

0/1 integer programming over polytopes of Chvátal rank one is as hard as lattice problems

Alberto Del Pia*

October 23, 2026

Abstract

A rational polyhedron has Chvátal rank at most one if a single round of Chvátal–Gomory cuts yields its integer hull. In 1973, Chvátal asked whether 0/1 integer programs over polytopes of bounded Chvátal rank can be solved in polynomial time. For polyhedra of bounded Chvátal rank, integer feasibility is in $\text{NP} \cap \text{coNP}$, so it is unlikely to be NP-hard. We answer Chvátal’s question negatively already in its most restricted form: for Chvátal rank one, for integer feasibility, and even for the SAT polytopes of formulas whose clauses have at most three literals. Our answer holds under either of two standard assumptions from lattice-based cryptography. First, a polynomial-time algorithm for integer feasibility on such SAT polytopes of Chvátal rank one would solve bounded distance decoding with polynomial factors in deterministic polynomial time, contradicting the conjectured hardness of approximating lattice problems within polynomial factors. Second, assuming the hardness of learning with errors, an average-case analog of bounded distance decoding, the problem is also hard on average, for a polynomial-time samplable distribution of formulas. This also gives negative answers to related questions of Cornuéjols, Lee, and Li. We first prove these results for parallelepipeds, where they rest on an elementary sufficient condition: a polyhedron has Chvátal rank at most one if its width is less than one along every row of some unimodular matrix. For our parallelepipeds, such a matrix exists but is hard to find. For SAT polytopes, we represent candidate lattice vectors by their residues modulo small integers, and binary arithmetic on additional 0/1 variables checks that these vectors lie in the lattice.

Keywords: Chvátal–Gomory cuts · Chvátal rank · 0/1 integer programming · satisfiability · promise problems · lattice problems · learning with errors

Mathematics Subject Classification (2020): Primary 90C09 · 90C10. Secondary 11H06 · 90C57 · 90C60.

1 Introduction

Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron and let $P_I = \text{conv}(P \cap \mathbb{Z}^n)$ be its *integer hull*. A *Chvátal–Gomory cut* (CG cut) for P is an inequality $c^\top x \leq \lfloor \delta \rfloor$, where $c \in \mathbb{Z}^n$ and $c^\top x \leq \delta$ is valid for P [22, 9]. The *Chvátal closure* P' of P is the set of vectors satisfying all CG cuts for P . We have $P_I \subseteq P' \subseteq P$, and P' is again a rational polyhedron [40]. Iterating the closure operation yields P_I after finitely many steps, as shown by Chvátal [9] for polytopes and by Schrijver [40] for rational polyhedra. The *Chvátal rank* of P is the number of steps needed.

*Department of Industrial and Systems Engineering & Wisconsin Institute for Discovery, University of Wisconsin–Madison, Madison, WI, USA. E-mail: delpia@wisc.edu.

Polyhedra of Chvátal rank zero are the *integral* polyhedra, and recognizing them is coNP-complete [34]. Polyhedra of Chvátal rank at most one include several classical examples from combinatorial optimization [42]: the fractional matching and edge cover polytopes and, more generally, the fractional b -matching and b -edge cover polyhedra of a graph, the fractional stable set polytope of a t -perfect graph, and the strong fractional stable set polytope of an h -perfect graph. More recent examples arise from a generalization of bidirected graphs [18], from the unit cube [13], and from binary polynomial optimization [17]. In general, recognizing polyhedra of Chvátal rank at most one is NP-hard [16, 14].

Deciding whether $P \cap \mathbb{Z}^n \neq \emptyset$, which we call *integer feasibility*, is NP-complete [41, Theorem 18.1]. If P is integral, it is easy: P contains an integral vector if and only if $P \neq \emptyset$, which can be decided in polynomial time by linear programming [41, Theorem 13.4]. Chvátal observed in 1973 that, for each fixed k , integer programs over polytopes of Chvátal rank at most k admit a good characterization [9, Section 7]. Boyd and Puleyblank showed in 1984 that, for each such k , integer feasibility is in $\text{NP} \cap \text{coNP}$ for polyhedra of Chvátal rank at most k [8, Theorem 5.4] (see also [41, Theorem 23.5 and Corollary 23.5a]). Chvátal also asked whether, for each fixed k , 0/1 integer programs over polytopes of Chvátal rank at most k can be solved in polynomial time [9, Section 7].

We consider the most restricted case of his question: Chvátal rank at most one, that is, $P' = P_I$, and integer feasibility in place of optimization. In this case, it is elementary that the problem is in $\text{NP} \cap \text{coNP}$.

Question 1. *Let $P = \{x \in [0, 1]^n : Cx \leq d\}$ with C and d rational, and suppose $P' = P_I$. Can we decide whether P contains an integral vector in time polynomial in the encoding size of (C, d) ?*

Question 1 is at the heart of this paper. Cornuéjols and Li stated the same question for general rational polyhedra [15, 16], and it is the main motivation of the systematic study of this class by Cornuéjols, Lee, and Li [14].

Question 2. *Let $P = \{x \in \mathbb{R}^n : Cx \leq d\}$ with C and d rational, and suppose $P' = P_I$. Can we decide whether P contains an integral vector in time polynomial in the encoding size of (C, d) ?*

Cornuéjols, Lee, and Li state Question 2 for polyhedra of Chvátal rank exactly one, and, for both questions, the versions for rank exactly one are equivalent to those above (Lemma 1).

We regard Questions 1 and 2 as *promise problems*, with the promise $P' = P_I$, as in [16, 14]. An algorithm solves such a problem in polynomial time if it runs in polynomial time on all inputs and answers correctly on those that satisfy the promise, while its answers on other inputs may be arbitrary.

The promise matters. Under the promise, $P \cap \mathbb{Z}^n = \emptyset$ if and only if $P' = \emptyset$. Without it, deciding whether $P' = \emptyset$ is NP-complete, even for polyhedra containing no integral vector [15, 14], and separating over P' is NP-hard [19]. With the promise, NP-hardness is unlikely: since these promise problems are in $\text{NP} \cap \text{coNP}$, a polynomial-time reduction from an NP-hard problem all of whose queries satisfy the promise would imply $\text{NP} = \text{coNP}$ [24] (see also [21, Theorem 7]).

Several problems in $\text{NP} \cap \text{coNP}$ were eventually shown to be polynomial, such as linear programming [41, Theorem 13.4] and primality testing [3]. Yet no algorithm is known to benefit from the promise $P' = P_I$. For instance, Cornuéjols, Lee, and Li prove a flatness theorem for compact convex sets with empty Chvátal closure, with an essentially tight bound, and observe that it yields a Lenstra-type algorithm [28] with running time $2^{O(n)}n^n$ times a polynomial in the encoding size [14, Theorem 14, Proposition 15, and Section 3.3]. This is slower than the randomized algorithm of Reis and Rothvoss for general integer programs, with running time $(\log n)^{O(n)}$ times a polynomial

in the encoding size [38]. Similarly, branch-and-bound with variable branching does not benefit from the promise: it can require exponentially many nodes even on polytopes with empty Chvátal closure [25].

Our results. We first answer Question 2 negatively, already for parallelepipeds, under two standard assumptions from lattice-based cryptography: that lattice problems such as the shortest vector problem are hard to approximate within polynomial factors, and that learning with errors, an average-case lattice problem, is hard. Lattice problems are a natural source of hardness here: like integer feasibility under the promise $P' = P_I$, they are not expected to be NP-hard, since the decision versions of approximating the norm of a shortest nonzero vector, or the distance of a target vector from the lattice, within a factor $c\sqrt{n}$ for a suitable constant c , are in $\text{NP} \cap \text{coNP}$ [4]. Under the same assumptions, we then answer Question 1, and with it Chvátal’s question, negatively, even for SAT polytopes of formulas whose clauses have at most three literals (Theorems C and D).

Let $L \subseteq \mathbb{R}^n$ be a full-rank lattice and let $\lambda_1(L)$ be the Euclidean norm of a shortest nonzero vector of L . In the *bounded distance decoding problem* with factor α (BDD_α) we are given a rational basis B of L and a rational target vector t whose distance from L is at most $\alpha\lambda_1(L)$, and we want to find a vector v of L closest to t , or equivalently the *error* $t - v$. In integer programming terms, we want to solve the integer least squares problem $\min\{\|Bx - t\| : x \in \mathbb{Z}^n\}$ under the promise that its optimal value is at most $\alpha\lambda_1(L)$.

Theorem A (see Proposition 6 and Theorem 7). *There is a deterministic polynomial-time reduction from $\text{BDD}_{1/(4n^2)}$ in dimension n to integer feasibility on rational parallelepipeds in $\mathbb{R}^n$ with $P' = P_I$. In particular, a polynomial-time algorithm for the latter gives a polynomial-time algorithm for $\text{BDD}_{1/(4n^2)}$.*

Combined with reductions of Lyubashevsky and Micciancio [29], Theorem A has two further consequences (Corollary 9). First, a polynomial-time algorithm for integer feasibility under the promise $P' = P_I$ would give a polynomial-time algorithm for uSVP_{4n^2} , that is, for finding a shortest nonzero vector of L when every vector of L of norm at most $4n^2\lambda_1(L)$ is an integer multiple of a shortest one. Second, it would give a randomized polynomial-time algorithm for GapSVP_γ with $\gamma = 4n^{5/2}/\sqrt{\log n}$, that is, for approximating $\lambda_1(L)$ within a factor γ .

The best known algorithms for BDD, uSVP, and GapSVP with polynomial factors run in time $2^{O(n)}$ [32, 2]. It is conjectured that no polynomial-time algorithm, even a randomized or quantum one, approximates lattice problems such as the shortest vector problem within polynomial factors [31, Conjectures 1 and 2]. This conjecture underlies lattice-based cryptography [37, 36]. Since GapSVP is the decision version of the shortest vector problem, Corollary 9 shows that a polynomial-time algorithm for integer feasibility under the promise $P' = P_I$, even a randomized one, would contradict this conjecture.

The reduction of Theorem A preserves the dimension (Theorem 7), and so do those of Lyubashevsky and Micciancio. Hence, even an algorithm for integer feasibility under the promise $P' = P_I$ running in time $2^{o(n)}$ times a polynomial in the encoding size would give algorithms with the same running time for BDD and uSVP, and a randomized one for GapSVP (Corollary 9).

Theorem A is a worst-case result: it concerns algorithms that are correct on every instance, and it does not show that hard instances are common, nor how to generate them. From *learning with errors* (LWE) [37], the average-case problem on which much of lattice-based cryptography is built, we obtain a distribution of hard instances that can be sampled in polynomial time. In LWE with modulus q , we are given a uniformly random matrix $A \in \mathbb{Z}_q^{n \times m}$ and $t \equiv A^\top s + e \pmod{q}$, where $s \in \mathbb{Z}_q^n$ is uniformly random and $e \in \mathbb{Z}^m$ is a short random error, and we want to find s , or

to distinguish t from a uniformly random vector (decision-LWE). Thus t is a target vector close to the lattice of vectors $y \in \mathbb{Z}^m$ with $y \equiv A^\top s \pmod{q}$ for some s , and LWE can be viewed as an average-case version of BDD.

Theorem B (see Proposition 6 and Theorem 12). *For every sufficiently large n , there is a polynomial-time samplable distribution of parallelepipeds in dimension $O(n \log n)$, given by integral constraints with coefficients bounded by a polynomial in n , that satisfy $P' = P_I$ with high probability. A randomized polynomial-time algorithm that decides integer feasibility correctly with probability at least $2/3$ on these distributions gives one for decision-LWE in dimension n with a polynomial modulus and Gaussian errors of inverse-polynomial rate.*

Combined with a result of Regev [37], Theorem B has a further consequence (Corollary 14): a randomized polynomial-time algorithm that decides integer feasibility correctly with probability at least $2/3$ on these distributions would give a polynomial-time quantum algorithm for GapSVP_γ and SIVP_γ on n -dimensional lattices with $\gamma = \tilde{O}(n^3)$, where $\tilde{O}$ hides polylogarithmic factors. Here SIVP_γ asks for n linearly independent vectors of the lattice whose largest norm is at most γ times the smallest possible. Such a quantum algorithm would contradict the conjecture that no polynomial-time quantum algorithm approximates lattice problems within polynomial factors [31, Conjecture 2].

Theorems A and B hold even under a stronger promise, that every rational subpolyhedron of P has Chvátal rank at most one (Proposition 6 and Corollary 9). Unlike $P' = P_I$, this promise is preserved by branching and by adding cutting planes, since both replace P by a rational subpolyhedron. It is strictly stronger than $P' = P_I$, even for $0/1$ polytopes (Remark 3).

Theorems A and B are incomparable. The negative answer from Theorem A relies on the conjecture only for classical algorithms, and the reduction of Theorem A preserves the dimension. The negative answer that we derive from Theorem B relies on the conjecture for quantum algorithms, and the reduction of Theorem B increases the dimension from n to $O(n \log n)$. However, Theorem B gives hard instances on average, with integer coefficients bounded by a polynomial in the dimension, so it also rules out pseudo-polynomial algorithms. Its distribution also gives test instances for integer programming solvers. With high probability, they satisfy the promise, and their answers are known to the generator: those in which t comes from LWE contain an integral vector that it can compute, and those in which t is uniformly random contain none.

The idea. The proofs of Theorems A and B rest on an elementary sufficient condition for Chvátal rank at most one (Lemma 2): if P has width less than one along the rows of a unimodular matrix U , then $P' = P_I$, and P_I is obtained by adding at most $2n$ CG cuts. We apply it to the rational parallelepipeds $\{x \in \mathbb{R}^n : \|Bx - t\|_\infty \leq r\}$ with $r > 0$, where B is a basis of a lattice L and t is a target vector (Figure 1). Their integral vectors x correspond, via $x \mapsto Bx$, to the vectors of L within ℓ_∞ -distance r of t . The unimodular matrices are exactly the matrices with rows $w_1^\top B, \dots, w_n^\top B$, where $w_1, \dots, w_n$ is a basis of the dual lattice $L^* = \{w \in \mathbb{R}^n : w^\top y \in \mathbb{Z} \text{ for all } y \in L\}$, and the width along $B^\top w_i$ is $2r\|w_i\|_1$. The dual lattice L^* has a basis with ℓ_1 -norms at most $n^2/\lambda_1(L)$, by a transference theorem of Banaszczyk [7, Theorem 2.1] (Lemma 5). Hence, for every t , these parallelepipeds and all their rational subpolyhedra have Chvátal rank at most one whenever $r < \lambda_1(L)/(2n^2)$, since widths can only decrease on subpolyhedra. We call such parallelepipeds *thin*.

For Theorem A, given an instance (B, t) of $\text{BDD}_{1/(4n^2)}$, we try radii r that are powers of two, and one of them gives a thin parallelepiped, which satisfies the promise by Proposition 6 and contains the integral vector corresponding to the closest vector. We then find its entries bit by bit, with queries for parallelepipeds defined by sublattices of L , which are thin as well.

For Theorem B, given an instance (A, t) of decision-LWE, the lattice of the vectors $y \in \mathbb{Z}^m$ with $y \equiv A^\top s \pmod{q}$ for some s depends only on A , which is uniformly random, so with high probability it has no short nonzero vectors (Lemma 10). Hence (A, t) yields a single parallelepiped that is thin, and so satisfies the promise, with high probability. If t comes from LWE, then with high probability the error is short, and so this parallelepiped contains an integral vector. If t is uniformly random, the parallelepiped contains none with high probability, by a counting argument. Thus a single call to an algorithm for integer feasibility distinguishes the two cases.

Easy with a short dual basis, hard without. For thin parallelepipeds, the promise holds because the dual lattice has a basis with small ℓ_1 -norms, and knowing such a basis makes the problem easy. With such a basis, Babai's round-off algorithm [6] finds the integral vector if there is one, and otherwise the basis gives a *refutation*, that is, a set of CG cuts with empty intersection (Remark 15). In the variables Ux , where U has rows $w_1^\top B, \dots, w_n^\top B$, the parallelepiped has width less than one along every coordinate direction, so each variable takes at most one integer value and branch-and-bound solves it with $O(n)$ nodes. This parallels the integer programs of Aardal, Hurkens, and Lenstra [1] and of Pataki, Tural, and Wong [35], which are difficult for branch-and-bound in the original variables but much easier after a reformulation computed by basis reduction. Here, however, finding such a basis is as hard as $BDD_{1/(4n^2)}$ for the parallelepipeds of Theorem A, and as hard as decision-LWE for those of Theorem B.

In proof complexity terms, CG cuts form the Cutting Planes proof system [12], and finding a refutation of size polynomial in that of a shortest one is NP-hard in general [23]. Under the promise $P' = P_I$, a rational polyhedron without integral vectors has $P' = \emptyset$. Since P' is the intersection of finitely many CG cuts [40], Helly's theorem then gives a refutation with at most $n + 1$ CG cuts. For our parallelepipeds, however, finding any refutation is as hard as $BDD_{1/(4n^2)}$ and decision-LWE (Remark 15).

Back to the unit cube. We now return to Question 1, through SAT polytopes. For a formula F in conjunctive normal form on variables $x_1, \dots, x_N$, the *SAT polytope* $P(F)$ is the set of vectors $x \in [0, 1]^N$ that satisfy $\sum_{i \in I} x_i + \sum_{i \in J} (1 - x_i) \geq 1$ for every clause of F , where I and J are the sets of indices of the variables that appear in the clause unnegated and negated, respectively. Its integral vectors are the satisfying assignments of F . Cornuéjols, Lee, and Li also asked whether satisfiability can be decided in polynomial time for formulas whose SAT polytope has Chvátal rank exactly one [14, Open Question 2]. Up to the equivalence between rank exactly one and rank at most one (Lemma 1), this is the special case of Question 1 for SAT polytopes. Under the promise $P(F)' = P(F)_I$, the problem is easy if every clause has at most two literals, since then satisfiability is polynomial even without the promise [5]. In this case, the Chvátal closure of the SAT polytope is also well understood [20]. The problem is also easy if every clause has at least three literals, since then the promise implies that F is satisfiable [14, Remark 3]. Cornuéjols, Lee, and Li point out that the remaining case consists of formulas with both clauses of two literals and clauses of at least three literals [14, Section 2.1]. We show that Theorems A and B extend, with different parameters, to formulas whose clauses have at most three literals, and the formulas in our reductions are of this remaining kind.

Theorem C (see Proposition 16 and Theorem 26). *There is a deterministic polynomial-time reduction from $BDD_{1/(4n^{5/2})}$ in dimension n to satisfiability of formulas whose clauses have at most three literals and whose SAT polytopes satisfy $P(F)' = P(F)_I$. In particular, a polynomial-time algorithm for the latter gives a polynomial-time algorithm for $BDD_{1/(4n^{5/2})}$.*

Theorem D (see Proposition 16 and Theorem 29). *For every sufficiently large n , there is a polynomial-time samplable distribution of formulas whose clauses have at most three literals, with $O(n^2 \log^4 n)$ variables, whose SAT polytopes satisfy $P(F)' = P(F)_I$ with high probability. A randomized polynomial-time algorithm that decides satisfiability correctly with probability at least $2/3$ on these distributions gives one for decision-LWE in dimension n with a polynomial modulus and Gaussian errors of inverse-polynomial rate.*

Theorems C and D answer the question of Cornuéjols, Lee, and Li on SAT polytopes negatively. Since SAT polytopes lie in the unit cube, they also answer Question 1 negatively, and with it Chvátal’s question. Thus Chvátal’s question has a negative answer already in its most restricted form: for Chvátal rank one, for integer feasibility, and even for SAT polytopes.

Like Theorems A and B, Theorems C and D hold even under the stronger promise that every rational subpolyhedron of $P(F)$ has Chvátal rank at most one (Proposition 16 and Corollary 27). They also hold for set partitioning polytopes with at most three ones in each row (Remark 32).

Theorems C and D are also incomparable. The negative answer from Theorem C relies on the conjecture only for classical algorithms. As with Theorem A, the reductions of Lyubashevsky and Micciancio show that a polynomial-time algorithm for satisfiability under the promise would give polynomial-time algorithms for $\text{uSVP}_{4n^{5/2}}$ and, with randomization, for GapSVP_γ with $\gamma = 4n^3/\sqrt{\log n}$ (Corollary 27). Unlike that of Theorem A, the reduction of Theorem C does not preserve the dimension, since the number of variables of its formulas depends on the encoding size of the input. The negative answer that we derive from Theorem D relies on the conjecture for quantum algorithms, through a result of Regev, with $\gamma = \tilde{O}(n^{5/2})$ (Corollary 30). However, Theorem D gives hard instances on average, and, as with Theorem B, its distribution gives test instances with known answers, here for SAT solvers.

SAT polytopes are rational polytopes, so Theorems C and D also give hardness for general polytopes, but they do not imply Theorems A and B: Theorem A has the better factor $1/(4n^2)$ and preserves the dimension, and the polytopes of Theorem B have dimension $O(n \log n)$ instead of $O(n^2 \log^4 n)$. In the other direction, Theorem D has an advantage over Theorem B: it gives the better factor $\tilde{O}(n^{5/2})$ for quantum algorithms, instead of $\tilde{O}(n^3)$. Moreover, since SAT polytopes have coefficients in $\{0, \pm 1\}$, even a pseudo-polynomial algorithm for them would contradict the conjecture for classical algorithms, through Theorem C, whereas for parallelepipeds our argument uses Theorem B, and so the conjecture for quantum algorithms.

The idea for SAT polytopes. The proofs of Theorems C and D need a different construction, since the promise for our SAT polytopes does not follow directly from Lemma 2. The satisfying assignments of our formulas correspond to the vectors of an integral lattice in a box around a target vector (Proposition 16). We represent the error, that is, the difference between the target and such a lattice vector, by the residues of its entries modulo small integers, each residue by a group of $0/1$ variables exactly one of which equals one. Additional $0/1$ variables carry out, in binary, the arithmetic that recovers these entries from their residues and checks that the corresponding vector lies in the lattice. The Chvátal closure of the SAT polytope lies in a face defined by equations that set a sum of terms x_i and $1 - x_i$ equal to one. Each step of this arithmetic is modeled by the standard extended formulation of the convex hull of its finitely many $0/1$ solutions. So the arithmetic holds as a linear identity on the whole face, not only at its integral vectors, and vectors of the dual lattice give integral directions along which this face has width less than one. As in the proof of Lemma 2, CG cuts along these directions fix all residues. We use residues because the errors can be exponentially large, and residues modulo small integers represent them compactly, in a form that one round of CG cuts can fix. As with the parallelepipeds, the vectors of the dual

lattice that give these cuts exist but are hard to find (Remark 31).

The paper is organized as follows. Section 2 contains preliminaries. Section 3 proves our sufficient condition for Chvátal rank at most one (Lemma 2) and applies it to parallelepipeds defined by lattices (Proposition 6). Section 4 proves Theorems A and B and their consequences. Section 5 constructs SAT polytopes defined by lattices, which play the role of the parallelepipeds (Proposition 16), and Section 6 proves Theorems C and D and their consequences.

2 Preliminaries

Notation. All vectors are column vectors. We write e_i for the i th unit vector, and $\mathbf{0}$ and $\mathbf{1}$ for the all-zeros and all-ones vectors. All norms without subscript are Euclidean, and $\lfloor \cdot \rfloor$ denotes rounding to a nearest integer. The notation $\tilde{O}$ hides polylogarithmic factors. A matrix $U \in \mathbb{Z}^{n \times n}$ is *unimodular* if $\det U = \pm 1$.

Polyhedra. CG cuts, the Chvátal closure P' , the integer hull P_I , and the Chvátal rank are defined in Section 1. For a polyhedron $P \subseteq \mathbb{R}^n$ and $c \in \mathbb{R}^n$, let $\delta_P(c) = \sup\{c^\top x : x \in P\}$, and let $\text{width}_P(c) = \delta_P(c) + \delta_P(-c)$ be the *width* of P along c . A *rational subpolyhedron* of P is a rational polyhedron contained in P .

Lattices. A *lattice* is a set $L = B\mathbb{Z}^n$, where $B \in \mathbb{R}^{n \times n}$ is nonsingular, and B is a *basis* of L . For $i = 1, \dots, n$, the i th *successive minimum* $\lambda_i(L)$ is the smallest λ such that L contains i linearly independent vectors of norm at most λ . For $t \in \mathbb{R}^n$, let $\text{dist}(t, L) = \min\{\|v - t\| : v \in L\}$. The *dual lattice* is $L^* = \{w \in \mathbb{R}^n : w^\top y \in \mathbb{Z} \text{ for all } y \in L\}$. We have $L^* = B^{-\top} \mathbb{Z}^n$ and $L^{**} = L$.

Promise problems. An algorithm *solves* a promise problem if it runs within its time bound on every input and answers correctly on every input that satisfies the promise [21, Section 1.2]. Its output on other inputs is arbitrary. A randomized or quantum algorithm solves a promise problem if it runs within its time bound on every input and answers correctly with probability at least $2/3$ on every input that satisfies the promise. Similarly, an oracle for a promise problem answers every query, and answers correctly the queries that satisfy the promise. A *reduction* from one promise problem to another is a deterministic or randomized algorithm that solves the first using an oracle for the second, whatever the oracle answers on queries that violate the promise. Each oracle call counts as one step, and the reduction is polynomial-time if this algorithm is. It *preserves the dimension* if, on inputs of dimension n , all its queries have dimension n .

Lattice problems. In the following problems, the input is a rational basis B of a lattice $L \subseteq \mathbb{R}^n$, together with a rational vector t for BDD and a rational number η for GapSVP. The parameters $\alpha \in (0, 1)$ and $\gamma \geq 1$ are functions of n .

- BDD $_\alpha$: under the promise $\text{dist}(t, L) \leq \alpha \lambda_1(L)$, find $v \in L$ with $\|v - t\| = \text{dist}(t, L)$.
- uSVP $_\gamma$: under the promise $\lambda_2(L) > \gamma \lambda_1(L)$, find $v \in L$ with $\|v\| = \lambda_1(L)$.
- GapSVP $_\gamma$: under the promise that $\lambda_1(L) \leq \eta$ or $\lambda_1(L) > \gamma \eta$, decide which of the two holds.
- SIVP $_\gamma$: find n linearly independent vectors of L of norm at most $\gamma \lambda_n(L)$.

Learning with errors. Let $q \geq 2$, and let $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$, whose elements we represent by integers in $\{0, \dots, q-1\}$. In the following problem, the input is a pair (A, t) with $A \in \mathbb{Z}_q^{n \times m}$ and $t \in \mathbb{Z}_q^m$. The modulus q , the number of samples m , and a probability distribution χ on $\mathbb{Z}$ are functions of n . An *LWE instance* and a *uniform instance* are both pairs (A, t) with A uniformly random. In an LWE instance, $t \equiv A^\top s + e \pmod{q}$, where $s \in \mathbb{Z}_q^n$ is uniformly random and $e \in \mathbb{Z}^m$ has independent entries distributed according to χ . In a uniform instance, t is uniformly random and independent of A .

- decision-LWE $_{n,q,\chi}$ with m samples: given an LWE instance or a uniform instance, decide which of the two it is, by outputting 1 or 0, respectively.

An algorithm *solves* this problem if the probabilities that it outputs 1 on an LWE instance and on a uniform instance differ by at least n^{-c} for some constant $c > 0$ and all sufficiently large n . In our main results, χ is the error distribution $\bar{\Psi}_\alpha$ of Regev [37, Section 2], lifted from $\mathbb{Z}_q$ to $\mathbb{Z}$: for $\alpha \in (0, 1)$, it is the distribution of $\lfloor X \rfloor$, where X is normal with mean zero and standard deviation $\alpha q / \sqrt{2\pi}$. It can be sampled in polynomial time.

Chvátal rank exactly one. The questions of Cornuéjols, Lee, and Li are stated for polyhedra of Chvátal rank exactly one, while Questions 1 and 2 also allow rank zero. The following standard lemma shows that this makes no difference.

Lemma 1. *There is a polynomial-time algorithm that, given a system $Cx \leq d$ with C and d rational, defining a polyhedron P , either decides whether P contains an integral vector, or concludes that P is not integral. In particular, if $P' = P_I$ and the algorithm concludes that P is not integral, then P has Chvátal rank exactly one.*

Proof. A rational polyhedron is integral if and only if each of its minimal faces contains an integral vector [41, Section 16.3]. We decide whether $P = \emptyset$ by linear programming [41, Theorem 13.4]. If $P = \emptyset$, then P contains no integral vector. Otherwise, we compute a system of linear equations $C'x = d'$ that defines a minimal face of P [41, proof of Theorem 16.2], and we decide whether this face contains an integral vector [41, Corollary 5.3b]. If it does, then so does P . Otherwise this minimal face contains no integral vector, so P is not integral. If moreover $P' = P_I$, then $P \neq P_I = P'$, so P has Chvátal rank exactly one. $\square$

Hence the versions of Questions 1 and 2 for polyhedra of Chvátal rank exactly one are equivalent to ours. Indeed, an algorithm for our version applies in particular to polyhedra of Chvátal rank exactly one. Conversely, given an algorithm for the version with Chvátal rank exactly one, we first process the input with Lemma 1, and we call this algorithm only if Lemma 1 concludes that the input is not integral.

3 Parallelepipeds of Chvátal rank at most one

In this section we prove Lemma 2, the sufficient condition for Chvátal rank at most one on which the proofs of Theorems A and B rest, and we apply it to parallelepipeds defined by lattices (Proposition 6).

3.1 A sufficient condition

Lemma 2. *Let $U \in \mathbb{Z}^{n \times n}$ be unimodular with rows $u_1^\top, \dots, u_n^\top$, and let $P \subseteq \mathbb{R}^n$ be a rational polyhedron with $\text{width}_P(u_i) < 1$ for $i = 1, \dots, n$. Then $P' = P_I$, P_I is the intersection of P*

with at most $2n$ CG cuts, and P contains at most one integral vector. Moreover, every rational subpolyhedron of P has Chvátal rank at most one.

Proof. We may assume $P \neq \emptyset$. Then $\delta_P(\pm u_i)$ is finite, since $\text{width}_P(u_i) < 1$. Let P_0 be the intersection of P with the $2n$ CG cuts $u_i^\top x \leq \lfloor \delta_P(u_i) \rfloor$ and $u_i^\top x \geq -\lfloor \delta_P(-u_i) \rfloor$. The difference of these bounds is $\lfloor \delta_P(u_i) \rfloor + \lfloor \delta_P(-u_i) \rfloor$, an integer at most $\delta_P(u_i) + \delta_P(-u_i) = \text{width}_P(u_i) < 1$, and hence at most zero. If $P_0 \neq \emptyset$, it follows that, for every i , there is an integer γ_i with $u_i^\top x = \gamma_i$ for all $x \in P_0$, so $P_0 = \{U^{-1}\gamma\}$. Then $U^{-1}\gamma$ is an integral vector of P , since U is unimodular. Thus $P_0 \subseteq P_I$, which also holds if $P_0 = \emptyset$. Since also $P_I \subseteq P' \subseteq P_0$, these three sets are equal. In particular, P contains at most one integral vector, since $P \cap \mathbb{Z}^n \subseteq P_I = P_0$. Let R be a rational subpolyhedron of P . Then $\text{width}_R \leq \text{width}_P$, so the width assumption holds for R , and the first statement applies to R . $\square$

Remark 3. The strict inequality in Lemma 2 cannot be weakened to $\text{width}_P(u_i) \leq 1$, and the last statement of Lemma 2 is not implied by $P' = P_I$. Both facts are shown by the *cropped cube*

$$\left\{ x \in [0, 1]^n : \sum_{j \in J} x_j + \sum_{j \notin J} (1 - x_j) \geq \frac{1}{2} \text{ for all } J \subseteq \{1, \dots, n\} \right\},$$

which has Chvátal rank at least n [10, Lemma 7.2], and so at least two for $n \geq 2$. For such n , it has width one along every e_i , which shows the first fact, and it is a rational subpolyhedron of the integral cube $[0, 1]^n$, which shows the second.

3.2 Parallelepipeds defined by lattices

We now apply Lemma 2 to parallelepipeds defined by lattices: we compute their widths along the directions given by dual bases, and we show that these parallelepipeds satisfy the promise when they are thin (Proposition 6). For a basis $B \in \mathbb{Q}^{n \times n}$ of a lattice L , $t \in \mathbb{Q}^n$, and $r > 0$, let

$$Q(B, t, r) = \{x \in \mathbb{R}^n : \|Bx - t\|_\infty \leq r\} = \{x \in \mathbb{R}^n : t - r\mathbf{1} \leq Bx \leq t + r\mathbf{1}\},$$

which is a parallelepiped (Figure 1). The map $x \mapsto Bx$ is a bijection between $Q(B, t, r) \cap \mathbb{Z}^n$ and the set of vectors of L within ℓ_∞ -distance r of t . To apply Lemma 2 to $Q(B, t, r)$, we need a unimodular matrix along whose rows $Q(B, t, r)$ has width less than one. The next lemma shows that every basis $w_1, \dots, w_n$ of the dual lattice L^* gives a candidate, with rows $w_1^\top B, \dots, w_n^\top B$, and computes the widths along these rows.

Lemma 4. *Let $B \in \mathbb{Q}^{n \times n}$ be a basis of a lattice L , let $t \in \mathbb{Q}^n$, let $r > 0$, and let $w_1, \dots, w_n$ be a basis of L^* . Then $w_1^\top B, \dots, w_n^\top B$ are the rows of a unimodular matrix, every unimodular matrix arises in this way from a basis of L^* , and $\text{width}_{Q(B, t, r)}(B^\top w) = 2r\|w\|_1$ for every $w \in \mathbb{R}^n$.*

Proof. The map $w \mapsto B^\top w$ is a linear bijection from L^* to $\mathbb{Z}^n$, hence it maps the bases of L^* bijectively to the bases of $\mathbb{Z}^n$. The transposes of the vectors of a basis of $\mathbb{Z}^n$ are the rows of a unimodular matrix, and every unimodular matrix arises in this way. Moreover, $(B^\top w)^\top x = w^\top Bx$ and $\{Bx : x \in Q(B, t, r)\} = t + r[-1, 1]^n$. Hence $\delta_{Q(B, t, r)}(B^\top w) = w^\top t + r\|w\|_1$ for every $w \in \mathbb{R}^n$, and adding this identity for w and $-w$ gives the claim. $\square$

To make these widths less than one, we need a basis of L^* with ℓ_1 -norms less than $1/(2r)$. The next lemma gives a basis with ℓ_1 -norms at most $n^2/\lambda_1(L)$, which suffices when $r < \lambda_1(L)/(2n^2)$.

Lemma 5. *For every lattice $L \subseteq \mathbb{R}^n$, the dual lattice L^* has a basis $w_1, \dots, w_n$ with $\|w_i\|_1 \leq n^2/\lambda_1(L)$ for $i = 1, \dots, n$.*



Figure 1: The parallelepiped $Q(B, t, r)$ for $B = \begin{pmatrix} 5 & 9 \\ 4 & 4 \end{pmatrix}$, $t = (4.5, -0.4)$, and $r = 1.5$. The vectors $w_1 = (\frac{1}{4}, -\frac{1}{16})$ and $w_2 = (0, \frac{1}{4})$ form a basis of L^* , where $L = B\mathbb{Z}^2$, and $u_1 = B^\top w_1 = (1, 2)$ and $u_2 = B^\top w_2 = (1, 1)$. Left: the lattice L , the square $t + r[-1, 1]^2$, which contains only the vector $v = (4, 0)$ of L , and the lines $w_1^\top y = 0, 1, 2$ (dashed) and $w_2^\top y = -1, 0, 1$ (dotted). Right: $Q(B, t, r)$, whose only integral vector is $B^{-1}v = (-1, 1)$, and the lines $u_1^\top x = 0, 1, 2$ (dashed) and $u_2^\top x = -1, 0, 1$ (dotted), which the map $x \mapsto Bx$ sends to the lines on the left. By Lemma 4, the widths of $Q(B, t, r)$ along u_1 and u_2 are $\frac{15}{16}$ and $\frac{3}{4}$, so Lemma 2 applies, whereas its widths along e_1 and e_2 are $\frac{39}{16}$ and $\frac{27}{16}$.

Proof. We have $\lambda_n(L^*) \leq n/\lambda_1(L)$, by a transference theorem of Banaszczyk [7, Theorem 2.1]. Moreover, by [30, Corollary 7.2], L^* has a basis $w_1, \dots, w_n$ such that, for $i = 1, \dots, n$,

$$\|w_i\| \leq \max\{1, \sqrt{i}/2\} \lambda_i(L^*).$$

Hence $\|w_i\| \leq \max\{1, \sqrt{i}/2\} \lambda_i(L^*) \leq \sqrt{n} \lambda_n(L^*) \leq n^{3/2}/\lambda_1(L)$, and $\|w_i\|_1 \leq \sqrt{n} \|w_i\| \leq n^2/\lambda_1(L)$. $\square$

We call $Q(B, t, r)$ *thin* if $r \in \mathbb{Q}$ and $0 < r < \lambda_1(L)/(2n^2)$, where $L \subseteq \mathbb{R}^n$ is the lattice with basis B . Combining Lemmas 2, 4, and 5, we obtain the following.

Proposition 6. *If $Q(B, t, r)$ is thin, then every rational subpolyhedron of $Q(B, t, r)$ has Chvátal rank at most one, and $Q(B, t, r)$ contains at most one integral vector.*

Proof. Let $w_1, \dots, w_n$ be the basis of L^* given by Lemma 5. By Lemma 4, the matrix with rows $w_1^\top B, \dots, w_n^\top B$ is unimodular, and, for each i , the width of $Q(B, t, r)$ along $B^\top w_i$ is $2r\|w_i\|_1 \leq 2rn^2/\lambda_1(L) < 1$. Since B , t , and r are rational, $Q(B, t, r)$ is a rational polyhedron, and the statement follows from Lemma 2. $\square$

4 Hardness for parallelepipeds

In this section we reduce worst-case lattice problems and learning with errors to integer feasibility under the promise $P' = P_I$, using the parallelepipeds of Proposition 6. This proves Theorems A and B, through Theorems 7 and 12, and gives their consequences for lattice problems (Corollaries 9 and 14).

4.1 Hardness from worst-case lattice problems

To solve BDD, we try radii r that are powers of two, and we find the closest vector coordinate by coordinate (Theorem 7). Combined with reductions of Lyubashevsky and Micciancio, this gives consequences for uSVP and GapSVP (Corollary 9).

Theorem 7. *$BDD_{1/(4n^2)}$ can be solved in deterministic polynomial time with an oracle for integer feasibility under the promise that the query is a thin parallelepiped in $\mathbb{R}^n$. Every query is a rational parallelepiped in $\mathbb{R}^n$.*

Proof. Let $B \in \mathbb{Q}^{n \times n}$ be a basis of a lattice L and let $t \in \mathbb{Q}^n$, with total encoding size σ . Let $\rho = \text{dist}(t, L)$, and let v^* be a vector of L closest to t . For an integer j , let $Q_j = Q(B, t, 2^j)$. The idea is to guess ρ up to a factor of two by trying the radii 2^j . If the input satisfies the promise of BDD, some j satisfies $\rho \leq 2^j < \lambda_1(L)/(2n^2)$. For such j , Q_j is thin, and $B^{-1}v^*$ is its only integral vector, and we find its entries bit by bit, with queries for thin parallelepipeds defined by sublattices of L .

We first construct integers $j_0 \leq j_1$ such that, under the promise, some j^* between them satisfies $\rho \leq 2^{j^*} < \lambda_1(L)/(2n^2)$. To do so, it suffices that $2^{j_0} \leq \lambda_1(L)/(4n^2)$ and $2^{j_1} \geq \rho$. Indeed, let j^* be the smallest integer $j \geq j_0$ with $2^j \geq \rho$, so that $j^* \leq j_1$. If $j^* = j_0$, then $2^{j^*} \leq \lambda_1(L)/(4n^2)$. Otherwise $2^{j^*-1} < \rho$ by the choice of j^* , and so $2^{j^*} < 2\rho$, which is at most $\lambda_1(L)/(2n^2)$ under the promise $\rho \leq \lambda_1(L)/(4n^2)$. We now construct such j_0 and j_1 . Since $\mathbf{0} \in L$, we have $\rho \leq \|t\| \leq 2^\sigma$, so $j_1 = \sigma$ satisfies $2^{j_1} \geq \rho$. The least common denominator of the entries of B is at most 2^σ , and multiplying L by it gives a sublattice of $\mathbb{Z}^n$, whose nonzero vectors have norm at least one. Thus $\lambda_1(L) \geq 2^{-\sigma}$. Hence $j_0 = -\sigma - \lceil \log_2(4n^2) \rceil$ satisfies $2^{j_0} \leq \lambda_1(L)/(4n^2)$.

Next we describe the queries and bound the running time. We consider $j = j_0, j_0 + 1, \dots, j_1$ and do the following for each j . We ask the oracle whether Q_j contains an integral vector, and if it reports that Q_j contains none, we proceed to the next j . If the oracle reports that Q_j contains an integral vector, then, for $i = 1, \dots, n$, we compute $I_i = \{x_i : x \in Q_j\}$ by linear programming, we let $c_i = \lceil \min I_i \rceil$, and we let K_i be the smallest nonnegative integer with $2^{K_i} > \max I_i - c_i$. For $k = 1, \dots, K_i$, let $B^{(i,k)}$ be the matrix obtained from B by multiplying its i th column b_i by 2^k . Starting from $u_{i,0} = 0$, for $k = 1, \dots, K_i$ in turn, we ask the oracle whether $Q(B^{(i,k)}, t - (c_i + u_{i,k-1})b_i, 2^j)$ contains an integral vector, and we let $u_{i,k} = u_{i,k-1}$ if it reports that it does and $u_{i,k} = u_{i,k-1} + 2^{k-1}$ otherwise. We let $z_i = c_i + u_{i,K_i}$, output Bz , and stop. If no output is produced for $j \leq j_1$, we output $\mathbf{0}$. Each query is a parallelepiped $Q(B', t', 2^j)$ in $\mathbb{R}^n$, given by the $2n$ inequalities $t' - 2^j \mathbf{1} \leq B'x \leq t' + 2^j \mathbf{1}$, where B' is B or some $B^{(i,k)}$, and so is rational, since B' , t' , and 2^j are. There are $j_1 - j_0 + 1 = O(\sigma)$ values of j , and 2^j has encoding size $O(\sigma)$ for each of them. The endpoints of I_i are optimal values of linear programs over Q_j , so they have encoding size polynomial in σ [41, Corollary 10.2a], so K_i is polynomial in σ , and $B^{(i,k)}$ and $t - (c_i + u_{i,k-1})b_i$ have encoding size polynomial in σ . Hence, on every input, the oracle is called a number of times polynomial in σ , on queries of encoding size polynomial in σ , and all other computations take time polynomial in σ .

We now show that the output is v^* if the input satisfies the promise $\rho \leq \lambda_1(L)/(4n^2)$. The vector $z^* = B^{-1}v^*$ is integral, since $v^* \in B\mathbb{Z}^n$, and it lies in Q_{j^*} , since $\|Bz^* - t\|_\infty = \|v^* - t\|_\infty \leq \|v^* - t\| = \rho \leq 2^{j^*}$. Since $2^{j^*} < \lambda_1(L)/(2n^2)$, the parallelepiped Q_{j^*} is thin, so z^* is its unique integral vector by Proposition 6. For $j \leq j^*$, the parallelepiped Q_j is thin, since $2^j \leq 2^{j^*}$, so the oracle answers the query Q_j correctly, and $Q_j \cap \mathbb{Z}^n \subseteq Q_{j^*} \cap \mathbb{Z}^n = \{z^*\}$. Hence, if Q_j contains no integral vector, we proceed to the next j . Otherwise $Q_j \cap \mathbb{Z}^n = \{z^*\}$, that is, Bz^* is the only vector of L within ℓ_∞ -distance 2^j of t . Fix i and k , and let $u = u_{i,k-1}$. Since $B^{(i,k)}x + (c_i + u)b_i = By$ for $y_i = 2^k x_i + c_i + u$ and $y_l = x_l$ for $l \neq i$, the integral vectors x of $Q(B^{(i,k)}, t - (c_i + u)b_i, 2^j)$ correspond to the integral vectors y with $\|By - t\|_\infty \leq 2^j$ and $y_i \equiv c_i + u \pmod{2^k}$. So this parallelepiped contains an integral vector if and only if $z_i^* \equiv c_i + u \pmod{2^k}$. It is thin, since $B^{(i,k)}$ is a basis of a sublattice of L , whose nonzero vectors have norm at least $\lambda_1(L) > 2n^2 \cdot 2^j$, so the oracle answers it correctly. Since z_i^* is an integer in I_i , we have $0 \leq z_i^* - c_i < 2^{K_i}$, and induction on k gives $u_{i,k} = (z_i^* - c_i) \bmod 2^k$, so $z_i = z_i^*$. Hence $z = z^*$. Since Q_{j^*} contains z^* , the procedure stops at some $j \leq j^*$ and outputs $Bz^* = v^*$. This is why we try the values of j in increasing order: for $j > j^*$, the queries may violate the promise, and the oracle may answer them incorrectly. $\square$

Proof of Theorem A. By Proposition 6, every thin parallelepiped in $\mathbb{R}^n$ satisfies $P' = P_I$, and it

is rational by definition. By Theorem 7, every query is a rational parallelepiped in $\mathbb{R}^n$. Hence Theorem 7 gives Theorem A. $\square$

We use the following reductions of Lyubashevsky and Micciancio.

Theorem 8 ([29, Theorems 2 and 4 and their proofs]). *Let $\gamma = \gamma(n)$ be bounded by a polynomial in n . There is a deterministic polynomial-time reduction from $uSVP_\gamma$ to $BDD_{1/\gamma}$ that preserves the dimension. If $n \geq 2$ and $\gamma > 2\sqrt{n/\log n}$, there is a randomized polynomial-time reduction from $GapSVP_\gamma$ to $BDD_{\sqrt{n/\log n}/\gamma}$ that preserves the dimension.*

The proof of [29, Theorem 2] assumes that the oracle returns a lattice vector on every query. Every oracle for BDD can be made to do so by replacing each answer v to a query with basis B such that $B^{-1}v \notin \mathbb{Z}^n$ by $\mathbf{0}$. This does not change the answers to queries that satisfy the promise, so both results give reductions in the sense of Section 2. Replacing the oracle by an algorithm and combining Theorem 7 with Theorem 8, we obtain the further consequences stated in the introduction.

Corollary 9. *Suppose that an algorithm solves integer feasibility for rational parallelepipeds under the promise that every rational subpolyhedron of the input, and in particular the input itself, has Chvátal rank at most one, in time $T(n, \sigma')$, where n is the dimension, σ' is the encoding size of the input, and T is nondecreasing. If T is bounded by a polynomial, then $BDD_{1/(4n^2)}$ and $uSVP_{4n^2}$ can be solved in deterministic polynomial time, and, for $n \geq 2$, $GapSVP_\gamma$ with $\gamma = 4n^{5/2}/\sqrt{\log n}$ can be solved in randomized polynomial time. If $T(n, \sigma') = 2^{o(n)} \text{poly}(\sigma')$, then these problems can be solved in time $2^{o(n)}$ times a polynomial in the encoding size, deterministically for BDD and $uSVP$ and with randomization for $GapSVP$. If the algorithm is randomized, then all these statements hold with randomized algorithms for the three problems.*

Proof. By Proposition 6, every thin parallelepiped satisfies this promise, so, as in the proof of Theorem A, the algorithm can serve as the oracle in Theorem 7. Since the queries are in dimension n and have encoding size polynomial in the encoding size σ of (B, t) , and T is nondecreasing, we obtain a deterministic algorithm for $BDD_{1/(4n^2)}$ that runs in time $T(n, \text{poly}(\sigma)) \text{poly}(\sigma)$ on every input. The statements about $uSVP$ and $GapSVP$ follow from Theorem 8 with $\gamma = 4n^2$ and $\gamma = 4n^{5/2}/\sqrt{\log n}$, respectively, since in both cases the BDD factor is $1/(4n^2)$. The statements about running times $2^{o(n)}$ hold since all the reductions preserve the dimension and make a polynomial number of queries of polynomial encoding size.

Suppose now that the algorithm is randomized and answers correctly with probability at least $2/3$. The algorithms above make a number K of queries polynomial in the encoding size of their input. We answer each new query by the majority answer of $O(\log K)$ independent runs of the algorithm, and give the same answer if the query is repeated, so that, by Hoeffding's inequality, each query that satisfies the promise is answered correctly with probability at least $1 - 1/(12K)$. By the union bound, all such queries are answered correctly with probability at least $11/12$. Hence BDD and $uSVP$ are solved correctly with probability at least $11/12$, and $GapSVP$ with probability at least $2/3 - 1/12 = 7/12$. Taking the majority answer of a constant number of independent runs of this algorithm for $GapSVP$ raises this probability to $2/3$, again by Hoeffding's inequality. $\square$

4.2 Hardness from learning with errors

To solve decision-LWE, we use a single parallelepiped defined by a random q -ary lattice, that is, a lattice L with $q\mathbb{Z}^m \subseteq L \subseteq \mathbb{Z}^m$ [31, Section 2]. With high probability, it is thin, and so satisfies the promise, and it contains an integral vector for LWE instances but not for uniform instances

(Proposition 11). Thus a single call to an algorithm for integer feasibility distinguishes the two cases (Theorem 12). Combined with a result of Regev, this gives consequences for GapSVP and SIVP for quantum algorithms (Corollary 14).

For $A \in \mathbb{Z}_q^{n \times m}$ we consider the q -ary lattice

$$\Lambda_q(A^\top) = \{y \in \mathbb{Z}^m : y \equiv A^\top s \pmod{q} \text{ for some } s \in \mathbb{Z}_q^n\},$$

which contains $q\mathbb{Z}^m$ and hence has full rank. For an LWE instance (A, t) with error e , we have $t - e \in \Lambda_q(A^\top)$, so t is close to $\Lambda_q(A^\top)$ when e is short. In this subsection, n is the dimension of s , and also of the lattice problems in Theorem 13, while the lattices $\Lambda_q(A^\top)$ and the parallelepipeds $Q(B, t, r)$ have dimension m .

By Proposition 6, the parallelepipeds defined by $\Lambda_q(A^\top)$ satisfy the promise whenever they are thin, and they are thin if their radius is small compared with $\lambda_1(\Lambda_q(A^\top))$. We first show, by a standard counting argument, that $\lambda_1(\Lambda_q(A^\top))$ is large with high probability over the choice of A .

Lemma 10. *Let q be prime, and let $A \in \mathbb{Z}_q^{n \times m}$ be uniformly random. Then $\lambda_1(\Lambda_q(A^\top)) > q/8$ with probability at least $1 - q^n 2^{-m}$.*

Proof. Let $L = \Lambda_q(A^\top)$. A nonzero vector of $q\mathbb{Z}^m$ has ℓ_∞ -norm at least q , and every other vector of L is congruent modulo q to $A^\top s$ for some nonzero $s \in \mathbb{Z}_q^n$. For fixed nonzero s , the vector $A^\top s$ is uniformly random in $\mathbb{Z}_q^m$, since q is prime. At most $2 \lfloor q/8 \rfloor + 1 \leq q/2$ residues modulo q have a representative c with $|c| \leq q/8$. Hence, for fixed nonzero s , all entries of $A^\top s$ are such residues with probability at most 2^{-m} . By the union bound over the fewer than q^n nonzero s , the probability that L has a nonzero vector of ℓ_∞ -norm at most $q/8$ is at most $q^n 2^{-m}$. Otherwise every nonzero $y \in L$ satisfies $\|y\| \geq \|y\|_\infty > q/8$, and so $\lambda_1(L) > q/8$. $\square$

We now show that the parallelepiped defined by an instance (A, t) is thin with high probability, and that it distinguishes LWE instances from uniform instances.

Proposition 11. *Let q be prime, let $k = \lceil \log_2 q \rceil$ and $m = 3nk$, and assume $q > 16m^2$. Let $r = \lfloor (q-1)/(16m^2) \rfloor$, let χ be a probability distribution on $\mathbb{Z}$, and let ε_χ be the probability that a vector $e \in \mathbb{Z}^m$ with independent entries distributed according to χ satisfies $\|e\|_\infty > r$. Given A , a basis $B \in \{0, \dots, q\}^{m \times m}$ of $\Lambda_q(A^\top)$ can be computed in time polynomial in n and $\log q$. For $t \in \mathbb{Z}_q^m$, let $Q = Q(B, t, r)$. If (A, t) is an LWE instance or a uniform instance, then Q is thin with probability at least $1 - 2^{-2nk}$, and Q contains an integral vector with probability at least $1 - \varepsilon_\chi$ in the first case and at most 2^{-m} in the second.*

Proof. We first compute B and Q . We let B be the Hermite normal form basis of the lattice generated by the columns of $A^\top$ and the vectors $qe_1, \dots, qe_m$, that is, of $\Lambda_q(A^\top)$, which can be computed in time $\text{poly}(n, \log q)$ [41, Theorem 5.3]. For every i , we have $qe_i \in \Lambda_q(A^\top)$, so $qe_i = By$ for some $y \in \mathbb{Z}^m$. Since B is lower triangular, the first i rows of this system give $y_1 = \dots = y_{i-1} = 0$ and $B_{ii}y_i = q$. Hence each diagonal entry of B divides q , and the other entries in its row are nonnegative and smaller, so $B \in \{0, \dots, q\}^{m \times m}$. Then $Q = \{x : Bx \leq t + r\mathbf{1}, -Bx \leq -t + r\mathbf{1}\}$ has integral data, $2m$ inequalities, and encoding size polynomial in n and $\log q$.

Next we show that Q is thin with high probability. In LWE instances and in uniform instances alike, A is uniformly random, so $\lambda_1(\Lambda_q(A^\top)) > q/8$ with probability at least $1 - q^n 2^{-m}$ by Lemma 10. Moreover, $q^n \leq 2^{nk} = 2^{m/3}$, so this probability is at least $1 - 2^{-2nk}$. Since $q > 16m^2$, we have $r \geq 1$ and $r < q/(16m^2)$. Hence, if $\lambda_1(\Lambda_q(A^\top)) > q/8$, then $r < (q/8)/(2m^2) < \lambda_1(\Lambda_q(A^\top))/(2m^2)$, and so Q is thin.

We then show that, for an LWE instance, Q contains an integral vector with probability at least $1 - \varepsilon_\chi$. By the definition of ε_χ , the error e satisfies $\|e\|_\infty \leq r$ with probability at least $1 - \varepsilon_\chi$. In this case, since $t \equiv A^\top s + e \pmod{q}$, the vector $t - e \in \Lambda_q(A^\top)$ lies within ℓ_∞ -distance r of t , and so Q contains an integral vector.

Finally, we show that, for a uniform instance, Q contains an integral vector with probability at most 2^{-m} . If Q contains an integral vector, then $t \equiv A^\top s + e' \pmod{q}$ for some $s \in \mathbb{Z}_q^n$ and $e' \in \{-r, \dots, r\}^m$. There are at most $q^n(2r+1)^m$ such residues, and so Q contains an integral vector with probability at most $q^n(2r+1)^m/q^m$. Since $q > 16m^2 \geq 16$, we have $2r+1 \leq q/(8m^2)+1 \leq q/4$, and $q^n \leq 2^{nk} = 2^{m/3}$, so this probability is at most $2^{m/3}4^{-m} \leq 2^{-m}$. $\square$

For the error distribution $\bar{\Psi}_\alpha$ and suitable parameters, Proposition 11 gives an explicit distribution of hard instances.

Theorem 12. *Let n be sufficiently large, let $q = q(n)$ be a prime in $[n^3, 2n^3]$, which exists by Bertrand's postulate, let $k = \lceil \log_2 q \rceil$, let $\alpha = 1/(200n^2k^2 \log n)$, and let $m = 3nk$. Let $\mathcal{D}$ be the distribution of the parallelepiped Q of Proposition 11 with $\chi = \bar{\Psi}_\alpha$ when (A, t) is an LWE instance or a uniform instance, each with probability $1/2$. The distribution $\mathcal{D}$ can be sampled in polynomial time, its samples are parallelepipeds in $\mathbb{R}^m$ given by integral constraints with coefficients of absolute value less than $2q$, and a sample of $\mathcal{D}$ is thin with probability at least $1 - 2^{-2nk}$. Suppose that a randomized polynomial-time algorithm decides integer feasibility correctly with probability at least $1/2 + n^{-c}$ for some constant $c > 0$, over the choice of Q from $\mathcal{D}$ and its own randomness. Then there is a randomized polynomial-time algorithm that solves decision-LWE $_{n,q,\bar{\Psi}_\alpha}$ with m samples.*

Proof. Since n is sufficiently large, we have $q > 16m^2$, so Proposition 11 applies with $\chi = \bar{\Psi}_\alpha$.

We first prove the properties of $\mathcal{D}$ stated in the theorem. The distribution $\mathcal{D}$ can be sampled in polynomial time, since q can be found in time polynomial in n by testing the integers in $[n^3, 2n^3]$ for primality, $\bar{\Psi}_\alpha$ can be sampled in polynomial time, and Q can be computed in polynomial time. Its samples are the parallelepipeds Q in $\mathbb{R}^m$, given by the integral constraints $t - r\mathbf{1} \leq Bx \leq t + r\mathbf{1}$, whose coefficients have absolute value less than $2q$, since $B \in \{0, \dots, q\}^{m \times m}$, $t \in \{0, \dots, q-1\}^m$, and $r < q$. By Proposition 11, a sample of $\mathcal{D}$ is thin with probability at least $1 - 2^{-2nk}$.

Next we show that, for an LWE instance with error e , Q contains an integral vector with probability at least $1 - 2m \exp(-\pi \log^2 n)$. By Proposition 11, this probability is at least $1 - \varepsilon_\chi$. Moreover, $r = \lfloor (q-1)/(16m^2) \rfloor \geq \alpha q \log n$ for sufficiently large n , since $\alpha q \log n = q/(200n^2k^2)$, $16m^2 = 144n^2k^2$, $q \geq n^3$, and $k = O(\log n)$. Each entry of e is distributed according to $\bar{\Psi}_\alpha$, that is, it equals $\lfloor X \rfloor$ with X normal with mean zero and standard deviation $\bar{\sigma} = \alpha q / \sqrt{2\pi}$. Then each entry has absolute value greater than r with probability at most $\Pr[|X| \geq r + \frac{1}{2}] \leq \Pr[|X| \geq \alpha q \log n] = \Pr[|X| \geq \sqrt{2\pi} \log n \cdot \bar{\sigma}] \leq 2 \exp(-\pi \log^2 n)$, by the Gaussian tail bound $\Pr[|X| \geq \tau \bar{\sigma}] \leq 2 \exp(-\tau^2/2)$ for $\tau \geq 0$. Hence, by the union bound over the m entries, $\varepsilon_\chi \leq 2m \exp(-\pi \log^2 n)$, which is smaller than every inverse polynomial in n for sufficiently large n .

It remains to construct the algorithm for decision-LWE. Given (A, t) , our algorithm computes Q as in Proposition 11, runs the given algorithm on Q , and outputs 1 if and only if the given algorithm reports that Q contains an integral vector. Since $q \leq 2n^3$, our algorithm runs in time polynomial in n . Let β_1 and β_0 be the probabilities that the given algorithm answers incorrectly on Q for LWE instances and for uniform instances, respectively. Then $\beta_1 + \beta_0 \leq 1 - 2n^{-c}$, since the given algorithm answers incorrectly with probability at most $1/2 - n^{-c}$ for Q drawn from $\mathcal{D}$. By Proposition 11, the output is 1 with probability at least $1 - \varepsilon_\chi - \beta_1$ on LWE instances and at most $2^{-m} + \beta_0$ on uniform instances. These probabilities differ by at least $2n^{-c} - \varepsilon_\chi - 2^{-m}$, which is at least n^{-c} for sufficiently large n . $\square$

Proof of Theorem B. By Proposition 6, thin parallelepipeds satisfy $P' = P_I$. So a sample of $\mathcal{D}$ satisfies $P' = P_I$ with high probability. Moreover, $2/3 \geq 1/2 + 1/n$ for $n \geq 6$. Hence Theorem 12 gives Theorem B. $\square$

We use the following result of Regev.

Theorem 13 ([37, Theorem 3.1, Section 3.3, and Lemma 4.4]). *Let $q = q(n)$ be a prime and $m = m(n)$ an integer, both bounded by a polynomial in n , and let $\alpha = \alpha(n) \in (0, 1)$ with $\alpha q > 2\sqrt{n}$ for all sufficiently large n . If there is a randomized polynomial-time algorithm that solves decision-LWE $_{n,q,\bar{\Psi}_\alpha}$ with m samples, then there is a polynomial-time quantum algorithm that solves GapSVP $_\gamma$ and SIVP $_\gamma$ on n -dimensional lattices for some $\gamma = \tilde{O}(n/\alpha)$.*

Theorem 13 is stated with our notion of solving decision-LWE. Regev's results apply to it since, by averaging over s , an algorithm that solves decision-LWE in our sense distinguishes, for a fraction at least $n^{-c}/2$ of all $s \in \mathbb{Z}_q^n$, the LWE instances with secret s from uniform instances, with output probabilities that differ by at least $n^{-c}/2$, as required in [37, Lemma 4.4]. Combining Theorem 12 with Theorem 13, we obtain the consequence for quantum algorithms stated in the introduction.

Corollary 14. *Suppose that there is a randomized polynomial-time algorithm as in Theorem 12. Then, for some $\gamma = \tilde{O}(n^3)$, there is a polynomial-time quantum algorithm that solves GapSVP $_\gamma$ and SIVP $_\gamma$ on n -dimensional lattices.*

Proof. Let q, k, α , and m be as in Theorem 12. By Theorem 12, there is a randomized polynomial-time algorithm that solves decision-LWE $_{n,q,\bar{\Psi}_\alpha}$ with m samples. Theorem 13 applies, since $q \leq 2n^3$ is prime, q and $m = 3nk$ are bounded by a polynomial in n , and $\alpha q \geq n^3/(200n^2k^2 \log n) > 2\sqrt{n}$ for sufficiently large n , and it gives $\gamma = \tilde{O}(n/\alpha) = \tilde{O}(n^3)$. $\square$

4.3 Refutations

We conclude with a remark on *refutations*, that is, sets of CG cuts for a polyhedron with empty intersection.

Remark 15. By Proposition 6, thin parallelepipeds and their rational subpolyhedra that contain no integral vector have empty Chvátal closure. Since the Chvátal closure is the intersection of finitely many CG cuts [40], Helly's theorem then gives them refutations with at most $n+1$ CG cuts. Given a short dual basis, refutations can be computed in polynomial time, but both refutations and short dual bases are hard to find.

We first show that a short dual basis gives a refutation. Let $Q(B, t, r)$ be a thin parallelepiped, let L be the lattice with basis B , let $P \subseteq Q(B, t, r)$ be a rational polyhedron without integral vectors, and let $w_1, \dots, w_n$ be the basis of L^* given by Lemma 5. If $P = \emptyset$, then the CG cut $\mathbf{0}^\top x \leq -1$ is a refutation, so we may assume $P \neq \emptyset$. As in the proof of Proposition 6, P has width less than one along each $u_i = B^\top w_i$, so the CG cuts for P along u_i and $-u_i$ either have empty intersection or together give an equation $u_i^\top x = \gamma_i$ with $\gamma_i \in \mathbb{Z}$. If they give an equation for every i , the equations define a single integral vector z , since $u_1^\top, \dots, u_n^\top$ are the rows of a unimodular matrix, and $z \notin P$. An inequality defining P that z violates, scaled to have integral coefficients and with its right-hand side rounded down, is a CG cut violated by z , and together with the n equations it gives a contradiction. Hence P has a refutation with at most $2n+1$ CG cuts. Given $w_1, \dots, w_n$, these cuts can be computed in polynomial time. For $P = Q(B, t, r)$, the cuts along u_i are $u_i^\top x \leq \lfloor w_i^\top t + r \|w_i\|_1 \rfloor$ and $u_i^\top x \geq -\lfloor -w_i^\top t + r \|w_i\|_1 \rfloor$, as in Babai's round-off algorithm [6].

We now show that refutations and short dual bases are hard to find. Given an algorithm that finds a refutation whenever one exists, we decide integer feasibility for thin parallelepipeds

by running it and checking by linear programming whether its output is a refutation. So, by Theorem 7, finding any refutation is as hard as $\text{BDD}_{1/(4n^2)}$. The same holds for finding $w_1, \dots, w_n$, or any unimodular matrix along whose rows $Q(B, t, r)$ has width less than one, since such a matrix gives, by linear programming, the CG cuts of the proof of Lemma 2, and thus decides whether $Q(B, t, r)$ contains an integral vector. Similarly, since the parallelepipeds of Theorem 12 are thin with high probability, finding any refutation, or such a matrix, for them is as hard as decision-LWE.

5 SAT polytopes of Chvátal rank at most one

In this section we construct SAT polytopes defined by lattices, on which the proofs of Theorems C and D rest, and we show that they satisfy the promise when the lattice has no short nonzero vectors (Proposition 16).

A *literal* on variables $x_1, \dots, x_N$ is an affine function x_i or $\bar{x}_i = 1 - x_i$. For a literal ℓ , we write $\bar{\ell} = 1 - \ell$, which is again a literal. A *formula* F in conjunctive normal form is a finite family of *clauses*, each of which is a nonempty set of literals. The *SAT polytope* of F is

$$P(F) = \left\{ x \in [0, 1]^N : \sum_{\ell \in T} \ell(x) \geq 1 \text{ for every } T \in F \right\}.$$

Its integral vectors are the *satisfying assignments* of F . Without the promise $P(F)' = P(F)_I$, satisfiability is NP-complete for formulas whose clauses have at most three literals [11].

The proofs of Theorems C and D have the same structure as those of Theorems A and B. The parallelepipeds $Q(B, t, r)$ are replaced by the formulas of Proposition 16 below, which also takes the place of Proposition 6. Unlike Proposition 6, Proposition 16 does not follow directly from Lemma 2, and we prove it in four parts. In Section 5.1, we show that the Chvátal closure of our SAT polytopes lies in a face defined by *exact-one equations*, which set a sum of literals equal to one. In Section 5.2, we model arithmetic on binary numbers by such equations, through extended formulations, and we show that its identities hold on the whole face. These identities give the integral linear functions along which we take CG cuts. In Section 5.3, we encode each entry of the error by its residues modulo small pairwise coprime integers, and binary arithmetic recovers the error and checks that the corresponding vector lies in the lattice. In Section 5.4, we show that, for each residue, a vector of the dual lattice close to a unit vector divided by the modulus gives integral linear functions of width less than one. As in the proof of Lemma 2, CG cuts along them fix all residues, and hence leave at most one integral vector (Proposition 24).

Residues are needed because the errors can be exponentially large: our formulas need an integral lattice, and making a general rational lattice integral can create such errors (Lemma 25). Such errors cannot be written in unary with polynomially many 0/1 variables. In binary, the digits of the errors depend on residues modulo powers of two as large as the errors themselves, which our CG cuts cannot fix. Residues modulo small integers represent the errors compactly, and one round of CG cuts can fix these residues. The price is a factor equal to the largest modulus in the condition on λ_1 in Proposition 16(b).

The lattices in our formulas are given by congruences. For integers $D \geq 1$, $m \geq 1$, and $m' \geq 1$, and for $H \in \mathbb{Z}^{m' \times m}$, let

$$\Lambda_D^\perp(H) = \{y \in \mathbb{Z}^m : Hy \equiv \mathbf{0} \pmod{D}\}.$$

This is a D -ary lattice, that is, $D\mathbb{Z}^m \subseteq \Lambda_D^\perp(H) \subseteq \mathbb{Z}^m$. For $h \geq 1$, let $\mathcal{B}_h = \{-(2^{h-1}-1), \dots, 2^{h-1}\}^m$, a box with 2^h integers in each coordinate.

Proposition 16. *Let $D \geq 2$, $m \geq 1$, and $m' \geq 1$ be integers, let $H \in \{0, \dots, D-1\}^{m' \times m}$, let $M = \Lambda_D^\perp(H)$, and let $t \in \mathbb{Z}^m$. Let $h \geq 1$, let $p = (p_1, \dots, p_s)$, where $p_1, \dots, p_s \geq 2$ are pairwise coprime integers with $p_1 \cdots p_s \geq 2^h$, and let $p_{\max} = \max_j p_j$. In time polynomial in $m, m', \sum_j p_j, \log D$, and the encoding size of t , we can compute a formula $F = F(p, h, H, D, t)$ whose clauses have at most three literals, with the following properties.*

- (a) *The satisfying assignments of F correspond one-to-one to the vectors $\tilde{y} \in M$ with $t - \tilde{y} \in \mathcal{B}_h$.*
- (b) *If $\lambda_1(M) > m^{3/2} p_{\max}(2^h - 1)/2$, then every rational subpolyhedron of $P(F)$ has Chvátal rank at most one, and F has at most one satisfying assignment.*
- (c) *F has $O(m(\sum_j p_j) \log(p_1 \cdots p_s) + m' m h \log D)$ variables.*

We call the formula $F(p, h, H, D, t)$ *thin* if $\lambda_1(\Lambda_D^\perp(H)) > m^{3/2} p_{\max}(2^h - 1)/2$, that is, if it satisfies the assumption of Proposition 16(b). Sections 5.1–5.4 prove Proposition 16, and Theorems 26 and 29 use only Proposition 16 from this section.

5.1 Exact-one systems

In this subsection we show that SAT polytopes can express systems of equations that set sums of literals equal to one, and that this preserves the promise (Lemma 18). We first show that adding variables that are integral affine functions of the others commutes with taking integer hulls and Chvátal closures.

Lemma 17. *Let $P \subseteq \mathbb{R}^N$ be a rational polyhedron, let $E \in \mathbb{Z}^{N' \times N}$, let $f \in \mathbb{Z}^{N'}$, and let $\varphi(x) = (x, Ex + f)$. Then φ maps $P \cap \mathbb{Z}^N$ bijectively onto $\varphi(P) \cap \mathbb{Z}^{N+N'}$, $\varphi(P)_I = \varphi(P_I)$, and $\varphi(P)' = \varphi(P')$.*

Proof. We may assume $P \neq \emptyset$. The map φ is injective, and a vector $\varphi(x)$ is integral if and only if x is integral. Hence φ maps $P \cap \mathbb{Z}^N$ bijectively onto $\varphi(P) \cap \mathbb{Z}^{N+N'}$, and so $\varphi(P)_I = \varphi(P_I)$.

We now show that $\varphi(P)' = \varphi(P')$. Since $\varphi(P)$ is a rational polyhedron, $\varphi(P)' \subseteq \varphi(P)$, so every vector of $\varphi(P)'$ is $\varphi(x)$ for some $x \in P$. Let $a = (a_1, a_2) \in \mathbb{Z}^{N+N'}$ and $c = a_1 + E^\top a_2 \in \mathbb{Z}^N$. Then $a^\top \varphi(x) = c^\top x + a_2^\top f$ and $\delta_{\varphi(P)}(a) = \delta_P(c) + a_2^\top f$, where $a_2^\top f$ is an integer. Hence the CG cut for $\varphi(P)$ along a holds at $\varphi(x)$ if and only if the CG cut for P along c holds at x . Since every $c \in \mathbb{Z}^N$ arises in this way, for instance with $a_2 = \mathbf{0}$, the vector $\varphi(x)$ lies in $\varphi(P)'$ if and only if x lies in P' . $\square$

For a family $\mathcal{T}$ of nonempty sets of literals on variables $x_1, \dots, x_N$, let

$$X(\mathcal{T}) = \left\{ x \in [0, 1]^N : \sum_{\ell \in T} \ell(x) = 1 \text{ for every } T \in \mathcal{T} \right\}.$$

We call the equations of $X(\mathcal{T})$ *exact-one equations*, and $X(\mathcal{T})$ an *exact-one system*.

Lemma 18. *Let $\mathcal{T}$ be a family of nonempty sets of literals on variables $x_1, \dots, x_N$. In time polynomial in $N + \sum_{T \in \mathcal{T}} |T|$, we can compute a formula F with at most $N + \sum_{T \in \mathcal{T}} |T|$ variables, whose clauses have at most three literals, and whose satisfying assignments correspond one-to-one to the integral vectors of $X(\mathcal{T})$. If every rational subpolyhedron of $X(\mathcal{T})$ has Chvátal rank at most one, then so does every rational subpolyhedron of $P(F)$.*

Proof. We first construct a family $\mathcal{T}_3$ of sets of at most three literals, on the variables $x_1, \dots, x_N$ and N' new variables, and a map $\varphi(x) = (x, Ex + f)$ with $E \in \mathbb{Z}^{N' \times N}$ and $f \in \mathbb{Z}^{N'}$, such that $X(\mathcal{T}_3) = \varphi(X(\mathcal{T}))$. Let $T = \{\ell_1, \dots, \ell_\tau\} \in \mathcal{T}$ with $\tau \geq 4$. We introduce new variables $x'_1, \dots, x'_{\tau-3}$ and replace T by the $\tau - 2$ sets

$$\{\ell_1, \ell_2, x'_1\}, \{\bar{x}'_1, \ell_3, x'_2\}, \dots, \{\bar{x}'_{\tau-4}, \ell_{\tau-2}, x'_{\tau-3}\}, \{\bar{x}'_{\tau-3}, \ell_{\tau-1}, \ell_\tau\}.$$

A vector satisfies the new equations if and only if $x'_j = 1 - \ell_1 - \dots - \ell_{j+1}$ for $j = 1, \dots, \tau - 3$ and $\ell_1 + \dots + \ell_\tau = 1$. Moreover, if the ℓ_i are nonnegative and sum to one, then these values of x'_j lie in $[0, 1]$. Doing this for every set with more than three literals gives $\mathcal{T}_3$. We let φ append to x the values of the new variables x'_j . Since each x'_j is an integral affine function of x , the map φ has this form, and $X(\mathcal{T}_3) = \varphi(X(\mathcal{T}))$.

Next we encode $\mathcal{T}_3$ by a formula and prove the first statement. Let F consist of the clause $\{\ell_1\}$ for every $\{\ell_1\} \in \mathcal{T}_3$, the clauses $\{\ell_1, \ell_2\}$ and $\{\bar{\ell}_1, \bar{\ell}_2\}$ for every $\{\ell_1, \ell_2\} \in \mathcal{T}_3$, and the clauses $\{\ell_1, \ell_2, \ell_3\}$, $\{\bar{\ell}_1, \bar{\ell}_2\}$, $\{\bar{\ell}_1, \bar{\ell}_3\}$, and $\{\bar{\ell}_2, \bar{\ell}_3\}$ for every $\{\ell_1, \ell_2, \ell_3\} \in \mathcal{T}_3$. Let S be the set of vectors of $P(F)$ that satisfy $\ell_1 + \ell_2 + \ell_3 = 1$ for every $\{\ell_1, \ell_2, \ell_3\} \in \mathcal{T}_3$. Then $S = X(\mathcal{T}_3)$. Indeed, the clauses of a set with one or two literals hold together if and only if its literals sum to one. For a set $\{\ell_1, \ell_2, \ell_3\}$, if $\ell_1 + \ell_2 + \ell_3 = 1$ and the literals are nonnegative, then $\ell_1 + \ell_2 \leq 1$, $\ell_1 + \ell_3 \leq 1$, and $\ell_2 + \ell_3 \leq 1$, which are the inequalities of the other three clauses. Every integral vector of $P(F)$ lies in S , since there the clauses of each set $\{\ell_1, \ell_2, \ell_3\}$ force exactly one of its literals to be one. So, by Lemma 17, the satisfying assignments of F are the vectors $\varphi(x)$ with x an integral vector of $X(\mathcal{T})$, and so they correspond one-to-one to these vectors. Since $\mathcal{T}_3$ has at most $\sum_{T \in \mathcal{T}} |T|$ sets, each giving at most four clauses, F can be computed in time polynomial in $N + \sum_{T \in \mathcal{T}} |T|$. Moreover, F has $N + N' \leq N + \sum_{T \in \mathcal{T}} |T|$ variables, since a set with $\tau \geq 4$ literals adds $\tau - 3$ new variables.

We now prove the second statement. Let R be a rational subpolyhedron of $P(F)$, and assume that every rational subpolyhedron of $X(\mathcal{T})$ has Chvátal rank at most one. We show that R' lies in the face $R \cap S$ of R , and that this face is the image under φ of a rational subpolyhedron of $X(\mathcal{T})$, to which the assumption applies. For every $\{\ell_1, \ell_2, \ell_3\} \in \mathcal{T}_3$, the three inequalities $\ell_1 + \ell_2 \leq 1$, $\ell_1 + \ell_3 \leq 1$, and $\ell_2 + \ell_3 \leq 1$ are valid for $P(F)$, and hence for R , and their sum is $2(\ell_1 + \ell_2 + \ell_3) \leq 3$. Since $\ell_1 + \ell_2 + \ell_3$ is an affine function with integral coefficients and an integral constant term, this gives the CG cut $\ell_1 + \ell_2 + \ell_3 \leq 1$ for R . Hence $R' \subseteq R \cap S$. Since $\ell_1 + \ell_2 + \ell_3 \geq 1$ is valid for R for every such set, the set $R \cap S$ is a face of R . The Chvátal closure of a face of a rational polyhedron is the intersection of the face with the Chvátal closure of the polyhedron [41, Section 23.1], so $(R \cap S)' = R' \cap S = R'$. Every integral vector of R lies in S , so $R_I = (R \cap S)_I$. Since $R \cap S \subseteq X(\mathcal{T}_3) = \varphi(X(\mathcal{T}))$, we have $R \cap S = \varphi(\tilde{R})$, where $\tilde{R} = \{x : \varphi(x) \in R \cap S\}$ is a rational subpolyhedron of $X(\mathcal{T})$. By Lemma 17 and our assumption, $(R \cap S)' = \varphi(\tilde{R}') = \varphi(\tilde{R}_I) = (R \cap S)_I$. Hence $R' = R_I$. $\square$

5.2 Binary arithmetic

We now model arithmetic on binary numbers by exact-one systems. We introduce constraints that set some 0/1 variables equal to a given function of earlier ones, and we write each of them with the standard extended formulation of the convex hull of its solutions. With them, we compute weighted sums modulo an integer in binary (Lemma 20), which Section 5.3 uses to recover the entries from their residues and to check that a vector lies in the lattice.

We define *functional formulations*, the systems built from these constraints. We write x_u for the variable with index u , and we also call it the variable u . A functional formulation X starts from *free* variables, with *multiple-choice constraints* $\sum_{u \in I} x_u = 1$ for some disjoint sets I of free variables, and adds a finite sequence of functional constraints and fixings, as follows. A *functional constraint* g introduces new variables, its *outputs*, as a function of at most three earlier variables, its *inputs*, that is, of free variables or variables introduced by earlier functional constraints. Writing I_g and O_g for the sequences of its inputs and outputs, and x_{I_g} and x_{O_g} for the corresponding vectors of variables, it requires $x_{O_g} = f_g(x_{I_g})$ for a given function $f_g : \{0, 1\}^{|I_g|} \rightarrow \{0, 1\}^{|O_g|}$. Its *solutions* are the $2^{|I_g|}$ vectors $(v, f_g(v))$ with $v \in \{0, 1\}^{|I_g|}$. We write it with the standard extended formulation of

the convex hull of its solutions, that is, with new variables $x_{g,v}$ for $v \in \{0,1\}^{|I_g|}$ and the constraints

$$x_{g,v} \geq 0, \quad \sum_v x_{g,v} = 1, \quad (x_{I_g}, x_{O_g}) = \sum_v x_{g,v} (v, f_g(v)). \quad (1)$$

For instance, for a functional constraint g with inputs u, w , output o , and $f_g(v_1, v_2) = v_1 v_2$, the solutions are $(0, 0, 0)$, $(0, 1, 0)$, $(1, 0, 0)$, and $(1, 1, 1)$, and the last equation in (1) reads $x_u = x_{g,(1,0)} + x_{g,(1,1)}$, $x_w = x_{g,(0,1)} + x_{g,(1,1)}$, and $x_o = x_{g,(1,1)}$. Finally, some outputs are *fixed* to 0 or 1. The variables of X are its free variables, the outputs, and the variables $x_{g,v}$. The functional formulation X is the system

$$\begin{aligned} \sum_{u \in I} x_u &= 1 && \text{for every set } I \text{ of a multiple-choice constraint,} \\ (1) &&& \text{for every functional constraint } g, \\ x_u &= \epsilon_u && \text{for every output } u \text{ fixed to } \epsilon_u \in \{0, 1\}, \\ 0 &\leq x_u \leq 1 && \text{for every free variable } u, \end{aligned}$$

and we also denote by X the polyhedron it defines. All these equations are exact-one equations in the literals x_u and $\bar{x}_u = 1 - x_u$. Indeed, each entry x_u of (x_{I_g}, x_{O_g}) gives, in the last equation in (1), an equation $x_u = \sum_{v: u(v)=1} x_{g,v}$, which we write as $\bar{x}_u + \sum_{v: u(v)=1} x_{g,v} = 1$, where $u(v)$ is the corresponding entry of $(v, f_g(v))$. We write fixing u to 1 or 0 as $x_u = 1$ or $\bar{x}_u = 1$, respectively. Thus, since $0 \leq x_u \leq 1$ for every variable u and every $x \in X$, the functional formulation X is an exact-one system, that is, $X = X(\mathcal{T})$ as in Section 5.1, where $\mathcal{T}$ is the family of the sets of literals of these equations. For $x \in X$, we denote by $b(x)$ the vector of its free variables. An *integral linear function* of the variables of X is a linear combination of these variables with integer coefficients. For a polyhedron P and an affine function ψ , we write $\delta_P(\psi) = \sup\{\psi(x) : x \in P\}$ and $\text{width}_P(\psi) = \delta_P(\psi) + \delta_P(-\psi)$.

The next lemma gives the two properties of functional formulations that we use. Part (a) is the reason for using extended formulations: for a general formulation of the same integral vectors, such as one given by clauses, an affine identity that holds at the solutions of a functional constraint can fail at fractional vectors.

Lemma 19. *Let X be a functional formulation.*

(a) *Let g be a functional constraint of X , and let η be an affine function that vanishes at the solutions $(v, f_g(v))$ of g . Then $\eta(x_{I_g}, x_{O_g}) = 0$ for every $x \in X$.*

(b) *Let $x \in X$ be such that $b(x)$ is integral. Then x is integral, and it is the only vector of X with free variables $b(x)$.*

Proof. (a) For every $x \in X$, the vector (x_{I_g}, x_{O_g}) is a convex combination of the solutions of g , and η vanishes at them, and hence at this convex combination, since η is affine.

(b) Going through the functional constraints in order, suppose that the inputs of g take 0/1 values v^* in x , which holds for free inputs by assumption and for the other inputs by induction. Then $v^* = \sum_v x_{g,v} v$, and since v^* is a vertex of $[0, 1]^{|I_g|}$, we obtain $x_{g,v^*} = 1$ and $x_{g,v} = 0$ for $v \neq v^*$, and so $x_{O_g} = f_g(v^*)$. Hence all variables of x are integral and determined by $b(x)$. $\square$

The next lemma computes in binary the residue of a weighted sum of variables modulo an integer, and writes the quotient as an integral linear function.

Lemma 20. *Let X_0 be a functional formulation, let $k \geq 1$, let $u_1, \dots, u_k$ be variables of X_0 with integer weights $\alpha_1, \dots, \alpha_k \geq 0$, let $K \geq 1$ be an integer, and let L be the smallest integer with $2^L \geq 2K$. In time polynomial in k , $\log K$, and $\log(1 + \max_i \alpha_i)$, we can add to X_0 new functional*

constraints with at most three inputs and at most two outputs each, $O(kL)$ in total, and new fixings, obtaining a functional formulation X with the same free variables and multiple-choice constraints, outputs $\rho_0, \dots, \rho_{L-1}$ of X for the residue, and an integral linear function κ of the variables of X for the quotient, with the following properties.

(a) We have $\sum_i \alpha_i x_{u_i} = K\kappa(x) + \sum_l 2^l x_{\rho_l}$ for every $x \in X$.

(b) Every integral vector of X_0 extends to exactly one integral vector of X , and in it $\rho_0, \dots, \rho_{L-1}$ take the binary digits of $(\sum_i \alpha_i x_{u_i}) \bmod K$.

Proof. We may assume that $\alpha_i < K$ for every i . Indeed, writing $\alpha_i = K \lfloor \alpha_i/K \rfloor + \alpha'_i$ with $\alpha'_i \in \{0, \dots, K-1\}$, we have $\sum_i \alpha_i x_{u_i} = \sum_i \alpha'_i x_{u_i} + K \sum_i \lfloor \alpha_i/K \rfloor x_{u_i}$. Since $\alpha'_i \equiv \alpha_i \pmod{K}$, the residue in (b) is the same for both weights. So it suffices to prove the lemma for the weights α'_i , and then to add $\sum_i \lfloor \alpha_i/K \rfloor x_{u_i}$ to κ .

We first describe how to add a fixed multiple of a variable to a number whose binary digits are variables, and how to subtract it. Here, for a sequence $r = (r_0, \dots, r_{L-1})$ of outputs, the number is $r(x) = \sum_l 2^l x_{r_l}$, and we use the same notation for every such sequence. The multiple is θx_u for a variable u of X and an integer $\theta \in \{0, \dots, 2^L - 1\}$ with binary digits $\theta_0, \dots, \theta_{L-1}$. Some inputs of these operations are constant. To treat them like the other inputs, we obtain the constants 0 and 1 as the outputs of two functional constraints without inputs. To add θ times u to r , that is, to obtain a sequence $s = (s_0, \dots, s_{L-1})$ of outputs with $s(x) = r(x) + \theta x_u$, we introduce, for $l = 0, \dots, L-1$, a functional constraint with inputs r_l , u , and o_l , and with outputs s_l and o_{l+1} , the *digit* and the *carry*. Its function maps the values $\hat{x}_{r_l}, \hat{x}_u, \hat{x}_{o_l} \in \{0, 1\}$ of the inputs to the values $\hat{x}_{s_l}, \hat{x}_{o_{l+1}} \in \{0, 1\}$ with $\hat{x}_{r_l} + \theta_l \hat{x}_u + \hat{x}_{o_l} = \hat{x}_{s_l} + 2\hat{x}_{o_{l+1}}$. The first carry o_0 is the constant 0, and we fix the last carry o_L to 0. By Lemma 19(a), we have $x_{r_l} + \theta_l x_u + x_{o_l} = x_{s_l} + 2x_{o_{l+1}}$ for every $x \in X$ and every l , and multiplying by 2^l and summing over l gives $s(x) = r(x) + \theta x_u$, since $x_{o_0} = 0$ and o_L is fixed to 0. Let $\hat{x}$ be a 0/1 vector on the variables of X in which each output takes the value of its function at the values of its inputs, and in which r takes the binary digits of an integer z . If $z + \theta \hat{x}_u < 2^L$, then, as in schoolbook addition, s takes the binary digits of $z + \theta \hat{x}_u$ and $\hat{x}_{o_L} = 0$. To subtract θ times u from r , that is, to obtain s with $s(x) = r(x) - \theta x_u$, we do the same, but with $\hat{x}_{r_l} - \theta_l \hat{x}_u - \hat{x}_{o_l} = \hat{x}_{s_l} - 2\hat{x}_{o_{l+1}}$. As with additions, $s(x) = r(x) - \theta x_u$ for every $x \in X$. For $\hat{x}$ and z as above, if $z - \theta \hat{x}_u \geq 0$, then, as in schoolbook subtraction, s takes the binary digits of $z - \theta \hat{x}_u$ and $\hat{x}_{o_L} = 0$.

We now use these operations to compute the residue, by adding the terms $\alpha_i x_{u_i}$ one at a time and reducing modulo K after each addition. Let every entry of ρ^0 be the constant 0. For $i = 1, \dots, k$, we obtain ρ^i from ρ^{i-1} in four steps. We add α_i times u_i to ρ^{i-1} , with result s^i . To compare $s^i(x)$ with K , we subtract K times the constant 1 from s^i . We use only the last carry of this subtraction, which we call β_i and do not fix. For integral $x \in X$, it takes the value 1 exactly when $s^i(x) < K$. We introduce a functional constraint with input β_i and output q_i , with $\hat{x}_{q_i} = 1 - \hat{x}_{\beta_i}$, so that q_i indicates whether we must subtract K . Finally, we subtract K times q_i from s^i , with result ρ^i . We let $\rho = \rho^k$ and $\kappa(x) = \sum_i x_{q_i}$. Each of the $3k$ additions and subtractions introduces L functional constraints, and there are $k+2$ others, so there are $O(kL)$ new functional constraints, and they can be computed in the stated time.

We now prove (a). By the identities for additions and subtractions, we have $s^i(x) = \rho^{i-1}(x) + \alpha_i x_{u_i}$ and $\rho^i(x) = s^i(x) - K x_{q_i}$ for every $x \in X$ and every i , and summing over i gives (a), since $\rho^0(x) = 0$ and $\rho = \rho^k$.

We now prove (b). Let x^0 be an integral vector of X_0 . We extend it to a 0/1 vector $\hat{x}$ on the variables of X : each new output takes the value of its function at the values of its inputs, one functional constraint after the other, and each new variable $x_{g,v}$ takes the value 1 if v is the vector of values of the inputs of g , and 0 otherwise. Let $z_i = (\sum_{i' \leq i} \alpha_{i'} \hat{x}_{u_{i'}}) \bmod K$. We show by

induction on i that, in $\hat{x}$, ρ^i takes the binary digits of z_i and the fixings introduced for the first i terms are satisfied. This holds for $i = 0$. Let $i \geq 1$. Since $z_{i-1} < K$ and $\alpha_i < K$, we have $z_{i-1} + \alpha_i \hat{x}_{u_i} < 2K \leq 2^L$, so, as shown above for additions, s^i takes the binary digits of $z_{i-1} + \alpha_i \hat{x}_{u_i}$, and the last carry of this addition, which is fixed to 0, takes the value 0. Then, as in schoolbook subtraction, β_i takes the value 1 if and only if $z_{i-1} + \alpha_i \hat{x}_{u_i} < K$, so q_i takes the value 1 if and only if $z_{i-1} + \alpha_i \hat{x}_{u_i} \geq K$. Hence $z_{i-1} + \alpha_i \hat{x}_{u_i} - K \hat{x}_{q_i}$ lies in $\{0, \dots, K-1\}$ and is congruent to z_i modulo K , so it equals z_i . Since it is nonnegative, as shown above for subtractions, ρ^i takes its binary digits, and the last carry of this subtraction takes the value 0. For $i = k$, this shows that $\hat{x}$ satisfies the new fixings. It also satisfies the extended formulations of the new functional constraints, by construction, and the constraints of X_0 , since $x^0 \in X_0$, so $\hat{x} \in X$. By Lemma 19(b), $\hat{x}$ is the only vector of X with free variables $b(x^0)$, and so the only integral vector of X that extends x^0 . In it, $\rho = \rho^k$ takes the binary digits of z_k , which is $(\sum_i \alpha_i x_{u_i}^0) \bmod K$. $\square$

5.3 Residue encodings

We now construct the functional formulations from which we obtain the formulas $F(p, h, H, D, t)$ of Proposition 16. We encode vectors with possibly exponentially large entries by their residues. Let $p = (p_1, \dots, p_s)$, where $p_1, \dots, p_s \geq 2$ are pairwise coprime integers, let $\Pi = p_1 \cdots p_s$, and let $p_{\max} = \max_j p_j$. An integer in $\{0, \dots, \Pi - 1\}$ is determined by its residues modulo $p_1, \dots, p_s$, and we encode the residue modulo p_j by p_j free variables with a multiple-choice constraint. For each j , let $\iota_j \in \{1, \dots, \Pi - 1\}$ with $\iota_j \equiv 1 \pmod{p_j}$ and $\iota_j \equiv 0 \pmod{p_{j'}}$ for $j' \neq j$. By the Chinese remainder theorem, the integer in $\{0, \dots, \Pi - 1\}$ with residues $a_1, \dots, a_s$ modulo $p_1, \dots, p_s$ is $(\sum_j \iota_j a_j) \bmod \Pi$. We compute it in binary with Lemma 20, so that, for every vector satisfying the resulting formulation, the number written by its binary digits equals a linear function of the free variables minus Π times an integral linear function.

Let $h \geq 1$ with $2^h \leq \Pi$, let $D \geq 2$, $m \geq 1$, and $m' \geq 1$ be integers, let $H \in \{0, \dots, D-1\}^{m' \times m}$, let $M = \Lambda_D^\perp(H)$, and let $t \in \mathbb{Z}^m$. Let $d \in \{0, \dots, D-1\}^{m'}$ with $d \equiv Ht \pmod{D}$. The functional formulation $X = X(p, h, H, D, t)$ is built in three parts.

First, for $i = 1, \dots, m$ and $j = 1, \dots, s$, it has free variables $x_{i,j,a}$ for $a = 0, \dots, p_j - 1$, with the multiple-choice constraint $\sum_a x_{i,j,a} = 1$. A 0/1 vector x satisfying these constraints *encodes* the vector in $\{0, \dots, \Pi - 1\}^m$ whose i th entry has residue a modulo p_j whenever $x_{i,j,a} = 1$. For instance, for $p = (3, 4)$, if $x_{i,1,1} = x_{i,2,3} = 1$, then the i th entry of the encoded vector is 7, since $7 \equiv 1 \pmod{3}$ and $7 \equiv 3 \pmod{4}$.

Second, for $i = 1, \dots, m$ in turn, we apply Lemma 20 to the current formulation, with the variables $x_{i,j,a}$ for all j and a , the weight $\iota_j a$ for $x_{i,j,a}$, and $K = \Pi$. This gives outputs $\rho_{i,0}, \rho_{i,1}, \dots$ and an integral linear function κ_i . We fix the outputs $\rho_{i,l}$ with $l \geq h$ to 0, and we let $c_i(x) = \sum_{l < h} 2^l x_{\rho_{i,l}}$. For integral x , these fixings force the encoded vector to lie in $\{0, \dots, 2^h - 1\}^m$, and hence to equal $c(x) = (c_1(x), \dots, c_m(x))$.

Third, for $r = 1, \dots, m'$ in turn, we apply Lemma 20 to the current formulation, with the variables $\rho_{i,l}$ for $i = 1, \dots, m$ and $l < h$, the weight $H'_{r,(i,l)}$ for $\rho_{i,l}$, where $H'_{r,(i,l)} \in \{0, \dots, D-1\}$ and $H'_{r,(i,l)} \equiv 2^l H_{ri} \pmod{D}$, and $K = D$. This gives outputs $\sigma_{r,0}, \sigma_{r,1}, \dots$, which we fix to the binary digits of d_r , and an integral linear function κ'_r . For integral x , these fixings force $Hc(x) \equiv Ht \pmod{D}$, that is, $c(x) \in t + M$.

The next lemma gives the properties of $X(p, h, H, D, t)$ that we use.

Lemma 21. *Let $X = X(p, h, H, D, t)$ and $c(x) = (c_1(x), \dots, c_m(x))$, and let $V_i(x) = \sum_{j,a} \iota_j a x_{i,j,a}$ for every i .*

(a) We have $c_i(x) = V_i(x) - \Pi\kappa_i(x)$ and $0 \leq c_i(x) \leq 2^h - 1$ for every $x \in X$ and every i . Moreover, there are integral linear functions $\zeta_1, \dots, \zeta_{m'}$ of the variables of X with $(Hc(x) - d)/D = (\zeta_1(x), \dots, \zeta_{m'}(x))$ for every $x \in X$.

(b) The integral vectors of X correspond one-to-one to the vectors in $(t + M) \cap \{0, \dots, 2^h - 1\}^m$.

(c) We have $X = X(\mathcal{T})$ for a family $\mathcal{T}$ of sets of at most $\max\{9, p_{\max}\}$ literals, and X has $O(m(\sum_j p_j) \log \Pi + m'mh \log D)$ functional constraints. Moreover, $\mathcal{T}$ can be computed in time polynomial in $m, m', \sum_j p_j, \log D$, and the encoding size of t .

Proof. Let X_1 and X_2 be the functional formulations obtained after the first and the second part of the construction of X , so that X is obtained from X_2 by the third part. Each part only adds variables and constraints, so every vector of X restricts to vectors of X_1 and X_2 . Let ρ be the family of the outputs $\rho_{i,l}$ with $l < h$, indexed by the pairs (i, l) , and let $G \in \mathbb{Z}^{m \times mh}$ have entry 2^l in row i and column (i, l) , and 0 elsewhere, so that $c(x) = Gx_\rho$ and $H' \equiv HG \pmod{D}$.

(a) By Lemma 20(a), $V_i(x) = \Pi\kappa_i(x) + \sum_l 2^l x_{\rho_{i,l}}$ for every $x \in X_2$, and hence for every $x \in X$. Since the outputs $\rho_{i,l}$ with $l \geq h$ are fixed to 0, we obtain $V_i(x) = \Pi\kappa_i(x) + c_i(x)$ for every $x \in X$. Moreover, $0 \leq c_i(x) \leq 2^h - 1$, since all variables lie in $[0, 1]$. By Lemma 20(a), $H'_r x_\rho = D\kappa'_r(x) + \sum_l 2^l x_{\sigma_{r,l}}$ for every $x \in X$ and every r , where H'_r is the r th row of H' . The outputs $\sigma_{r,l}$ are fixed to the binary digits of d_r , so $H'_r x_\rho = D\kappa'_r(x) + d_r$ for every $x \in X$. Writing $HG = H' + DW$ with W integral, we obtain $H_r c(x) = H_r G x_\rho = d_r + D(\kappa'_r(x) + W_r x_\rho)$, where H_r and W_r are the r th rows of H and W . Hence, for every $x \in X$, the r th entry of $(Hc(x) - d)/D$ equals $\zeta_r(x) = \kappa'_r(x) + W_r x_\rho$, which is an integral linear function, since κ'_r is an integral linear function and W is integral.

(b) The integral vectors of X_1 are the 0/1 vectors that satisfy the multiple-choice constraints. Such a vector has exactly one $a = a_{ij}$ with $x_{i,j,a} = 1$ for all i and j , and it encodes the vector $c \in \{0, \dots, \Pi - 1\}^m$ with $c_i \equiv a_{ij} \pmod{p_j}$ for all j . By the Chinese remainder theorem, this gives a bijection from the integral vectors of X_1 to $\{0, \dots, \Pi - 1\}^m$. Since $\sum_j \iota_j a_{ij} \equiv c_i \pmod{\Pi}$, Lemma 20(b), applied for $i = 1, \dots, m$, shows that each integral vector of X_1 extends to exactly one integral vector satisfying the formulation obtained from X_1 by the applications of Lemma 20 in the second part, and in it the outputs $\rho_{i,l}$ take the binary digits of c_i . This vector satisfies the fixings of the second part if and only if $c \in \{0, \dots, 2^h - 1\}^m$. Hence the integral vectors of X_1 with c in this box extend to exactly one integral vector of X_2 each, in which the outputs $\rho_{i,l}$ with $l < h$ take the binary digits of c_i , and the others extend to none. By Lemma 20(b), applied for $r = 1, \dots, m'$, each integral vector of X_2 extends to exactly one integral vector satisfying the formulation obtained from X_2 by the applications of Lemma 20 in the third part, and in it the outputs $\sigma_{r,l}$ take the binary digits of $(H_r c) \bmod D$, since $H' \equiv HG \pmod{D}$. This vector satisfies the fixings of the third part, and so lies in X , if and only if $Hc \equiv d \equiv Ht \pmod{D}$, that is, $c \in t + M$. Since every integral vector of X restricts to integral vectors of X_1 and X_2 , the integral vectors of X correspond one-to-one to the vectors $c \in (t + M) \cap \{0, \dots, 2^h - 1\}^m$, which lie in $\{0, \dots, \Pi - 1\}^m$ since $2^h \leq \Pi$.

(c) The functional constraints have at most three inputs, so the exact-one equations of their extended formulations have at most 9 literals: $\sum_v x_{g,v} = 1$ has at most 8, and each coordinate equation has at most 8 variables $x_{g,v}$ and one more literal. The multiple-choice constraints have p_j literals. The second part applies Lemma 20 m times with $k = \sum_j p_j$ and $K = \Pi$, and the third part applies it m' times with $k = mh$ and $K = D$. Since each application adds $O(kL)$ functional constraints, where $L = O(\log K)$, this gives the stated number of functional constraints. This number is polynomial in $m, m', \sum_j p_j$, and $\log D$, since $h \leq \log_2 \Pi \leq \sum_j p_j$. The numbers ι_j can be computed by the extended Euclidean algorithm, and H' and d by reducing $2^l H_{r,i}$ and Ht modulo D . $\square$

5.4 Thin formulas

We now show that thin formulas satisfy the promise, which completes the proof of Proposition 16. We use the following description of the dual lattice of $\Lambda_D^\perp(H)$, which is stated in [31, Section 2], and we include its short proof.

Lemma 22. *Let $D \geq 1$, $m \geq 1$, and $m' \geq 1$ be integers, let $H \in \mathbb{Z}^{m' \times m}$, and let $M = \Lambda_D^\perp(H)$. Then $DM^* = H^\top \mathbb{Z}^{m'} + D\mathbb{Z}^m$.*

Proof. Let $L_0 = H^\top \mathbb{Z}^{m'} + D\mathbb{Z}^m$, and let L_0^* be its dual lattice. A vector $y \in \mathbb{R}^m$ lies in DL_0^* if and only if $v^\top y \in D\mathbb{Z}$ for every $v \in L_0$. Since this condition is preserved under integer combinations of v , it suffices to check it for the vectors De_i with $i = 1, \dots, m$ and $H^\top z$ with $z \in \mathbb{Z}^{m'}$, which generate L_0 . For the vectors De_i , this condition holds if and only if $y \in \mathbb{Z}^m$, and for the vectors $H^\top z$, it holds if and only if $Hy \in D\mathbb{Z}^{m'}$. Hence $DL_0^* = M$. Since $(DL)^* = D^{-1}L^*$ for every lattice L , we obtain $DM^* = D(DL_0^*)^* = L_0^{**} = L_0$. $\square$

The next lemma plays the role of Lemma 5: instead of short vectors of the dual lattice, we need vectors of the dual lattice close to given vectors.

Lemma 23. *Let $M \subseteq \mathbb{R}^m$ be a lattice. For every $y \in \mathbb{R}^m$, there is $w \in M^*$ with $\|w - y\|_1 \leq m^{3/2}/(2\lambda_1(M))$.*

Proof. By a transference theorem of Banaszczyk [7, Theorem 2.2], $\text{dist}(y, M^*) \leq m/(2\lambda_1(M))$. Let $w \in M^*$ with $\|w - y\| = \text{dist}(y, M^*)$. Then $\|w - y\|_1 \leq \sqrt{m}\|w - y\| \leq m^{3/2}/(2\lambda_1(M))$. $\square$

The next proposition is the main step in the proof of Proposition 16. It plays the role of Proposition 6 for the functional formulations.

Proposition 24. *Let $X = X(p, h, H, D, t)$ and $M = \Lambda_D^\perp(H)$. If $\lambda_1(M) > m^{3/2}p_{\max}(2^h - 1)/2$, then every rational subpolyhedron of X has Chvátal rank at most one.*

Proof. Let R be a rational subpolyhedron of X . We may assume $R \neq \emptyset$. We find integral linear functions along which R has width less than one, such that every free variable is the difference of two of them. Then CG cuts along these functions fix all free variables, and hence, by Lemma 19(b), leave at most one integral vector.

We obtain these functions from affine functions, defined separately for each i and j . Fix i and j . Most objects defined below depend on i and j , but until the last paragraph we do not show this in the notation. By Lemma 23, there is $w \in M^*$ with $\|w - e_i/p_j\|_1 \leq m^{3/2}/(2\lambda_1(M)) < 1/(p_j(2^h - 1))$. Let $w' = w - e_i/p_j$. For $l = 0, \dots, p_j$, let $f_l(a) = a/p_j$ for $a < l$ and $f_l(a) = a/p_j - 1$ for $a \geq l$, and let

$$\psi_l(x) = \sum_{a=0}^{p_j-1} f_l(a) x_{i,j,a} + w'^\top c(x).$$

The functions f_l have three properties: $f_l(a) - a/p_j$ is an integer, $f_{l+1}(a) - f_l(a)$ is 1 if $a = l$ and 0 otherwise, and the values of f_l lie in an interval of length less than one. For instance, for $p_j = 3$, the vectors $(f_l(0), f_l(1), f_l(2))$ for $l = 0, 1, 2, 3$ are $(-1, -\frac{2}{3}, -\frac{1}{3})$, $(0, -\frac{2}{3}, -\frac{1}{3})$, $(0, \frac{1}{3}, -\frac{1}{3})$, and $(0, \frac{1}{3}, \frac{2}{3})$. We use them, in turn, to show that, for every $x \in X$, each $\psi_l(x)$ equals an integral linear function of x plus a constant, that consecutive differences of these integral linear functions are the free variables, and that R has width less than one along them.

We first show that, for every $x \in X$, each $\psi_l(x)$ equals an integral linear function of x plus a constant that does not depend on l . By Lemma 22, $Dw = H^\top z + Dz'$ for some $z \in \mathbb{Z}^{m'}$

and $z' \in \mathbb{Z}^m$. By Lemma 21(a), for every $x \in X$, we have $w^\top c(x) = z^\top Hc(x)/D + z'^\top c(x) = z^\top d/D + z^\top \zeta(x) + z'^\top c(x)$, where $\zeta = (\zeta_1, \dots, \zeta_{m'})$, and $z^\top \zeta + z'^\top c$ is an integral linear function, since the entries of ζ and of c are. By Lemma 21(a) again, for every $x \in X$,

$$\frac{c_i(x)}{p_j} = \frac{V_i(x) - \Pi \kappa_i(x)}{p_j} = \sum_a \frac{\iota_j a}{p_j} x_{i,j,a} + \sum_{j' \neq j} \sum_a \frac{\iota_{j'} a}{p_j} x_{i,j',a} - \frac{\Pi}{p_j} \kappa_i(x).$$

We have $\iota_{j'} \equiv 0 \pmod{p_j}$ for $j' \neq j$, the integer p_j divides Π , and $\iota_j - 1$ is a multiple of p_j . Hence $c_i(x)/p_j = \sum_a (a/p_j) x_{i,j,a} + \omega(x)$ for every $x \in X$, with ω an integral linear function. Since $e_i^\top c(x) = c_i(x)$, for every $x \in X$ we obtain

$$\psi_l(x) = \sum_a (f_l(a) - a/p_j) x_{i,j,a} - \omega(x) + w^\top c(x) = \psi'_l(x) + z^\top d/D,$$

where ψ'_l is an integral linear function, since $f_l(a) - a/p_j \in \{0, -1\}$, and $z^\top d/D$ does not depend on l . Moreover, $\psi'_{l+1} - \psi'_l = x_{i,j,l}$ for $l = 0, \dots, p_j - 1$, by the second property of the functions f_l .

Next we show that R has width less than one along each function ψ'_l . For every $x \in X$, the vector $(x_{i,j,a})_a$ lies in the standard simplex, so the first term of $\psi_l(x)$ lies in $[\min_a f_l(a), \max_a f_l(a)]$. This interval has length at most $1 - 1/p_j$. Indeed, its endpoints are $l/p_j - 1$ and $(l-1)/p_j$ for $1 \leq l \leq p_j - 1$, and they are -1 and $-1/p_j$, or 0 and $(p_j - 1)/p_j$, for $l = 0$ or $l = p_j$, respectively. By Lemma 21(a), $c(x) \in [0, 2^h - 1]^m$ for every $x \in X$, so $w^\top c(x)$ varies by at most $\|w'\|_1(2^h - 1) < 1/p_j$ as x ranges over X . Hence, since $R \subseteq X$ and $\psi_l(x) - \psi'_l(x) = z^\top d/D$ for every $x \in X$, $\text{width}_R(\psi'_l) \leq \text{width}_X(\psi'_l) = \text{width}_X(\psi_l) < 1$.

Finally, we show that CG cuts along these functions fix all free variables. Let R_0 be the intersection of R with the CG cuts $\psi'_{i,j,l} \leq \left\lfloor \delta_R(\psi'_{i,j,l}) \right\rfloor$ and $\psi'_{i,j,l} \geq -\left\lfloor \delta_R(-\psi'_{i,j,l}) \right\rfloor$ for all i, j , and l . If $R_0 \neq \emptyset$, then every $\psi'_{i,j,l}$ takes the same integral value at every vector of R_0 , since the bounds $\left\lfloor \delta_R(\psi'_{i,j,l}) \right\rfloor$ and $-\left\lfloor \delta_R(-\psi'_{i,j,l}) \right\rfloor$ differ by an integer at most $\text{width}_R(\psi'_{i,j,l}) < 1$. Hence every free variable $x_{i,j,l} = \psi'_{i,j,l+1} - \psi'_{i,j,l}$ takes the same integral value at every vector of R_0 , that is, 0 or 1. By Lemma 19(b), R_0 is empty or a single integral vector, which then lies in R . Thus $R_0 \subseteq R_I$. Since also $R_I \subseteq R' \subseteq R_0$, these three sets are equal, so R has Chvátal rank at most one. $\square$

Proof of Proposition 16. Let $X = X(p, h, H, D, t + (2^{h-1} - 1)\mathbf{1})$, and let $F = F(p, h, H, D, t)$ be the formula given by Lemma 18 for X . By Lemmas 18 and 21(c), its clauses have at most three literals, and it can be computed in the stated time.

(a) By Lemma 18, the satisfying assignments of F correspond one-to-one to the integral vectors of X . By Lemma 21(b), these correspond one-to-one to the vectors $c \in (t + (2^{h-1} - 1)\mathbf{1} + M) \cap \{0, \dots, 2^h - 1\}^m$. Writing $c = t - \tilde{y} + (2^{h-1} - 1)\mathbf{1}$, these are the vectors $\tilde{y} \in M$ with $t - \tilde{y} \in \mathcal{B}_h$.

(b) By Proposition 24, every rational subpolyhedron of X has Chvátal rank at most one, and so, by Lemma 18, does every rational subpolyhedron of $P(F)$. By (a), the satisfying assignments of F correspond to the vectors $\tilde{y} \in M$ with $t - \tilde{y} \in \mathcal{B}_h$. For two such vectors $\tilde{y}$ and $\tilde{y}'$, we have $\tilde{y} - \tilde{y}' \in M$ and $\|\tilde{y} - \tilde{y}'\|_\infty \leq 2^h - 1$, since $\mathcal{B}_h$ has 2^h consecutive integers in each coordinate. Hence, since $p_{\max} \geq 2$, we have $\|\tilde{y} - \tilde{y}'\| \leq \sqrt{m}(2^h - 1) \leq m^{3/2} p_{\max}(2^h - 1)/2 < \lambda_1(M)$, and so $\tilde{y} = \tilde{y}'$.

(c) Let $\Pi = p_1 \cdots p_s$, and let $\mathcal{T}$ be the family of Lemma 21(c), so that $X = X(\mathcal{T})$. By Lemma 21(c), X has $O(m(\sum_j p_j) \log \Pi + m' m h \log D)$ functional constraints. By Lemma 20, each of them has at most three inputs and at most two outputs, so it adds at most ten variables, at most six exact-one equations of at most nine literals each, and at most two fixings. The $m \sum_j p_j$ free variables lie in multiple-choice constraints with $m \sum_j p_j$ literals in total. Hence, since $\log \Pi \geq \log 2$, the number of variables of X plus $\sum_{T \in \mathcal{T}} |T|$ is $O(m(\sum_j p_j) \log \Pi + m' m h \log D)$, and (c) follows from Lemma 18. $\square$

6 Hardness for SAT polytopes

In this section we reduce worst-case lattice problems and learning with errors to satisfiability under the promise $P(F)' = P(F)_I$, using the formulas of Proposition 16. This proves Theorems C and D, through Theorems 26 and 29, and gives their consequences for lattice problems (Corollaries 27 and 30).

6.1 Hardness from worst-case lattice problems

To solve BDD, we make the lattice integral (Lemma 25), we try error bounds that are powers of two, and we find the closest vector coordinate by coordinate (Theorem 26). Combined with reductions of Lyubashevsky and Micciancio, this gives consequences for uSVP and GapSVP (Corollary 27).

To make the lattice integral, we scale a basis and round its entries. The argument is standard, and we include it because we need explicit bounds.

Lemma 25. *There is a deterministic polynomial-time algorithm that, given a basis $B \in \mathbb{Q}^{n \times n}$ of a lattice L , a target $t \in \mathbb{Q}^n$, and a rational $\nu > \sqrt{n}$, computes a basis B' of L , a vector $v_0 \in L$, a rational $S > 0$, a nonsingular matrix $M_0 \in \mathbb{Z}^{n \times n}$, and $t' \in \mathbb{Z}^n$, of encoding size polynomial in that of (B, t, ν) , with the following properties. We have $\nu(3/2)^n \leq S\lambda_1(L) \leq \nu(3/2)^n 2^{(n+1)/2}$ and $\lambda_1(M_0\mathbb{Z}^n) \geq (1 - \sqrt{n}/\nu)S\lambda_1(L)$. Moreover, let $\rho = \text{dist}(t, L)$, let $v^* \in L$ with $\|t - v^*\| = \rho$, and let $y^* \in \mathbb{Z}^n$ with $v^* = v_0 + B'y^*$. Then $e^* = t' - M_0y^*$ is an integral vector with $\|e^*\|_\infty \leq (1 + 1/\nu)S\rho + \frac{1}{2} + \frac{1}{2}(3/2)^n$.*

Proof. We first compute B' , S , and v_0 , with the LLL algorithm with parameter $3/4$ [27] and the nearest plane algorithm [6], as presented in [30, Chapter 2]. The LLL algorithm computes a basis B' of L with Gram-Schmidt vectors $b_1^\circ, \dots, b_n^\circ$ and coefficients μ_{kl} with $|\mu_{kl}| \leq \frac{1}{2}$ for $l < k$. Its first vector b_1' satisfies $\|b_1'\| \leq 2^{(k-1)/2}\|b_k^\circ\|$ for every k . Let $\tau = \min_k \|b_k^\circ\|$. Every nonzero vector $B'y$ of L satisfies $\|B'y\| \geq |y_k| \|b_k^\circ\| \geq \tau$ for the largest k with $y_k \neq 0$, so $\tau \leq \lambda_1(L)$. Hence $\tau \leq \lambda_1(L) \leq \|b_1'\| \leq 2^{(n-1)/2}\tau$. Since τ may be irrational, we use a power of two in place of $1/\tau$. We compute the smallest integer k_0 with $4^{k_0}\tau^2 \geq 1$, which is possible since τ^2 is rational, and we let $S = \nu(3/2)^{n2^{k_0}}$. Let $S_0 = \nu(3/2)^n/\tau$. Since $2^{k_0} \geq 1/\tau > 2^{k_0-1}$, we have $S_0 \leq S < 2S_0$. By the nearest plane algorithm, we compute $v_0 \in L$ such that $t_0 = t - v_0 = \sum_k \beta_k b_k^\circ$ with $|\beta_k| \leq \frac{1}{2}$.

Next we bound the coefficients of lattice vectors in the basis B' . We need this because rounding SB' changes $SB'y$ by a vector of norm at most proportional to $\|y\|_1$. For $y \in \mathbb{R}^n$, write $B'y = \sum_k \eta_k b_k^\circ$ with $\eta_k = y_k + \sum_{l>k} \mu_{lk} y_l$. Then $|\eta_k| \leq \|B'y\|/\|b_k^\circ\|$. If $|\eta_l| \leq \xi$ for all l , the recursion $|y_k| \leq |\eta_k| + \frac{1}{2} \sum_{l>k} |y_l|$ gives, by induction from $k = n$ down, $|y_k| \leq \xi(3/2)^{n-k}$, and hence $\|y\|_1 \leq 2\xi(3/2)^n$. Applied with $\xi = \|B'y\|/\tau$, this gives $\|y\|_1 \leq 2(3/2)^n \|B'y\|/\tau$ for every y . For $y = y^*$, we have $B'y^* = t_0 - (t - v^*)$, so $\eta_k = \beta_k - (t - v^*)^\top b_k^\circ / \|b_k^\circ\|^2$ and $|\eta_k| \leq \frac{1}{2} + \rho/\tau$. Hence $\|y^*\|_1 \leq (1 + 2\rho/\tau)(3/2)^n$.

Finally, we round, and we prove the bounds of the statement. Let $M_0 = \lfloor SB' \rfloor$ and $t' = \lfloor St_0 \rfloor$, rounding entrywise, and let $E_0 = M_0 - SB'$, whose entries lie in $[-\frac{1}{2}, \frac{1}{2}]$. For $y \in \mathbb{Z}^n \setminus \{0\}$, we have $\|E_0 y\| \leq \frac{\sqrt{n}}{2} \|y\|_1 \leq \sqrt{n}(3/2)^n \|B'y\|/\tau = (\sqrt{n}/\nu)S_0 \|B'y\| \leq (\sqrt{n}/\nu)S \|B'y\|$. Hence $\|M_0 y\| \geq (1 - \sqrt{n}/\nu)S \|B'y\| \geq (1 - \sqrt{n}/\nu)S\lambda_1(L) > 0$. In particular, M_0 is nonsingular, and $\lambda_1(M_0\mathbb{Z}^n) \geq (1 - \sqrt{n}/\nu)S\lambda_1(L)$. Moreover, $e^* = t' - M_0 y^*$ is integral, since t' , M_0 , and y^* are, and $e^* = (t' - St_0) + S(t - v^*) - E_0 y^*$, so $\|e^*\|_\infty \leq \frac{1}{2} + S\rho + \frac{1}{2} \|y^*\|_1 \leq \frac{1}{2} + S\rho + \frac{1}{2}(3/2)^n + (3/2)^n \rho/\tau$, which gives the stated bound, since $(3/2)^n \rho/\tau = S_0 \rho/\nu \leq S\rho/\nu$. The bounds on $S\lambda_1(L)$ follow from $S_0 \tau = \nu(3/2)^n$, $S_0 \leq S < 2S_0$, and $\tau \leq \lambda_1(L) \leq 2^{(n-1)/2}\tau$. The algorithm runs in polynomial time, since the LLL and nearest plane algorithms do, and the encoding sizes are polynomial, since those of B' , v_0 , and τ^2 , and hence of k_0 , S , M_0 , and t' , are. $\square$

Theorem 26. $BDD_{1/(4n^{5/2})}$ can be solved in deterministic polynomial time with an oracle for satisfiability under the promise that the query is a thin formula $F(p, h, H, D, t)$. Every query is a formula $F(p, h, H, D, t)$ of Proposition 16.

Proof. Let $B \in \mathbb{Q}^{n \times n}$ be a basis of a lattice L and let $t \in \mathbb{Q}^n$, with total encoding size σ . If $n < 150$, we compute a vector of L closest to t with Kannan's algorithm [26], in time polynomial in σ , since n is bounded by a constant. So let $n \geq 150$.

We use the following parameters. Let $h_{\max} = h_{\max}(n)$ be the smallest integer with $2^{h_{\max}} \geq 8n^4(3/2)^{n2^{(n+1)/2}} + 4(3/2)^n + 8$. Then $h_{\max} \leq n \log_2(3/\sqrt{2}) + 4 \log_2 n + 6$, where $\log_2(3/\sqrt{2}) < 1.09$, and hence $3h_{\max} \leq 4n$, since $n \geq 150$. It follows that, since $2h_{\max}/n^{5/2} \leq \frac{8}{3}n^{-3/2}$,

$$\frac{2}{3}(1 + n^{-4}) + 2h_{\max}/n^{5/2} \leq 1 - n^{-7/2}. \quad (2)$$

We have $h_{\max} \geq 7$. For each prime at most $h_{\max}$, we take the largest power of this prime that is at most $h_{\max}$, and we let p be the vector of these prime powers. For instance, for $h_{\max} = 10$, they would be $8 = 2^3$, $9 = 3^2$, 5 , and 7 . These moduli are pairwise coprime, and their product Π is $\text{lcm}(1, \dots, h_{\max}) \geq 2^{h_{\max}}$, by Nair's inequality [33]. Moreover, there are at most $h_{\max}$ moduli, each at most $h_{\max}$, so $p_{\max} \leq h_{\max}$ and $\sum_j p_j \leq h_{\max}^2$.

Let $\alpha_0 = 1/(3n^{3/2}h_{\max})$. Since $3h_{\max} \leq 4n$, we have $1/(4n^{5/2}) \leq \alpha_0$, so it suffices to solve BDD_{α_0} . As in the proof of Theorem 7, we try the error bounds 2^{h-1} for $h = 1, \dots, h_{\max}$, in increasing order. The first formula reported satisfiable has a unique satisfying assignment, which determines the closest vector, and we find this vector bit by bit, with queries for thin formulas defined by sublattices.

We first make the lattice integral and construct the formulas. Apply Lemma 25 with $\nu = n^4$, and let $M = M_0\mathbb{Z}^n$ and $D = |\det M_0|$. Then $\lambda_1(M) \geq (1 - \sqrt{n}/\nu)\nu(3/2)^n > 1$, so $M \neq \mathbb{Z}^n$ and $D \geq 2$. Let $H \in \{0, \dots, D-1\}^{n \times n}$ with $H \equiv \text{adj}(M_0) \pmod{D}$. Then $M = \Lambda_D^\perp(H)$, since $M_0^{-1} = \text{adj}(M_0)/\det M_0$ and $|\det M_0| = D$, so $y \in M_0\mathbb{Z}^n$ if and only if $\text{adj}(M_0)y \equiv \mathbf{0} \pmod{D}$. For $h = 1, \dots, h_{\max}$, let $F_h = F(p, h, H, D, t')$, with $m = m' = n$, which is possible by Proposition 16, since the product Π of the moduli satisfies $\Pi \geq 2^{h_{\max}} \geq 2^h$. By Proposition 16(a), the satisfying assignments of F_h correspond to the vectors $\tilde{y} \in M$ with $t' - \tilde{y} \in \mathcal{B}_h$.

Next we describe the queries and bound the running time. We ask the oracle whether $F_1, F_2, \dots, F_{h_{\max}}$ are satisfiable, until it reports that some F_h is. If it reports none, we output v_0 . Otherwise, for $i = 1, \dots, n$, we compute $I_i = \{y_i : y \in \mathbb{R}^n, t' - M_0y \in [-(2^{h-1}-1), 2^{h-1}]^n\}$ by linear programming, we let $c_i = \lceil \min I_i \rceil$, and we let K_i be the smallest nonnegative integer with $2^{K_i} > \max I_i - c_i$. For $k = 1, \dots, K_i$, let $S_{i,k}$ be the diagonal matrix with i th diagonal entry 2^k and all other diagonal entries 1, let $D_{i,k} = 2^k D$, and let $H_{i,k} \in \{0, \dots, D_{i,k}-1\}^{n \times n}$ with $H_{i,k} \equiv \text{adj}(M_0 S_{i,k}) \pmod{D_{i,k}}$, so that $M_0 S_{i,k} \mathbb{Z}^n = \Lambda_{D_{i,k}}^\perp(H_{i,k})$, as for M , since $|\det(M_0 S_{i,k})| = D_{i,k}$. Starting from $u_{i,0} = 0$, for $k = 1, \dots, K_i$ in turn, we ask the oracle whether $F(p, h, H_{i,k}, D_{i,k}, t' - (c_i + u_{i,k-1})M_0 e_i)$ is satisfiable, and we let $u_{i,k} = u_{i,k-1}$ if it reports that it is and $u_{i,k} = u_{i,k-1} + 2^{k-1}$ otherwise. We let $z_i = c_i + u_{i,K_i}$, and we output $v_0 + B'z$. Every query is a formula of Proposition 16, namely some F_h or some $F(p, h, H_{i,k}, D_{i,k}, t' - (c_i + u_{i,k-1})M_0 e_i)$. By Lemma 25, $\log D$ and the encoding sizes of M_0 and t' are polynomial in σ . The endpoints of I_i are optimal values of linear programs, so they have encoding size polynomial in σ [41, Corollary 10.2a], and so K_i and $\log D_{i,k} = \log D + k$ are polynomial in σ . Since also $\sum_j p_j \leq h_{\max}^2$, every query can be computed in time polynomial in σ , by Proposition 16. Hence, on every input, the oracle is called at most $h_{\max} + \sum_i K_i$ times, on queries of encoding size polynomial in σ , and all other computations take time polynomial in σ .

We now show that the output is v^* if the input satisfies the promise $\rho \leq \alpha_0 \lambda_1(L)$, where ρ , v^* , y^* , and e^* are as in Lemma 25. Let h^* be the smallest $h \geq 1$ with $2^{h-1} > \|e^*\|_\infty$. Then

$\|e^*\|_\infty \leq 2^{h^*-1} - 1$, since e^* is integral, and so $e^* \in \mathcal{B}_{h^*}$. If $h^* = 1$, then $e^* = \mathbf{0}$. If $h^* \geq 2$, then $2^{h^*-2} \leq \|e^*\|_\infty$, and so $2^{h^*} - 1 < 4\|e^*\|_\infty$.

We start by showing that $h^* \leq h_{\max}$. By Lemma 25, $\|e^*\|_\infty \leq 2S\rho + 1 + (3/2)^n$, and, by the promise, $S\rho < S\lambda_1(L) \leq \nu(3/2)^n 2^{(n+1)/2}$. Hence $2^{h^*} \leq 4\|e^*\|_\infty + 2 < 8n^4(3/2)^n 2^{(n+1)/2} + 4(3/2)^n + 6 < 2^{h_{\max}}$.

Next we show that all queries made for F_h with $h \leq h^*$ satisfy the promise. We claim that F_{h^*} is thin, that is, $\lambda_1(M) > n^{3/2}p_{\max}(2^{h^*} - 1)/2$. By Lemma 25, $\lambda_1(M) \geq (1 - \sqrt{n}/\nu)S\lambda_1(L)$ and $S\lambda_1(L) \geq \nu(3/2)^n$. If $h^* = 1$, the claim holds since $\lambda_1(M) \geq \frac{1}{2}n^4(3/2)^n > \frac{1}{2}n^{3/2}h_{\max} \geq \frac{1}{2}n^{3/2}p_{\max}$. If $h^* \geq 2$, then

$$\begin{aligned} \frac{n^{3/2}p_{\max}(2^{h^*} - 1)}{2} &< 2n^{3/2}h_{\max}\|e^*\|_\infty \leq 2n^{3/2}h_{\max}\left((1 + \frac{1}{\nu})S\rho + \frac{1}{2} + \frac{1}{2}(3/2)^n\right) \\ &\leq \left(\frac{2}{3}(1 + \frac{1}{\nu}) + \frac{2n^{3/2}h_{\max}}{\nu}\right)S\lambda_1(L), \end{aligned}$$

using $S\rho \leq S\lambda_1(L)/(3n^{3/2}h_{\max})$ and $1 + (3/2)^n \leq 2(3/2)^n \leq \frac{2}{\nu}S\lambda_1(L)$. Since $\nu = n^4$, the right-hand side is at most $(1 - \sqrt{n}/\nu)S\lambda_1(L) \leq \lambda_1(M)$, by (2). Hence, for every $h \leq h^*$, the formula F_h is thin, since $\lambda_1(M) > n^{3/2}p_{\max}(2^h - 1)/2$. So are the formulas queried after F_h , which have the same p and h , since their lattices $M_0S_{i,k}\mathbb{Z}^n$ are contained in M , and so $\lambda_1(M_0S_{i,k}\mathbb{Z}^n) \geq \lambda_1(M)$. Hence the oracle answers all queries made for F_h correctly.

Finally, we show that the procedure outputs v^* . By Proposition 16(a), the satisfying assignments of F_{h^*} correspond to the vectors $\tilde{y} \in M$ with $t' - \tilde{y} \in \mathcal{B}_{h^*}$. One of them is M_0y^* , since $t' - M_0y^* = e^* \in \mathcal{B}_{h^*}$, and so, by Proposition 16(b), it is the only one. For $h \leq h^*$, we have $\mathcal{B}_h \subseteq \mathcal{B}_{h^*}$, so the satisfying assignments of F_h correspond to vectors among these, that is, to M_0y^* at most. Thus, for $h \leq h^*$, the formula F_h has at most one satisfying assignment, and F_{h^*} has one. Hence the procedure stops at some $h \leq h^*$, and M_0y^* is the only vector $\tilde{y} \in M$ with $t' - \tilde{y} \in \mathcal{B}_h$. Fix i and k , and let $u = u_{i,k-1}$. By Proposition 16(a), the satisfying assignments of $F(p, h, H_{i,k}, D_{i,k}, t' - (c_i + u)M_0e_i)$ correspond to the vectors $M_0S_{i,k}x$ with $x \in \mathbb{Z}^n$ and $t' - (c_i + u)M_0e_i - M_0S_{i,k}x \in \mathcal{B}_h$. Since $M_0S_{i,k}x + (c_i + u)M_0e_i = M_0y$ for $y_i = 2^k x_i + c_i + u$ and $y_l = x_l$ for $l \neq i$, these correspond to the vectors M_0y with $y \in \mathbb{Z}^n$, $y_i \equiv c_i + u \pmod{2^k}$, and $t' - M_0y \in \mathcal{B}_h$. So this formula is satisfiable if and only if $y_i^* \equiv c_i + u \pmod{2^k}$. Since $t' - M_0y^* \in \mathcal{B}_h$, y_i^* is an integer in I_i , so $0 \leq y_i^* - c_i < 2^{K_i}$, and induction on k gives $u_{i,k} = (y_i^* - c_i) \bmod 2^k$, so $z_i = y_i^*$. Hence $z = y^*$, and the output is $v_0 + B'y^* = v^*$. $\square$

Proof of Theorem C. By Proposition 16, every formula $F(p, h, H, D, t)$ has clauses with at most three literals, and, if it is thin, it satisfies $P(F)' = P(F)_I$. By Theorem 26, every query is such a formula. Hence Theorem 26 gives Theorem C. $\square$

Replacing the oracle by an algorithm and combining Theorem 26 with Theorem 8, we obtain the further consequences stated in the introduction.

Corollary 27. *Suppose that an algorithm solves satisfiability in polynomial time for formulas F whose clauses have at most three literals, under the promise that every rational subpolyhedron of $P(F)$, and in particular $P(F)$ itself, has Chvátal rank at most one. Then $BDD_{1/(4n^{5/2})}$ and $uSVP_{4n^{5/2}}$ can be solved in deterministic polynomial time, and, for $n \geq 2$, GapSVP_γ with $\gamma = 4n^3/\sqrt{\log n}$ can be solved in randomized polynomial time. If the algorithm is randomized, then these statements hold with randomized algorithms for the three problems.*

Proof. By Proposition 16(b), every thin formula satisfies this promise, so, as in the proof of Theorem C, the algorithm can serve as the oracle in Theorem 26. The statements then follow as in

the proof of Corollary 9, using Theorem 8 with $\gamma = 4n^{5/2}$ and $\gamma = 4n^3/\sqrt{\log n}$, for which the BDD factor is $1/(4n^{5/2})$. $\square$

The factor $1/(4n^{5/2})$ is worse than the factor $1/(4n^2)$ of Theorem 7 by a factor $\sqrt{n}$. The loss comes from the moduli: the errors after Lemma 25 can be as large as $2^{O(n)}$, so the largest modulus must be at least of order n , which costs a factor of order n in the condition of Proposition 16(b), partly offset by a factor of order $\sqrt{n}$ gained in Lemma 23, which, unlike Lemma 5, does not need a basis of the dual lattice. Moreover, unlike that of Theorem 7, the reduction of Theorem 26 does not preserve the dimension: the number of variables of its formulas depends on the encoding size of the input, through the determinant of the integral lattice given by Lemma 25, so Corollary 27 has no analog of the statement of Corollary 9 on running times $2^{o(n)}$.

6.2 Hardness from learning with errors

To solve decision-LWE, we use a single formula defined by the q -ary lattice $\Lambda_q(A^\top)$. This lattice is integral and the errors are polynomially bounded, so we apply Proposition 16 directly, with moduli of logarithmic size. With high probability, this formula is thin, and so satisfies the promise, and it is satisfiable for LWE instances but not for uniform instances. Thus a single call to an algorithm for satisfiability distinguishes the two cases (Theorem 29). Combined with a result of Regev, this gives consequences for GapSVP and SIVP for quantum algorithms (Corollary 30). In this subsection, n is the dimension of s , and also of the lattice problems in Theorem 13, while the lattice $\Lambda_q(A^\top)$ has dimension m .

To apply Proposition 16, we write $\Lambda_q(A^\top)$ in the form $\Lambda_q^\perp(H)$.

Lemma 28. *Let q be a prime, let $n < m$, and let $A \in \mathbb{Z}_q^{n \times m}$. In time polynomial in m and $\log q$, we can compute a matrix $H \in \{0, \dots, q-1\}^{m' \times m}$ with $1 \leq m' \leq m$ and $\Lambda_q^\perp(H) = \Lambda_q(A^\top)$.*

Proof. Since q is prime, we can compute by Gaussian elimination over $\mathbb{Z}_q$ a matrix $H \in \{0, \dots, q-1\}^{m' \times m}$ whose rows form a basis of the kernel of A over $\mathbb{Z}_q$. This kernel has dimension at least $m - n \geq 1$, so $1 \leq m' \leq m$. Over the field $\mathbb{Z}_q$, the orthogonal complement of this kernel is the row space of A . Hence, for $y \in \mathbb{Z}_q^m$, we have $Hy \equiv \mathbf{0} \pmod{q}$ if and only if $y \equiv A^\top s \pmod{q}$ for some $s \in \mathbb{Z}_q^n$, that is, $\Lambda_q^\perp(H) = \Lambda_q(A^\top)$. $\square$

For the error distribution $\bar{\Psi}_\alpha$ and suitable parameters, the formulas of Proposition 16 give an explicit distribution of hard instances.

Theorem 29. *Let n be sufficiently large, let $q = q(n)$ be a prime in $[n^3, 2n^3]$, which exists by Bertrand's postulate, let $k = \lceil \log_2 q \rceil$, let $\alpha = 1/(200n^{3/2}k^{5/2} \log n)$, and let $m = 3nk$. Let h be the largest integer with $h2^h \leq q/(4m^{3/2})$. For each prime at most h , take the largest power of this prime that is at most h , and let p be the vector of these prime powers. For $A \in \mathbb{Z}_q^{n \times m}$ and $t \in \mathbb{Z}_q^m$, let $F(A, t) = F(p, h, H, q, t)$, where H is the matrix given by Lemma 28 for A , and let $\mathcal{F}$ be the distribution of $F(A, t)$ when (A, t) is an LWE instance with $\chi = \bar{\Psi}_\alpha$ or a uniform instance, each with probability $1/2$. The distribution $\mathcal{F}$ can be sampled in polynomial time, its samples are formulas with $O(n^2 \log^4 n)$ variables whose clauses have at most three literals, and a sample of $\mathcal{F}$ is thin with probability at least $1 - 2^{-2nk}$. Suppose that a randomized polynomial-time algorithm decides satisfiability correctly with probability at least $1/2 + n^{-c}$ for some constant $c > 0$, over the choice of F from $\mathcal{F}$ and its own randomness. Then there is a randomized polynomial-time algorithm that solves decision-LWE $_{n,q,\bar{\Psi}_\alpha}$ with m samples.*

Proof. Let $M = \Lambda_q(A^\top)$, so that $M = \Lambda_q^\perp(H)$ by Lemma 28.

We first show that $F(A, t)$, and hence a sample of $\mathcal{F}$, can be computed in time polynomial in n . Since n is sufficiently large, we have $q/(4m^{3/2}) \geq n^3/(12\sqrt{3}n^{3/2}k^{3/2}) \geq 7 \cdot 2^7$, so $h \geq 7$, and $h \leq \log_2 q \leq k$. The product Π of the moduli is $\text{lcm}(1, \dots, h) \geq 2^h$, by Nair's inequality [33]. Moreover, there are at most h moduli, each at most h , so $p_{\max} \leq h$, $\sum_j p_j \leq h^2$, and $\Pi \leq h^h$. Hence, by Proposition 16, $F(A, t)$ is a formula whose clauses have at most three literals, and it can be computed in time polynomial in n , and so can a sample of $\mathcal{F}$, since q can be found in time polynomial in n by testing the integers in $[n^3, 2n^3]$ for primality, and Ψ_α can be sampled in polynomial time.

The formulas $F(A, t)$ have $O(n^2 \log^4 n)$ variables. Indeed, by Proposition 16(c), since $h \leq k$, $\sum_j p_j \leq h^2$, $\log_2 \Pi \leq h \log_2 h$, $m' \leq m$, and $D = q$, they have $O(mh^3 \log h + m^2 h \log q) = O(n^2 k^4)$ variables.

Next we show that $F(A, t)$ is thin with high probability. In LWE instances and in uniform instances alike, A is uniformly random, so $\lambda_1(M) > q/8$ with probability at least $1 - q^n 2^{-m} \geq 1 - 2^{-2nk}$, by Lemma 10 and $q^n \leq 2^{nk} = 2^{m/3}$. If $\lambda_1(M) > q/8$, then, since $p_{\max} \leq h$ and $h2^h \leq q/(4m^{3/2})$,

$$\frac{m^{3/2} p_{\max} (2^h - 1)}{2} < \frac{m^{3/2} h 2^h}{2} \leq \frac{q}{8} < \lambda_1(M),$$

and so $F(A, t)$ is thin.

We then show that, for an LWE instance, $F(A, t)$ is satisfiable with probability at least $1 - 2m \exp(-\pi \log^2 n)$. By Proposition 16(a), the satisfying assignments of $F(A, t)$ correspond to the vectors $\tilde{y} \in M$ with $t - \tilde{y} \in \mathcal{B}_h$. Let (A, t) be an LWE instance with error e , and let ε_χ be the probability that $\|e\|_\infty > 2^{h-1} - 1$. Then $t - e \in M$, so $F(A, t)$ is satisfiable if $\|e\|_\infty \leq 2^{h-1} - 1$, and hence with probability at least $1 - \varepsilon_\chi$. By the choice of h , we have $(h+1)2^{h+1} > q/(4m^{3/2})$, and so, since $h+1 \leq 2k$ and $16m^{3/2} = 48\sqrt{3}n^{3/2}k^{3/2}$,

$$2^{h-1} > \frac{q}{16m^{3/2}(h+1)} \geq \frac{q}{96\sqrt{3}n^{3/2}k^{5/2}}.$$

Since $q \geq n^3$ and n is sufficiently large, it follows that $2^{h-1} - \frac{1}{2} \geq q/(200n^{3/2}k^{5/2}) = \alpha q \log n$. As in the proof of Theorem 12, each entry of e has absolute value greater than $2^{h-1} - 1$ with probability at most $\Pr[|X| \geq \alpha q \log n] \leq 2 \exp(-\pi \log^2 n)$, where X is normal with mean zero and standard deviation $\alpha q / \sqrt{2\pi}$. Hence, by the union bound over the m entries, $\varepsilon_\chi \leq 2m \exp(-\pi \log^2 n)$.

We also show that, for a uniform instance, $F(A, t)$ is satisfiable with probability at most 2^{-m} . Let (A, t) be a uniform instance. If $F(A, t)$ is satisfiable, then $t \equiv A^\top s + e' \pmod{q}$ for some $s \in \mathbb{Z}_q^n$ and $e' \in \mathcal{B}_h$. There are at most $q^n 2^{hm}$ such residues, so $F(A, t)$ is satisfiable with probability at most $q^n (2^h/q)^m$. Since $2^h \leq q/(4m^{3/2}h) \leq q/4$ and $q^n \leq 2^{m/3}$, this probability is at most $2^{m/3} 4^{-m} \leq 2^{-m}$.

It remains to construct the algorithm for decision-LWE. Given (A, t) , our algorithm computes $F(A, t)$, runs the given algorithm on it, and outputs 1 if and only if the given algorithm reports that $F(A, t)$ is satisfiable. As in the proof of Theorem 12, the probabilities that it outputs 1 on LWE instances and on uniform instances differ by at least $2n^{-c} - \varepsilon_\chi - 2^{-m}$, which is at least n^{-c} for sufficiently large n . $\square$

Proof of Theorem D. By Proposition 16(b), the SAT polytopes of thin formulas satisfy $P(F)' = P(F)_I$. So the SAT polytope of a sample of $\mathcal{F}$ satisfies $P(F)' = P(F)_I$ with high probability. Moreover, $2/3 \geq 1/2 + 1/n$ for $n \geq 6$. Hence Theorem 29 gives Theorem D. $\square$

Combining Theorem 29 with Theorem 13, we obtain the consequence for quantum algorithms stated in the introduction.

Corollary 30. *Suppose that there is a randomized polynomial-time algorithm as in Theorem 29. Then, for some $\gamma = \tilde{O}(n^{5/2})$, there is a polynomial-time quantum algorithm that solves GapSVP_γ and SIVP_γ on n -dimensional lattices.*

Proof. Let q, k, α , and m be as in Theorem 29. By Theorem 29, there is a randomized polynomial-time algorithm that solves decision-LWE $_{n,q,\tilde{\Psi}_\alpha}$ with m samples. Theorem 13 applies, since $q \leq 2n^3$ is prime, q and $m = 3nk$ are bounded by a polynomial in n , and $\alpha q \geq n^3/(200n^{3/2}k^{5/2} \log n) > 2\sqrt{n}$ for sufficiently large n , and it gives $\gamma = \tilde{O}(n/\alpha) = \tilde{O}(n^{5/2})$. $\square$

6.3 Refutations and set partitioning polytopes

We conclude with two remarks, on refutations and on set partitioning polytopes.

Remark 31. As with thin parallelepipeds in Remark 15, the SAT polytopes of thin formulas $F = F(p, h, H, D, t)$ and their rational subpolyhedra that contain no integral vector have empty Chvátal closure, by Proposition 16(b), and so, by Helly's theorem, refutations with at most $N + 1$ CG cuts, where N is the number of variables of F . The vectors w of the proof of Proposition 24 play the role of the short dual basis, and both these vectors and refutations are hard to find.

We now show that the vectors w and refutations are hard to find. Given the vectors w , we compute z and z' with $Dw = H^T z + Dz'$, as in the proof of Proposition 24, by solving a system of linear Diophantine equations [41, Corollary 5.3b], and then the CG cuts of that proof by linear programming. As in that proof, these cuts decide whether the functional formulation from which F is obtained has an integral vector. By Lemma 18, this decides satisfiability of F . So, by Theorems 26 and 29, finding the vectors w is as hard as $\text{BDD}_{1/(4n^{5/2})}$ and, since the formulas of Theorem 29 are thin with high probability, as hard as decision-LWE. The same holds for finding any refutation, since, as in Remark 15, an algorithm that finds a refutation whenever one exists decides satisfiability for thin formulas.

Remark 32. Theorems C and D also hold for integer feasibility on every class of polytopes that contains the SAT polytopes of formulas whose clauses have at most three literals, for instance the polytopes in $[0, 1]^N$ defined by inequalities with coefficients in $\{0, \pm 1\}$, integral right-hand sides, and at most three nonzero coefficients each. They also hold for *set partitioning polytopes* $\{x \geq \mathbf{0} : Wx = \mathbf{1}\}$, where W is a 0/1 matrix, with at most three ones in each row, although this class does not contain these SAT polytopes. Without the promise, integer feasibility on these set partitioning polytopes is NP-complete, even with exactly three ones in each row [39].

We now prove the statement for set partitioning polytopes. Let F be one of the formulas of Proposition 16, and let $X(\mathcal{T}_3)$ be the face of $P(F)$ from the proof of Lemma 18. Here $\mathcal{T}_3$ is a family of sets of at most three literals, and $X(\mathcal{T}_3)$ contains all integral vectors of $P(F)$. Adding a variable $\bar{x}_i$ and the equation $x_i + \bar{x}_i = 1$ for every i , and writing every literal $\bar{x}_i$ as this variable, transforms $X(\mathcal{T}_3)$ into a set partitioning polytope $\{x \geq \mathbf{0} : Wx = \mathbf{1}\}$ whose matrix W has at most three ones in each row. This transformation is a map as in Lemma 17, so the integral vectors correspond one-to-one. Every rational subpolyhedron of $X(\mathcal{T}_3)$ is one of $P(F)$, so, by Lemma 17, every rational subpolyhedron of the set partitioning polytope has Chvátal rank at most one whenever every rational subpolyhedron of $P(F)$ does. Every query of Theorem 26 and every formula $F(A, t)$ of Theorem 29 is a formula of Proposition 16, so this applies to all of them.

Funding. A. Del Pia is partially supported by AFOSR Grant FA9550-23-1-0433 and ONR Grant N00014-25-1-2490. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Air Force Office of Scientific Research or the Office of Naval Research.

AI declaration. AI tools (Claude, developed by Anthropic) were used for proofreading and suggesting improvements to the exposition. The author takes full responsibility for the content of the paper.

References

- [1] Karen Aardal, Cor A. J. Hurkens, and Arjen K. Lenstra. Solving a system of linear Diophantine equations with lower and upper bounds on the variables. *Mathematics of Operations Research*, 25(3):427–442, 2000.
- [2] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2^n time via discrete Gaussian sampling. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 733–742, 2015.
- [3] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004.
- [4] Dorit Aharonov and Oded Regev. Lattice problems in $\text{NP} \cap \text{coNP}$. *Journal of the ACM*, 52(5):749–765, 2005.
- [5] Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified Boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.
- [6] László Babai. On Lovász’ lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
- [7] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993.
- [8] Sylvia Boyd and William R. Pulleyblank. Facet generating techniques. In William Cook, László Lovász, and Jens Vygen, editors, *Research Trends in Combinatorial Optimization*, pages 33–55. Springer, 2009.
- [9] Vašek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973.
- [10] Vašek Chvátal, William J. Cook, and Mark E. Hartmann. On cutting-plane proofs in combinatorial optimization. *Linear Algebra and its Applications*, 114–115:455–499, 1989.
- [11] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 1971.
- [12] William Cook, Collette R. Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987.
- [13] Gérard Cornuéjols and Dabeen Lee. On some polytopes contained in the 0,1 hypercube that have a small Chvátal rank. *Mathematical Programming*, 172:467–503, 2018.
- [14] Gérard Cornuéjols, Dabeen Lee, and Yanjun Li. On the rational polytopes with Chvátal rank 1. *Mathematical Programming*, 179(1–2):21–46, 2020.
- [15] Gérard Cornuéjols and Yanjun Li. Deciding emptiness of the Gomory–Chvátal closure is NP-complete, even for a rational polyhedron containing no integer point. In *Integer Programming and Combinatorial Optimization (IPCO 2016)*, volume 9682 of *Lecture Notes in Computer Science*, pages 387–397. Springer, 2016.
- [16] Gérard Cornuéjols and Yanjun Li. When the Gomory–Chvátal closure coincides with the integer hull. *Operations Research Letters*, 46(2):251–256, 2018.
- [17] Alberto Del Pia and Silvia Di Gregorio. Chvátal rank in binary polynomial optimization. *INFORMS Journal on Optimization*, 3(4):315–349, 2021.
- [18] Alberto Del Pia, Antoine Musitelli, and Giacomo Zambelli. On matrices with the Edmonds–Johnson property arising from bidirected graphs. *Journal of Combinatorial Theory, Series B*, 130:49–91, 2018.

- [19] Friedrich Eisenbrand. On the membership problem for the elementary closure of a polyhedron. *Combinatorica*, 19(2):297–300, 1999.
- [20] A. M. H. Gerards and Alexander Schrijver. Matrices with the Edmonds–Johnson property. *Combinatorica*, 6(4):365–379, 1986.
- [21] Oded Goldreich. On promise problems: a survey. In Oded Goldreich, Arnold L. Rosenberg, and Alan L. Selman, editors, *Theoretical Computer Science: Essays in Memory of Shimon Even*, volume 3895 of *Lecture Notes in Computer Science*, pages 254–290. Springer, 2006.
- [22] Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958.
- [23] Mika Göös, Sajin Korothe, Ian Mertz, and Toniann Pitassi. Automating cutting planes is NP-hard. In *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 68–77, 2020.
- [24] Joachim Grollmann and Alan L. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988.
- [25] Robert G. Jeroslow. Trivial integer programs unsolvable by branch-and-bound. *Mathematical Programming*, 6(1):105–109, 1974.
- [26] Ravi Kannan. Minkowski’s convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.
- [27] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
- [28] Hendrik W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
- [29] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2009.
- [30] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 2002.
- [31] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer, 2009.
- [32] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. *SIAM Journal on Computing*, 42(3):1364–1391, 2013.
- [33] M. Nair. On Chebyshev-type inequalities for primes. *The American Mathematical Monthly*, 89(2):126–129, 1982.
- [34] Christos H. Papadimitriou and Mihalis Yannakakis. On recognizing integer polyhedra. *Combinatorica*, 10(1):107–109, 1990.
- [35] Gábor Pataki, Mustafa Tural, and Erick B. Wong. Basis reduction and the complexity of branch-and-bound. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1254–1261, 2010.
- [36] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science*, 10(4):283–424, 2016.
- [37] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34:1–34:40, 2009.
- [38] Victor Reis and Thomas Rothvoss. The subspace flatness conjecture and faster integer programming. In *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 974–988, 2023.

- [39] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.
- [40] Alexander Schrijver. On cutting planes. *Annals of Discrete Mathematics*, 9:291–296, 1980.
- [41] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.
- [42] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.