases of complexity emerge, which sometimes surprisingly lead even to polynomial time algorithms.

We also cover recent successful approaches for more general classes of problems. Though no positive theoretical efficiency results are available, nor are they likely to ever be available, these seem to be the currently most successful and interesting approaches for solving practical problems.

It is our belief that the study of algorithms motivated by theoretical considerations and those motivated by our desire to solve practical instances should and do inform one another. So it is with this viewpoint that we present the subject, and it is in this direction that we hope to spark further research.

Raymond Hemmecke
Otto-von-Guericke-Universität Magdeburg, FMA/IMO, Universitätsplatz 2, 39106 Magdeburg,
Germany, e-mail: hemmecke@imo.math.uni-magdeburg.de

Matthias Köppe
University of California, Davis, Dept. of Mathematics, One Shields Avenue, Davis, CA, 95616,
USA, e-mail: mkoeppe@math.ucdavis.edu

Jon Lee
IBM T.J. Watson Research Center, PO Box 218, Yorktown Heights, NY, 10598, USA, e-mail: jon-lee@us.ibm.com

Robert Weismantel
Otto-von-Guericke-Universität Magdeburg, FMA/IMO, Universitätsplatz 2, 39106 Magdeburg,
Germany, e-mail: weismant@imo.math.uni-magdeburg.de

* To appear in: M. Jünger, T. Liebling, D. Naddef, G. Nemhauser, W. Pulleyblank, G. Reinelt, G. Rinaldi, and L. Wolsey (eds.), *50 Years of Integer Programming 1958–2008: The Early Years and State-of-the-Art Surveys*, Springer-Verlag, 2009, ISBN 3540682740.

1 Overview

In the past decade, *nonlinear* integer programming has gained a lot of mindshare. Obviously many important applications demand that we be able to handle nonlinear objective functions and constraints. Traditionally, nonlinear mixed-integer programs have been handled in the context of the field of global optimization, where the main focus is on numerical algorithms to solve nonlinear continuous optimization problems and where integrality constraints were considered as an afterthought, using branch-and-bound over the integer variables. In the past few years, however, researchers from the field of integer programming have increasingly studied nonlinear mixed-integer programs from their point of view. Nevertheless, this is generally considered a very young field, and most of the problems and methods are not as well-understood or stable as in the case of linear mixed-integer programs.

Any contemporary review of nonlinear mixed-integer programming will therefore be relatively short-lived. For this reason, our primary focus is on a classification of nonlinear mixed-integer problems from the point of view of computational complexity, presenting theory and algorithms for the efficiently solvable cases. The hope is that at least this part of the chapter will still be valuable in the next few decades. However, we also cover recent successful approaches for more general classes of problems. Though no positive theoretical efficiency results are available — nor are they likely to ever be available, these seem to be the currently most successful and interesting approaches for solving practical problems. It is our belief that the study of algorithms motivated by theoretical considerations and those motivated by our desire to solve practical instances should and do inform one another. So it is with this viewpoint that we present the subject, and it is in this direction that we hope to spark further research.

Let us however also remark that the selection of the material that we discuss in this chapter is subjective. There are topics that some researchers associate with “nonlinear integer programming” that are not covered here. Among them are pseudo-Boolean optimization, max-cut and quadratic assignment as well as general 0/1 polynomial programming. There is no doubt that these topics are interesting, but, in order to keep this chapter focused, we refrain from going into these topics. Instead we refer the interested reader to the references [55] on max-cut, [32] for recent advances in general 0/1 polynomial programming, and the excellent surveys [29] on pseudo-Boolean optimization and [103, 34] on the quadratic assignment problem.

A general model of mixed-integer programming could be written as

$$\begin{aligned}
 \max/\min \quad & f(x_1, \dots, x_n) \\
 \text{s.t.} \quad & g_1(x_1, \dots, x_n) \leq 0 \\
 & \vdots \\
 & g_m(x_1, \dots, x_n) \leq 0 \\
 & \mathbf{x} \in \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2},
 \end{aligned} \tag{1}$$

where $f, g_1, \dots, g_m: \mathbb{R}^n \rightarrow \mathbb{R}$ are arbitrary nonlinear functions. However, in parts of the chapter, we study a rather restricted model of nonlinear integer programming, where the nonlinearity is confined to the objective function, i.e., the following model:

$$\begin{aligned} & \text{max/min} && f(x_1, \dots, x_n) \\ & \text{subject to} && \mathbf{Ax} \leq \mathbf{b} \\ & && \mathbf{x} \in \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}, \end{aligned} \tag{2}$$

where A is a rational matrix and \mathbf{b} is a rational vector. It is clear that this model is still NP-hard, and that it is much more expressive and much harder to solve than integer linear programs.

We start out with a few fundamental hardness results that help to get a picture of the complexity situation of the problem.

Even in the pure continuous case, nonlinear optimization is known to be hard.

Theorem 1. *Pure continuous polynomial optimization over polytopes ($n_2 = 0$) in varying dimension is NP-hard. Moreover, there does not exist a fully polynomial time approximation scheme (FPTAS) (unless $P = NP$).*

Indeed the max-cut problem can be modeled as minimizing a quadratic form over the cube $[-1, 1]^n$, and thus inapproximability follows from a result by Håstad [65]. On the other hand, pure continuous polynomial optimization problems over polytopes ($n_2 = 0$) can be solved in polynomial time when the dimension n_1 is fixed. This follows from a much more general result on the computational complexity of approximating the solutions to general algebraic formulae over the real numbers by Renegar [111].

However, as soon as we add just two integer variables, we get a hard problem again:

Theorem 2. *The problem of minimizing a degree-4 polynomial over the lattice points of a convex polygon is NP-hard.*

This is based on the NP-completeness of the problem whether there exists a positive integer $x < c$ with $x^2 \equiv a \pmod{b}$; see [53, 41].

We also get hardness results that are much worse than just NP-hardness. The negative solution of Hilbert's tenth problem by Matiyasevich [95, 96], based on earlier work by Davis, Putnam, and Robinson, implies that nonlinear integer programming is *incomputable*, i.e., there cannot exist any general algorithm. (It is clear that for cases where finite bounds for all variables are known, an algorithm trivially exists.) Due to a later strengthening of the negative result by Matiyasevich (published in [76]), there also cannot exist any such general algorithm for even a small fixed number of integer variables; see [41].

Theorem 3. *The problem of minimizing a linear form over polynomial constraints in at most 10 integer variables is not computable by a recursive function.*

Another consequence, as shown by Jeroslow [75], is that even integer quadratic programming is incomputable.

Theorem 4. *The problem of minimizing a linear form over quadratic constraints in integer variables is not computable by a recursive function.*

How do we get positive complexity results and practical methods? One way is to consider subclasses of the objective functions and constraints. First of all, for the problem of *concave minimization* or *convex maximization* which we study in Section 2, we can make use of the property that optimal solutions can always be found on the boundary (actually on the set of vertices) of the convex hull of the feasible solutions. On the other hand, as in the pure continuous case, *convex minimization*, which we address in Section 3), is much easier to handle, from both theoretical and practical viewpoints, than the general case. Next, in Section 4, we study the general case of polynomial optimization, as well as practical approaches for handling the important case of quadratic functions. Finally, in Section 5, we briefly describe the practical approach of global optimization.

For each of these subclasses covered in sections 2–5, we discuss positive complexity results, such as *polynomiality results in fixed dimension*, if available (Sections 2.1, 3.1, 4.1), including some *boundary cases of complexity* in Sections 2.2, 3.2, and 5.2, and discuss *practical algorithms* (Sections 2.3, 3.3, 4.2, 4.3, 5.1).

We end the chapter with conclusions (Section 6), including a table that summarizes the complexity situation of the problem (Table 1).

2 Convex integer maximization

2.1 Fixed dimension

Maximizing a convex function over the integer points in a polytope in fixed dimension can be done in polynomial time. To see this, note that the optimal value is taken on at a vertex of the convex hull of all feasible integer points. But when the dimension is fixed, there is only a polynomial number of vertices, as Cook et al. [38] showed.

Theorem 5. *Let $P = \{ \mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b} \}$ be a rational polyhedron with $A \in \mathbb{Q}^{m \times n}$ and let ϕ be the largest binary encoding size of any of the rows of the system $A\mathbf{x} \leq \mathbf{b}$. Let $P^{\mathbb{I}} = \text{conv}(P \cap \mathbb{Z}^n)$ be the integer hull of P . Then the number of vertices of $P^{\mathbb{I}}$ is at most $2m^n (6n^2 \phi)^{n-1}$.*

Moreover, Hartmann [64] gave an algorithm for enumerating all the vertices, which runs in polynomial time in fixed dimension.

By using Hartmann’s algorithm, we can therefore compute all the vertices of the integer hull $P^{\mathbb{I}}$, evaluate the convex objective function on each of them and pick the best. This simple method already provides a polynomial-time algorithm.

2.2 Boundary cases of complexity

In the past fifteen years algebraic geometry and commutative algebra tools have shown their exciting potential to study problems in integer optimization (see [19, 131] and references therein). The presentation in this section is based on the papers [43, 101].

The first key lemma, extending results of [101] for combinatorial optimization, shows that when a suitable geometric condition holds, it is possible to efficiently reduce the convex integer maximization problem to the solution of polynomially many linear integer programming counterparts. As we will see, this condition holds naturally for a broad class of problems in variable dimension. To state this result, we need the following terminology. A *direction* of an edge (i.e., a one-dimensional face) e of a polyhedron P is any nonzero scalar multiple of $\mathbf{u} - \mathbf{v}$ with \mathbf{u}, \mathbf{v} any two distinct points in e . A set of vectors *covers all edge-directions of P* if it contains a direction of each edge of P . A *linear integer programming oracle* for matrix $A \in \mathbb{Z}^{m \times n}$ and vector $\mathbf{b} \in \mathbb{Z}^m$ is one that, queried on $\mathbf{w} \in \mathbb{Z}^n$, solves the linear integer program $\max\{\mathbf{w}^\top \mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{N}^n\}$, that is, either returns an optimal solution $\mathbf{x} \in \mathbb{N}^n$, or asserts that the program is infeasible, or asserts that the objective function $\mathbf{w}^\top \mathbf{x}$ is unbounded.

Lemma 1. *For any fixed d there is a strongly polynomial oracle-time algorithm that, given any vectors $\mathbf{w}_1, \dots, \mathbf{w}_d \in \mathbb{Z}^n$, matrix $A \in \mathbb{Z}^{m \times n}$ and vector $\mathbf{b} \in \mathbb{Z}^m$ endowed with a linear integer programming oracle, finite set $E \subset \mathbb{Z}^n$ covering all edge-directions of the polyhedron $\text{conv}\{\mathbf{x} \in \mathbb{N}^n : \mathbf{A}\mathbf{x} = \mathbf{b}\}$, and convex functional $c : \mathbb{R}^d \rightarrow \mathbb{R}$ presented by a comparison oracle, solves the convex integer program*

$$\max\{c(\mathbf{w}_1^\top \mathbf{x}, \dots, \mathbf{w}_d^\top \mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{N}^n\}.$$

Here, *solving* the program means that the algorithm either returns an optimal solution $\mathbf{x} \in \mathbb{N}^n$, or asserts the problem is infeasible, or asserts the polyhedron $\{\mathbf{x} \in \mathbb{R}_+^n : \mathbf{A}\mathbf{x} = \mathbf{b}\}$ is unbounded; and *strongly polynomial oracle-time* means that the number of arithmetic operations and calls to the oracles are polynomially bounded in m and n , and the size of the numbers occurring throughout the algorithm is polynomially bounded in the size of the input (which is the number of bits in the binary representation of the entries of $\mathbf{w}_1, \dots, \mathbf{w}_d, A, \mathbf{b}, E$).

Let us outline the main ideas behind the proof to Lemma 1, and let us point out the difficulties that one has to overcome. Given data for a convex integer maximization problem $\max\{c(\mathbf{w}_1^\top \mathbf{x}, \dots, \mathbf{w}_d^\top \mathbf{x}) : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{N}^n\}$, consider the polyhedron $P := \text{conv}\{\mathbf{x} \in \mathbb{N}^n : \mathbf{A}\mathbf{x} = \mathbf{b}\} \subseteq \mathbb{R}^n$ and its linear transformation $Q := \{(\mathbf{w}_1^\top \mathbf{x}, \dots, \mathbf{w}_d^\top \mathbf{x}) : \mathbf{x} \in P\} \subseteq \mathbb{R}^d$. Note that P has typically exponentially many vertices and is not accessible computationally. Note also that, because c is convex, there is an optimal solution \mathbf{x} whose image $(\mathbf{w}_1^\top \mathbf{x}, \dots, \mathbf{w}_d^\top \mathbf{x})$ is a vertex of Q . So an important ingredient in the solution is to construct the vertices of Q . Unfortunately, Q may also have exponentially many vertices even though it lives in a space \mathbb{R}^d of fixed dimension. However, it can be shown that, when the number of *edge-directions* of P is polynomial, the number of vertices of Q is polynomial. Nonetheless, even in this

case, it is not possible to construct these vertices directly, because the number of vertices of P may still be exponential. This difficulty can finally be overcome by using a suitable *zonotope*. See [43, 101] for more details.

Let us now apply Lemma 1 to a broad (in fact, *universal*) class of convex integer maximization problems. Lemma 1 implies that these problems can be solved in polynomial time. Given an $(r+s) \times t$ matrix A , let A_1 be its $r \times t$ sub-matrix consisting of the first r rows and let A_2 be its $s \times t$ sub-matrix consisting of the last s rows. Define the n -fold matrix of A to be the following $(r+ns) \times nt$ matrix,

$$A^{(n)} := (\mathbf{1}_n \otimes A_1) \oplus (I_n \otimes A_2) = \begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}.$$

Note that $A^{(n)}$ depends on r and s : these will be indicated by referring to A as an “ $(r+s) \times t$ matrix.”

We establish the following theorem, which asserts that convex integer maximization over n -fold systems of a fixed matrix A , in variable dimension nt , are solvable in polynomial time.

Theorem 6. *For any fixed positive integer d and fixed $(r+s) \times t$ integer matrix A there is a polynomial oracle-time algorithm that, given n , vectors $\mathbf{w}_1, \dots, \mathbf{w}_d \in \mathbb{Z}^{nt}$ and $\mathbf{b} \in \mathbb{Z}^{r+ns}$, and convex function $c: \mathbb{R}^d \rightarrow \mathbb{R}$ presented by a comparison oracle, solves the convex n -fold integer maximization problem*

$$\max \{c(\mathbf{w}_1^\top \mathbf{x}, \dots, \mathbf{w}_d^\top \mathbf{x}) : A^{(n)} \mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{N}^{nt}\}.$$

The equations defined by an n -fold matrix have the following, perhaps more illuminating, interpretation: splitting the variable vector and the right-hand side vector into components of suitable sizes, $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^n)$ and $\mathbf{b} = (\mathbf{b}^0, \mathbf{b}^1, \dots, \mathbf{b}^n)$, where $\mathbf{b}^0 \in \mathbb{Z}^r$ and $\mathbf{x}^k \in \mathbb{N}^t$ and $\mathbf{b}^k \in \mathbb{Z}^s$ for $k = 1, \dots, n$, the equations become $A_1(\sum_{k=1}^n \mathbf{x}^k) = \mathbf{b}^0$ and $A_2 \mathbf{x}^k = \mathbf{b}^k$ for $k = 1, \dots, n$. Thus, each component \mathbf{x}^k satisfies a system of constraints defined by A_2 with its own right-hand side \mathbf{b}^k , and the sum $\sum_{k=1}^n \mathbf{x}^k$ obeys constraints determined by A_1 and \mathbf{b}^0 restricting the “common resources shared by all components”.

Theorem 6 has various applications, including multiway transportation problems, packing problems, vector partitioning and clustering. For example, we have the following corollary providing the first polynomial time solution of convex 3-way transportation.

Corollary 1. (convex 3-way transportation) *For any fixed d, p, q there is a polynomial oracle-time algorithm that, given n , arrays $\mathbf{w}_1, \dots, \mathbf{w}_d \in \mathbb{Z}^{p \times q \times n}$, $\mathbf{u} \in \mathbb{Z}^{p \times q}$, $\mathbf{v} \in \mathbb{Z}^{p \times n}$, $\mathbf{z} \in \mathbb{Z}^{q \times n}$, and convex $c: \mathbb{R}^d \rightarrow \mathbb{R}$ presented by comparison oracle, solves the convex integer 3-way transportation problem*

$$\max\{c(\mathbf{w}_1^\top \mathbf{x}, \dots, \mathbf{w}_d^\top \mathbf{x}) : \mathbf{x} \in \mathbb{N}^{p \times q \times n}, \sum_i x_{i,j,k} = z_{j,k}, \sum_j x_{i,j,k} = v_{i,k}, \sum_k x_{i,j,k} = u_{i,j}\}.$$

Note that in contrast, if the dimensions of two sides of the tables are variable, say, q and n , then already the standard *linear* integer 3-way transportation problem over such tables is NP-hard, see [44, 45, 46].

In order to prove Theorem 6, we need to recall some definitions. The *Graver basis* of an integer matrix $A \in \mathbb{Z}^{m \times n}$, introduced in [59], is a canonical finite set $\mathcal{G}(A)$ that can be defined as follows. For each of the 2^n orthants \mathcal{O}_j of \mathbb{R}^n let H_j denote the inclusion-minimal Hilbert basis of the pointed rational polyhedral cone $\ker(A) \cap \mathcal{O}_j$. Then the Graver basis $\mathcal{G}(A)$ is defined to be the union $\mathcal{G}(A) = \cup_{j=1}^{2^n} H_j \setminus \{\mathbf{0}\}$ over all these 2^n Hilbert bases. By this definition, the Graver basis $\mathcal{G}(A)$ has a nice representation property: every $\mathbf{z} \in \ker(A) \cap \mathbb{Z}^n$ can be written as a sign-compatible nonnegative integer linear combination $\mathbf{z} = \sum_i \alpha_i \mathbf{g}_i$ of Graver basis elements $\mathbf{g}_i \in \mathcal{G}(A)$. This follows from the simple observation that \mathbf{z} has to belong to some orthant \mathcal{O}_j of \mathbb{R}^n and thus it can be represented as a sign-compatible nonnegative integer linear combination of elements in $H_j \subseteq \mathcal{G}(A)$. For more details on Graver bases and the currently fastest procedure for computing them see [124, 67, 130].

Graver bases have another nice property: They contain all edge directions in the integer hulls within the polytopes $P_{\mathbf{b}} = \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ as \mathbf{b} is varying. We include a direct proof here.

Lemma 2. *For every integer matrix $A \in \mathbb{Z}^{m \times n}$ and every integer vector $\mathbf{b} \in \mathbb{N}^m$, the Graver basis $\mathcal{G}(A)$ of A covers all edge-directions of the polyhedron $\text{conv}\{\mathbf{x} \in \mathbb{N}^n : A\mathbf{x} = \mathbf{b}\}$ defined by A and \mathbf{b} .*

Proof. Consider any edge e of $P := \text{conv}\{\mathbf{x} \in \mathbb{N}^n : A\mathbf{x} = \mathbf{b}\}$ and pick two distinct points $\mathbf{u}, \mathbf{v} \in e \cap \mathbb{N}^n$. Then $\mathbf{g} := \mathbf{u} - \mathbf{v}$ is in $\ker(A) \cap \mathbb{Z}^n \setminus \{\mathbf{0}\}$ and hence, by the representation property of the Graver basis $\mathcal{G}(A)$, \mathbf{g} can be written as a finite *sign-compatible* sum $\mathbf{g} = \sum \mathbf{g}^i$ with $\mathbf{g}^i \in \mathcal{G}(A)$ for all i . Now, we claim that $\mathbf{u} - \mathbf{g}^i \in P$ for all i . To see this, note first that $\mathbf{g}^i \in \mathcal{G}(A) \subseteq \ker(A)$ implies $A\mathbf{g}^i = \mathbf{0}$ and hence $A(\mathbf{u} - \mathbf{g}^i) = A\mathbf{u} = \mathbf{b}$; and second, note that $\mathbf{u} - \mathbf{g}^i \geq \mathbf{0}$: indeed, if $g_j^i \leq 0$ then $u_j - g_j^i \geq u_j \geq 0$; and if $g_j^i > 0$ then sign-compatibility implies $g_j^i \leq g_j$ and therefore $u_j - g_j^i \geq u_j - g_j = v_j \geq 0$.

Now let $\mathbf{w} \in \mathbb{R}^n$ be a linear functional uniquely maximized over P at the edge e . Then for all i , as just proved, $\mathbf{u} - \mathbf{g}^i \in P$ and hence $\mathbf{w}^\top \mathbf{g}^i \geq 0$. But $\sum \mathbf{w}^\top \mathbf{g}^i = \mathbf{w}^\top \mathbf{g} = \mathbf{w}^\top \mathbf{u} - \mathbf{w}^\top \mathbf{v} = 0$, implying that in fact, for all i , we have $\mathbf{w}^\top \mathbf{g}^i = 0$ and therefore $\mathbf{u} - \mathbf{g}^i \in e$. This implies that each \mathbf{g}^i is a direction of the edge e (in fact, moreover, all \mathbf{g}^i are the same, so \mathbf{g} is a multiple of some Graver basis element).

In Section 3.2, we show that for fixed matrix A , the size of the Graver basis of $A^{(n)}$ increases only polynomially in n implying Theorem 14 that states that certain integer convex n -fold *minimization* problems can be solved in polynomial time when the matrix A is kept fixed. As a special case, this implies that the integer *linear* n -fold *maximization* problem can be solved in polynomial time when the matrix A is kept fixed. Finally, combining these results with Lemmas 1 and 2, we can now prove Theorem 6.

Proof (of Theorem 6). The algorithm underlying Proposition 14 provides a polynomial time realization of a linear integer programming oracle for $A^{(n)}$ and \mathbf{b} . The algorithm underlying Proposition 2 allows to compute the Graver basis $\mathcal{G}(A^{(n)})$ in time which is polynomial in the input. By Lemma 2, this set $E := \mathcal{G}(A^{(n)})$ covers all edge-directions of the polyhedron $\text{conv}\{\mathbf{x} \in \mathbb{N}^{nt} : A^{(n)}\mathbf{x} = \mathbf{b}\}$ underlying the convex integer program. Thus, the hypothesis of Lemma 1 is satisfied and hence the algorithm underlying Lemma 1 can be used to solve the convex integer maximization problem in polynomial time.

2.3 Reduction to linear integer programming

In this section it is our goal to develop a basic understanding about when a discrete polynomial programming problem can be tackled with classical linear integer optimization techniques. We begin to study polyhedra related to polynomial integer programming. The presented approach applies to problems of the kind

$$\max\{f(\mathbf{x}) : A\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{Z}_+^n\}$$

with convex polynomial function f , as well as to models such as

$$\max\{\mathbf{c}^\top \mathbf{x} : \mathbf{x} \in K_I\}, \quad (3)$$

where K_I denotes the set of all integer points of a compact *basic-closed* semi-algebraic set K described by a finite number of polynomial inequalities, i. e.,

$$K = \{\mathbf{x} \in \mathbb{R}^n : p_i(\mathbf{x}) \leq 0, i \in M, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}.$$

We assume that $p_i \in \mathbb{Z}[\mathbf{x}] := \mathbb{Z}[x_1, \dots, x_n]$, for all $i \in M = \{1, \dots, m\}$, and $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^n$.

One natural idea is to derive a linear description of the convex hull of K_I . Unfortunately, $\text{conv}(K_I)$ might contain interior integer points that are not elements of K , see Figure 1. On the other hand, if a description of $\text{conv}(K_I)$ is at hand, then the mathematical optimization problem of solving (3) can be reduced to optimizing the linear objective function $\mathbf{c}^\top \mathbf{x}$ over the polyhedron $\text{conv}(K_I)$. This is our first topic of interest. In what follows we denote for a set D the projection of D to a set of variables x by the symbol D_x .

Definition 1. For polynomials $p_1, \dots, p_m : \mathbb{Z}^n \rightarrow \mathbb{Z}$ we define the polyhedron associated with the vector $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_m(\mathbf{x}))$ of polynomials as

$$P_{\mathbf{p}} = \text{conv}\left(\{(\mathbf{x}, \mathbf{p}(\mathbf{x})) \in \mathbb{R}^{n+m} \mid \mathbf{x} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n\}\right).$$

The significance of the results below is that they allow us to reformulate the non-linear integer optimization problem $\max\{f(\mathbf{x}) : A\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{Z}_+^n\}$ as the optimization problem $\min\{\pi : A\mathbf{x} = \mathbf{b}, f(\mathbf{x}) \leq \pi, \mathbf{x} \in \mathbb{Z}_+^n\}$. This in turn has the same objective

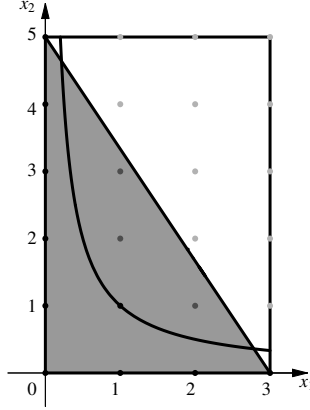


Fig. 1 Let $K = \{\mathbf{x} \in \mathbb{R}^2 \mid x_1 x_2 - 1 \leq 0, 0 \leq x_1 \leq 3, 0 \leq x_2 \leq 5\}$. The point $(1, 2)$ is contained in $\text{conv}(K_I)$. But it violates the constraint $x_1 x_2 - 1 \leq 0$.

function value as the linear integer program: $\min\{\boldsymbol{\pi} : \mathbf{A}\boldsymbol{\pi} = \mathbf{b}, (\mathbf{x}, \boldsymbol{\pi}) \in P_f, \mathbf{x} \in \mathbb{Z}_+^n\}$. In this situation, an H-description or V-description of the polyhedron P_f is sufficient to reduce the original nonlinear optimization problem to a linear integer problem.

Proposition 1. For a vector of polynomials $\mathbf{p} \in \mathbb{Z}[\mathbf{x}]^m$, let

$$K_I = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{p}(\mathbf{x}) \leq \mathbf{0}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}.$$

Then,

$$\text{conv}(K_I) \subseteq (P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) \in \mathbb{R}^{n+m} \mid \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}}. \quad (4)$$

Proof. It suffices to show that $K_I \subseteq (P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) \in \mathbb{R}^{n+m} : \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}}$. Let us consider $\mathbf{x} \in K_I \subseteq [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n$. By definition, $(\mathbf{x}, \mathbf{p}(\mathbf{x})) \in P_{\mathbf{p}}$. Moreover, we have $\mathbf{p}(\mathbf{x}) \leq \mathbf{0}$. This implies $(\mathbf{x}, \mathbf{p}(\mathbf{x})) \in (P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) \in \mathbb{R}^{n+m} : \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}}$, and thus,

$$\mathbf{x} \in (P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) \in \mathbb{R}^{n+m} : \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}}.$$

Note that even in the univariate case, equality in Formula (4) of Proposition 1 does not always hold. For instance, if $K_I = \{x \in \mathbb{Z} : x^2 - 5 \leq 0, -3 \leq x \leq 5\}$, then

$$\text{conv}(K_I) = [-2, 2] \neq [-2.2, 2.2] \subseteq (P_{x^2} \cap \{(x, \boldsymbol{\pi}) \in \mathbb{R}^2 : \boldsymbol{\pi} - 5 \leq 0\})_{\mathbf{x}}.$$

Although even in very simple cases the sets $\text{conv}(K_I)$ and $(P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) : \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}}$ differ, it is still possible that the integer points in K_I and $(P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) : \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}}$ are equal. In our example with $K_I = \{x \in \mathbb{Z} : x^2 - 5 \leq 0, -3 \leq x \leq 5\}$, we then obtain,

$$K_I = \{-2, -1, 0, 1, 2\} = [-2.2, 2.2] \cap \mathbb{Z}.$$

Of course, for any $\mathbf{p} \in \mathbb{Z}[\mathbf{x}]^m$ we have that

$$K_I = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{p}(\mathbf{x}) \leq \mathbf{0}, \mathbf{1} \leq \mathbf{x} \leq \mathbf{u}\} \subseteq (P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) : \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}} \cap \mathbb{Z}^n. \quad (5)$$

The key question here is when equality holds in Formula (5).

Theorem 7. *Let $\mathbf{p} \in \mathbb{Z}[\mathbf{x}]^m$ and $K_I = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{p}(\mathbf{x}) \leq \mathbf{0}, \mathbf{1} \leq \mathbf{x} \leq \mathbf{u}\}$. Then,*

$$K_I = (P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) : \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}} \cap \mathbb{Z}^n$$

holds if every polynomial $p' \in \{p_i : i = 1, \dots, m\}$ satisfies the following condition

$$p' \left(\sum_{\mathbf{k}} \lambda_{\mathbf{k}} \mathbf{k} \right) - \sum_{\mathbf{k}} \lambda_{\mathbf{k}} p'(\mathbf{k}) < 1, \quad (6)$$

for all $\lambda_{\mathbf{k}} \geq 0$, $\mathbf{k} \in [\mathbf{1}, \mathbf{u}] \cap \mathbb{Z}^n$, $\sum_{\mathbf{k}} \lambda_{\mathbf{k}} = 1$ and $\sum_{\mathbf{k}} \lambda_{\mathbf{k}} \mathbf{k} \in \mathbb{Z}^n$.

Proof. Using Formula (5), we have to show that $(P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) : \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}} \cap \mathbb{Z}^n \subseteq K_I$ if all p_i , $i \in \{1, \dots, m\}$, satisfy Equation (6). Let $\mathbf{x} \in (P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) \in \mathbb{R}^{m+n} : \boldsymbol{\pi} \leq \mathbf{0}\})_{\mathbf{x}} \cap \mathbb{Z}^n$. Then, there exists a $\boldsymbol{\pi} \in \mathbb{R}^m$ such that

$$(\mathbf{x}, \boldsymbol{\pi}) \in P_{\mathbf{p}} \cap \{(\mathbf{x}, \boldsymbol{\pi}) \in \mathbb{R}^{n+m} : \boldsymbol{\pi} \leq \mathbf{0}\}.$$

By definition, $\boldsymbol{\pi} \leq \mathbf{0}$. Furthermore, there must exist nonnegative real numbers $\lambda_{\mathbf{k}} \geq 0$, $\mathbf{k} \in [\mathbf{1}, \mathbf{u}] \cap \mathbb{Z}^n$, such that $\sum_{\mathbf{k}} \lambda_{\mathbf{k}} = 1$ and $(\mathbf{x}, \boldsymbol{\pi}) = \sum_{\mathbf{k}} \lambda_{\mathbf{k}} (\mathbf{k}, \mathbf{p}(\mathbf{k}))$. Suppose that there exists an index i_0 such that the inequality $p_{i_0}(\mathbf{x}) \leq 0$ is violated. The fact that $p_{i_0} \in \mathbb{Z}[\mathbf{x}]$ and $\mathbf{x} \in \mathbb{Z}^n$, implies that $p_{i_0}(\mathbf{x}) \geq 1$. Thus, we obtain

$$\sum_{\mathbf{k}} \lambda_{\mathbf{k}} p_{i_0}(\mathbf{k}) = \pi_{i_0} \leq 0 < 1 \leq p_{i_0}(\mathbf{x}) = p_{i_0} \left(\sum_{\mathbf{k}} \lambda_{\mathbf{k}} \mathbf{k} \right),$$

or equivalently, $p_{i_0} \left(\sum_{\mathbf{k}} \lambda_{\mathbf{k}} \mathbf{k} \right) - \sum_{\mathbf{k}} \lambda_{\mathbf{k}} p_{i_0}(\mathbf{k}) \geq 1$. Because this is a contradiction to our assumption, we have that $p_i(\mathbf{x}) \leq 0$ for all i . Hence, $\mathbf{x} \in K_I$. This proves the claim.

The next example illustrates the statement of Theorem 7.

Example 1. Let $p \in \mathbb{Z}[\mathbf{x}]$, $\mathbf{x} \mapsto p(\mathbf{x}) := 3x_1^2 + 2x_2^2 - 19$. We consider the semi-algebraic set

$$K = \{\mathbf{x} \in \mathbb{R}^2 \mid p(\mathbf{x}) \leq 0, 0 \leq x_1 \leq 3, 0 \leq x_2 \leq 3\} \quad \text{and} \quad K_I = K \cap \mathbb{Z}^2.$$

It turns out that the convex hull of K_I is described by $x_1 + x_2 \leq 3$, $0 \leq x_1 \leq 2$ and $0 \leq x_2$. Notice that the polynomial p is convex. This condition ensures that p satisfies Equation (6) of Theorem 7. We obtain in this case

$$\begin{aligned} (P_p \cap \{(\mathbf{x}, \boldsymbol{\pi}) \in \mathbb{R}^3 : \boldsymbol{\pi} \leq 0\})_{\mathbf{x}} = \{ \mathbf{x} \in \mathbb{R}^2 : & 9x_1 + 6x_2 \leq 29, \\ & 3x_1 + 10x_2 \leq 31, \\ & 9x_1 + 10x_2 \leq 37, \\ & 15x_1 + 2x_2 \leq 37, \\ & 15x_1 + 6x_2 \leq 41, \\ & -x_1 \leq 0, 0 \leq x_2 \leq 3 \}. \end{aligned}$$

The sets K , $\text{conv}(K_I)$ and $(P_p \cap \{(\mathbf{x}, \boldsymbol{\pi}) \in \mathbb{R}^3 : \boldsymbol{\pi} \leq 0\})_{\mathbf{x}}$ are illustrated in Figure 2. Note that here $K_I = (P_p \cap \{(\mathbf{x}, \boldsymbol{\pi}) : \boldsymbol{\pi} \leq 0\})_{\mathbf{x}} \cap \mathbb{Z}^2$.

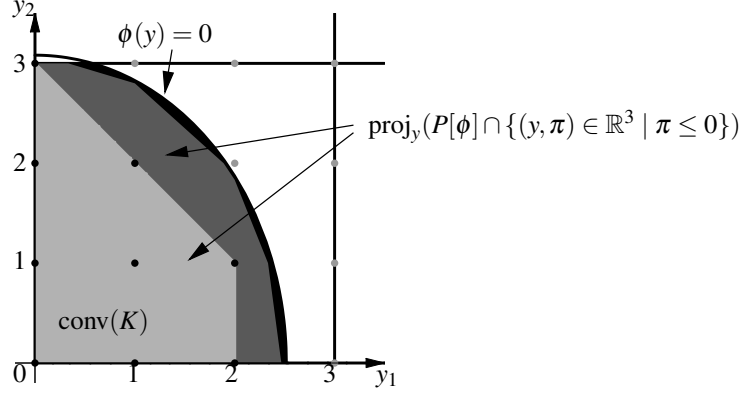


Fig. 2 Illustration of Theorem 7 in Example 1.

Next we introduce a large class of nonlinear functions for which one can ensure that equality holds in Formula (5).

Definition 2. (*Integer-Convex Polynomials*)

A polynomial $f \in \mathbb{Z}[\mathbf{x}]$ is said to be *integer-convex* on $[\mathbf{l}, \mathbf{u}] \subseteq \mathbb{R}^n$, if for any finite subset of nonnegative real numbers $\{\lambda_{\mathbf{k}}\}_{\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n} \subseteq \mathbb{R}_+$ with $\sum_{\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n} \lambda_{\mathbf{k}} = 1$ and $\sum_{\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n} \lambda_{\mathbf{k}} \mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n$, the following inequality holds:

$$f\left(\sum_{\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n} \lambda_{\mathbf{k}} \mathbf{k}\right) \leq \sum_{\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n} \lambda_{\mathbf{k}} f(\mathbf{k}). \quad (7)$$

If (7) holds strictly for all $\{\lambda_{\mathbf{k}}\}_{\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n} \subseteq \mathbb{R}_+$, and $\mathbf{x} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n$ such that $\sum_{\mathbf{k}} \lambda_{\mathbf{k}} = 1$, $\mathbf{x} = \sum_{\mathbf{k}} \lambda_{\mathbf{k}} \mathbf{k}$, and $\lambda_{\mathbf{x}} < 1$, then the polynomial f is called *strictly integer-convex* on $[\mathbf{l}, \mathbf{u}]$.

By definition, a (strictly) convex polynomial is (strictly) integer-convex. Conversely, a (strictly) integer-convex polynomial is not necessarily (strictly) convex. Figure 3 gives an example.

Integer convexity is inherited under taking conic combinations and applying a composition rule.

- (a) For any finite number of integer-convex polynomials $f_s \in \mathbb{Z}[\mathbf{x}]$, $s \in \{1, \dots, t\}$, on $[\mathbf{l}, \mathbf{u}]$, and nonnegative integers $a_s \in \mathbb{Z}_+$, $s \in \{1, \dots, t\}$, the polynomial $f \in \mathbb{Z}[\mathbf{x}]$, $\mathbf{x} \mapsto f(\mathbf{x}) := \sum_{s=1}^t a_s f_s(\mathbf{x})$, is integer-convex on $[\mathbf{l}, \mathbf{u}]$.

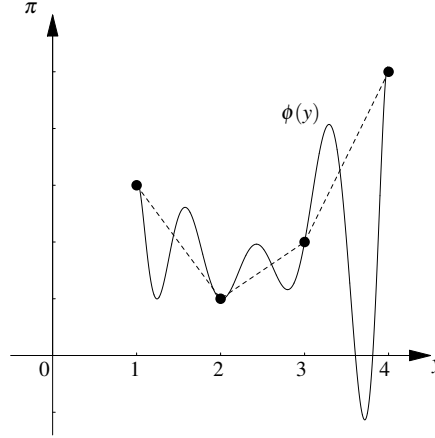


Fig. 3 The graph of an integer-convex polynomial p on $[1, 4]$.

- (b) Let $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^n$ and let $h \in \mathbb{Z}[\mathbf{x}]$, $\mathbf{x} \mapsto h(\mathbf{x}) := \mathbf{c}^\top \mathbf{x} + \gamma$, be a linear function. Setting $W = \{\mathbf{c}^\top \mathbf{x} + \gamma : \mathbf{x} \in [\mathbf{l}, \mathbf{u}]\}$, for every integer-convex univariate polynomial $q \in \mathbb{Z}[w]$, the function $p \in \mathbb{Z}[\mathbf{x}]$, $\mathbf{x} \mapsto p(\mathbf{x}) := q(h(\mathbf{x}))$ is integer-convex on $[\mathbf{l}, \mathbf{u}]$.

Indeed, integer-convex polynomial functions capture a lot of combinatorial structure. In particular, we can characterize the set of all vertices in an associated polyhedron. Most importantly, if f is integer-convex on $[\mathbf{l}, \mathbf{u}]$, then this ensures that for any integer point $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$ the value $f(\mathbf{x})$ is not underestimated by all $\pi \in \mathbb{R}$ with $(\mathbf{x}, \pi) \in P_f$, where P_f is the polytope associated with the graph of the polynomial $f \in \mathbb{Z}[\mathbf{x}]$.

Theorem 8. For a polynomial $f \in \mathbb{Z}[\mathbf{x}]$ and $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^n$, $\mathbf{l} + \mathbf{1} < \mathbf{u}$, let

$$P_f = \text{conv} \left(\left\{ (\mathbf{x}, f(\mathbf{x})) \in \mathbb{Z}^{n+1} \mid \mathbf{x} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n \right\} \right).$$

Then, f is integer-convex on $[\mathbf{l}, \mathbf{u}]$ is equivalent to the condition that for all $(\mathbf{x}, \pi) \in P_f$, $\mathbf{x} \in \mathbb{Z}^n$ we have that $f(\mathbf{x}) \leq \pi$. Moreover, if f is strictly integer-convex on $[\mathbf{l}, \mathbf{u}]$, then for every $\mathbf{x} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n$, the point $(\mathbf{x}, f(\mathbf{x}))$ is a vertex of P_f .

Proof. First let us assume that f is integer-convex on $[\mathbf{l}, \mathbf{u}]$. Let $(\mathbf{x}, \pi) \in P_f$ such that $\mathbf{x} \in \mathbb{Z}^n$. Then, there exist nonnegative real numbers $\{\lambda_{\mathbf{k}}\}_{\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n} \subseteq \mathbb{R}_+$, $\sum_{\mathbf{k}} \lambda_{\mathbf{k}} = 1$, such that $(\mathbf{x}, \pi) = \sum_{\mathbf{k}} \lambda_{\mathbf{k}} (\mathbf{k}, f(\mathbf{k}))$. It follows that

$$f(\mathbf{x}) = f \left(\sum_{\mathbf{k}} \lambda_{\mathbf{k}} \mathbf{k} \right) \leq \sum_{\mathbf{k}} \lambda_{\mathbf{k}} f(\mathbf{k}) = \pi.$$

Next we assume that f is not integer-convex on $[\mathbf{l}, \mathbf{u}]$. Then, there exists a subset of nonnegative real numbers $\{\lambda_{\mathbf{k}}\}_{\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n} \subseteq \mathbb{R}_+$ with $\sum_{\mathbf{k}} \lambda_{\mathbf{k}} = 1$ such that

$$\mathbf{x} := \sum_{\mathbf{k}} \lambda_{\mathbf{k}} \mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n \text{ and } \pi := \sum_{\mathbf{k}} \lambda_{\mathbf{k}} f(\mathbf{k}) < f\left(\sum_{\mathbf{k}} \lambda_{\mathbf{k}} \mathbf{k}\right) = f(\mathbf{x}).$$

But then, $(\mathbf{x}, \pi) = \sum_{\mathbf{k}} \lambda_{\mathbf{k}} (\mathbf{k}, f(\mathbf{k})) \in P_f$ violates the inequality $f(\mathbf{x}) \leq \pi$. This is a contradiction to the assumption.

If f is strictly integer-convex on $[\mathbf{l}, \mathbf{u}]$, then for each $\mathbf{x} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n$, we have that

$$f(\mathbf{x}) < \sum_{\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n \setminus \{\mathbf{x}\}} \lambda_{\mathbf{k}} f(\mathbf{k}),$$

for all $\lambda_{\mathbf{k}} \in \mathbb{R}_+$, $\mathbf{k} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n \setminus \{\mathbf{x}\}$, with $\mathbf{x} = \sum_{\mathbf{k}} \lambda_{\mathbf{k}} \mathbf{k}$ and $\sum_{\mathbf{k}} \lambda_{\mathbf{k}} = 1$. Thus, every point $(\mathbf{x}, f(\mathbf{x}))$, $\mathbf{x} \in [\mathbf{l}, \mathbf{u}] \cap \mathbb{Z}^n$, is a vertex of P_f .

3 Convex integer minimization

The complexity of the case of convex integer minimization is set apart from the general case of integer polynomial optimization by the existence of bounding results for the coordinates of optimal solutions. Once a finite bound can be computed, it is clear that an algorithm for minimization exists. Thus the fundamental incomputability result for integer polynomial optimization (Theorem 3) does not apply to the case of convex integer minimization.

The first bounds for the optimal integer solutions to convex minimization problems were proved by [78, 125]. We present the sharpened bound that was obtained by [11, 10] for the more general case of quasi-convex polynomials. This bound is a consequence of an efficient theory of quantifier elimination over the reals; see [110].

Theorem 9. *Let $f, g_1, \dots, g_m \in \mathbb{Z}[x_1, \dots, x_n]$ be quasi-convex polynomials of degree at most $d \geq 2$, whose coefficients have a binary encoding length of at most ℓ . Let*

$$F = \{ \mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0 \text{ for } i = 1, \dots, m \}$$

be the (continuous) feasible region. If the integer minimization problem $\min\{f(\mathbf{x}) : \mathbf{x} \in F \cap \mathbb{Z}^n\}$ is bounded, there exists a radius $R \in \mathbb{Z}_+$ of binary encoding length at most $(md)^{O(n)} \ell$ such that

$$\min\{f(\mathbf{x}) : \mathbf{x} \in F \cap \mathbb{Z}^n\} = \min\{f(\mathbf{x}) : \mathbf{x} \in F \cap \mathbb{Z}^n, \|\mathbf{x}\| \leq R\}.$$

3.1 Fixed dimension

In fixed dimension, the problem of convex integer minimization can be solved using variants of Lenstra's algorithm [87] for integer programming. Indeed, when the dimension n is fixed, the bound R given by Theorem 9 has a binary encoding size that is bounded polynomially by the input data. Thus, a Lenstra-type algorithm can

be started with a “small” (polynomial-size) initial outer ellipsoid that includes a bounded part of the feasible region containing an optimal integer solution.

The first algorithm of this kind for convex integer minimization was announced by Khachiyan [78]. In the following we present the variant of Lenstra’s algorithm due to Heinz [66], which seems to yield the best known complexity bound for the problem. The complexity result is the following.

Theorem 10. *Let $f, g_1, \dots, g_m \in \mathbb{Z}[x_1, \dots, x_n]$ be quasi-convex polynomials of degree at most $d \geq 2$, whose coefficients have a binary encoding length of at most ℓ . There exists an algorithm running in time $m\ell^{O(1)}d^{O(n)}2^{O(n^3)}$ that computes a minimizer $\mathbf{x}^* \in \mathbb{Z}^n$ of the problem (1) or reports that no minimizer exists. If the algorithm outputs a minimizer \mathbf{x}^* , its binary encoding size is $\ell d^{O(n)}$.*

A complexity result of greater generality was presented by Khachiyan and Porkolab [79]. It covers the case of minimization of convex polynomials over the integer points in convex semialgebraic sets given by *arbitrary* (not necessarily quasi-convex) polynomials.

Theorem 11. *Let $Y \subseteq \mathbb{R}^k$ be a convex set given by*

$$Y = \{ \mathbf{y} \in \mathbb{R}^k : \mathbf{Q}_1 \mathbf{x}^1 \in \mathbb{R}^{n_1} : \dots : \mathbf{Q}_\omega \mathbf{x}^\omega \in \mathbb{R}^{n_\omega} : P(\mathbf{y}, \mathbf{x}^1, \dots, \mathbf{x}^\omega) \}$$

with quantifiers $\mathbf{Q}_i \in \{\exists, \forall\}$, where P is a Boolean combination of polynomial inequalities

$$g_i(\mathbf{y}, \mathbf{x}^1, \dots, \mathbf{x}^\omega) \leq 0, \quad i = 1, \dots, m$$

with degrees at most $d \geq 2$ and coefficients of binary encoding size at most ℓ . There exists an algorithm for solving the problem $\min\{y_k : \mathbf{y} \in Y \cap \mathbb{Z}^k\}$ in time $\ell^{O(1)}(md)^{O(k^4)}\prod_{i=1}^\omega O(n_i)$.

When the dimension $k + \sum_{i=1}^\omega n_i$ is fixed, the algorithm runs in polynomial time. For the case of convex minimization where the feasible region is described by convex polynomials, the complexity bound of Theorem 11, however, translates to $\ell^{O(1)}m^{O(n^2)}d^{O(n^4)}$, which is worse than the bound of Theorem 10 [66].

In the remainder of this subsection, we describe the ingredients of the variant of Lenstra’s algorithm due to Heinz. The algorithm starts out by “rounding” the feasible region, by applying the shallow-cut ellipsoid method to find proportional inscribed and circumscribed ellipsoids. It is well-known [60] that the shallow-cut ellipsoid method only needs an initial circumscribed ellipsoid that is “small enough” (of polynomial binary encoding size – this follows from Theorem 9) and an implementation of a *shallow separation oracle*, which we describe below.

For a positive-definite matrix A we denote by $\mathcal{E}(A, \hat{\mathbf{x}})$ the ellipsoid $\{\mathbf{x} \in \mathbb{R}^n : (\mathbf{x} - \hat{\mathbf{x}})^\top A (\mathbf{x} - \hat{\mathbf{x}}) \leq 1\}$.

Lemma 3 (Shallow separation oracle). *Let $g_0, \dots, g_{m+1} \in \mathbb{Z}[\mathbf{x}]$ be quasi-convex polynomials of degree at most d , the binary encoding sizes of whose coefficients are at most r . Let the (continuous) feasible region $F = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) < 0\}$ be contained*

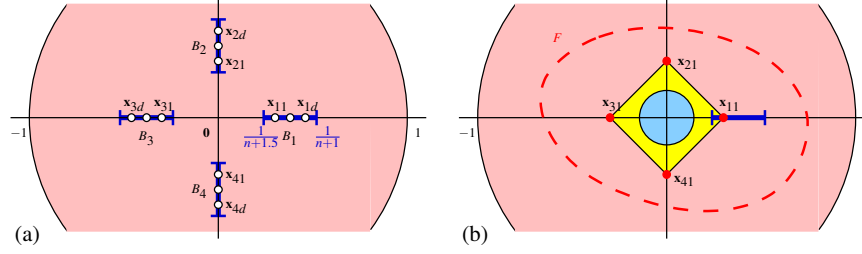


Fig. 4 The implementation of the shallow separation oracle. **(a)** Test points \mathbf{x}_{ij} in the circumscribed ball $\mathcal{E}(1, \mathbf{0})$. **(b)** Case I: All test points \mathbf{x}_{ij} are (continuously) feasible; so their convex hull (a cross-polytope) and its inscribed ball $\mathcal{E}((n+1)^{-3}, \mathbf{0})$ are contained in the (continuous) feasible region F .

in the ellipsoid $\mathcal{E}(A, \hat{\mathbf{x}})$, where A and $\hat{\mathbf{x}}$ have binary encoding size at most ℓ . There exists an algorithm with running time $m(\ln r)^{O(1)} d^{O(n)}$ that outputs

(a) “true” if

$$\mathcal{E}((n+1)^{-3}A, \hat{\mathbf{x}}) \subseteq F \subseteq \mathcal{E}(A, \hat{\mathbf{x}}); \quad (8)$$

(b) otherwise, a vector $\mathbf{c} \in \mathbb{Q}^n \setminus \{\mathbf{0}\}$ of binary encoding length $(l+r)(dn)^{O(1)}$ with

$$F \subseteq \mathcal{E}(A, \hat{\mathbf{x}}) \cap \{ \mathbf{x} \in \mathbb{R}^n : \mathbf{c}^\top (\mathbf{x} - \hat{\mathbf{x}}) \leq \frac{1}{n+1} (\mathbf{c}^\top A \mathbf{c})^{1/2} \}. \quad (9)$$

Proof. We give a simplified sketch of the proof, without hard complexity estimates. By applying an affine transformation to $F \subseteq \mathcal{E}(A, \hat{\mathbf{x}})$, we can assume that F is contained in the unit ball $\mathcal{E}(1, \mathbf{0})$. Let us denote as usual by $\mathbf{e}_1, \dots, \mathbf{e}_n$ the unit vectors and by $\mathbf{e}_{n+1}, \dots, \mathbf{e}_{2n}$ their negatives. The algorithm first constructs numbers $\lambda_{i1}, \dots, \lambda_{id} > 0$ with

$$\frac{1}{n + \frac{3}{2}} < \lambda_{i1} < \dots < \lambda_{id} < \frac{1}{n+1} \quad (10)$$

and the corresponding point sets $B_i = \{ \mathbf{x}_{ij} := \lambda_{ij} \mathbf{e}_i : j = 1, \dots, d \}$; see Figure 4(a). The choice of the bounds (10) for λ_{ij} will ensure that we either find a large enough inscribed ball for (a) or a deep enough cut for (b). Then the algorithm determines the (continuous) feasibility of the center $\mathbf{0}$ and the $2n$ innermost points $\mathbf{x}_{i,1}$.

Case I. If $\mathbf{x}_{i,1} \in F$ for $i = 1, \dots, 2n$, then the cross-polytope $\text{conv}\{ \mathbf{x}_{i,1} : i = 1, \dots, 2n \}$ is contained in F ; see Figure 4(b). An easy calculation shows that the ball $\mathcal{E}((n+1)^{-3}, \mathbf{0})$ is contained in the cross-polytope and thus in F ; see Figure 4. Hence the condition in (a) is satisfied and the algorithm outputs “true”.

Case II. We now discuss the case when the center $\mathbf{0}$ violates a polynomial inequality $g_0(\mathbf{x}) < 0$ (say). Let $F_0 = \{ \mathbf{x} \in \mathbb{R}^n : g_0(\mathbf{x}) < 0 \} \supseteq F$. Due to convexity of F_0 , for all $i = 1, \dots, n$, one set of each pair $B_i \cap F_0$ and $B_{n+i} \cap F_0$ must be empty; see Figure 5(a). Without loss of generality, let us assume $B_{n+i} \cap F_0 = \emptyset$ for all i . We can determine whether a n -variate polynomial function of known maximum degree d is constant by evaluating it on $(d+1)^n$ suitable points (this is a consequence of the Fundamental Theorem of Algebra). For our case of quasi-convex polynomials, this

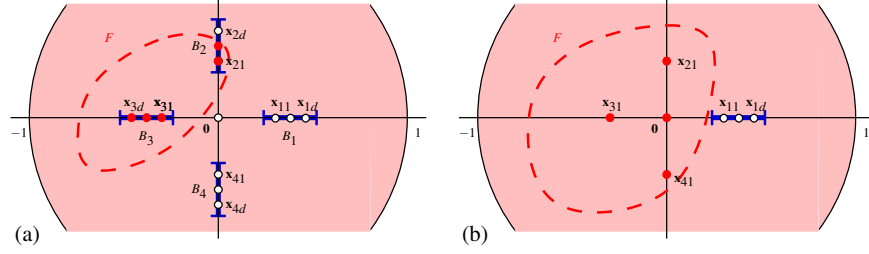


Fig. 5 The implementation of the shallow separation oracle. **(a)** Case II: The center $\mathbf{0}$ violates a polynomial inequality $g_0(\mathbf{x}) < 0$ (say). Due to convexity, for all $i = 1, \dots, n$, one set of each pair $B_i \cap F$ and $B_{n+i} \cap F$ must be empty. **(b)** Case III: A test point \mathbf{x}_{k1} is infeasible, as it violates an inequality $g_0(\mathbf{x}) < 0$ (say). However, the center $\mathbf{0}$ is feasible at least for this inequality.

can be improved; indeed, it suffices to test whether the gradient ∇g_0 vanishes on the nd points in the set $B_1 \cup \dots \cup B_n$. If it does, we know that g_0 is constant, thus $F = \emptyset$, and so can we return an arbitrary vector \mathbf{c} . Otherwise, there is a point $\mathbf{x}_{ij} \in B_i$ with $\mathbf{c} := \nabla g_0(\mathbf{x}_{ij}) \neq \mathbf{0}$; we return this vector as the desired normal vector of a shallow cut. Due to the choice of λ_{ij} as a number smaller than $\frac{1}{n+1}$, the cut is deep enough into the ellipsoid $\mathcal{E}(A, \hat{\mathbf{x}})$, so that (9) holds.

Case III. The remaining case to discuss is when $\mathbf{0} \in F$ but there exists a $k \in \{1, \dots, 2n\}$ with $\mathbf{x}_{k,1} \notin F$. Without loss of generality, let $k = 1$, and let $\mathbf{x}_{1,1}$ violate the polynomial inequality $g_0(\mathbf{x}) < 0$, i.e., $g_0(\mathbf{x}_{1,1}) \geq 0$; see Figure 5 (b). We consider the univariate polynomial $\phi(\lambda) = g_0(\lambda \mathbf{e}_1)$. We have $\phi(0) = g_0(\mathbf{0}) < 0$ and $\phi(\lambda_{1,1}) \geq 0$, so ϕ is not constant. Because ϕ has degree at most d , its derivative ϕ' has degree at most $d - 1$, so ϕ' has at most $d - 1$ roots. Thus, for at least one of the d different values $\lambda_{1,1}, \dots, \lambda_{1,d}$, say $\lambda_{1,j}$, we must have $\phi'(\lambda_{1,j}) \neq 0$. This implies that $\mathbf{c} := \nabla g_0(\mathbf{x}_{1,j}) \neq \mathbf{0}$. By convexity, we have $\mathbf{x}_{1,j} \notin F$, so we can use \mathbf{c} as the normal vector of a shallow cut.

By using this oracle in the shallow-cut ellipsoid method, one obtains the following result.

Corollary 2. *Let $g_0, \dots, g_m \in \mathbb{Z}[\mathbf{x}]$ be quasi-convex polynomials of degree at most $d \geq 2$. Let the (continuous) feasible region $F = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0\}$ be contained in the ellipsoid $\mathcal{E}(A_0, \mathbf{0})$, given by the positive-definite matrix $A_0 \in \mathbb{Q}^{n \times n}$. Let $\varepsilon \in \mathbb{Q}_{>0}$ be given. Let the entries of A_0 and the coefficients of all monomials of g_0, \dots, g_m have binary encoding size at most ℓ .*

There exists an algorithm with running time $m(\ell n)^{O(1)} d^{O(n)}$ that computes a positive-definite matrix $A \in \mathbb{Q}^{n \times n}$ and a point $\hat{\mathbf{x}} \in \mathbb{Q}^n$ with

- (a) *either $\mathcal{E}((n+1)^{-3}A, \hat{\mathbf{x}}) \subseteq F \subseteq \mathcal{E}(A, \hat{\mathbf{x}})$*
- (b) *or $F \subseteq \mathcal{E}(A, \hat{\mathbf{x}})$ and $\text{vol } \mathcal{E}(A, \hat{\mathbf{x}}) < \varepsilon$.*

Finally, there is a lower bound for the volume of a continuous feasible region F that can contain an integer point.

Lemma 4. *Under the assumptions of Theorem 2, if $F \cap \mathbb{Z}^n \neq \emptyset$, then there exists an $\varepsilon \in \mathbb{Q}_{>0}$ of binary encoding size $\ell(dn)^{O(1)}$ with $\text{vol} F > \varepsilon$.*

On the basis of these results, one obtains a Lenstra-type algorithm for the decision version of the convex integer minimization problem with the desired complexity. By applying binary search, the optimization problem can be solved, which provides a proof of Theorem 10.

3.2 Boundary cases of complexity

In this section we present an optimality certificate for problems of the form

$$\min\{f(\mathbf{x}) : A\mathbf{x} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n\},$$

where $A \in \mathbb{Z}^{d \times n}$, $\mathbf{b} \in \mathbb{Z}^d$, $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^n$, and where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a separable convex function, that is, $f(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$ with convex functions $f_i : \mathbb{R} \rightarrow \mathbb{R}$, $i = 1, \dots, n$. This certificate then immediately leads us to a oracle-polynomial time algorithm to solve the separable convex integer minimization problem at hand. Applied to separable convex n -fold integer minimization problems, this gives a polynomial time algorithm for their solution [69].

For the construction of the optimality certificate, we exploit a nice super-additivity property of separable convex functions.

Lemma 5. *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a separable convex function and let $\mathbf{h}_1, \dots, \mathbf{h}_k \in \mathbb{R}^n$ belong to a common orthant of \mathbb{R}^n , that is, they all have the same sign pattern from $\{\geq 0, \leq 0\}^n$. Then, for any $\mathbf{x} \in \mathbb{R}^n$ we have*

$$f\left(\mathbf{x} + \sum_{i=1}^k \mathbf{h}_i\right) - f(\mathbf{x}) \geq \sum_{i=1}^k [f(\mathbf{x} + \mathbf{h}_i) - f(\mathbf{x})].$$

Proof. The claim is easy to show for $n = 1$ by induction. If, w.l.o.g., $h_2 \geq h_1 \geq 0$ then convexity of f implies $[f(x+h_1+h_2) - f(x+h_2)]/h_1 \geq [f(x+h_1) - f(x)]/h_1$, and thus $f(x+h_1+h_2) - f(x) \geq [f(x+h_2) - f(x)] + [f(x+h_1) - f(x)]$. The claim for general n then follows from the separability of f by adding the superadditivity relations of each one-parametric convex summand of f .

A crucial role in the following theorem is again played by the Graver basis $\mathcal{G}(A)$ of A . Let us remind the reader that the Graver basis $\mathcal{G}(A)$ has a nice representation property due to its definition: every $\mathbf{z} \in \ker(A) \cap \mathbb{Z}^n$ can be written as a sign-compatible nonnegative integer linear combination $\mathbf{z} = \sum_i \alpha_i \mathbf{g}_i$ of Graver basis elements $\mathbf{g}_i \in \mathcal{G}(A)$. This followed from the simple observation that \mathbf{z} has to belong to some orthant \mathcal{O}_j of \mathbb{R}^n and thus it can be represented as a sign-compatible nonnegative integer linear combination of elements in $H_j \subseteq \mathcal{G}(A)$ belonging to this orthant. Note that by the integer Carathéodory property of Hilbert bases, at most

$2 \cdot \dim(\ker(A)) - 2$ vectors are needed in such a representation [118]. It is precisely this simple representation property of $\mathcal{G}(A)$ combined with the superadditivity of the separable convex function f that turns $\mathcal{G}(A)$ into an optimality certificate for $\min\{f(\mathbf{x}) : \mathbf{Ax} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n\}$.

Theorem 12. *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a separable convex function given by a comparison oracle that when queried on $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^n$ decides whether $f(\mathbf{x}) < f(\mathbf{y})$, $f(\mathbf{x}) = f(\mathbf{y})$, or $f(\mathbf{x}) > f(\mathbf{y})$. Then \mathbf{x}_0 is an optimal feasible solution to $\min\{f(\mathbf{x}) : \mathbf{Ax} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n\}$ if and only if for all $\mathbf{g} \in \mathcal{G}(A)$ the vector $\mathbf{x}_0 + \mathbf{g}$ is not feasible or $f(\mathbf{x}_0 + \mathbf{g}) \geq f(\mathbf{x}_0)$.*

Proof. Assume that \mathbf{x}_0 is not optimal and let \mathbf{x}_{\min} be an optimal solution to the given problem. Then $\mathbf{x}_{\min} - \mathbf{x}_0 \in \ker(A)$ and thus it can be written as a sign-compatible nonnegative integer linear combination $\mathbf{x}_{\min} - \mathbf{x}_0 = \sum_i \alpha_i \mathbf{g}_i$ of Graver basis elements $\mathbf{g}_i \in \mathcal{G}(A)$. We show that one of the \mathbf{g}_i must be an improving vector, that is, for some \mathbf{g}_i we have that $\mathbf{x}_0 + \mathbf{g}_i$ is feasible and $f(\mathbf{x}_0 + \mathbf{g}_i) < f(\mathbf{x}_0)$.

For all i , the vector \mathbf{g}_i has the same sign-pattern as $\mathbf{x}_{\min} - \mathbf{x}_0$ and it is now easy to check that the coordinates of $\mathbf{x}_0 + \mathbf{g}_i$ lie between the corresponding coordinates of \mathbf{x}_0 and \mathbf{x}_{\min} . This implies in particular $\mathbf{l} \leq \mathbf{x}_0 + \mathbf{g}_i \leq \mathbf{u}$. Because $\mathbf{g}_i \in \ker(A)$, we also have $A(\mathbf{x}_0 + \mathbf{g}_i) = \mathbf{b}$ for all i . Consequently, for all i the vector $\mathbf{x}_0 + \mathbf{g}_i$ would be a feasible solution. It remains to show that one of these vectors has a strictly smaller objective value than \mathbf{x}_0 .

Due to the superadditivity from Lemma 5, we have

$$0 \geq f(\mathbf{x}_{\min}) - f(\mathbf{x}_0) = f\left(\mathbf{x}_0 + \sum_{i=1}^{2n-2} \alpha_i \mathbf{g}_i\right) - f(\mathbf{x}_0) \geq \sum_{i=1}^k \alpha_i [f(\mathbf{x}_0 + \mathbf{g}_i) - f(\mathbf{x}_0)].$$

Thus, at least one of the summands $f(\mathbf{x}_0 + \mathbf{g}_i) - f(\mathbf{x}_0)$ must be negative and we have found an improving vector for \mathbf{z}_0 in $\mathcal{G}(A)$.

We now turn this optimality certificate into a polynomial oracle-time algorithm to solve the separable convex integer minimization problem $\min\{f(\mathbf{x}) : \mathbf{Ax} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n\}$. For this, we call $\alpha \mathbf{g}$ a *greedy* Graver improving vector if $\mathbf{x}_0 + \alpha \mathbf{g}$ is feasible and such that $f(\mathbf{x}_0 + \alpha \mathbf{g})$ is minimal among all such choices of $\alpha \in \mathbb{Z}_+$ and $\mathbf{g} \in \mathcal{G}(A)$. Then the following result holds.

Theorem 13. *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a separable convex function given by a comparison oracle. Moreover, assume that $|f(\mathbf{x})| < M$ for all $\mathbf{x} \in \{\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n\}$. Then any feasible solution \mathbf{x}_0 to $\min\{f(\mathbf{x}) : \mathbf{Ax} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n\}$ can be augmented to optimality by a number of greedy Graver augmentation steps that is polynomially bounded in the encoding lengths of $A, \mathbf{b}, \mathbf{l}, \mathbf{u}, M$, and \mathbf{x}_0 .*

Proof. Assume that \mathbf{x}_0 is not optimal and let \mathbf{x}_{\min} be an optimal solution to the given problem. Then $\mathbf{x}_{\min} - \mathbf{x}_0 \in \ker(A)$ and thus it can be written as a sign-compatible nonnegative integer linear combination $\mathbf{x}_{\min} - \mathbf{x}_0 = \sum_i \alpha_i \mathbf{g}_i$ of at most $2n - 2$ Graver basis elements $\mathbf{g}_i \in \mathcal{G}(A)$. As in the proof of Theorem 12, sign-compatibility implies that for all i the coordinates of $\mathbf{x}_0 + \alpha_i \mathbf{g}_i$ lie between the corresponding coordinates

of \mathbf{x}_0 and \mathbf{x}_{\min} . Consequently, we have $\mathbf{l} \leq \mathbf{x}_0 + \alpha_i \mathbf{g}_i \leq \mathbf{u}$. Because $\mathbf{g}_i \in \ker(A)$, we also have $A(\mathbf{x}_0 + \alpha_i \mathbf{g}_i) = \mathbf{b}$ for all i . Consequently, for all i the vector $\mathbf{x}_0 + \alpha_i \mathbf{g}_i$ would be a feasible solution.

Due to the superadditivity from Lemma 5, we have

$$0 \geq f(\mathbf{x}_{\min}) - f(\mathbf{x}_0) = f\left(\mathbf{x}_0 + \sum_{i=1}^{2n-2} \alpha_i \mathbf{g}_i\right) - f(\mathbf{x}_0) \geq \sum_{i=1}^k [f(\mathbf{x}_0 + \alpha_i \mathbf{g}_i) - f(\mathbf{x}_0)].$$

Thus, at least one of the summands $f(\mathbf{x}_0 + \alpha_i \mathbf{g}_i) - f(\mathbf{x}_0)$ must be smaller than $\frac{1}{2n-2}[f(\mathbf{x}_{\min}) - f(\mathbf{x}_0)]$, giving an improvement that is at least $\frac{1}{2n-2}$ times the maximal possible improvement $f(\mathbf{x}_{\min}) - f(\mathbf{x}_0)$. Such a geometric improvement, however, implies that the optimum is reached in a number of greedy augmentation steps which is polynomial in the encoding lengths of A , \mathbf{b} , \mathbf{l} , \mathbf{u} , M , and \mathbf{x}_0 [3].

Thus, once we have a polynomial size Graver basis, we get a polynomial time algorithm to solve the convex integer minimization problem at hand.

For this, let us consider again n -fold systems (introduced in Section 2.2). Two nice stabilization results established by Hoşten and Sullivant [71] and Santos and Sturmfels [113] immediately imply that if A_1 and A_2 are kept fixed, then the size of the Graver basis increases only polynomially in the number n of copies of A_1 and A_2 .

Proposition 2. *For any fixed $(r+s) \times t$ integer matrix A there is a polynomial time algorithm that, given any n , computes the Graver basis $\mathcal{G}(A^{(n)})$ of the n -fold matrix $A^{(n)} = (\mathbf{1}_n \otimes A_1) \oplus (I_n \otimes A_2)$.*

Combining this proposition with Theorem 13, we get following nice result from [69].

Theorem 14. *Let A be a fixed integer $(r+s) \times t$ matrix and let $f : \mathbb{R}^m \rightarrow \mathbb{R}$ be any separable convex function given by a comparison oracle. Then there is a polynomial time algorithm that, given n , a right-hand side vector $\mathbf{b} \in \mathbb{Z}^{r+ns}$ and some bound $|f(\mathbf{x})| < M$ on f over the feasible region, solves the n -fold convex integer programming problem*

$$\min\{f(\mathbf{x}) : A^{(n)}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{N}^m\}.$$

Note that by applying an approach similar to Phase I of the simplex method one can also compute an initial feasible solution x_0 to the n -fold integer program in polynomial time based on greedy Graver basis directions [43, 67].

We wish to point out that the presented approach can be generalized to the mixed-integer situation and also to more general objective functions that satisfy a certain superadditivity/subadditivity condition, see [68, 86] for more details. Note that for mixed-integer convex problems one may only expect an approximation result, as there need not exist a rational optimum. In fact, already a mixed-integer greedy augmentation vector can be computed only approximately. Nonetheless, the technical difficulties when adjusting the proofs for the pure integer case to the mixed-integer

situation can be overcome [68]. It should be noted, however, that the Graver basis of n -fold matrices does not show a stability result similar to the pure integer case as presented in [71, 113]. Thus, we do not get a nice polynomial time algorithm for solving mixed-integer convex n -fold problems.

3.3 Practical algorithms

In this section, the methods that we look at, aimed at formulations having convex continuous relaxations, are driven by O.R./engineering approaches, transporting and motivated by successful mixed-integer linear programming technology and smooth continuous nonlinear programming technology. In Section 3.3.1 we discuss general algorithms that make few assumptions beyond those that are typical for convex continuous nonlinear programming. In Section 3.3.2 we present some more specialized techniques aimed at convex quadratics.

3.3.1 General algorithms

Practical, broadly applicable approaches to general mixed-integer nonlinear programs are aimed at problems involving convex minimization over a convex set with some additional integrality restriction. Additionally, for the sake of obtaining well-behaved continuous relaxations, a certain amount of smoothness is usually assumed. Thus, in this section, the model that we focus on is

$$\begin{aligned}
 \min \quad & f(\mathbf{x}, \mathbf{y}) \\
 \text{s.t.} \quad & g(\mathbf{x}, \mathbf{y}) \leq \mathbf{0} \\
 & \mathbf{l} \leq \mathbf{y} \leq \mathbf{u} \\
 & \mathbf{x} \in \mathbb{R}^{n_1}, \mathbf{y} \in \mathbb{Z}^{n_2},
 \end{aligned} \tag{P[\mathbf{l}, \mathbf{u}]}$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$ are twice continuously-differentiable convex functions, $\mathbf{l} \in (\mathbb{Z} \cup \{-\infty\})^{n_2}$, $\mathbf{u} \in (\mathbb{Z} \cup \{+\infty\})^{n_2}$, and $\mathbf{l} \leq \mathbf{u}$. It is also helpful to assume that the feasible region of the relaxation of (P[\mathbf{l}, \mathbf{u}]) obtained by replacing $\mathbf{y} \in \mathbb{Z}^{n_2}$ with $\mathbf{y} \in \mathbb{R}^{n_2}$ is bounded. We denote this continuous relaxation by (P_ℝ[\mathbf{l}, \mathbf{u}]).

To describe the various algorithmic approaches, it is helpful to define some related subproblems of (P[\mathbf{l}, \mathbf{u}]) and associated relaxations. Our notation is already designed for this. For vector $\mathbf{l}' \in (\mathbb{Z} \cup \{-\infty\})^{n_2}$ and $\mathbf{u}' \in (\mathbb{Z} \cup \{+\infty\})^{n_2}$, with $\mathbf{l} \leq \mathbf{l}' \leq \mathbf{u}' \leq \mathbf{u}$, we have the *subproblem* (P[\mathbf{l}', \mathbf{u}']) and its associated continuous relaxation (P_ℝ[\mathbf{l}', \mathbf{u}']).

Already, we can see how the family of relaxations (P_ℝ[\mathbf{l}', \mathbf{u}']) leads to the obvious extension of the Branch-and-Bound Algorithm of mixed-integer linear programming. Indeed, this approach was experimented with in [63]. The *Branch-and-Bound Algorithm* for mixed-integer nonlinear programming has been implemented as MINLP-BB [88], with continuous nonlinear-programming subproblem relax-

ations solved with the active-set solver `filterSQP` and also as SBB, with associated subproblems solved with any of `CONOPT`, `SNOPT` and `MINOS`. Moreover, Branch-and-Bound is available as an algorithmic option in the actively developed code `Bonmin` [21, 25, 23], which can be used as a callable library, as a stand-alone solver, via the modeling languages `AMPL` and `GAMS`, and is available in source-code form, under the Common Public License, from COIN-OR [28], available for running on NEOS [27]. By default, relaxations of subproblems are solved with the interior-point solver `Ipopt` (whose availability options include all of those for `Bonmin`), though there is also an interface to `filterSQP`. The Branch-and-Bound Algorithm in `Bonmin` includes effective strong branching and SOS branching. It can also be used as a robust heuristic on problems for which the relaxation $(P_{\mathbb{R}})$ does not have a convex feasible region, by setting negative ‘cutoff gaps’.

Another type of algorithmic approach emphasizes continuous nonlinear programming over just the continuous variables of the formulation. For fixed $\bar{\mathbf{y}} \in \mathbb{Z}^{n_2}$, we define

$$\begin{aligned} \min \quad & f(\mathbf{x}, \mathbf{y}) \\ \text{s.t.} \quad & g(\mathbf{x}, \mathbf{y}) \leq \mathbf{0} \\ & \mathbf{y} = \bar{\mathbf{y}} \\ & \mathbf{x} \in \mathbb{R}^{n_1}. \end{aligned} \tag{P $\bar{\mathbf{y}}$ }$$

Clearly any feasible solution to such a continuous nonlinear-programming subproblem $(P^{\bar{\mathbf{y}}})$ yields an upper bound on the optimal value of $(P[\mathbf{l}, \mathbf{u}])$. When $(P^{\bar{\mathbf{y}}})$ is infeasible, we may consider the continuous nonlinear-programming feasibility subproblem

$$\begin{aligned} \min \quad & \sum_{i=1}^m w_i \\ \text{s.t.} \quad & g(\mathbf{x}, \mathbf{y}) \leq \mathbf{w} \\ & \mathbf{y} = \bar{\mathbf{y}} \\ & \mathbf{x} \in \mathbb{R}^{n_1} \\ & \mathbf{w} \in \mathbb{R}_+^m. \end{aligned} \tag{F $\bar{\mathbf{y}}$ }$$

If we can find a way to couple the solution of upper-bounding problems $(P^{\bar{\mathbf{y}}})$ (and the closely related feasibility subproblems $(F^{\bar{\mathbf{y}}})$) with a lower-bounding procedure exploiting the convexity assumptions, then we can hope to build an iterative procedure that will converge to a global optimum of $(P[\mathbf{l}, \mathbf{u}])$. Indeed, such a procedure is the *Outer-Approximation (OA) Algorithm* [48, 49]. Toward this end, for a finite set of ‘linearization points’

$$\mathcal{K} := \left\{ \left(\mathbf{x}^k \in \mathbb{R}^{n_1}, \mathbf{y}^k \in \mathbb{R}^{n_2} \right) : k = 1, \dots, K \right\},$$

we define the mixed-integer *linear* programming relaxation

$$\begin{aligned}
& \min && z \\
& \text{s.t.} && \nabla f(\mathbf{x}^k, \mathbf{y}^k)^\top \begin{pmatrix} \mathbf{x} - \mathbf{x}^k \\ \mathbf{y} - \mathbf{y}^k \end{pmatrix} + f(\mathbf{x}^k, \mathbf{y}^k) \leq z, \quad \forall (\mathbf{x}^k, \mathbf{y}^k) \in \mathcal{K} \\
& && \nabla g(\mathbf{x}^k, \mathbf{y}^k)^\top \begin{pmatrix} \mathbf{x} - \mathbf{x}^k \\ \mathbf{y} - \mathbf{y}^k \end{pmatrix} + g(\mathbf{x}^k, \mathbf{y}^k) \leq 0, \quad \forall (\mathbf{x}^k, \mathbf{y}^k) \in \mathcal{K} \quad (P^{\mathcal{K}}[\mathbf{l}, \mathbf{u}]) \\
& && \mathbf{x} \in \mathbb{R}^{n_1} \\
& && \mathbf{y} \in \mathbb{R}^{n_2}, \quad \mathbf{l} \leq \mathbf{y} \leq \mathbf{u} \\
& && z \in \mathbb{R}.
\end{aligned}$$

We are now able to concisely state the basic OA Algorithm.

Algorithm 1 (OA Algorithm)

Input: *The mixed-integer nonlinear program* $(P[\mathbf{l}, \mathbf{u}])$.

Output: *An optimal solution* $(\mathbf{x}^*, \mathbf{y}^*)$.

1. *Solve the nonlinear-programming relaxation* $(P_{\mathbb{R}})$, let $(\mathbf{x}^1, \mathbf{y}^1)$ be an optimal solution, and let $K := 1$, so that initially we have $\mathcal{K} = \{(\mathbf{x}^1, \mathbf{y}^1)\}$.
2. *Solve the mixed-integer linear programming relaxation* $(P^{\mathcal{K}}[\mathbf{l}, \mathbf{u}])$, and let $(\mathbf{x}^*, \mathbf{y}^*, z^*)$ be an optimal solution. *If* $(\mathbf{x}^*, \mathbf{y}^*, z^*)$ *corresponds to a feasible solution of* $(P[\mathbf{l}, \mathbf{u}])$ *(i.e., if* $f(\mathbf{x}^*, \mathbf{y}^*) \leq z^*$ *and* $g(\mathbf{x}^*, \mathbf{y}^*) \leq \mathbf{0}$ *), then STOP (with the optimal solution* $(\mathbf{x}^*, \mathbf{y}^*)$ *of* $(P[\mathbf{l}, \mathbf{u}])$ *).*
3. *Solve the continuous nonlinear-programming subproblem* $(P^{\bar{\mathbf{y}}^*})$.
 - i. *Either a feasible solution* $(\mathbf{x}^*, \mathbf{y}^*, z^*)$ *is obtained,*
 - ii. *or* $(P^{\bar{\mathbf{y}}^*})$ *is infeasible, in which case we solve the nonlinear-programming feasibility subproblem* $(F^{\bar{\mathbf{y}}^*})$, *and let its solution be* $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{u}^*)$
4. *In either case, we augment the set* \mathcal{K} *of linearization points, by letting* $K := K + 1$ *and* $(\mathbf{x}^K, \mathbf{y}^K) := (\mathbf{x}^*, \mathbf{y}^*)$.
5. *GOTO* 2.

Each iteration of Steps 3-4 generate a linear cut that can improve the mixed-integer linear programming relaxation $(P^{\mathcal{K}}[\mathbf{l}, \mathbf{u}])$ that is repeatedly solved in Step 2. So clearly the sequence of optimal objective values for $(P^{\mathcal{K}}[\mathbf{l}, \mathbf{u}])$ obtained in Step 2 corresponds to a nondecreasing sequence of lower bounds on the optimum value of $(P[\mathbf{l}, \mathbf{u}])$. Moreover each linear cut returned from Steps 3-4 cuts off the previous solution of $(P^{\mathcal{K}}[\mathbf{l}, \mathbf{u}])$ from Step 2. A precise proof of convergence (see for example [21]) uses these simple observations, but it also requires an additional assumption that is standard in continuous nonlinear programming (i.e. a ‘‘constraint qualification’’).

Implementations of OA include DICOPT [39] which can be used with either of the mixed-integer linear programs codes Cplex and Xpress-MP, in conjunction with any of the continuous nonlinear programming codes CONOPT, SNOPT and MINOS and is available with GAMS. Additionally Bonmin has OA as an algorithmic option, which can use Cplex or the COIN-OR code Cbc as its mixed-integer

linear programming solver, and Ipopt or FilterSQP as its continuous nonlinear programming solver.

Generalized Benders Decomposition [54] is a technique that is closely related to and substantially predates the OA Algorithm. In fact, one can regard OA as a proper strengthening of Generalized Benders Decomposition (see [48, 49]), so as a practical tool, we view it as superseded by OA.

Substantially postdating the development of the OA Algorithm is the simpler and closely related *Extended Cutting Plane (ECP) Algorithm* introduced in [135]. The original ECP Algorithm is a straightforward generalization of *Kelley's Cutting-Plane Algorithm* [77] for convex continuous nonlinear programming (which predates the development of the OA Algorithm). Subsequently, the ECP Algorithm has been enhanced and further developed (see, for example [136, 134]) to handle, for example, even pseudo-convex functions.

The motivation for the ECP Algorithm is that continuous nonlinear programs are expensive to solve, and all that the associated solutions give us are further linearization points for $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$. So the ECP Algorithm dispenses altogether with the solution of continuous nonlinear programs. Rather, in the most rudimentary version, after each solution of the mixed-integer linear program $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$, the most violated constraint (i.e., of $f(\mathbf{x}^*, \mathbf{y}^*) \leq z$ and $g(\mathbf{x}^*, \mathbf{y}^*) \leq \mathbf{0}$) is linearized and appended to $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$. This simple iteration is enough to easily establish convergence (see [135]). It should be noted that for the case in which there are no integer-constrained variables, then at each step $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$ is just a continuous linear program and we exactly recover Kelley's Cutting-Plane Algorithm for convex continuous nonlinear programming.

It is interesting to note that Kelley, in his seminal paper [77], already considered application of his approach to *integer* nonlinear programs. In fact, Kelley cited Gomory's seminal work on integer programming [58, 57] which was also taking place in the same time period, and he discussed how the approaches could be integrated.

Of course, many practical improvements can be made to the rudimentary ECP Algorithm. For example, more constraints can be linearized at each iteration. An implementation of the ECP Algorithm is the code `Alpha-ECP` (see [134]) which uses `Cplex` as its mixed-integer linear programming solver and is available with `GAMS`. The general experience is that for mildly nonlinear problems, an ECP Algorithm can outperform an OA Algorithm. But for a highly nonlinear problem, the performance of the ECP Algorithm is limited by the performance of Kelley's Cutting-Plane Algorithm, which can be quite poor on highly-nonlinear purely continuous problems. In such cases, it is typically better to use an OA Algorithm, which will handle the nonlinearity in a more sophisticated manner.

In considering again the performance of an OA Algorithm on a mixed-integer nonlinear program $(P[\mathbf{l}, \mathbf{u}])$, rather than the convex continuous nonlinear programming problems $(P^{\mathbf{y}^*})$ and $(F^{\mathbf{y}})$ being too time consuming to solve (which led us to the ECP Algorithm), it can be the case that solution of the mixed-integer linear programming problems $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$ dominate the running time. Such a situation led to the *Quesada-Grossmann Branch-and-Cut Algorithm* [107]. The viewpoint is that the mixed-integer linear programming problems $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$ are solved by a Branch-

and-Bound or Branch-and-Cut Algorithm. During the solution of the mixed-integer linear programming problem $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$, whenever a new solution is found (i.e., one that has the variables \mathbf{y} integer), we interrupt the solution process for $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$, and we solve the convex continuous nonlinear programming problems $(P^{\bar{\mathbf{y}}^*})$ to derive new outer approximation cuts that are appended to mixed-integer linear programming problem $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$. We then continue with the solution process for $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$. The Quesada-Grossmann Branch-and-Cut Algorithm is available as an option in `Bonmin`.

Finally, it is clear that the essential scheme of the Quesada-Grossmann Branch-and-Cut Algorithm admits enormous flexibility. The *Hybrid Algorithm* [21] incorporates two important enhancements.

First, we can seek to further improve the linearization $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$ by solving convex continuous nonlinear programming problems at additional nodes of the mixed-integer linear programming Branch-and-Cut tree for $(P^{\mathcal{X}}[\mathbf{l}, \mathbf{u}])$ — that is, not just when solutions are found having \mathbf{y} integer. In particular, at any node $(P^{\mathcal{X}}[\mathbf{l}', \mathbf{u}'])$ of the mixed-integer linear programming Branch-and-Cut tree, we can solve the associated convex continuous nonlinear programming subproblem $(P_{\mathbb{R}}[\mathbf{l}', \mathbf{u}'])$: Then, if in the solution $(\mathbf{x}^*, \mathbf{y}^*)$ we have that \mathbf{y}^* is integer, we may update the incumbent and fathom the node; otherwise, we append $(\mathbf{x}^*, \mathbf{y}^*)$ to the set \mathcal{X} of linearization points. In the extreme case, if we solve these continuous nonlinear programming subproblems at every node, we essentially have the Branch-and-Bound Algorithm for mixed-integer nonlinear programming.

A second enhancement is based on working harder to find a solution $(\mathbf{x}^*, \mathbf{y}^*)$ with \mathbf{y}^* integer at selected nodes $(P^{\mathcal{X}}[\mathbf{l}', \mathbf{u}'])$ of the mixed-integer linear programming Branch-and-Cut tree. The idea is that at a node $(P^{\mathcal{X}}[\mathbf{l}', \mathbf{u}'])$, we perform a time-limited mixed-integer linear programming Branch-and-Bound Algorithm. If we are successful, then we will have found a solution to the node with $(\mathbf{x}^*, \mathbf{y}^*)$ with \mathbf{y}^* integer, and then we perform an OA iteration (i.e., Steps 3-4) on $(P[\mathbf{l}', \mathbf{u}'])$ which will improve the linearization $(P^{\mathcal{X}}[\mathbf{l}', \mathbf{u}'])$. We can then repeat this until we have solved the mixed-integer nonlinear program $(P[\mathbf{l}', \mathbf{u}'])$ associated with the node. If we do this without time limit at the root node $(P[\mathbf{l}, \mathbf{u}])$, then the entire procedure reduces to the OA Algorithm. The Hybrid Algorithm was developed for and first made available as part of `Bonmin`.

`FilMint` [1] is another successful modern code, also based on enhancing the general framework of the Quesada-Grossmann Branch-and-Cut Algorithm. The main additional innovation introduced with `FilMint` is the idea of using ECP cuts rather than only performing OA iterations for getting cuts to improve the linearizations $(P^{\mathcal{X}}[\mathbf{l}', \mathbf{u}'])$. Subsequently, this feature was also added to `Bonmin`. `FilMint` was put together from the continuous nonlinear programming active-set code `FilterSQP`, and the mixed-integer linear programming code `MINTO`. Currently, `FilMint` is only generally available via NEOS [50].

It is worth mentioning that just as for mixed-integer linear programming, effective heuristics can and should be used to provide good upper bounds quickly. This can markedly improve the performance of any of the algorithms described above. Some examples of work in this direction are [24] and [22].

3.3.2 Convex quadratics and second-order cone programming

Though we will not go into any details, there is considerable algorithmic work and associated software that seeks to leverage more specialized (but still rather general and powerful) nonlinear models and existing convex continuous nonlinear-programming algorithms for the associated relaxations. In this direction, recent work has focused on conic programming relaxations (in particular, the semi-definite and second-order cones). On the software side, we point to work on the binary quadratic and max-cut problems (via semi-definite relaxation) [108, 109] with the code `Biq Mac` [20]. We also note that `Cplex` (v11) has a capability aimed at solving mixed-integer quadratically-constrained programs that have a convex continuous relaxation.

One important direction for approaching quadratic models is at the modeling level. This is particularly useful for the convex case, where there is a strong and appealing relationship between quadratically constrained programming and second-order cone programming (SOCP). A *second-order cone constraint* is one that expresses that the Euclidean norm of an affine function should be no more than another affine function. An *SOCP* problem consists of minimizing a linear function over a finite set of second-order cone constraints. Our interest in SOCP stems from the fact that (continuous) convex quadratically constrained programming problems can be reformulated as SOCP problems (see [92]). The appeal is that very efficient interior-point algorithms have been developed for solving SOCP problems (see [56], for example), and there is considerable mature software available that has functionality for efficient handling of SOCP problems; see, for example: `SDPT3` [117] (GNU GPL open-source license; Matlab), `SeDuMi` [119] (GNU GPL open-source license; Matlab), `LOQO` [93] (proprietary; C library with interfaces to AMPL and Matlab), `MOSEK` [99] (proprietary; C library with interface to Matlab), `Cplex` [72] (proprietary; C library). Note also that `MOSEK` and `Cplex` can handle integer variables as well; one can expect that the approaches essentially marry specialized SOCP solvers with Branch-and-Bound and/or Outer-Approximation Algorithms. Further branch-and-cut methods for mixed-integer SOCP, employing linear and convex quadratic cuts [36] and a careful treatment of the non-differentiability inherent in the SOCP constraints, have recently been proposed [47].

Also in this vein is recent work by Günlük and Linderoth [61, 62]. Among other things, they demonstrated that many practical mixed-integer quadratically constrained programming formulations have substructures that admit extended formulations that can be easily strengthened as certain integer SOCP problems. This approach is well known in the mixed-integer linear programming literature. Let

$$\mathcal{Q} := \left\{ w \in \mathbb{R}, \mathbf{x} \in \mathbb{R}_+^n, \mathbf{z} \in \{0, 1\}^n : w \geq \sum_{i=1}^n r_i x_i^2, u_i z_i \geq x_i \geq l_i z_i, i = 1, 2, \dots, n \right\},$$

where $r_i \in \mathbb{R}_+$ and $u_i, l_i \in \mathbb{R}$ for all $i = 1, 2, \dots, n$. The set \mathcal{Q} appears in several formulations as a substructure. Consider the following extended formulation of \mathcal{Q}

$$\bar{Q} := \left\{ w \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{R}^n : w \geq \sum_i r_i y_i, \right. \\ \left. (x_i, y_i, z_i) \in S_i, i = 1, 2, \dots, n \right\},$$

where

$$S_i := \left\{ (x_i, y_i, z_i) \in \mathbb{R}^2 \times \{0, 1\} : y_i \geq x_i^2, u_i z_i \geq x_i \geq l_i z_i, x_i \geq 0 \right\},$$

and $u_i, l_i \in \mathbb{R}$. The convex hull of each S_i has the form

$$S_i^c := \left\{ (x_i, y_i, z_i) \in \mathbb{R}^3 : y_i z_i \geq x_i^2, u_i z_i \geq x_i \geq l_i z_i, 1 \geq z_i \geq 0, x_i, y_i \geq 0 \right\}$$

(see [35, 61, 62, 129]). Note that $x_i^2 - y_i z_i$ is *not* a convex function, but nonetheless S_i^c is a convex set. Finally, we can state the result of [61, 62], which also follows from a more general result of [70], that the convex hull of the extended formulation \bar{Q} has the form

$$\bar{Q}^c := \left\{ w \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^n, \mathbf{z} \in \mathbb{R}^n : w \geq \sum_{i=1}^n r_i y_i, \right. \\ \left. (x_i, y_i, z_i) \in S_i^c, i = 1, 2, \dots, n \right\}.$$

Note that all of the nonlinear constraints describing the S_i^c and \bar{Q}^c are rather simple quadratic constraints. Generally, it is well known that even the “restricted hyperbolic constraint”

$$y_i z_i \geq \sum_{k=1}^n x_k^2, \mathbf{x} \in \mathbb{R}^n, y_i \geq 0, z_i \geq 0$$

(more general than the nonconvexity in S_i^c) can be reformulated as the second-order cone constraint

$$\left\| \begin{pmatrix} 2\mathbf{x} \\ y_i - z_i \end{pmatrix} \right\|_2 \leq y_i + z_i.$$

In this subsection, in the interest of concreteness and brevity, we have focused our attention on convex quadratics and second-order cone programming. However, it should be noted that a related approach, with broader applicability (to all convex objective functions) is presented in [51], and a computational comparison is available in [52]. Also, it is relevant that many convex non-quadratic functions are representable as second-order cone programs (see [4, 15]).

4 Polynomial optimization

In this section, we focus our attention on the study of optimization models involving polynomials only, but without any assumptions on convexity or concavity. It is worth emphasizing the fundamental result of Jeroslow (Theorem 4) that even pure integer

quadratically constrained programming is *undecidable*. One can however avoid this daunting pitfall by bounding the variables, and this is in fact an assumption that we should and do make for the purpose of designing practical approaches. From the point of view of most applications that we are aware of, this is a very reasonable assumption. We must be cognizant of the fact that the geometry of even quadratics on boxes is daunting from the point of view of mathematical programming; see Figure 6. Specifically, the convex envelope of the graph of the product x_1x_2 on a box deviates badly from the graph, so relaxation-based methods are intrinsically handicapped. It is easy to see, for example, that for $\delta_1, \delta_2 > 0$, x_1x_2 is strictly convex on the line segment joining $(0,0)$ and (δ_1, δ_2) ; while x_1x_2 is strictly concave on the line segment joining $(\delta_1, 0)$ and $(0, \delta_2)$.

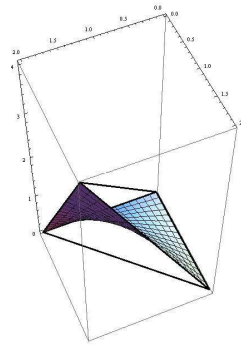


Fig. 6 Tetrahedral convex envelope of the graph of the product x_1x_2 on a box

Despite these difficulties, we have positive results. In Section 4.1, the highlight is a *fully polynomial time approximation scheme (FPTAS)* for problems involving maximization of a polynomial in fixed dimension, over the mixed-integer points in a polytope. In Section 4.2, we broaden our focus to allow feasible regions defined by inequalities in polynomials (i.e., semi-algebraic sets). In this setting, we do not present (nor could we expect) complexity results as strong as for linear constraints, but rather we show how tools of semi-definite programming are being developed to provide, in a systematic manner, strengthened relaxations. Finally, in Section 4.3, we describe recent computational advances in the special case of semi-algebraic programming for which all of the functions are quadratic — i.e., *mixed-integer quadratically constrained programming (MIQCP)*.

4.1 Fixed dimension and linear constraints: An FPTAS

As we pointed out in the introduction (Theorem 2), optimizing degree-4 polynomials over problems with two integer variables 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.

Definition 3. (FPTAS)

- (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 $f_{\mathcal{A}} \geq (1 - \varepsilon) \cdot f_{\max}$.
- (b) A family $\{\mathcal{A}_\varepsilon\}_\varepsilon$ of ε -approximation algorithms is a *fully polynomial time approximation scheme (FPTAS)* if the running time of \mathcal{A}_ε is polynomial in the encoding size of the instance and $1/\varepsilon$.

Indeed it is possible to obtain an FPTAS for general polynomial optimization of mixed-integer feasible sets in polytopes [41, 40, 42]. To explain the method of the FPTAS, we need to review the theory of *short rational generating functions* pioneered by Barvinok [12, 13]. The FPTAS itself appears in Section 4.1.3.

4.1.1 Introduction to rational generating functions

We explain the theory on a simple, one-dimensional example. Let us consider the set S of integers in the interval $P = [0, \dots, n]$; see the top of Figure 7 (a). We associate with S the polynomial $g(S; z) = z^0 + z^1 + \dots + z^{n-1} + z^n$; 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 $0, 1, \dots, n-1, n$ would be. However, we can observe that $g(S; z)$ is a finite geometric series, so there exists a simple summation formula that expresses it in a much more compact way:

$$g(S; z) = z^0 + z^1 + \dots + z^{n-1} + z^n = \frac{1 - z^{n+1}}{1 - z}. \quad (11)$$

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 $|S| = g(S; 1) = 1^0 + 1^1 + \dots + 1^{n-1} + 1^n = n + 1$. We can do the same on the shorter, rational-function formula if we are careful with the (removable) singularity $z = 1$. We just compute the limit

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

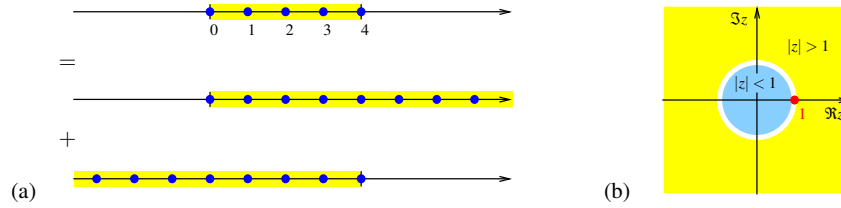


Fig. 7 (a) One-dimensional Brion theorem. (b) The domains of convergence of the Laurent series.

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.

The summation formula (11) can also be written in a slightly different way:

$$g(S; z) = \frac{1}{1-z} - \frac{z^{n+1}}{1-z} = \frac{1}{1-z} + \frac{z^n}{1-z^{-1}} \tag{12}$$

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

$$g_1(z) = \frac{1}{1-z} = z^0 + z^1 + z^2 + \dots, \tag{13a}$$

$$g_2(z) = \frac{z^n}{1-z^{-1}} = z^n + z^{n-1} + z^{n-2} + \dots \tag{13b}$$

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 7. We remark that all integer points in the interval $[0, n]$ are covered twice, and also all integer points outside the interval are covered once. This phenomenon 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 of convergence (Figure 7): For $|z| < 1$, the power series $z^0 + z^1 + z^2 + \dots$ converges to $g_1(z)$. As a matter of fact, it converges absolutely and uniformly on every compact subset of the open circle $\{z \in \mathbb{C} : |z| < 1\}$. For $|z| > 1$, however, the series diverges. On the other hand, the Laurent series $-z^{-1} - z^{-2} - z^{-3} - \dots$ converges (absolutely and compact-uniformly) on the open circular ring $\{z \in \mathbb{C} : |z| > 1\}$ to the function $g_1(z)$, whereas it diverges for $|z| < 1$. The same holds for $g_2(z)$. Altogether we have:

$$g_1(z) = \begin{cases} z^0 + z^1 + z^2 + \dots & \text{for } |z| < 1 \\ -z^{-1} - z^{-2} - z^{-3} - \dots & \text{for } |z| > 1 \end{cases} \tag{14}$$

$$g_2(z) = \begin{cases} -z^{n+1} - z^{n+2} - z^{n+3} - \dots & \text{for } |z| < 1 \\ z^n + z^{n-1} + z^{n-2} + \dots & \text{for } |z| > 1 \end{cases} \tag{15}$$

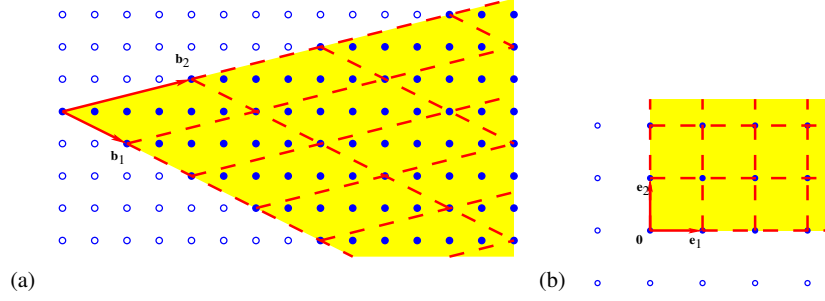


Fig. 8 (a) Tiling a rational two-dimensional cone with copies of the fundamental parallelepiped. (b) The semigroup $S \subseteq \mathbb{Z}^2$ generated by \mathbf{b}_1 and \mathbf{b}_2 is a linear image of \mathbb{Z}_+^2 .

We can now see that the phenomenon we observed in formula (13) and Figure 7 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; the situation of formula (13) and Figure 7 appears again in the d -dimensional case as *Brion's Theorem*.

Let us now consider a *two-dimensional* cone C spanned by the vectors $\mathbf{b}_1 = (\alpha, -1)$ and $\mathbf{b}_2 = (\beta, 1)$; see Figure 8 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*, $\Pi = \{ \lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 : \lambda_1 \in [0, 1), \lambda_2 \in [0, 1) \}$, the cone can be *tilled*:

$$C = \bigcup_{\mathbf{s} \in S} (\mathbf{s} + \Pi) \quad \text{where} \quad S = \{ \mu_1 \mathbf{b}_1 + \mu_2 \mathbf{b}_2 : (\mu_1, \mu_2) \in \mathbb{Z}_+^2 \} \quad (16)$$

(a disjoint union). Because we have chosen *integral* generators $\mathbf{b}_1, \mathbf{b}_2$, 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 $\Pi \cap \mathbb{Z}^2$; on the other hand, we can see $C \cap \mathbb{Z}^2$ as a finite disjoint union of copies of S , shifted by the integer points of Π :

$$C \cap \mathbb{Z}^2 = \bigcup_{\mathbf{s} \in S} (\mathbf{s} + (\Pi \cap \mathbb{Z}^2)) = \bigcup_{\mathbf{x} \in \Pi \cap \mathbb{Z}^2} (\mathbf{x} + S). \quad (17)$$

The set S is just the image of \mathbb{Z}_+^2 under the matrix $(\mathbf{b}_1, \mathbf{b}_2) = \begin{pmatrix} \alpha & \beta \\ -1 & 1 \end{pmatrix}$; cf. Figure 8. Now \mathbb{Z}_+^2 is the direct product of \mathbb{Z}_+ with itself, whose generating function is the geometric series $g(\mathbb{Z}_+; z) = z^0 + z^1 + z^2 + z^3 + \dots = \frac{1}{1-z}$. We thus obtain the generating function as a product, $g(\mathbb{Z}_+^2; z_1, z_2) = \frac{1}{1-z_1} \cdot \frac{1}{1-z_2}$. Applying the linear transformation $(\mathbf{b}_1, \mathbf{b}_2)$,

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

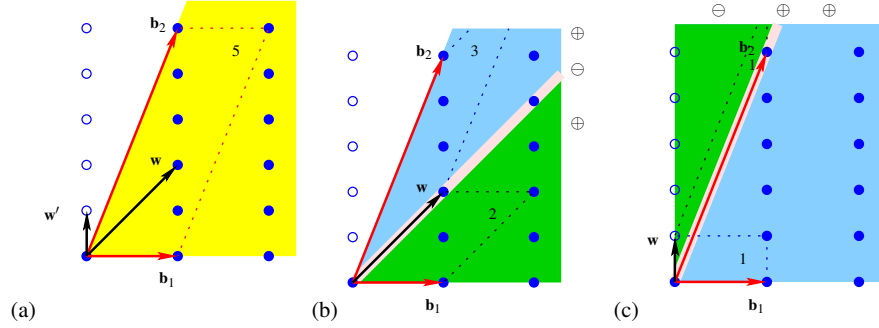


Fig. 9 (a) A cone of index 5 generated by \mathbf{b}^1 and \mathbf{b}^2 . (b) A triangulation of the cone into the two cones spanned by $\{\mathbf{b}^1, \mathbf{w}\}$ and $\{\mathbf{b}^2, \mathbf{w}\}$, having an index of 2 and 3, respectively. We have the inclusion-exclusion formula $g(\text{cone}\{\mathbf{b}_1, \mathbf{b}_2\}; \mathbf{z}) = g(\text{cone}\{\mathbf{b}_1, \mathbf{w}\}; \mathbf{z}) + g(\text{cone}\{\mathbf{b}_2, \mathbf{w}\}; \mathbf{z}) - g(\text{cone}\{\mathbf{w}\}; \mathbf{z})$; here the one-dimensional cone spanned by \mathbf{w} needed to be subtracted. (c) A signed decomposition into the two unimodular cones spanned by $\{\mathbf{b}^1, \mathbf{w}'\}$ and $\{\mathbf{b}^2, \mathbf{w}'\}$. We have the inclusion-exclusion formula $g(\text{cone}\{\mathbf{b}_1, \mathbf{b}_2\}; \mathbf{z}) = g(\text{cone}\{\mathbf{b}_1, \mathbf{w}'\}; \mathbf{z}) - g(\text{cone}\{\mathbf{b}_2, \mathbf{w}'\}; \mathbf{z}) + g(\text{cone}\{\mathbf{w}'\}; \mathbf{z})$.

From (17) it is now clear that $g(C; z_1, z_2) = \sum_{\mathbf{x} \in \Pi \cap \mathbb{Z}^2} z_1^{x_1} z_2^{x_2} g(S; z_1, z_2)$; 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 $\Pi \cap \mathbb{Z}^2 = \{(i, 0) : i = 0, \dots, \alpha + \beta - 1\}$. Thus

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

Unfortunately, this formula has an exponential size as the numerator contains $\alpha + \beta$ summands. To make the formula shorter, we need to recursively break the cone into “smaller” cones, each of which have a much shorter formula. We have observed that the length of the formula is determined by the number of integer points in the fundamental parallelepiped, the *index* of the cone. *Triangulations* usually do not help to reduce the index significantly, as another two-dimensional example shows. Consider the cone C' generated by $\mathbf{b}_1 = (1, 0)$ and $\mathbf{b}_2 = (1, \alpha)$; see Figure 9. We have $\Pi' \cap \mathbb{Z}^2 = \{(0, 0)\} \cup \{(1, i) : i = 1, \dots, \alpha - 1\}$, so the rational generating function would have α summands in the numerator, and thus have exponential size. 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 9, 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! The important new idea by Barvinok 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 9. Using these cones, we have the inclusion-exclusion formula

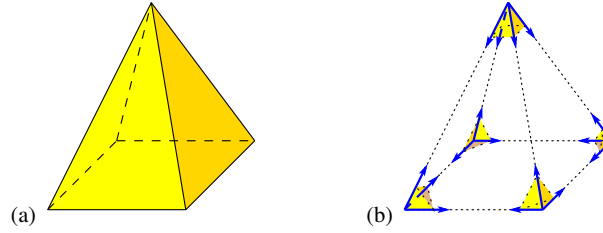


Fig. 10 Brion's theorem, expressing the generating function of a polyhedron to those of the supporting cones of all vertices

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

It turns out that all cones C_1 and C_2 are unimodular, and we obtain the rational generating function by summing up those of the subcones,

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

4.1.2 Barvinok's algorithm for short rational generating functions

We now present the general definitions and results. Let $P \subseteq \mathbb{R}^d$ be a rational polyhedron. We first define its *generating function* as the *formal Laurent series* $\tilde{g}(P; \mathbf{z}) = \sum_{\alpha \in P \cap \mathbb{Z}^d} \mathbf{z}^\alpha \in \mathbb{Z}[[z_1, \dots, z_d, z_1^{-1}, \dots, z_d^{-1}]]$, i.e., without any consideration of convergence properties. (A formal *power series* is not enough because monomials with negative exponents can appear.) As we remarked above, this encoding of a set of lattice points 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. Now if P is a polytope, then $\tilde{g}(P; \mathbf{z})$ is a *Laurent polynomial* (i.e., a *finite* sum of monomials with positive or negative integer exponents), so it can be naturally identified with a rational function $g(P; \mathbf{z})$. 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 a *pointed* polyhedron P , i.e., P does not contain a straight line.

Theorem 15. *Let $P \subseteq \mathbb{R}^d$ be a pointed rational polyhedron. Then there exists a non-empty open subset $U \subseteq \mathbb{C}^d$ such that the series $\tilde{g}(P; \mathbf{z})$ converges absolutely and uniformly on every compact subset of U to a rational function $g(P; \mathbf{z}) \in \mathbb{Q}(z_1, \dots, z_d)$.*

Finally, when P contains an integer point and also a straight line, there does not exist any point $\mathbf{z} \in \mathbb{C}^d$ where the series $\tilde{g}(P; \mathbf{z})$ converges absolutely. In this case we set $g(P; \mathbf{z}) = 0$; this turns out to be a consistent choice (making the map $P \mapsto g(P; \mathbf{z})$ a *valuation*, i.e., a finitely additive measure). The rational function $g(P; \mathbf{z}) \in \mathbb{Q}(z_1, \dots, z_d)$ defined as described above is called the *rational generating function* of $P \cap \mathbb{Z}^d$.

Theorem 16 (Brion [30]). *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 $C_P(\mathbf{v}) = \mathbf{v} + \text{cone}(P - \mathbf{v})$ is the supporting cone of the vertex \mathbf{v} ; see Figure 10.*

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.

Barvinok's algorithm computes the rational generating function of a polyhedron P as follows. By Brion's theorem, the rational generating function of a polyhedron can be expressed as the sum of the rational generating functions of the supporting cones of its vertices. Every supporting cone $\mathbf{v}_i + C_i$ can be triangulated to obtain simplicial cones $\mathbf{v}_i + C_{ij}$. If the dimension is fixed, these polyhedral computations all run in polynomial time.

Now let K be one of these simplicial cones, whose *basis vectors* $\mathbf{b}_1, \dots, \mathbf{b}_d$ (i.e., primitive representatives of its extreme rays) are the columns of some matrix $B \in \mathbb{Z}^{d \times d}$; then the index of K is $|\det B|$. Barvinok's algorithm now computes a *signed decomposition* of K to produce simplicial cones with smaller index. To this end, it constructs a vector $\mathbf{w} = \alpha_1 \mathbf{b}_1 + \dots + \alpha_d \mathbf{b}_d \in \mathbb{Z}^d \setminus \{\mathbf{0}\}$ with $|\alpha_i| \leq |\det B|^{-1/d}$. The existence of such a vector follows from Minkowski's first theorem, and it can be constructed in polynomial time using integer programming or lattice basis reduction followed by enumeration. The cone is then decomposed into cones spanned by d vectors from the set $\{\mathbf{b}_1, \dots, \mathbf{b}_d, \mathbf{w}\}$; each of the resulting cones then has an index at most $(\text{ind } K)^{(d-1)/d}$. In general, these cones form a signed decomposition of K ; only if \mathbf{w} lies inside K , they form a triangulation (see Figure 9). The resulting cones and their intersecting proper faces (arising in an inclusion-exclusion formula) are recursively processed, until cones of low index (for instance unimodular cones) are obtained. Finally, for a unimodular cone $\mathbf{v} + B\mathbb{R}_+^d$, the rational generating function is $\mathbf{z}^{\mathbf{a}} / \prod_{j=1}^d (1 - \mathbf{z}^{\mathbf{b}_j})$, where \mathbf{a} is the unique integer point in the fundamental parallelepiped. We summarize Barvinok's algorithm below.

Algorithm 2 (Barvinok's algorithm)

Input: A polyhedron $P \subset \mathbb{R}^d$ given by rational inequalities.

Output: The rational generating function for $P \cap \mathbb{Z}^d$ in the form

$$g_P(\mathbf{z}) = \sum_{i \in I} \varepsilon_i \frac{\mathbf{z}^{\mathbf{a}_i}}{\prod_{j=1}^d (1 - \mathbf{z}^{\mathbf{b}_{ij}})} \tag{18}$$

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

1. Compute all vertices \mathbf{v}_i and corresponding supporting cones C_i of P .
2. Triangulate C_i into simplicial cones C_{ij} , keeping track of all the intersecting proper faces.
3. Apply signed decomposition to the cones $\mathbf{v}_i + C_{ij}$ to obtain unimodular cones $\mathbf{v}_i + C_{ijl}$, keeping track of all the intersecting proper faces.
4. Compute the unique integer point \mathbf{a}_i in the fundamental parallelepiped of every resulting cone $\mathbf{v}_i + C_{ijl}$.
5. Write down the formula (18).

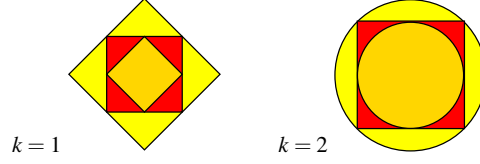


Fig. 11 Approximation properties of ℓ_k -norms

We remark that it is possible to avoid computations with the intersecting proper faces of cones (step 2 of the algorithm) entirely, using techniques such as polarization, irrational decomposition [80], or half-open decomposition [31, 81].

Due to the descent of the indices in the signed decomposition procedure, the depth of the decomposition tree is at most $\lceil 1 + \frac{\log_2 \log_2 D}{\log_2 \frac{d}{d-1}} \rceil$, where $D = |\det B|$. Because at each decomposition step at most $O(2^d)$ cones are created and the depth of the tree is doubly logarithmic in the index of the input cone, Barvinok could obtain a polynomiality result *in fixed dimension*:

Theorem 17 (Barvinok [12]). *Let d be fixed. There exists a polynomial-time algorithm for computing the rational generating function (18) of a polyhedron $P \subseteq \mathbb{R}^d$ given by rational inequalities.*

4.1.3 The FPTAS for polynomial optimization

We now describe the fully polynomial-time approximation scheme, which appeared in [41, 40, 42]. It makes use of the elementary relation

$$\max\{s_1, \dots, s_N\} = \lim_{k \rightarrow \infty} \sqrt[k]{s_1^k + \dots + s_N^k}, \quad (19)$$

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 $P \cap \mathbb{Z}^d$, let $\mathbf{x}^1, \dots, \mathbf{x}^N$ denote all the feasible integer solutions in $P \cap \mathbb{Z}^d$ and collect their objective function values $s_i = f(\mathbf{x}^i)$ in a vector $\mathbf{s} \in \mathbb{Q}^N$. Then, comparing the unit balls of the ℓ_k -norm and the ℓ_∞ -norm (Figure 11), we get the relation

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

Thus, for obtaining a good approximation of the maximum, it suffices to solve a summation problem of the polynomial function $h = f^k$ on $P \cap \mathbb{Z}^d$ for a value of k that is large enough. Indeed, for $k = \lceil (1 + 1/\varepsilon) \log N \rceil$, we obtain $U_k - L_k \leq \varepsilon f(\mathbf{x}^{\max})$. On the other hand, this choice of k is polynomial in the input size (because $1/\varepsilon$ 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.

Solving the summation problem can be accomplished using short rational generating functions as follows. Let $g(P; \mathbf{z})$ be the rational generating function of $P \cap \mathbb{Z}^d$, computed using Barvinok's algorithm. By symbolically applying differential operators to $g(P; \mathbf{z})$, we can compute a short rational function representation of the Laurent polynomial $g(P, h; \mathbf{z}) = \sum_{\alpha \in P \cap \mathbb{Z}^d} h(\alpha) \mathbf{z}^\alpha$, where each monomial \mathbf{z}^α corresponding to an integer point $\alpha \in P \cap \mathbb{Z}^d$ has a coefficient that is the value $h(\alpha)$. To illustrate this, consider again the generating function 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 $z \frac{d}{dz}$ and obtain

$$\left(z \frac{d}{dz} \right) g_P(z) = 1z^1 + 2z^2 + 3z^3 + 4z^4 = \frac{1}{(1-z)^2} - \frac{-4z^5 + 5z^4}{(1-z)^2}$$

Applying the same differential operator again, we obtain

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

We have thus evaluated the monomial function $h(\alpha) = \alpha^2$ for $\alpha = 0, \dots, 4$; the results appear as the coefficients of the respective monomials. The same works for several variables, using the partial differential operators $z_i \frac{\partial}{\partial z_i}$ for $i = 1, \dots, d$. In fixed dimension, the size of the rational function expressions occurring in the symbolic calculation can be bounded polynomially. Thus one obtains the following result.

Theorem 18. (a) Let $h(x_1, \dots, x_d) = \sum_{\beta} c_{\beta} \mathbf{x}^{\beta} \in \mathbb{Q}[x_1, \dots, x_d]$ be a polynomial. Define the differential operator

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

Then D_h maps the generating function $g(P; \mathbf{z}) = \sum_{\alpha \in P \cap \mathbb{Z}^d} \mathbf{z}^{\alpha}$ to the weighted generating function $(D_h g)(\mathbf{z}) = g(P, h; \mathbf{z}) = \sum_{\alpha \in P \cap \mathbb{Z}^d} h(\alpha) \mathbf{z}^{\alpha}$.

(b) Let the dimension d be fixed. Let $g(P; \mathbf{z})$ be the Barvinok representation of the generating function $\sum_{\alpha \in P \cap \mathbb{Z}^d} \mathbf{z}^{\alpha}$ of $P \cap \mathbb{Z}^d$. Let $h \in \mathbb{Q}[x_1, \dots, x_d]$ be a polynomial, given as a list of monomials with rational coefficients c_{β} encoded in binary and exponents β encoded in unary. We can compute in polynomial time a Barvinok representation $g(P, h; \mathbf{z})$ for the weighted generating function $\sum_{\alpha \in P \cap \mathbb{Z}^d} h(\alpha) \mathbf{z}^{\alpha}$.

Thus, we can implement the following algorithm in polynomial time.

Algorithm 3 (Computation of bounds for the optimal value)

Input: A rational convex polytope $P \subset \mathbb{R}^d$; a polynomial objective function $f \in \mathbb{Q}[x_1, \dots, x_d]$ that is non-negative over $P \cap \mathbb{Z}^d$, given as a list of monomials with

rational coefficients $c_{\boldsymbol{\beta}}$ encoded in binary and exponents $\boldsymbol{\beta}$ encoded in unary; an index k , encoded in unary.

Output: A lower bound L_k and an upper bound U_k for the maximal function value f^* of f over $P \cap \mathbb{Z}^d$. The bounds L_k form a nondecreasing, the bounds U_k a nonincreasing sequence of bounds that both reach f^* in a finite number of steps.

1. Compute a short rational function expression for the generating function $g(P; \mathbf{z}) = \sum_{\boldsymbol{\alpha} \in P \cap \mathbb{Z}^d} \mathbf{z}^{\boldsymbol{\alpha}}$. Using residue techniques, compute $|P \cap \mathbb{Z}^d| = g(P; \mathbf{1})$ from $g(P; \mathbf{z})$.
2. Compute the polynomial f^k from f .
3. From the rational function $g(P; \mathbf{z})$ compute the rational function representation of $g(P, f^k; \mathbf{z})$ of $\sum_{\boldsymbol{\alpha} \in P \cap \mathbb{Z}^d} f^k(\boldsymbol{\alpha}) \mathbf{z}^{\boldsymbol{\alpha}}$ by Theorem 18. Using residue techniques, compute

$$L_k := \left\lfloor \sqrt[k]{g(P, f^k; \mathbf{1}) / g(P; \mathbf{1})} \right\rfloor \quad \text{and} \quad U_k := \left\lceil \sqrt[k]{g(P, f^k; \mathbf{1})} \right\rceil.$$

Taking the discussion of the convergence of the bounds into consideration, one obtains the following result.

Theorem 19 (Fully polynomial-time approximation scheme). *Let the dimension d be fixed. Let $P \subset \mathbb{R}^d$ be a rational convex polytope. Let f be a polynomial with rational coefficients that is non-negative on $P \cap \mathbb{Z}^d$, given as a list of monomials with rational coefficients $c_{\boldsymbol{\beta}}$ encoded in binary and exponents $\boldsymbol{\beta}$ encoded in unary.*

- (i) *Algorithm 3 computes the bounds L_k, U_k in time polynomial in k , the input size of P and f , and the total degree D . The bounds satisfy the following inequality:*

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

- (ii) *For $k = (1 + 1/\varepsilon) \log(|P \cap \mathbb{Z}^d|)$ (a number bounded by a polynomial in the input size), L_k is a $(1 - \varepsilon)$ -approximation to the optimal value f^* and it can be computed in time polynomial in the input size, the total degree D , and $1/\varepsilon$. Similarly, U_k gives a $(1 + \varepsilon)$ -approximation to f^* .*
- (iii) *With the same complexity, by iterated bisection of P , we can also find a feasible solution $\mathbf{x}_\varepsilon \in P \cap \mathbb{Z}^d$ with $|f(\mathbf{x}_\varepsilon) - f^*| \leq \varepsilon f^*$.*

The mixed-integer case can be handled by *discretization* of the continuous variables. We illustrate on an example that one needs to be careful to pick a sequence of discretizations that actually converges. Consider the mixed-integer *linear* optimization problem depicted in Figure 12, whose feasible region consists of the point $(\frac{1}{2}, 1)$ and the segment $\{(x, 0) : x \in [0, 1]\}$. The unique optimal solution is $x = \frac{1}{2}, z = 1$. Now consider the sequence of grid approximations where $x \in \frac{1}{m} \mathbb{Z}_{\geq 0}$. 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 because it has two limit points; see Figure 12.

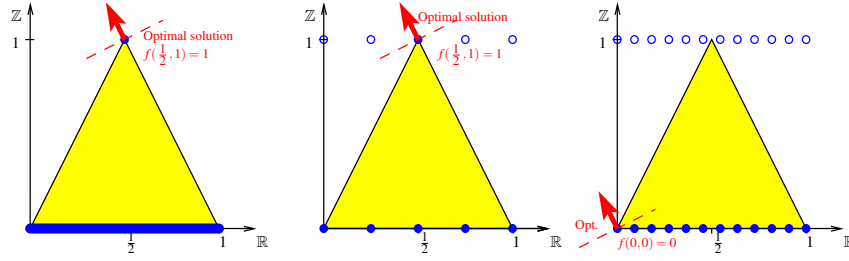


Fig. 12 A mixed-integer linear optimization problem and a sequence of optimal solutions to grid problems with two limit points, for even m and for odd m

To handle polynomial objective functions that take arbitrary (positive and negative) values on the feasible region, one can shift the objective function by a constant that is large enough. Then, to obtain a strong approximation result, one iteratively reduces the constant by a factor. Altogether we have the following result.

Theorem 20 (Fully polynomial-time approximation schemes). *Let the dimension $n = n_1 + n_2$ be fixed. Let an optimization problem (2) of a polynomial function f over the mixed-integer points of a polytope P and an error bound ϵ be given, where*

- (I₁) f is given as a list of monomials with rational coefficients c_{β} encoded in binary and exponents β encoded in unary,
- (I₂) P is given by rational inequalities in binary encoding,
- (I₃) the rational number $\frac{1}{\epsilon}$ is given in unary encoding.

(a) *There exists a fully polynomial time approximation scheme (FPTAS) for the maximization problem for all polynomial functions $f(\mathbf{x}, \mathbf{z})$ that are non-negative on the feasible region. That is, there exists a polynomial-time algorithm that, given the above data, computes a feasible solution $(\mathbf{x}_{\epsilon}, \mathbf{z}_{\epsilon}) \in P \cap (\mathbb{R}^{n_1} \times \mathbb{Z}^{n_2})$ with*

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

(b) *There exists a polynomial-time algorithm that, given the above data, computes a feasible solution $(\mathbf{x}_{\epsilon}, \mathbf{z}_{\epsilon}) \in P \cap (\mathbb{R}^{n_1} \times \mathbb{Z}^{n_2})$ with*

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

4.2 Semi-algebraic sets and SOS programming

In this section we use results from algebraic geometry over the reals to provide a convergent (and in the case of binary optimization, finite) sequence of semi-definite

relaxations for the general polynomial optimization problem over semi-algebraic sets.

$$\begin{aligned} Z^* = \text{minimize } & f(\mathbf{x}) \\ \text{s.t. } & g_i(\mathbf{x}) \geq 0, \quad i = 1, \dots, m, \\ & \mathbf{x} \in \mathbb{R}^n, \end{aligned} \quad (20)$$

where $f, g_i \in \mathbb{R}[\mathbf{x}]$ are polynomials defined as:

$$f(\mathbf{x}) = \sum_{\alpha \in \mathbb{Z}_+^n} f_{\alpha} \mathbf{x}^{\alpha}, \quad g_i(\mathbf{x}) = \sum_{\alpha \in \mathbb{Z}_+^n} g_{i,\alpha} \mathbf{x}^{\alpha},$$

where there are only finitely many nonzero coefficients f_{α} and $g_{i,\alpha}$. Moreover, let $K = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \geq 0, i = 1, \dots, m\}$ denote the set of feasible solutions. Note that problem (20) can model binary optimization, by taking $f(\mathbf{x}) = \mathbf{c}^{\top} \mathbf{x}$, and taking as the polynomials $g_i(\mathbf{x}), \mathbf{a}_i^{\top} \mathbf{x} - b_i, x_j^2 - x_j$ and $-x_j^2 + x_j$ (to model $x_j^2 - x_j = 0$). Problem (20) can also model bounded integer optimization (using for example the equation $(x_j - l_j)(x_j - l_j + 1) \cdots (x_j - u_j) = 0$ to model $l_j \leq x_j \leq u_j$), as well as bounded mixed-integer nonlinear optimization. Problem (20) can be written as:

$$\begin{aligned} \text{maximize } & \gamma \\ \text{s.t. } & f(\mathbf{x}) - \gamma \geq 0, \quad \forall \mathbf{x} \in K. \end{aligned} \quad (21)$$

This leads us to consider conditions for polynomials to be nonnegative over a set K .

Definition 4. Let $p \in \mathbb{R}[\mathbf{x}]$ where $\mathbf{x} = (x_1, \dots, x_n)^{\top}$. The polynomial p is called *sos* (sum of squares), if there exist polynomials $h_1, \dots, h_k \in \mathbb{R}[\mathbf{x}]$ such that $p = \sum_{i=1}^k h_i^2$.

Clearly, in multiple dimensions if a polynomial can be written as a sum of squares of other polynomials, then it is nonnegative. However, is it possible for a polynomial in higher dimensions to be nonnegative without being a sum of squares? The answer is yes. The most well-known example is probably the Motzkin-polynomial $M(x, y, z) = x^4 y^2 + x^2 y^4 + z^6 - 3x^2 y^2 z^2$, which is nonnegative without being a sum of squares of polynomials.

The following theorem establishes a certificate of positivity of a polynomial on the set K , under a certain assumption on K .

Theorem 21 ([106] [74]). *Suppose that the set K is compact and there exists a polynomial $h(\mathbf{x})$ of the form*

$$h(\mathbf{x}) = h_0(\mathbf{x}) + \sum_{i=1}^m h_i(\mathbf{x}) g_i(\mathbf{x}),$$

such that $\{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) \geq 0\}$ is compact and $h_i(\mathbf{x}), i = 0, 1, \dots, m$, are polynomials that have a sum of squares representation. Then, if the polynomial g is strictly positive over K , then there exist $p_i \in \mathbb{R}[\mathbf{x}], i = 0, 1, \dots, m$, that are sums of squares such that

$$g(\mathbf{x}) = p_0(\mathbf{x}) + \sum_{i=1}^m p_i(\mathbf{x}) g_i(\mathbf{x}). \quad (22)$$

Note that the number of terms in Equation (22) is linear. While the assumption of Theorem 21 may seem restrictive, it is satisfied in several cases:

- (a) For binary optimization problems, that is, when K includes the inequalities $x_j^2 \geq x_j$ and $x_j \geq x_j^2$ for all $j = 1, \dots, n$.
- (b) If all the g_j 's are linear, i.e., K is a polyhedron.
- (c) If there is one polynomial g_k such that the set $\{\mathbf{x} \in \mathbb{R}^n \mid g_k(\mathbf{x}) \geq 0\}$ is compact.

More generally, one way to ensure that the assumption of Theorem 21 holds is to add to K the extra quadratic constraint $g_{m+1}(\mathbf{x}) = a^2 - \|\mathbf{x}\|^2 \geq 0$ for some a sufficiently large. It is also important to emphasize that we do not assume that K is convex. Notice that it may even be disconnected.

Let us now investigate algorithmically when a polynomial is a sum of squares. As we will see this question is strongly connected to semi-definite optimization. The idea of using semi-definite optimization for solving optimization problems over polynomials is due to [120] and further expanded in [82] and [104]. We consider the vector

$$\mathbf{v}_d(\mathbf{x}) = (\mathbf{x}^\alpha)_{|\alpha| \leq d} = (1, x_1, \dots, x_n, x_1^2, x_1x_2, \dots, x_{n-1}x_n, x_n^2, \dots, x_1^d, \dots, x_n^d)^\top,$$

of all the monomials \mathbf{x}^α of degree less than or equal to d , which has dimension $s = \sum_{i=0}^d \binom{n}{i} = \binom{n+d}{d}$.

Proposition 3 ([37]). *The polynomial $g(\mathbf{x})$ of degree $2d$ has a sum of squares decomposition if and only if there exists a positive semi-definite matrix Q for which $g(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})^\top Q \mathbf{v}_d(\mathbf{x})$.*

Proof. Suppose there exists an $s \times s$ matrix $Q \succeq 0$ for which $g(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})^\top Q \mathbf{v}_d(\mathbf{x})$. Then $Q = HH^\top$ for some $s \times k$ matrix H , and thus,

$$g(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})^\top HH^\top \mathbf{v}_d(\mathbf{x}) = \sum_{i=1}^k (H^\top \mathbf{v}_d(\mathbf{x}))_i^2.$$

Because $(H^\top \mathbf{v}_d(\mathbf{x}))_i$ is a polynomial, then $g(\mathbf{x})$ is expressed as a sum of squares of the polynomials $(H^\top \mathbf{v}_d(\mathbf{x}))_i$.

Conversely, suppose that $g(\mathbf{x})$ has a sum of squares decomposition $g(\mathbf{x}) = \sum_{i=1}^\ell h_i(\mathbf{x})^2$. Let \mathbf{h}_i be the vector of coefficients of the polynomial $h_i(\mathbf{x})$, i.e., $h_i(\mathbf{x}) = \mathbf{h}_i^\top \mathbf{v}_d(\mathbf{x})$. Thus,

$$g(\mathbf{x}) = \sum_{i=1}^\ell \mathbf{v}_d(\mathbf{x})^\top \mathbf{h}_i \mathbf{h}_i^\top \mathbf{v}_d(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})^\top Q \mathbf{v}_d(\mathbf{x}),$$

with $Q = \sum_{i=1}^\ell \mathbf{h}_i \mathbf{h}_i^\top \succeq 0$, and the proposition follows. \square

Proposition 3 gives rise to an algorithm. Given a polynomial $f(\mathbf{x}) \in \mathbb{R}[x_1, \dots, x_n]$ of degree $2d$. In order to compute the minimum value $f^* = \min\{f(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n\}$ we introduce an artificial variable λ and determine

$$\max\{\lambda \mid \lambda \in \mathbb{R}, f(\mathbf{x}) - \lambda \geq 0\}.$$

With the developments above, we realize that we can determine a lower bound for f^* by computing the value

$$p^{sos} = \max\{\lambda \mid \lambda \in \mathbb{R}, f(\mathbf{x}) - \lambda \text{ is sos}\} \leq f^*.$$

The latter task can be accomplished by setting up a semi-definite program. In fact, if we denote by f_α the coefficient of the monomial \mathbf{x}^α in the polynomial f , then $f(\mathbf{x}) - \lambda$ is sos if and only if there exists an $s \times s$ matrix $Q \succeq 0$ for which $f(\mathbf{x}) - \lambda = \mathbf{v}_d(\mathbf{x})^\top Q \mathbf{v}_d(\mathbf{x})$. Now we can compare the coefficients on both sides of the latter equation. This leads to the SOS-program

$$\begin{aligned} p^{sos} = \max \lambda \\ \text{st. } f_0 - \lambda &= Q_{\mathbf{0}, \mathbf{0}} \\ \sum_{\beta, \gamma, \beta + \gamma = \alpha} Q_{\beta, \gamma} &= f_\alpha \\ Q = (Q_{\beta, \gamma})_{\beta, \gamma} &\succeq 0. \end{aligned}$$

In a similar vein, Theorem 21 and Proposition 3 jointly imply that we can use semi-definite optimization to provide a sequence of semi-definite relaxations for the optimization problem (21). Assuming that the set K satisfies the assumption of Theorem 21, then if $f(\mathbf{x}) - \gamma > 0$ for all $\mathbf{x} \in K$, then

$$f(\mathbf{x}) - \gamma = p_0(\mathbf{x}) + \sum_{i=1}^m p_i(\mathbf{x})g_i(\mathbf{x}), \quad (23)$$

where $p_i(\mathbf{x})$, $i = 0, 1, \dots, m$ have a sum of squares representation. Theorem 21 does not specify the degree of the polynomials $p_i(\mathbf{x})$. Thus, we select a bound $2d$ on the degree of the polynomials $p_i(\mathbf{x})$, and we apply Proposition 3 to each of the polynomials $p_i(\mathbf{x})$, that is, $p_i(\mathbf{x})$ is a sum of squares if and only if $p_i(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})^\top Q_i \mathbf{v}_d(\mathbf{x})$ with $Q_i \succeq 0$, $i = 0, 1, \dots, m$. Substituting to Eq. (23), we obtain that γ, Q_i , $i = 0, 1, \dots, m$, satisfy linear equations that we denote as $L(\gamma, Q_0, Q_1, \dots, Q_m) = 0$. Thus, we can find a lower bound to problem (20) by solving the semi-definite optimization problem

$$\begin{aligned} Z_d = \max \gamma \\ \text{st. } L(\gamma, Q_0, Q_1, \dots, Q_m) &= 0, \\ Q_i \succeq 0, i = 0, 1, \dots, m. \end{aligned} \quad (24)$$

Problem (24) involves semi-definite optimization over $m + 1$ $s \times s$ matrices. From the construction we get the relation $Z_d \leq Z^*$. It turns out that as d increases, Z_d converges to Z^* . Moreover, for binary optimization, there exists a finite d for which $Z_d = Z^*$ [83].

Problem (24) provides a systematic way to find convergent semi-definite relaxations to problem (20). While the approach is both general (it applies to very general nonconvex problems including nonlinear mixed-integer optimization problems) and insightful from a theoretical point of view, it is only practical for values of $d = 1, 2$,

as large scale semi-definite optimization problems cannot be solved in practice. In many situations, however, Z_1 or Z_2 provide strong bounds. Let us consider an example.

Example 2. Let us minimize $f(x_1, x_2) = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$ over \mathbb{R}^2 . We attempt to write

$$\begin{aligned} f(x_1, x_2) &= 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4 \\ &= \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{pmatrix}^\top \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{pmatrix} \\ &= q_{11}x_1^4 + q_{22}x_2^4 + (q_{13} + 2q_{12})x_1^2x_2^2 + 2q_{13}x_1^3x_2 + 2q_{23}x_1x_2^3. \end{aligned}$$

In order to have an identity, we obtain

$$q_{11} = 2, \quad q_{22} = 5, \quad q_{33} + 2q_{12} = -1, \quad 2q_{13} = 2, \quad q_{23} = 0.$$

Using semi-definite optimization, we find a particular solution such that $Q \succeq 0$ is given by

$$Q = \begin{bmatrix} 2 & -3 & 1 \\ -3 & 5 & 0 \\ 1 & 0 & 5 \end{bmatrix} = HH^\top, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & 0 \\ -3 & 1 \\ 1 & 3 \end{bmatrix}.$$

It follows that $f(x_1, x_2) = \frac{1}{2}(2x_1^2 - 3x_2^2 + x_1x_2)^2 + \frac{1}{2}(x_2^2 + 3x_1x_2)^2$, and thus the optimal solution value is $\gamma^* = 0$ and the optimal solution is $x_1^* = x_2^* = 0$.

4.3 Quadratic functions

In this section, we focus on instances of polynomial programming where the functions are all quadratic. The specific form of the *mixed-integer quadratically constrained programming* problem that we consider is

$$\begin{aligned} \min \quad & q_0(\mathbf{x}) \\ \text{s.t.} \quad & q(\mathbf{x}) \leq 0 \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} && \text{(MIQCP}[\mathbf{l}, \mathbf{u}]) \\ & x_i \in \mathbb{R} && \text{for } i = 1, \dots, k \\ & x_i \in \mathbb{Z} && \text{for } i = k + 1, \dots, n, \end{aligned}$$

where $q_0: \mathbb{R}^n \rightarrow \mathbb{R}$ and $q: \mathbb{R}^n \rightarrow \mathbb{R}^m$ are quadratic, $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^n$, and $\mathbf{l} \leq \mathbf{u}$. We denote the continuous relaxation by (MIQCP $_{\mathbb{R}}$ [\mathbf{l}, \mathbf{u}]). We emphasize that we are not generally making any convexity/concavity assumptions on the quadratic functions q_i , so when we do require any such assumptions we will state so explicitly.

Of course one can write a binary constraint $y_i \in \{0, 1\}$ as the (nonconvex) quadratic inequality $y_i(1 - y_i) \leq 0$ in the bound-constrained variable $0 \leq y_i \leq 1$. So, in this way, the case of binary variables y_i can be seen as the special case of (MIQCP[\mathbf{l}, \mathbf{u}]) with no discrete variables (i.e., $k = n$). So, in a sense, the topic of *mixed-binary quadratically constrained programming* can be seen as a special case of (*purely continuous*) *quadratically constrained programming*. We are not saying that it is necessarily useful to do this from a computational viewpoint, but it makes it clear that the scope of even the purely continuous quadratic model includes quadratic models having both binary and continuous variables, and in particular mixed- $\{0, 1\}$ linear programming.

In addition to the natural mathematical interest in studying mixed-integer quadratically constrained programming, there is a wealth of applications that have motivated the development of practical approaches; for example: Trimloss problems (see [84], for example), portfolio optimization (see [26], for example), Max-Cut and other binary quadratic models (see [108, 109] and the references therein).

In the remainder of this section, we describe some recent work on practical computational approaches to nonconvex quadratic optimization models. Rather than attempt a detailed survey, our goal is to present a few recent and promising techniques. One could regard these techniques as belonging more to the field of global optimization, but in Section 5 we present material on global optimization aimed at more general unstructured nonlinear integer programming problems.

4.3.1 Disjunctive programming

It is not surprising that integer variables in a mixed-integer *nonlinear* program can be treated with disjunctive programming [7, 9]. A corresponding branch-and-cut method was first described in [123] in the context of 0/1 mixed convex programming.

Here we describe an intriguing result from [114, 115, 116], which shows that one can also make useful disjunctions from nonconvex quadratic functions in a mixed-integer quadratically-constrained programming problem or even in a purely continuous quadratically-constrained programming problem. The starting point for this approach is that we can take a quadratic form $\mathbf{x}^\top A_i \mathbf{x}$ in $\mathbf{x} \in \mathbb{R}^n$, and rewrite it via an extended formulation as the linear form $\langle A_i, \mathbf{X} \rangle$, using the matrix variable $\mathbf{X} \in \mathbb{R}^{n \times n}$, and the nonlinear equation $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$. The standard approach is to relax $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$ to the convex inequality $\mathbf{X} \succeq \mathbf{x}\mathbf{x}^\top$. But the approach of [114, 115, 116] involves working with the nonconvex inequality $\mathbf{X} \preceq \mathbf{x}\mathbf{x}^\top$. This basic idea is as follows. Let $\mathbf{v} \in \mathbb{R}^n$ be arbitrary (for now). We have the equation

$$\langle \mathbf{v}\mathbf{v}^\top, \mathbf{X} \rangle = \langle \mathbf{v}\mathbf{v}^\top, \mathbf{x}\mathbf{x}^\top \rangle = (\mathbf{v}^\top \mathbf{x})^2,$$

which we relax as the *concave* inequality

$$(\mathbf{v}^\top \mathbf{x})^2 \geq \langle \mathbf{v}\mathbf{v}^\top, \mathbf{X} \rangle. \quad (\Omega)$$

If we have a point $(\hat{\mathbf{x}}, \hat{\mathbf{X}})$ that satisfies the convex inequality $\mathbf{X} \succeq \mathbf{x}\mathbf{x}^\top$, but for which $\hat{\mathbf{X}} \neq \hat{\mathbf{x}}\hat{\mathbf{x}}^\top$, then it is the case that $\hat{\mathbf{X}} - \hat{\mathbf{x}}\hat{\mathbf{x}}^\top$ has a positive eigenvalue λ . Let \mathbf{v} denote a unit-length eigenvector belonging to λ . Then

$$\begin{aligned} \lambda &= \lambda \|\mathbf{v}\|_2^2 \\ &= \langle \mathbf{v}\mathbf{v}^\top, \hat{\mathbf{X}} - \hat{\mathbf{x}}\hat{\mathbf{x}}^\top \rangle. \end{aligned}$$

So, $\lambda > 0$ if and only if $(\mathbf{v}^\top \hat{\mathbf{x}})^2 < \langle \mathbf{v}\mathbf{v}^\top, \hat{\mathbf{X}} \rangle$. That is, every positive eigenvalue of $\hat{\mathbf{X}} - \hat{\mathbf{x}}\hat{\mathbf{x}}^\top$ yields an inequality of the form (Ω) that is violated by $(\hat{\mathbf{x}}, \hat{\mathbf{X}})$. Next, we make a disjunction on this violated nonconvex inequality (Ω) . First, we choose a suitable polyhedral relaxation \mathcal{P} of the feasible region, and we let $[\eta_L, \eta_U]$ be the range of $\mathbf{v}^\top \mathbf{x}$ as (\mathbf{x}, \mathbf{X}) varies over the relaxation \mathcal{P} . Next, we choose a value $\theta \in (\eta_L, \eta_U)$ (e.g., the midpoint), and we get the polyhedral disjunction:

$$\left\{ (\mathbf{x}, \mathbf{X}) \in \mathcal{P} : \begin{array}{l} \eta_L(\mathbf{v}) \leq \mathbf{v}^\top \mathbf{x} \leq \theta \\ (\mathbf{v}^\top \mathbf{x})(\eta_L(\mathbf{v}) + \theta) - \theta \eta_L(\mathbf{v}) \geq \langle \mathbf{v}\mathbf{v}^\top, \mathbf{X} \rangle \end{array} \right\}$$

or

$$\left\{ (\mathbf{x}, \mathbf{X}) \in \mathcal{P} : \begin{array}{l} \theta \leq \mathbf{v}^\top \mathbf{x} \leq \eta_U(\mathbf{v}) \\ (\mathbf{v}^\top \mathbf{x})(\eta_U(\mathbf{v}) + \theta) - \theta \eta_U(\mathbf{v}) \geq \langle \mathbf{v}\mathbf{v}^\top, \mathbf{X} \rangle. \end{array} \right\}.$$

Notice that the second part of the first (resp., second) half of the disjunction corresponds to a secant inequality over the interval between the point θ and the lower (resp., upper) bound for $\mathbf{v}^\top \mathbf{x}$. Finally, we use the linear-programming technology of ordinary disjunctive programming to separate, via a linear inequality, the point $(\hat{\mathbf{x}}, \hat{\mathbf{X}})$ from the convex closure of the two halves of the disjunction. Details and extensions of this idea appear in [114, 115, 116].

4.3.2 Branch and cut

A branch-and-cut scheme for optimization of a nonconvex quadratic form over a box was recently developed by Vandebussche and Nemhauser [133, 132]. They use a formulation of Balas via linear programming with complementarity conditions, based on the necessary optimality conditions of continuous quadratic programming (see [8]). Specifically, they consider the problem

$$\begin{array}{ll} \min & \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & \mathbf{x} \in [0, 1]^n, \end{array} \quad (\text{BoxQP}[\mathbf{Q}, \mathbf{c}])$$

where \mathbf{Q} is an $n \times n$ symmetric, non positive semi-definite matrix, and $\mathbf{c} \in \mathbb{R}^n$. The KKT necessary optimality conditions for $(\text{BoxQP}[\mathbf{Q}, \mathbf{c}])$ are

$$\mathbf{y} - Q\mathbf{x} - \mathbf{z} = \mathbf{c}, \quad (25)$$

$$\mathbf{y}^\top (\mathbf{1} - \mathbf{x}) = 0 \quad (26)$$

$$\mathbf{z}^\top \mathbf{x} = 0 \quad (27)$$

$$\mathbf{x} \in [0, 1]^n \quad (28)$$

$$\mathbf{y}, \mathbf{z} \in \mathbb{R}_+^n. \quad (29)$$

Vandenbussche and Nemhauser, appealing to a result of Balas, define $\mathcal{P}(Q, \mathbf{c})$ as the polyhedron defined as the *convex hull* of solutions to (25-29), and they work with the reformulation of (BoxQP) as the linear program

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{c}^\top \mathbf{x} + \frac{1}{2} \mathbf{1}^\top \mathbf{y} \\ \text{s.t.} \quad & (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathcal{P}(Q, \mathbf{c}). \end{aligned} \quad (\text{Balas}[Q, \mathbf{c}])$$

The main tactic of Vandenbussche and Nemhauser is to develop cutting planes for $\mathcal{P}(Q, c)$.

Burer and Vandenbussche pursue a similar direction, but they allow general polyhedral constraints and employ semi-definite relaxations [33].

4.3.3 Branch and bound

Linderoth also looks at quadratically-constrained programs that are not convex [91]. He develops a novel method for repeatedly partitioning the continuous feasible region into the Cartesian product of triangles and rectangles. What is particularly interesting is that to do this effectively, Linderoth develops convex envelopes of bilinear functions over rectangles and triangles (also see Anstreicher and Burer's paper [6]), and then he demonstrates that these envelopes involve hyperbolic constraints which can be reformulated as the second-order cone constraints. It is interesting to compare this with the similar use of second-order cone constraints for convex quadratics (see Section 3.3.2).

One can view the technique of Linderoth as being a specialized "Spatial Branch-and-Bound Algorithm." In Section 5 we will describe the Spatial Branch-and-Bound Algorithm for global optimization in its full generality.

5 Global optimization

In the present section we take up the subject of global optimization of rather general nonlinear functions. This is an enormous subject, and so we will point to just a couple of directions that we view as promising. On the practical side, in Section 5.1 we describe the Spatial Branch-and-Bound Algorithm which is one of the most successful computational approaches in this area. In Section 5.2, from the viewpoint of complexity theory, with a goal of trying to elucidate the boundary between tractable

and intractable, we describe some very recent work on global optimization of a very general class of nonlinear functions over an independence system.

5.1 *Spatial Branch-and-Bound*

In this section we address methods for global optimization of rather general mixed-integer nonlinear programs having non-convex relaxations. Again, to have any hope at all, we assume that the variables are bounded. There is a very large body of work on solution techniques in this space. We will not attempt to make any kind of detailed survey. Rather we will describe one very successful methodology, namely the *Spatial Branch-and-Bound Algorithm*. We refer to [122, 127] and the references therein.

The Spatial Branch-and-Bound Algorithm for mixed-integer nonlinear programming has many similarities to the ordinary branch-and-bound employed for the solution of mixed-integer linear programs, but there are many additional wrinkles. Moreover, the techniques can be integrated. In what follows, we will concentrate on how continuous nonlinearities are handled. We leave it to the reader to see how these techniques would be integrated with the associated techniques for mixed-integer linear programs.

One main difference with the mixed-integer linear case is that all nonlinear functions in a problem instance are symbolically and recursively decomposed via simple operators, until we arrive at simple functions. The simple operators should be in a limited library. For example: sum, product, quotient, exponentiation, power, logarithm, sine, cosine, absolute value. Such a decomposition is usually represented via a collection of rooted directed acyclic graphs. At each root is a nonlinear function occurring in the problem formulation. Leaves are constants, affine functions and atomic variables. Each non-leaf node is thought of as an auxiliary variable and also as representing a simple operator, and its children are the arguments of that operator.

An inequality constraint in the problem formulation can be thought of as a bounding interval on a root. In addition, the objective function is associated with a root, and so lower and upper bounds on the optimal objective value can also be thought of as a bounding interval on a root. Simple bounds on a variable in the problem formulation can be thought of as a bounding interval on a leaf. In this way, we have an extended-variable reformulation of the given problem.

Bounds are propagated up and down each such rooted directed acyclic graph via interval arithmetic and a library of convex envelopes or at least linear convex relaxations of the graphs of simple nonlinear operators acting on one or two variables on simple domains (intervals for univariate operators and simple polygons for bivariate operators). So, in this way, we have a now tractable convex or even linear relaxation of the extended-variable reformulation, and this is used to get a lower bound on the minimum objective value.

The deficiency in our relaxation is localized to the graphs of simple functions that are only approximated by convex sets. We can seek to improve bounds by branching

on the interval for a variable and reconvexifying on the subproblems. For example, we may have a variable v that is a convex function f in a variable w on an interval $[l, u]$. Then the convex envelope of the graph $G[l, u] := \{(v, w) : v = f(w)\}$ is precisely

$$\tilde{G}[l, u] := \left\{ (v, w) : f(w) \leq v \leq f(l) + \left(\frac{f(u) - f(l)}{u - l} \right) (w - l) \right\}.$$

We may find that at a solution of the relaxation, the values of the variables (u, v) , say $(\hat{v}, \hat{w}) \in \tilde{G}[l, u]$, are far from $G[l, u]$. In such a case, we can *branch* by choosing a point $b \in [l, u]$ (perhaps at or near \hat{v}), and forming two subproblems in which the bounding interval for v is amended as $[l, b]$ in one and $[b, u]$ in the other. The value in branching on such a continuous variable is that we now reconvexify on the subintervals, effectively replacing $\tilde{G}[l, u]$ with the smaller set $\tilde{G}[l, b] \cup \tilde{G}[b, u]$. In particular, if we did choose $b = \hat{v}$, then $(\hat{v}, \hat{w}) \notin \tilde{G}[l, b] \cup \tilde{G}[b, u]$, and so the algorithm makes some progress. We note that a lot of work has gone into good branching strategies (see [14] for example).

Finally, a good Spatial Branch-and-Bound procedure should have an effective strategy for finding good feasible solutions, so as to improve the objective upper bound (for minimization problems). A good rudimentary strategy is to take the solution of a relaxation as a starting point for a continuous nonlinear-programming solver aimed at finding a locally-optimal solution of the continuous relaxation (of either the original or extended-variable formulation). Then if a feasible solution to this relaxation is obtained and if it happens to have integer values for the appropriate variables, then we have an opportunity to update the objective value upper bound. Alternatively, one can use a solver aimed mainly at mixed-integer nonlinear programs having *convex* relaxation as a heuristic also from such a starting point. In fact, the Branch-and-Bound Algorithm in `Bonmin` has options aimed at giving good solutions from such a starting point, even for non-convex problems.

The Spatial Branch-and-Bound Algorithm relies on the rapid and tight convexification of simple functions on simple domains. Therefore, considerable work has gone into developing closed-form expressions for such envelopes. This type of work has paralleled some research in mixed-integer linear programming that has focused on determining convex hulls for simple constraints. Useful results include: univariate functions [2, 121, 35], univariate monomials of odd degree [89, 90], bilinear functions [5, 97], trilinear functions [98], so-called $(n - 1)$ -convex functions [73], and fractional terms [126]. Further relevant work includes algorithms exploiting variable transformations and appropriate convex envelopes and relaxations. For example, for the case of “signomials” (i.e., terms of the form $a_0 x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$, with $a_i \in \mathbb{R}$), see [105].

We do not make any attempt to exhaustively review available software for global optimization. Rather we just mention that state-of-the-art codes implementing a Spatial Branch-and-Bound Algorithm include `Baron` [112, 127, 128] and the new open-source code `Couenne` [14].

5.2 Boundary cases of complexity

Now, we shift our attention back to the viewpoint of complexity theory. Our goal is to sample a bit of the recent work that is aimed at revealing the boundary between tractable and intractable instances of nonlinear discrete optimization problems. We describe some very recent work on global optimization of a very general class of nonlinear functions over an independence system (see [85]). Other work in this vein includes [16, 17].

Specifically, we consider the problem of optimizing a nonlinear objective function over a weighted independence system presented by a linear-optimization oracle. While this problem is generally intractable, we are able to provide a polynomial-time algorithm that determines an “ r -best” solution for nonlinear functions of the total weight of an independent set, where r is a constant that depends on certain Frobenius numbers of the individual weights and is independent of the size of the ground set.

An *independence system* is a nonempty set of vectors $S \subseteq \{0, 1\}^n$ with the property that $\mathbf{x} \in \{0, 1\}^n$, $\mathbf{x} \leq \mathbf{y} \in S$ implies $\mathbf{x} \in S$. The general nonlinear optimization problem over a multiply-weighted independence system is as follows. Given an independence system $S \subseteq \{0, 1\}^n$, weight vectors $\mathbf{w}_1, \dots, \mathbf{w}_d \in \mathbb{Z}^n$, and a function $f: \mathbb{Z}^d \rightarrow \mathbb{R}$, find $\mathbf{x} \in S$ minimizing the objective $f(\mathbf{w}_1^\top \mathbf{x}, \dots, \mathbf{w}_d^\top \mathbf{x})$.

The representation of the objective in the above composite form has several advantages. First, for $d > 1$, it can naturally be interpreted as *multi-criteria optimization*: the d given weight vectors $\mathbf{w}_1, \dots, \mathbf{w}_d$ represent d different criteria, where the value of $\mathbf{x} \in S$ under criterion i is its i -th total weight $\mathbf{w}_i^\top \mathbf{x}$ and the objective is to minimize the “balancing” $f(\mathbf{w}_1^\top \mathbf{x}, \dots, \mathbf{w}_d^\top \mathbf{x})$ of the d given criteria by the given function f . Second, it allows us to classify nonlinear optimization problems into a hierarchy of increasing generality and complexity: at the bottom lies standard linear optimization, recovered with $d = 1$ and f the identity on \mathbb{Z} ; and at the top lies the problem of maximizing an arbitrary function, which is typically intractable, arising with $d = n$ and $\mathbf{w}_i = \mathbf{1}_i$ the i -th standard unit vector in \mathbb{Z}^n for all i .

The computational complexity of the problem depends on the number d of weight vectors, on the weights $w_{i,j}$, on the type of function f and its presentation, and on the type of independence system S and its presentation. For example, when S is a *matroid*, the problem can be solved in polynomial time for any fixed d , any $\{0, 1, \dots, p\}$ -valued weights $w_{i,j}$ with p fixed, and any function f presented by a *comparison oracle*, even when S is presented by a mere *membership oracle*, see [16]. Also, for example, when S consists of the *matchings* in a given bipartite graph G , the problem can be solved in polynomial time for any fixed d , any weights $w_{i,j}$ presented in unary, and any *concave* function f , see [18]; but on the other hand, for *convex* f , already with fixed $d = 2$ and $\{0, 1\}$ -valued weights $w_{i,j}$, the problem includes as a special case the *exact matching problem* whose complexity is long open [100, 102].

In view of the difficulty of the problem already for $d = 2$, we take a first step and concentrate on *nonlinear optimization over a (singly) weighted independence system*, that is, with $d = 1$, single weight vector $\mathbf{w} = (w_1, \dots, w_n)$, and univariate func-

tion $f : \mathbb{Z} \rightarrow \mathbb{R}$. The function f can be arbitrary and is presented by a *comparison oracle* that, queried on $\mathbf{x}, \mathbf{y} \in \mathbb{Z}$, asserts whether or not $f(\mathbf{x}) \leq f(\mathbf{y})$. The weights w_j take on values in a p -tuple $\mathbf{a} = (a_1, \dots, a_p)$ of positive integers. Without loss of generality we assume that $\mathbf{a} = (a_1, \dots, a_p)$ is *primitive*, by which we mean that the a_i are distinct positive integers whose greatest common divisor $\gcd(\mathbf{a}) := \gcd(a_1, \dots, a_p)$ is 1. The independence system S is presented by a *linear-optimization oracle* that, queried on vector $\mathbf{c} \in \mathbb{Z}^n$, returns an element $\mathbf{x} \in S$ that maximizes the linear function $\mathbf{c}^\top \mathbf{x} = \sum_{j=1}^n c_j x_j$. It turns out that this problem is already quite intriguing, and so we settle for an approximative solution in the following sense, that is interesting in its own right. For a nonnegative integer r , we say that $\mathbf{x}^* \in S$ is an *r -best solution* to the optimization problem over S if there are at most r better objective values attained by feasible solutions. In particular, a 0-best solution is optimal. Recall that the *Frobenius number* of a primitive \mathbf{a} is the largest integer $F(\mathbf{a})$ that is not expressible as a nonnegative integer combination of the a_i . We prove the following theorem.

Theorem 22. *For every primitive p -tuple $\mathbf{a} = (a_1, \dots, a_p)$, there is a constant $r(\mathbf{a})$ and an algorithm that, given any independence system $S \subseteq \{0, 1\}^n$ presented by a linear-optimization oracle, weight vector $\mathbf{w} \in \{a_1, \dots, a_p\}^n$, and function $f : \mathbb{Z} \rightarrow \mathbb{R}$ presented by a comparison oracle, provides an $r(\mathbf{a})$ -best solution to the nonlinear problem $\min\{f(\mathbf{w}^\top \mathbf{x}) : \mathbf{x} \in S\}$, in time polynomial in n . Moreover:*

1. *If a_i divides a_{i+1} for $i = 1, \dots, p-1$, then the algorithm provides an optimal solution.*
2. *For $p = 2$, that is, for $\mathbf{a} = (a_1, a_2)$, the algorithm provides an $F(\mathbf{a})$ -best solution.*

In fact, we give an explicit upper bound on $r(\mathbf{a})$ in terms of the Frobenius numbers of certain subtuples derived from \mathbf{a} . An interesting special case is that of $\mathbf{a} = (2, 3)$. Because $F(2, 3) = 1$, the solution provided by our algorithm in that case is either optimal or second best.

The proof of Theorem 22 is pretty technical, so we only outline the main ideas. Below we present a naïve solution strategy that does not directly lead to a good approximation. However, this naïve approach is used as a basic building block. One partitions the independence system into suitable pieces, to each of which a suitable refinement of the naïve strategy is applied separately. Considering the monoid generated by $\{a_1, \dots, a_p\}$ allows one to show that the refined naïve strategy applied to each piece gives a good approximation within that piece. In this way, the approximation quality $r(\mathbf{a})$ can be bounded as follows, establishing a proof to Theorem 22.

Lemma 6. *Let $\mathbf{a} = (a_1, \dots, a_p)$ be any primitive p -tuple. Then the following hold:*

1. *An upper bound on $r(\mathbf{a})$ is given by $r(\mathbf{a}) \leq (2 \max(\mathbf{a}))^p$.*
2. *For divisible \mathbf{a} , we have $r(\mathbf{a}) = 0$.*
3. *For $p = 2$, that is, for $\mathbf{a} = (a_1, a_2)$, we have $r(\mathbf{a}) = F(\mathbf{a})$.*

Before we continue, let us fix some notation. The *indicator* of a subset $J \subseteq N$ is the vector $\mathbf{1}_J := \sum_{j \in J} \mathbf{1}_j \in \{0, 1\}^n$, so that $\text{supp}(\mathbf{1}_J) = J$. Unless otherwise specified,

\mathbf{x} denotes an element of $\{0, 1\}^n$ and $\boldsymbol{\lambda}, \boldsymbol{\tau}, \mathbf{v}$ denote elements of \mathbb{Z}_+^p . Throughout, $\mathbf{a} = (a_1, \dots, a_p)$ is a *primitive* p -tuple. We will be working with weights taking values in \mathbf{a} , that is, vectors $\mathbf{w} \in \{a_1, \dots, a_p\}^n$. With such a weight vector w being clear from the context, we let $N_i := \{j \in N : w_j = a_i\}$ for $i = 1, \dots, p$, so that $N = \bigsqcup_{i=1}^p N_i$. For $\mathbf{x} \in \{0, 1\}^n$ we let $\lambda_i(\mathbf{x}) := |\text{supp}(\mathbf{x}) \cap N_i|$ for $i = 1, \dots, p$, and $\boldsymbol{\lambda}(\mathbf{x}) := (\lambda_1(\mathbf{x}), \dots, \lambda_p(\mathbf{x}))$, so that $\mathbf{w}^\top \mathbf{x} = \boldsymbol{\lambda}(\mathbf{x})^\top \mathbf{a}$. For integers $z, s \in \mathbb{Z}$ and a set of integers $Z \subseteq \mathbb{Z}$, we define $z + sZ := \{z + sx : x \in Z\}$.

Let us now present the naïve strategy to solve the univariate nonlinear problem $\min\{f(\mathbf{w}^\top \mathbf{x}) : \mathbf{x} \in S\}$. Consider a set $S \subseteq \{0, 1\}^n$, weight vector $\mathbf{w} \in \{a_1, \dots, a_p\}^n$, and function $f : \mathbb{Z} \rightarrow \mathbb{R}$ presented by a comparison oracle. Define the *image* of S under \mathbf{w} to be the set of values $\mathbf{w}^\top \mathbf{x}$ taken by elements of S ; we denote it by $\mathbf{w} \cdot S$.

We point out the following simple observation.

Proposition 4. *A necessary condition for any algorithm to find an r -best solution to the problem $\min\{f(\mathbf{w}^\top \mathbf{x}) : \mathbf{x} \in S\}$, where the function f is presented by a comparison oracle only, is that it computes all but at most r values of the image $\mathbf{w} \cdot S$ of S under \mathbf{w} .*

Note that this necessary condition is also sufficient for computing the *objective value* $f(\mathbf{w}^\top \mathbf{x}^*)$ of an r -best solution, but not for computing an actual r -best solution $\mathbf{x}^* \in S$, which may be harder. Any point $\bar{\mathbf{x}}$ attaining $\max\{\mathbf{w}^\top \mathbf{x} : \mathbf{x} \in S\}$ provides an approximation of the image given by

$$\{\mathbf{w}^\top \mathbf{x} : \mathbf{x} \leq \bar{\mathbf{x}}\} \subseteq \mathbf{w} \cdot S \subseteq \{0, 1, \dots, \mathbf{w}^\top \bar{\mathbf{x}}\}. \quad (30)$$

This suggests the following natural naïve strategy for finding an approximative solution to the optimization problem over an independence system S that is presented by a linear-optimization oracle.

Algorithm 4 (Naïve Strategy)

input Independence system $S \subseteq \{0, 1\}^n$ presented by a linear-optimization oracle, $f : \mathbb{Z} \rightarrow \mathbb{R}$ presented by a comparison oracle, and $\mathbf{w} \in \{a_1, \dots, a_p\}^n$
obtain $\bar{\mathbf{x}}$ attaining $\max\{\mathbf{w}^\top \mathbf{x} : \mathbf{x} \in S\}$ using the linear-optimization oracle for S
output \mathbf{x}^* as one attaining $\min\{f(\mathbf{w}^\top \mathbf{x}) : \mathbf{x} \leq \bar{\mathbf{x}}\}$ using the algorithm of Lemma 7 below.

Unfortunately, as the next example shows, the number of values of the image that are missing from the approximating set on the left-hand side of equation (30) cannot generally be bounded by any constant. So by Proposition 4, this strategy cannot be used *as is* to obtain a provably good approximation.

Example 3. Let $\mathbf{a} := (1, 2)$, $n := 4m$, $\mathbf{y} := \sum_{i=1}^{2m} \mathbf{1}_i$, $\mathbf{z} := \sum_{i=2m+1}^{4m} \mathbf{1}_i$, and $\mathbf{w} := \mathbf{y} + 2\mathbf{z}$, that is,

$$\mathbf{y} = (1, \dots, 1, 0, \dots, 0), \quad \mathbf{z} = (0, \dots, 0, 1, \dots, 1), \quad \mathbf{w} = (1, \dots, 1, 2, \dots, 2),$$

define f on \mathbb{Z} by

$$f(k) := \begin{cases} k, & k \text{ odd;} \\ 2m, & k \text{ even,} \end{cases}$$

and let S be the independence system

$$S := \{\mathbf{x} \in \{0, 1\}^n : \mathbf{x} \leq \mathbf{y}\} \cup \{\mathbf{x} \in \{0, 1\}^n : \mathbf{x} \leq \mathbf{z}\}.$$

Then the unique optimal solution of the linear-objective problem $\max\{\mathbf{w}^\top \mathbf{x} : \mathbf{x} \in S\}$ is $\bar{\mathbf{x}} := \mathbf{z}$, with $\mathbf{w}^\top \bar{\mathbf{x}} = 4m$, and therefore

$$\begin{aligned} \{\mathbf{w}^\top \mathbf{x} : \mathbf{x} \leq \bar{\mathbf{x}}\} &= \{2i : i = 0, 1, \dots, 2m\} \text{ and} \\ \mathbf{w} \cdot S &= \{i : i = 0, 1, \dots, 2m\} \cup \{2i : i = 0, 1, \dots, 2m\}. \end{aligned}$$

So all m odd values (i.e., $1, 3, \dots, 2m - 1$) in the image $\mathbf{w} \cdot S$ are missing from the approximating set $\{\mathbf{w}^\top \mathbf{x} : \mathbf{x} \leq \bar{\mathbf{x}}\}$ on the left-hand side of (30), and \mathbf{x}^* attaining $\min\{f(\mathbf{w}^\top \mathbf{x}) : \mathbf{x} \leq \bar{\mathbf{x}}\}$ output by the above strategy has objective value $f(\mathbf{w}^\top \mathbf{x}^*) = 2m$, while there are $m = \frac{n}{4}$ better objective values (i.e., $1, 3, \dots, 2m - 1$) attainable by feasible points (e.g., $\sum_{i=1}^k \mathbf{1}_i$, for $k = 1, 3, \dots, 2m - 1$).

Nonetheless, a more sophisticated refinement of the naïve strategy, applied repeatedly to several suitably chosen subsets of S rather than S itself, will lead to a good approximation. Note that the naïve strategy can be efficiently implemented as follows.

Lemma 7. *For every fixed p -tuple \mathbf{a} , there is a polynomial-time algorithm that, given univariate function $f : \mathbb{Z} \rightarrow \mathbb{R}$ presented by a comparison oracle, weight vector $\mathbf{w} \in \{a_1, \dots, a_p\}^n$, and $\bar{\mathbf{x}} \in \{0, 1\}^n$, solves $\min\{f(\mathbf{w}^\top \mathbf{x}) : \mathbf{x} \leq \bar{\mathbf{x}}\}$.*

Proof. Consider the following algorithm:

Algorithm 5 **input** function $f : \mathbb{Z} \rightarrow \mathbb{R}$ presented by a comparison oracle, $\mathbf{w} \in \{a_1, \dots, a_p\}^n$ and $\bar{\mathbf{x}} \in \{0, 1\}^n$

let $N_i := \{j : w_j = a_i\}$ and $\tau_i := \lambda_i(\bar{\mathbf{x}}) = |\text{supp}(\bar{\mathbf{x}}) \cap N_i|$, $i = 1, \dots, p$

For every choice of $\mathbf{v} = (v_1, \dots, v_p) \leq (\tau_1, \dots, \tau_p) = \boldsymbol{\tau}$

determine some $\mathbf{x}_{\mathbf{v}} \leq \bar{\mathbf{x}}$ with $\lambda_i(\mathbf{x}_{\mathbf{v}}) = |\text{supp}(\mathbf{x}_{\mathbf{v}}) \cap N_i| = v_i$, $i = 1, \dots, p$

output \mathbf{x}^* as one minimizing $f(\mathbf{w}^\top \mathbf{x})$ among the $\mathbf{x}_{\mathbf{v}}$ by using the comparison oracle of f .

As the value $\mathbf{w}^\top \mathbf{x}$ depends only on the cardinalities $|\text{supp}(\mathbf{x}) \cap N_i|$, $i = 1, \dots, p$, it is clear that

$$\{\mathbf{w}^\top \mathbf{x} : \mathbf{x} \leq \bar{\mathbf{x}}\} = \{\mathbf{w}^\top \mathbf{x}_{\mathbf{v}} : \mathbf{v} \leq \boldsymbol{\tau}\}.$$

Clearly, for each choice $\mathbf{v} \leq \boldsymbol{\tau}$ it is easy to determine some $\mathbf{x}_{\mathbf{v}} \leq \bar{\mathbf{x}}$ by zeroing out suitable entries of $\bar{\mathbf{x}}$. The number of choices $\mathbf{v} \leq \boldsymbol{\tau}$ and hence of loop iterations and comparison-oracle queries of f to determine \mathbf{x}^* is

$$\prod_{i=1}^p (\tau_i + 1) \leq (n + 1)^p.$$

Table 1 Computational complexity and algorithms for nonlinear integer optimization.

Objective function	Constraints		
	Linear	Convex Polynomial	Arbitrary Polynomial
Linear	<p>Polynomial-time in fixed dimension:</p> <ul style="list-style-type: none"> – Lenstra’s algorithm [87] – Generalized basis reduction, Lovász–Scarf [94] – Short rational generating functions, Barvinok [12] 	<p>Polynomial-time in fixed dimension:</p> <p>Lenstra-type algorithms (§ 3.1)</p> <ul style="list-style-type: none"> – Khachiyan–Porkolab [79] – Heinz [66] 	<p>Incomputable:</p> <p>Hilbert’s 10th problem, Matiyasevich [95] (§ 1), even for:</p> <ul style="list-style-type: none"> – quadratic constraints, Jeroslow [75] – fixed dimension 10, Matiyasevich [76]
Convex max (§ 2)	<p>Polynomial-time in fixed dimension: Cook et al. [38] (§ 2.1)</p>		<p>Incomputable (§ 1)</p>
Convex min (§ 3)	<p>Polynomial-time in fixed dimension: Lenstra-type algorithms: Khachiyan–Porkolab [79], Heinz [66] (§ 3.1)</p>		<p>Incomputable (§ 1)</p>
Arbitrary Polynomial (§ 4)	<p>NP-hard, inapproximable, even for quadratic forms over hypercubes: MAX-CUT, Hästad [65] (§ 1)</p> <p>NP-hard, even for fixed dimension 2, degree 4 (§ 1)</p> <p>FPTAS in fixed dimension: Short rational generating functions, De Loera et al. [41] (§ 4.1)</p>		<p>Incomputable (§ 1)</p>

6 Conclusions

In this chapter, we hope to have succeeded in reviewing mixed-integer nonlinear programming from two important viewpoints.

We have reviewed the computational complexity of several important classes of mixed-integer nonlinear programs. Some of the negative complexity results (incomputability, NP-hardness) that appeared in Section 1 have been supplemented by polynomiality or approximability results in fixed dimension. Table 1 gives a summary. In addition to that, and not shown in the table, we have explored the boundary between tractable and intractable problems, by highlighting interesting cases in varying dimension where still polynomiality results can be obtained.

Additionally, we have reviewed a selection of practical algorithms that seem to have the greatest potential from today’s point of view. Many of these algorithms, at their core, are aimed at *integer convex minimization*. Here we have nonlinear branch-and-bound, outer approximation, the Quesada–Grossman algorithm, hybrid

algorithms, and generalized Benders decomposition. As we have reported, such approaches can be specialized and enhanced for problems with SDP constraints, SOCP constraints, and (convex) quadratics. For *integer polynomial programming* (without convexity assumptions), the toolbox of Positivstellensätze and SOS programming is available. For the case of *quadratics* (without convexity assumptions), specialized versions of disjunctive programming, branch-and-cut, and branch-and-bound have been devised. Finally, for general *global optimization*, spatial branch-and-bound is available as a technique, which relies heavily on convexification methods.

It is our hope that, by presenting these two viewpoints to the interested reader, this chapter will help to create a synergy between both viewpoints in the near future. May this lead to a better understanding of the field, and to much better algorithms than what we have today!

Acknowledgments

We would like to thank our coauthors, Jesús De Loera and Shmuel Onn, for their permission to base the presentation of some of the material in this chapter on our joint papers [43, 86, 85].

References

1. Abhishek, K., Leyffer, S., Linderoth, J.: Filmint: An outer-approximation-based solver for nonlinear mixed integer programs. Preprint ANL/MCS-P1374-0906 (2006)
2. Adjiman, C.: Global optimization techniques for process systems engineering. Ph.D. thesis, Princeton University (1998)
3. Ahuja, R., Magnanti, T., Orlin, J.: Network flows: theory, algorithms, and applications. Prentice-Hall, Inc., New Jersey (1993)
4. Aktürk, S., Atamtürk, A., Gürel, S.: A strong conic quadratic reformulation for machine-job assignment with controllable processing times. *Oper. Res. Lett.* **37**(3), 187–191 (2009)
5. Al-Khayyal, F., Falk, J.: Jointly constrained biconvex programming. *Mathematics of Operations Research* **8**(2), 273–286 (1983)
6. Anstreicher, K., Burer, S.: Computable representations for convex hulls of low-dimensional quadratic forms. Tech. rep., Dept. of Management Sciences, University of Iowa (2007)
7. Balas, E.: Disjunctive programming: Properties of the convex hull of feasible points. MSRR No. 330, Carnegie Mellon University (Pittsburgh PA, 1974)
8. Balas, E.: Nonconvex quadratic programming via generalized polars. *SIAM J. Appl. Math.* **28**, 335–349 (1975)
9. Balas, E.: Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics* **89**, 3–44 (1998)
10. Bank, B., Heintz, J., Krick, T., Mandel, R., Solernó, P.: Une borne optimale pour la programmation entière quasi-convexe. *Bull. Soc. math. France* **121**, 299–314 (1993)
11. Bank, B., Krick, T., Mandel, R., Solernó, P.: A geometrical bound for integer programming with polynomial constraints. In: *Fundamentals of Computation Theory, Lecture Notes In Computer Science*, vol. 529, pp. 121–125. Springer-Verlag (1991)
12. Barvinok, A.I.: A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research* **19**, 769–779 (1994)

13. Barvinok, A.I., Pommersheim, J.E.: An algorithmic theory of lattice points in polyhedra. In: L.J. Billera, A. Björner, C. Greene, R.E. Simion, R.P. Stanley (eds.) *New Perspectives in Algebraic Combinatorics*, *Math. Sci. Res. Inst. Publ.*, vol. 38, pp. 91–147. Cambridge Univ. Press, Cambridge (1999)
14. Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex MINLP (2008). IBM Research Report RC24620
15. Ben-Tal, A., Nemirovski, A.: *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, USA (2001)
16. Berstein, Y., Lee, J., Maruri-Aguilar, H., Onn, S., Riccomagno, E., Weismantel, R., Wynn, H.: Nonlinear matroid optimization and experimental design. *SIAM Journal on Discrete Mathematics* **22**(3), 901–919 (2008)
17. Berstein, Y., Lee, J., Onn, S., Weismantel, R.: Nonlinear optimization for matroid intersection and extensions. IBM Research Report RC24610 (2008)
18. Berstein, Y., Onn, S.: Nonlinear bipartite matching. *Discrete Optimization* **5**, 53–65 (2008)
19. Bertsimas, D., Weismantel, R.: *Optimization over Integers*. Dynamic Ideas, Belmont, MA (2005)
20. Biq-Mac Solver - Binary quadratic and Max cut Solver: biqmac.uni-klu.ac.at (2006)
21. Bonami, P., Biegler, L., Conn, A., Cornuéjols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A.: An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* **5**, 186–204 (2008)
22. Bonami, P., Cornuéjols, G., Lodi, A., Margot, F.: A feasibility pump for mixed integer nonlinear programs (2008)
23. Bonami, P., Forrest, J., Lee, J., Wächter, A.: Rapid development of an MINLP solver with COIN-OR. *Optima* **75**, 1–5 (2007)
24. Bonami, P., Gonçalves, J.P.M.: Primal heuristics for mixed integer nonlinear programs (2008). IBM Research Report RC24639
25. Bonami, P., Lee, J.: *Bonmin users' manual*. Tech. rep. (June 2006)
26. Bonami, P., Lejeune, M.: An exact solution approach for integer constrained portfolio optimization problems under stochastic constraints. To appear in *Operations Research*
27. Bonmin: neos.mcs.anl.gov/neos/solvers/minco:Bonmin/AMPL.html
28. Bonmin: projects.coin-or.org/Bonmin (v. 0.99)
29. Boros, E., Hammer, P.L.: Pseudo-Boolean optimization. *Discrete Applied Mathematics* **123**, 155–225 (2002)
30. Brion, M.: Points entiers dans les polyèdres convexes. *Ann. Sci. École Norm. Sup.* **21**(4), 653–663 (1988)
31. Brion, M., Vergne, M.: Residue formulae, vector partition functions and lattice points in rational polytopes. *J. Amer. Math. Soc.* **10**, 797–833 (1997)
32. Buchheim, C., Rinaldi, G.: Efficient reduction of polynomial zero-one optimization to the quadratic case. *SIAM Journal on Optimization* **18**, 1398–1413 (2007)
33. Burer, S., Vandembussche, D.: A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Mathematical Programming Ser. A*, **113**, 259–282 (2008)
34. Burkard, R.E., Çela, E., Pitsoulis, L.: The quadratic assignment problem. In: *Handbook of Combinatorial Optimization, Computer-aided chemical engineering*, pp. 241–339. Kluwer Academic Publishers, Dordrecht (1998)
35. Ceria, S., Soares, J.: Convex programming for disjunctive convex optimization. *Mathematical Programming* **86**(3), 595–614 (1999). DOI 10.1007/s101070050106
36. Çezik, M.T., Iyengar, G.: Cuts for mixed 0-1 conic programming. *Mathematical Programming* **104**(1), 179–202 (2005). DOI 10.1007/s10107-005-0578-3
37. Choi, M.D., Lam, T.Y., Reznick, B.: Sums of squares of real polynomials. *Proceedings of symposia in pure mathematics* **58**(2), 103–126 (1995)
38. Cook, W.J., Hartmann, M.E., Kannan, R., McDiarmid, C.: On integer points in polyhedra. *Combinatorica* **12**(1), 27–37 (1992)

39. Corp., G.D.: DICOPT. www.gams.com/dd/docs/solvers/dicopt.pdf
40. De Loera, J.A., Hemmecke, R., Köppe, M., Weismantel, R.: FPTAS for mixed-integer polynomial optimization with a fixed number of variables. In: 17th ACM-SIAM Symposium on Discrete Algorithms, pp. 743–748 (2006)
41. De Loera, J.A., Hemmecke, R., Köppe, M., Weismantel, R.: Integer polynomial optimization in fixed dimension. *Mathematics of Operations Research* **31**(1), 147–153 (2006)
42. De Loera, J.A., Hemmecke, R., Köppe, M., Weismantel, R.: FPTAS for optimizing polynomials over the mixed-integer points of polytopes in fixed dimension. *Mathematical Programming, Series A* **118**, 273–290 (2008). DOI 10.1007/s10107-007-0175-8
43. De Loera, J.A., Hemmecke, R., Onn, S., Weismantel, R.: N-fold integer programming. *Disc. Optim.*, to appear (2008)
44. De Loera, J.A., Onn, S.: The complexity of three-way statistical tables. *SIAM Journal of Computing* **33**, 819–836 (2004)
45. De Loera, J.A., Onn, S.: All linear and integer programs are slim 3-way transportation programs. *SIAM Journal of Optimization* **17**, 806–821 (2006)
46. De Loera, J.A., Onn, S.: Markov bases of three-way tables are arbitrarily complicated. *Journal of Symbolic Computation* **41**, 173–181 (2006)
47. Drewes, S., Ulbrich, S.: Mixed integer second order cone programming. IMA Hot Topics Workshop, Mixed-Integer Nonlinear Optimization: Algorithmic Advances and Applications, November 17-21, 2008
48. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Programming* **36**(3), 307–339 (1986)
49. Duran, M.A., Grossmann, I.E.: Erratum: “An outer-approximation algorithm for a class of mixed-integer nonlinear programs” [*Math. Programming* **36** (1986), no. 3, 307–339]. *Math. Programming* **39**(3), 337 (1987)
50. FilMINT: www-neos.mcs.anl.gov/neos/solvers/minco:FilMINT/AMPL.html
51. Frangioni, A., Gentile, C.: Perspective cuts for a class of convex 0-1 mixed integer programs. *Math. Program.* **106**(2, Ser. A), 225–236 (2006)
52. Frangioni, A., Gentile, C.: A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes. *Oper. Res. Lett.* **37**(3), 206–210 (2009)
53. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, New York, NY (1979)
54. Geoffrion, A.M.: Generalized Benders decomposition. *J. Optimization Theory Appl.* **10**, 237–260 (1972)
55. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM* **42**, 1115–1145 (1995)
56. Goldfarb, D., Liu, S.C., Wang, S.Y.: A logarithmic barrier function algorithm for quadratically constrained convex quadratic programming. *SIAM J. Optim.* **1**(2), 252–267 (1991)
57. Gomory, R.E.: An algorithm for integer solutions to linear programs. Princeton IBM Mathematics Research Project, Technical Report No. 1, Princeton University (November 17, 1958)
58. Gomory, R.E.: Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society* **64**, 275–278 (1958)
59. Graver, J.E.: On the foundations of linear and integer linear programming I. *Mathematical Programming* **8**, 207–226 (1975)
60. Grötschel, M., Lovász, L., Schrijver, A.: *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, Germany (1988)
61. Günlük, O., Linderoth, J.: Perspective relaxation of mixed integer nonlinear programs with indicator variables. In: A. Lodi, A. Panconesi, G. Rinaldi (eds.) *Integer Programming and Combinatorial Optimization 2008 – Bertinoro, Italy, Lecture Notes in Computer Science*, vol. 5035, pp. 1–16. Springer Berlin / Heidelberg (2008)
62. Günlük, O., Linderoth, J.: Perspective reformulations of mixed integer nonlinear programs with indicator variables. Optimization Technical Report, ISyE Department, University of Wisconsin-Madison (June 20, 2008)

63. Gupta, O.K., Ravindran, A.: Branch and bound experiments in convex nonlinear integer programming. *Management Sci.* **31**(12), 1533–1546 (1985)
64. Hartmann, M.E.: Cutting planes and the complexity of the integer hull. Phd thesis, Cornell University, Department of Operations Research and Industrial Engineering, Ithaca, NY (1989)
65. Håstad, J.: Some optimal inapproximability results. In: *Proceedings of the 29th Symposium on the Theory of Computing (STOC)*, pp. 1–10. ACM (1997)
66. Heinz, S.: Complexity of integer quasiconvex polynomial optimization. *Journal of Complexity* **21**, 543–556 (2005)
67. Hemmecke, R.: On the positive sum property and the computation of Graver test sets. *Math. Programming, Series B* **96**, 247–269 (2003)
68. Hemmecke, R., Köppe, M., Weismantel, R.: Oracle-polynomial time convex mixed-integer minimization. Manuscript (2008)
69. Hemmecke, R., Onn, S., Weismantel, R.: A polynomial oracle-time algorithm for convex integer minimization. Manuscript (2008)
70. Hiriart-Urruty, J.B., Lemarchal, C.: *Convex analysis and minimization algorithms II: Advanced theory and bundle methods*. Grundlehren der Mathematischen Wissenschaften. 306. Berlin: Springer-Verlag (1993)
71. Hoşten, S., Sullivant, S.: A finiteness theorem for Markov bases of hierarchical models. *J. Combin. Theory Ser. A* **114**(2), 311–321 (2007)
72. Ilog-Cplex: www.ilog.com/products/cplex (v. 10.1)
73. Jach, M., Michaels, D., Weismantel, R.: The convex envelope of $(n-1)$ -convex functions. *SIAM Journal on Optimization* (to appear) (2008)
74. Jacobi, T., Prestel, A.: Distinguished representations of strictly positive polynomials. *J. Reine Angew. Math.* **532**, 223–235 (2001)
75. Jeroslow, R.G.: There cannot be any algorithm for integer programming with quadratic constraints. *Operations Research* **21**(1), 221–224 (1973)
76. Jones, J.P.: Universal diophantine equation. *Journal of Symbolic Logic* **47**(3), 403–410 (1982)
77. Kelley, Jr., J.E.: The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics* **8**(4), 703–712 (1960)
78. Khachiyan, L.G.: Convexity and complexity in polynomial programming. In: Z. Ciesielski, C. Olech (eds.) *Proceedings of the International Congress of Mathematicians, August 16–24, 1983, Warszawa*, pp. 1569–1577. North-Holland, New York (1984)
79. Khachiyan, L.G., Porkolab, L.: Integer optimization on convex semialgebraic sets. *Discrete and Computational Geometry* **23**(2), 207–224 (2000)
80. Köppe, M.: A primal Barvinok algorithm based on irrational decompositions. *SIAM Journal on Discrete Mathematics* **21**(1), 220–236 (2007). DOI 10.1137/060664768
81. Köppe, M., Verdoolaege, S.: Computing parametric rational generating functions with a primal Barvinok algorithm. *The Electronic Journal of Combinatorics* **15**, 1–19 (2008). #R16
82. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization* **11**, 796–817 (2001)
83. Laurent, M.: A comparison of the Sherali–Adams, Lovász–Schrijver and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research* **28**(3), 470–496 (2003)
84. Lee, J.: In situ column generation for a cutting-stock problem. *Computers & Operations Research* **34**(8), 2345–2358 (2007)
85. Lee, J., Onn, S., Weismantel, R.: Nonlinear optimization over a weighted independence system. IBM Research Report RC24513 (2008)
86. Lee, J., Onn, S., Weismantel, R.: On test sets for nonlinear integer maximization. *Operations Research Letters* **36**, 439–443 (2008)
87. Lenstra, H.W.: Integer programming with a fixed number of variables. *Mathematics of Operations Research* **8**, 538–548 (1983)
88. Leyffer, S.: User manual for MINLP_BB. Tech. rep., University of Dundee, UK (1999)

89. Liberti, L.: Comparison of convex relaxations for monomials of odd degree. In: I. Tseveendorj, P. Pardalos, R. Enkhbat (eds.) *Optimization and Optimal Control*. World Scientific (2003)
90. Liberti, L., Pantelides, C.: Convex envelopes of monomials of odd degree. *Journal of Global Optimization* **25**, 157–168 (2003)
91. Linderoth, J.: A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Mathematical Programming* **103**(2), 251–282 (2005)
92. Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H.: Applications of second-order cone programming. *Linear Algebra Appl.* **284**(1-3), 193–228 (1998). ILAS Symposium on Fast Algorithms for Control, Signals and Image Processing (Winnipeg, MB, 1997)
93. LOQO: www.princeton.edu/~rvdb (v. 4.05)
94. Lovász, L., Scarf, H.E.: The generalized basis reduction algorithm. *Mathematics of Operations Research* **17**(3), 751–764 (1992)
95. Matiyasevich, Yu.V.: Enumerable sets are diophantine. *Doklady Akademii Nauk SSSR* **191**, 279–282 (1970). (Russian); English translation, *Soviet Mathematics Doklady*, vol. 11 (1970), pp. 354–357
96. Matiyasevich, Yu.V.: *Hilbert’s tenth problem*. The MIT Press, Cambridge, MA, USA (1993)
97. McCormick, G.: Computability of global solutions to factorable nonconvex programs: Part i — convex underestimating problems. *Mathematical Programming* **10**, 146–175 (1976)
98. Meyer, C., Floudas, C.: Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes. *Journal of Global Optimization* **29**, 125–155 (2004)
99. MOSEK: www.mosek.com (v. 5.0)
100. Mulmuley, K., Vazirani, U., Vazirani, V.: Matching is as easy as matrix inversion. *Combinatorica* **7**, 105–113 (1987)
101. Onn, S., Rothblum, U.G.: Convex combinatorial optimization. *Disc. Comp. Geom.* **32**, 549–566 (2004)
102. Papadimitriou, C., Yanakakis, M.: The complexity of restricted spanning tree problems. *Journal of the Association for Computing Machinery* **29**, 285–309 (1982)
103. Pardalos, P.M., Rendl, F., Wolkowicz, H.: The quadratic assignment problem: A survey and recent developments. In: P.M. Pardalos, H. Wolkowicz (eds.) *Quadratic Assignment and Related Problems*, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 16, pp. 1–42. American Mathematical Society (1994)
104. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming* **96**, 293–320 (2003)
105. Pörn, R., Björk, K.M., Westerlund, T.: Global solution of optimization problems with signomial parts. *Discrete Optimization* **5**, 108–120 (2008)
106. Putinar, M.: Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal* **42**, 969–984 (1993)
107. Quesada, I., Grossmann, I.: An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers Chem. Engng.* **16**, 937–947 (1992)
108. Rendl, F., Rinaldi, G., Wiegele, A.: A branch and bound algorithm for max-cut based on combining semidefinite and polyhedral relaxations. In: M. Fischetti, D. Williamson (eds.) *Integer Programming and Combinatorial Optimization 2007 – Ithaca, New York*, *Lecture Notes in Computer Science*, vol. 4513, pp. 295–309. Springer Berlin / Heidelberg (2007)
109. Rendl, F., Rinaldi, G., Wiegele, A.: Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Tech. rep., Alpen-Adria-Universität Klagenfurt, Inst. f. Mathematik (2008)
110. Renegar, J.: On the computational complexity and geometry of the first-order theory of the reals. part III: Quantifier elimination. *Journal of Symbolic Computation* **13**(3), 329–352 (1992)
111. Renegar, J.: On the computational complexity of approximating solutions for real algebraic formulae. *SIAM Journal on Computing* **21**(6), 1008–1025 (1992)
112. Sahinidis, N.: BARON: A general purpose global optimization software package. *Journal of Global Optimization* **8**, 201–205 (1996)

113. Santos, F., Sturmfels, B.: Higher Lawrence configurations. *J. Combin. Theory Ser. A* **103**, 151–164 (2003)
114. Saxena, A., Bonami, P., Lee, J.: Disjunctive cuts for non-convex mixed integer quadratically constrained programs. In: A. Lodi, A. Panconesi, G. Rinaldi (eds.) *Integer Programming and Combinatorial Optimization 2008 – Bertinoro, Italy, Lecture Notes in Computer Science*, vol. 5035, pp. 17–33. Springer Berlin / Heidelberg (2008)
115. Saxena, A., Bonami, P., Lee, J.: Convex relaxations of non-convex mixed integer quadratically constrained programs: Extended formulations (2008). IBM Research Report RC24621
116. Saxena, A., Bonami, P., Lee, J.: Convex relaxations of non-convex mixed integer quadratically constrained programs: Projected formulations (2008). IBM Research Report RC24695
117. SDPT3: www.math.nus.edu.sg/~mattohkc/sdpt3.html (v. 4.0 (beta))
118. Sebö, A.: Hilbert bases, Caratheodory's Theorem and combinatorial optimization. In: Proc. of the IPCO conference, Waterloo, Canada, pp. 431–455 (1990)
119. SeDuMi: sedumi.mcmaster.ca (v. 1.1)
120. Shor, N.Z.: An approach to obtaining global extremums in polynomial mathematical programming. *Kibernetika* **52**, 102–106 (1987)
121. Smith, E.: On the optimal design of continuous processes. Ph.D. thesis, Imperial College of Science, Technology and Medicine, University of London (1996)
122. Smith, E., Pantelides, C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering* **23**, 457–478 (1999)
123. Stubbs, R.A., Mehrotra, S.: A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming* **86**(3), 515–532 (1999). DOI 10.1007/s101070050103
124. Sturmfels, B.: Gröbner Bases and Convex Polytopes. American Mathematical Society, Providence, RI (1996)
125. Tarasov, S.P., Khachiyan, L.G.: Bounds of solutions and algorithmic complexity of systems of convex diophantine inequalities. *Soviet Math. Doklady* **22**(3), 700–704 (1980)
126. Tawarmalani, M., Sahinidis, N.: Convex extensions and envelopes of semi-continuous functions. *Mathematical Programming* **93**(2), 247–263 (2002)
127. Tawarmalani, M., Sahinidis, N.: Convexification and global optimization in continuous and mixed-integer nonlinear programming: Theory, algorithms, software and applications, *Non-convex Optimization and Its Applications*, vol. 65. Kluwer Academic Publishers, Dordrecht (2002)
128. Tawarmalani, M., Sahinidis, N.: Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming* **99**(3), 563–591 (2004)
129. Tawarmalani, M., Sahinidis, N.V.: Semidefinite Relaxations of Fractional Programs via Novel Convexification Techniques. *Journal of Global Optimization* **20**, 137–158 (2001)
130. 4ti2 team: 4ti2 – a software package for algebraic, geometric and combinatorial problems on linear spaces. Available at <http://www.4ti2.de>
131. Thomas, R.R.: Algebraic methods in integer programming. In: C. Floudas, P. Pardalos (eds.) *Encyclopedia of Optimization*. Kluwer Academic Publishers, Dordrecht (2001)
132. Vandembussche, D., Nemhauser, G.L.: A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. *Mathematical Programming* **102**(3), 559–575 (2005)
133. Vandembussche, D., Nemhauser, G.L.: A polyhedral study of nonconvex quadratic programs with box constraints. *Mathematical Programming* **102**(3), 531–557 (2005)
134. Westerlund, T., Lundqvist, K.: Alpha-ECP, version 5.101: An interactive MINLP-solver based on the extended cutting plane method. Tech. Rep. 01-178-A, Process Design Laboratory at Abo Akademi University (Updated version of 2005-10-21)
135. Westerlund, T., Pettersson, F.: An extended cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering* **19**(Suppl.), S131–S136 (1995)
136. Westerlund, T., Pörn, R.: Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optimization and Engineering* **3**, 253–280 (2002)