Bounded Integer Quadratic Programming via Symmetric Displacement Covers

Cinar Ari   Robert Hildebrand*

Grado Department of Industrial and Systems Engineering
Virginia Tech, Blacksburg, Virginia, USA

Abstract

We give an exact algorithm for minimizing an arbitrary rational quadratic polynomial $\boldsymbol{x}^\top Q\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} + \gamma$ over the integer points of a bounded rational polyhedron $A\boldsymbol{x} \leq \boldsymbol{b}$. For n variables and m inequalities, the running time is

$$2^{O(n\log(n+1))}(m+1)^{O(n)}\varphi_{A,Q}^{O(n)}(1+\varphi)^{O(1)},$$

where $\varphi_{A,Q}$ is one plus the maximum binary encoding length of an entry of A or Q, and φ is the full input encoding length. The polynomial degree in φ is absolute; the encoding of $\boldsymbol{b}, \boldsymbol{c}, \gamma$ enters only this fixed-degree factor. We construct a cover of the feasible integer points by cells with symmetric displacement sets that contain all cell differences and permit feasible moves in both directions. A negative integer displacement rules out the associated cell. Otherwise, the quadratic identity supplies supporting inequalities on integer points, allowing exact cell optimization by an integer-query convex feasibility algorithm. A refinement through scaled lattices solves the negative-displacement search in $2^{O(n\log(n+1))}$ times a fixed-degree polynomial in its input length. Boundary and directional-gradient localization, followed by integer sensitivity, removes the right-hand sides from the dimension-dependent encoding factor. Combining the bounded algorithm with a separate unboundedness test extends the method to arbitrary rational polyhedra.

Keywords: integer quadratic programming; fixed dimension; parameterized complexity; polyhedral covers; lattice algorithms.

Mathematics Subject Classification (2020): 90C10, 90C20, 68Q25.

1 Introduction

Let

$$P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}, \qquad f(\boldsymbol{x}) = \boldsymbol{x}^\top Q\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} + \gamma, \qquad q(\boldsymbol{d}) = \boldsymbol{d}^\top Q\boldsymbol{d},$$

Vectors are written in bold; their scalar coordinates and scalar indices are written in ordinary italics. Vectors are columns unless explicitly identified as matrix rows. Matrices retain ordinary uppercase notation. Here A has m rows, all input data are rational, $Q = Q^\top$, and P is bounded. We seek a point minimizing f on $P \cap \mathbb{Z}^n$, or determine that this set is empty. We take $n \geq 1$; in dimension zero, test the unique point directly. The gradient is $\nabla f(\boldsymbol{x}) = 2Q\boldsymbol{x} + \boldsymbol{c}$, and the quadratic identity is

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top(\boldsymbol{y} - \boldsymbol{x}) + q(\boldsymbol{y} - \boldsymbol{x}).$$

*Corresponding author: rhil@vt.edu.

The algorithm. We first cover the feasible integer points by integral parallelepipeds contained in P. In each parallelepiped, we construct smaller sets called cells that cover its integer points. For each cell, we construct a displacement set containing every difference of two points in the cell. Each displacement in this set can be added to or subtracted from every point of the cell while staying in P. If an integer displacement $\boldsymbol{d}$ in the set has $q(\boldsymbol{d}) < 0$, then for every integer point $\boldsymbol{x}$ in the cell, at least one of $\boldsymbol{x} + \boldsymbol{d}$ and $\boldsymbol{x} - \boldsymbol{d}$ has smaller objective value than $\boldsymbol{x}$. We can therefore discard that cell.

On each remaining cell, q is nonnegative on differences of integer points. The quadratic identity then gives a linear inequality separating an integer point of the cell from its integer points with smaller objective value. We use these inequalities with the integer-query convex feasibility theorem of Hildebrand and Göß [14] to find the cell minimum. Comparing the minima from all remaining nonempty cells gives a global minimizer.

This procedure already solves the bounded problem with polynomial running time in every fixed dimension. We then improve its encoding dependence. The localization argument in Section 5 replaces the original problem by bounded translated problems whose constraint encoding is controlled by A, Q. Applying the same procedure to these problems places the encoding of $\boldsymbol{b}, \boldsymbol{c}, \gamma$ entirely in a fixed-degree polynomial factor.

Encoding and running time. For a rational number $r = p/q$ in lowest terms with $q > 0$, define its binary encoding length by

$$\langle r \rangle = 1 + \lceil \log_2(1 + |p|) \rceil + \lceil \log_2(1 + q) \rceil.$$

For a list of matrices, vectors, or scalars, $\langle \cdot \rangle$ denotes its total binary encoding length, including dimensions. The notation $\langle \cdot \rangle_{\max}$ denotes the maximum encoding length of an individual scalar entry, taking one for an empty collection. Thus, for nonempty A,

$$\langle A \rangle_{\max} = \max_{i,j} \langle A_{ij} \rangle, \qquad \varphi_{A,Q} := 1 + \max\{\langle A \rangle_{\max}, \langle Q \rangle_{\max}\}.$$

The subscript max refers to the largest *entry encoding length*, not the largest coefficient magnitude or the total matrix encoding. Write $\varphi = \langle A, \boldsymbol{b}, Q, \boldsymbol{c}, \gamma \rangle$ for the full input length. All polynomial exponents written $O(1)$ are absolute and independent of n. All algorithms use exact rational arithmetic.

Theorem 1.1 (Bounded IQP with separated encoding)**.** *With $\varphi_{A,Q}$ as defined above, for bounded P, one can detect integer infeasibility or return an exact integer minimizer of f in time*

$$2^{O(n \log(n+1))}(m+1)^{O(n)} \varphi_{A,Q}^{O(n)} (1 + \varphi)^{O(1)}. \tag{1}$$

The coefficient-size factor $\varphi_{A,Q}^{O(n)}$ depends only on A, Q. The encoding of $\boldsymbol{b}, \boldsymbol{c}, \gamma$ occurs only in the fixed-degree polynomial factor. A bound using total structural encoding is obtained by replacing $(m+1)^{O(n)} \varphi_{A,Q}^{O(n)}$ with $(1 + \langle A, Q \rangle)^{O(n)}$.

The proof first establishes the following direct geometric solver and then uses localization to remove $\boldsymbol{b}$ from its dimension-dependent exponent.

Theorem 1.2 (Direct geometric solver)**.** *If P is bounded, integer quadratic minimization over P can be solved, including detection of integer infeasibility, in time*

$$2^{O(n \log(n+1))}(m+1)^{O(n)}(1 + \langle A, \boldsymbol{b} \rangle_{\max})^{O(n)}(1 + \varphi)^{O(1)}.$$

More precisely, let Π *be the Goemans–Rothvoß family of integral parallelepipeds constructed as in Theorem 2.4, let* $N := |\Pi|$ *be its number of pieces, and let* T_{GR} *be its construction cost, including clearing denominators for rational input. Then the total cost satisfies*

$$T_{\mathrm{bounded}} \leq T_{\mathrm{GR}} + 2^{O(n\log(n+1))} N (1 + \langle A, \boldsymbol{b}\rangle_{\max})^n (1+\varphi)^{O(1)}. \tag{2}$$

Integer linear programming and Lenstra-type bounds. Lenstra [17] established polynomial-time integer linear programming in fixed dimension. Kannan [16] improved the dimension dependence to $n^{O(n)}$, with a fixed-degree polynomial factor in the binary input length. Subsequent developments include Hildebrand and Köppe's Lenstra-type algorithm for quasiconvex polynomial integer minimization [15], Dadush's lattice algorithms [7], and the algorithm of Dadush, Eisenbrand, and Rothvoß [6]. Reis and Rothvoß [21] obtain a randomized ILP algorithm with running time $(\log(2n))^{O(n)}$ times a fixed-degree polynomial in the input length. For linear objectives, therefore, the full encoding of $A, \boldsymbol{b}, \boldsymbol{c}$ can occur outside the dimension-dependent factor. Our IQP bound has a different form: it places the encoding lengths of A, Q inside a factor raised to $O(n)$, while retaining a fixed-degree polynomial dependence on the remaining data.

Integer hulls and concave minimization. For a concave objective on a polytope containing an integer point, an integer minimum is attained at a vertex of $P_{\mathbb{Z}} = \mathrm{conv}(P \cap \mathbb{Z}^n)$. Indeed, expressing any integer point as a convex combination of integer-hull vertices and using concavity shows that at least one of those vertices has no greater value. Cook, Hartmann, Kannan, and McDiarmid [5] bound the number of integer-hull vertices by

$$2m^n(6n^2\varphi_{\mathrm{ineq}})^{n-1},$$

where $\varphi_{\mathrm{ineq}} \geq 1$ bounds the binary encoding length of one defining inequality, including its right-hand side. Together with Hartmann's fixed-dimensional vertex-enumeration algorithm [12], this gives polynomial-time concave integer minimization in fixed dimension, assuming exact polynomial-time objective evaluation. The vertex count already has the inequality encoding length in the base of a dimension-dependent power; it is not a Lenstra-type dimension-only parameter bound.

Ari and Hildebrand [1, Section 7, Proposition 7.4] give an explicit analysis that removes the right-hand sides from this geometric dependence. In our notation, their concave-minimization bound can be written as

$$n^{O(n)}(m+1)^{O(n)}(1+\langle A\rangle_{\max})^{O(n)}(1+\varphi)^{O(1)},$$

with the objective description and evaluation cost included in the input model. Their maximum row encoding length lies between the maximum entry encoding length and a polynomial in n times that length, so the two forms agree at this level of precision. Theorem 1.1 has the same form of encoding separation for arbitrary quadratic objectives, with Q entering the structural factor.

Earlier quadratic algorithms. Del Pia and Weismantel [9] gave a polynomial-time algorithm for minimizing an arbitrary quadratic polynomial over the integer points of a rational polyhedron in dimension two. Del Pia, Hildebrand, Weismantel, and Zemmer [10] extended this result to arbitrary cubic objectives. This completes the classification by degree in the plane: degree three is polynomial-time solvable, and degree four is NP-hard. These algorithms decompose the plane into regions on which suitable sublevel or superlevel sets are convex. The exact fixed-dimensional algorithm for arbitrary indefinite quadratics in bounded polyhedra given here also applies in dimensions $n \geq 3$.

Lokshtanov [18] and Zemmer [23] independently proved fixed-parameter tractability with parameters given by the dimension and the maximum numerical magnitude of the entries of A, Q.

Their bounds depend on coefficient magnitudes. Theorem 1.1 gives polynomial dependence on their binary encoding lengths for fixed dimension, and fixed-degree polynomial dependence on the rest of the input.

Hildebrand and Göß [14] study integer feasibility in structured reverse-convex sets through boundary hyperplane covers. They decompose these problems into convex and single-exclusion subproblems. Their integer-query convex feasibility theorem is the optimization tool used on our remaining cells.

Bounded entries and the number of rows. As Lokshtanov [18, Lemma 1] observes, if A is integral and $|A_{ij}| \le \alpha$ for an integer $\alpha \ge 1$, there are at most $(2\alpha+1)^n$ distinct coefficient rows. Among inequalities with the same row, retain only the smallest right-hand side. This polynomial-time preprocessing preserves even the real feasible region and leaves

$$m' \le (2\alpha + 1)^n.$$

Thus m' is bounded by a function of n, α, and by a function of n alone when the entry bound is fixed. The cost of reading the original system remains part of the input polynomial. This observation concerns integer coefficients (after normalization); bounded magnitudes of arbitrary rational entries alone would not bound the number of distinct rows.

Why dimension-dependent encoding may be necessary. Integer-hull enumeration motivates the encoding dependence but does not prove a lower bound for optimization. There is separate hardness evidence: Herrmann [13] proves W[1]-hardness of IQP parameterized by the number of variables, even for a separable concave objective with bounded quadratic coefficients. His construction can be made bounded by adding upper bounds on the auxiliary variables that retain every intended integer witness; see also the bounded construction in Ari and Hildebrand [1, Section 6]. Unless FPT = W[1], one therefore cannot replace all dependence on m and the structural encoding by a function of n while leaving a fixed-degree polynomial in the full input length. This supports the need for some dimension-dependent input-size dependence, but does not establish that our particular powers of $m+1$ and $1 + \langle A, Q\rangle_{\max}$ are individually optimal.

Localization and proximity. Our localization builds on the boundary/gradient alternatives in Lokshtanov [18, Lemmas 2–5] and the curvature-batching approach of Ari and Hildebrand [1]. We retain boundary and directional-gradient slabs as inequalities rather than enumerate their individual integer levels. We then obtain bounded representatives by linear integer sensitivity, as explained in Section 5.

Del Pia and Ma [8] show that even concave quadratic optimization has no general exact proximity bound depending only on dimension and constraint subdeterminants; their positive proximity results concern approximation. In our localization argument, the objective is constant on the points of a constructed region that have the same value under a specified linear map. Linear integer sensitivity gives a nearby integer point with that same image, and hence the same objective value.

Organization. Section 2 states the required cell decomposition and constructs it by refining a parallelepiped cover. Section 3 proves that a negative displacement excludes a cell and gives the optimizer for the other cells. Section 4 supplies the negative-displacement search, then assembles the bounded algorithm and proves its running-time bound. Section 5 proves the localization result and the improved encoding bound in Theorem 1.1. Section 6 explains the comparison-only variant and the extension using the companion unboundedness result. Appendix A gives the comparison-only implementation.

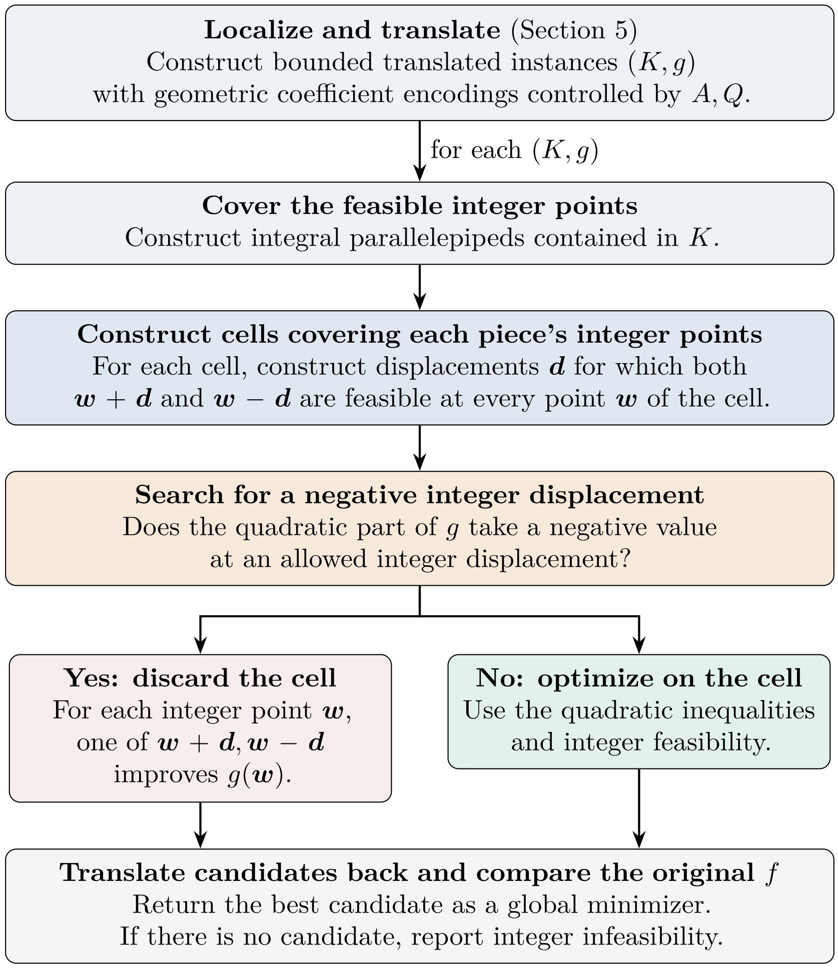


Figure 1: The complete bounded algorithm. Localization produces bounded translated instances (K, g). For each instance, the direct algorithm from Sections 2–4 covers the integer points, tests each cell for a negative displacement, and minimizes on the remaining nonempty cells. Candidates from all instances are translated back and compared using the original objective f.

2 A cell decomposition with feasible displacements

The algorithm needs a cover of the feasible integer points by cells on which differences and feasible two-sided displacements can be controlled by the same set. We first state this geometric result independently of its construction.

Definition 2.1 (Symmetric displacement cover)**.** Let $P \subseteq \mathbb{R}^n$ be a bounded rational polyhedron. A *symmetric displacement cover* of $P \cap \mathbb{Z}^n$ is a finite family $\mathcal{C}$ of pairs (C, D_C) such that:

1. Each cell $C \subseteq P$ is specified by rational linear equalities and inequalities, possibly strict, and the cells cover $P \cap \mathbb{Z}^n$.

2. Each D_C is a rational polytope containing $\mathbf{0}$ and symmetric about $\mathbf{0}$, with

$$C - C \subseteq D_C, \qquad C + D_C \subseteq P, \qquad C - D_C \subseteq P.$$

The inclusions hold for real points. The cells may overlap and need not cover all of P. Symmetry is required of the displacement sets; the cells themselves need not be symmetric.

The first inclusion places all cell differences in a common displacement set. The other two allow every displacement in that set to be added to or subtracted from every cell point while staying feasible; see Figure 2. These are the properties used by the quadratic argument in Section 3, and they do not require integral vertices.

Theorem 2.2 (Constructing a symmetric displacement cover)**.** *For a bounded rational polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \leq \boldsymbol{b}\}$ with m inequalities, one can construct a symmetric displacement cover with at most*

$$(m+1)^n n^{O(n)} (1 + \langle A, \boldsymbol{b} \rangle_{\max})^{2n}$$

pairs, in time

$$(m+1)^{O(n)} n^{O(n)} (1 + \langle A, \boldsymbol{b} \rangle_{\max})^{O(n)} (1 + \langle A, \boldsymbol{b} \rangle)^{O(1)}.$$

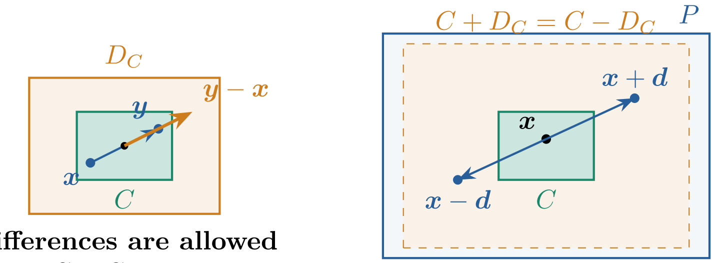
Each pair has description length at most $p(n)(1 + \langle A, \boldsymbol{b} \rangle_{\max})$ for an absolute polynomial p. Each D_C is defined in a rational subspace by at most n two-sided linear inequalities whose normals have trivial common kernel there. The final polynomial degree in the construction time is absolute.

We prove the theorem by refining a Goemans–Rothvoß cover. Proposition 2.9 gives the more precise cell count in terms of the number of parallelepipeds actually produced.

Why the right-hand-side dependence is compatible with the main bound. The cover theorem is an intermediate result: applying it directly to P allows the encoding of $\boldsymbol{b}$ inside the dimension-dependent exponent, as in Theorem 1.2. The full algorithm instead first applies Section 5. It constructs bounded translated instances $\widetilde{A}\boldsymbol{w} \leq \widetilde{\boldsymbol{b}}$ satisfying

$$1 + \langle \widetilde{A}, \widetilde{\boldsymbol{b}} \rangle_{\max} \leq p(n) \varphi_{A,Q}.$$

In each localized box, rows with large translated right-hand sides are redundant and are deleted; the remaining right-hand sides have this structural encoding bound. Applying the cover theorem to these instances therefore yields the factor $\varphi_{A,Q}^{O(n)}$. The original $\boldsymbol{b}$ still affects the translations and arithmetic, whose cost is charged to $(1+\varphi)^{O(1)}$, but not the final structural factor. Thus Theorem 2.2 is used, not replaced: it supplies the cover in the proof of the direct solver, Theorem 1.2, which Section 5 calls on each localized instance. Localization is necessary for the stronger encoding bound of Theorem 1.1; the direct bound of Theorem 1.2 does not require it.



Differences are allowed
$C - C \subseteq D_C$

Both moves stay feasible
$C \pm D_C \subseteq P$

Figure 2: The two requirements for each pair in a symmetric displacement cover. Here $C = [-7/10, 7/10] \times [-1/2, 1/2]$ and $D_C = 2C$, so $C - C = D_C$ and $C \pm D_C = 3C \subseteq P$. Left: every difference of cell points belongs to the displacement set. Right: any allowed displacement can be used in both directions from every point of C. A cover consists of enough such cells to contain all feasible integer points.

2.1 The integral parallelepiped cover

The covering theorem of Goemans–Rothvoß provides parallelepipeds with integer vertices. Every feasible integer point belongs to at least one of them, and every parallelepiped lies in P.

Definition 2.3 (Integral parallelepiped)**.** A k-dimensional parallelepiped in $\mathbb{R}^n$, $0 \le k \le n$, is a set

$$\pi = \boldsymbol{v} + B[0,1]^k$$

where $B \in \mathbb{R}^{n\times k}$ has linearly independent columns. It is *integral* if every vertex belongs to $\mathbb{Z}^n$. Equivalently, in this vertex-based representation one has $\boldsymbol{v} \in \mathbb{Z}^n$ and $B \in \mathbb{Z}^{n\times k}$. For $k = 0$ this means an integral singleton. No unimodularity is required.

Theorem 2.4 (Goemans–Rothvoß integral cover [11, Lemma 4.1])**.** *Let $A \in \mathbb{Z}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{Z}^m$, and suppose $P = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \le \boldsymbol{b}\}$ is a bounded polyhedron. Set*

$$\Delta = \max\{2, \max_{i,j} |A_{ij}|, \max_i |b_i|\}.$$

There is a finite family Π of integral parallelepipeds satisfying

$$P \cap \mathbb{Z}^n \;\subseteq\; \bigcup_{\pi\in\Pi} \pi \;\subseteq\; P, \qquad |\Pi| \le N_{\mathrm{GR}} := m^n n^{O(n)} (\log \Delta)^n.$$

Such a family can be computed in time $N_{\mathrm{GR}}^{O(1)}$. The pieces may have dimension less than n.

For rational $A, \boldsymbol{b}$, we apply the theorem after multiplying each row by a positive common denominator of that row and its right-hand side. This preserves P and makes the entries integral, with $\log \Delta \le p(n)\langle A, \boldsymbol{b}\rangle_{\max}$. The resulting number of pieces satisfies

$$N_{\mathrm{GR}} \le (m+1)^n n^{O(n)} (1 + \langle A, \boldsymbol{b}\rangle_{\max})^n. \tag{3}$$

Including this conversion, the construction takes time

$$T_{\mathrm{GR}} \le (1 + N_{\mathrm{GR}})^a (1 + \langle A, \boldsymbol{b}\rangle)^{O(1)} \tag{4}$$

for an absolute constant a. In the running-time analysis, T_{GR} denotes this construction cost and $N = |\Pi|$ denotes the number of pieces produced. Lemma 4.1 of the published journal version explicitly supplies both real containment and construction time $N_{\mathrm{GR}}^{O(1)}$, with an absolute exponent; it does not assert linear time in the actual number of output pieces. When $P \cap \mathbb{Z}^n$ is empty, the family may be empty. Figure 3 shows an example of the cover.

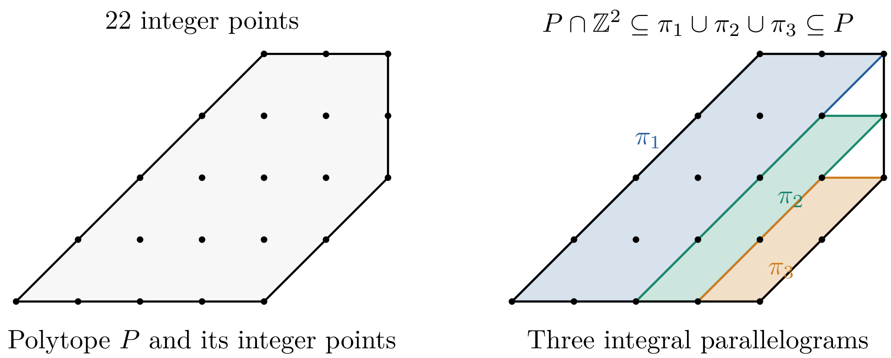


Figure 3: The three parallelograms cover every integer point of P and lie inside P. The example is obtained by applying the integral shear $(a, b) \mapsto (a + b, b)$ to $[0, 4]^2 \cap \{a + b \leq 6\}$ and to the rectangles $[0, 2] \times [0, 4]$, $[2, 3] \times [0, 3]$, and $[3, 4] \times [0, 2]$. The gaps between the parallelograms contain no integer points.

2.2 Constructing the cells

For each cell C in a parallelepiped π, we construct a displacement set D_C with two properties:

$$C - C \subseteq D_C, \qquad C + D_C,\ C - D_C \subseteq \pi.$$

The first property puts every difference of two cell points in D_C. The second allows every displacement in D_C to be used in both directions from every point of C. We obtain both properties by grouping points according to their slacks in the affine coordinates of π.

Affine coordinates and complementary slacks. Fix an integral cover piece $\pi = \boldsymbol{v} + B[0, 1]^k$ from Theorem 2.4, and put $V = \operatorname{span} B$. Choose a rational left inverse L with $LB = I_k$. For $\boldsymbol{x} \in \boldsymbol{v} + V$, define

$$\boldsymbol{\theta}(\boldsymbol{x}) = L(\boldsymbol{x} - \boldsymbol{v}).$$

The inequalities defining π in these coordinates are $0 \leq \theta_j(\boldsymbol{x}) \leq 1$. Their slacks are $\theta_j(\boldsymbol{x})$ and $1 - \theta_j(\boldsymbol{x})$. They sum to one and are nonnegative on π. For $\boldsymbol{d} \in V$, adding a displacement changes the coordinates by

$$\boldsymbol{\theta}(\boldsymbol{x} + \boldsymbol{d}) = \boldsymbol{\theta}(\boldsymbol{x}) + L\boldsymbol{d}.$$

Integrality refers to the ambient lattice $\mathbb{Z}^n$. The affine coordinates $\boldsymbol{\theta}(\boldsymbol{x})$ and $L\boldsymbol{d}$ may be fractional even when $\boldsymbol{x}$ and $\boldsymbol{d}$ are integral. We will divide positive slacks into intervals of the form $(a, 2a]$. Such an interval has width a, and every slack in it is greater than a. The same number can therefore bound both the difference between two slacks and the movement toward a boundary. Only finitely many such intervals are needed: the rational coordinate formulas give a lower bound on every positive slack at an integer point.

Let $D_j \geq 1$ be a common positive integer denominator of the coefficients and constant term of θ_j. Set

$$\ell_{\max} = 1 + \max\bigl(\{1\} \cup \{\lceil \log_2 D_j \rceil : 1 \leq j \leq k\}\bigr).$$

For $k = 0$, the coordinate family is empty and $\ell_{\max} = 2$.

Lemma 2.5 (Coordinate encoding and a finite cutoff)**.** *The affine-coordinate data above can be computed with encoding lengths $p(n)(1 + \langle A, \boldsymbol{b} \rangle_{\max})$, and $\ell_{\max} \leq p(n)(1 + \langle A, \boldsymbol{b} \rangle_{\max})$. At an ambient integer point of π, each positive complementary slack is strictly greater than $2^{-\ell_{\max}}$.*

Proof. Vertices of a bounded rational polyhedron with entry encoding lengths at most $\langle A, \boldsymbol{b}\rangle_{\max}$ have coordinate magnitudes at most $2^{p(n)(1+\langle A,\boldsymbol{b}\rangle_{\max})}$, by determinant bounds on vertex systems. The same magnitude bound holds throughout P, and hence for the integral vertices of every contained piece. Their coordinates and differences therefore have bit lengths $p(n)(1 + \langle A, \boldsymbol{b}\rangle_{\max})$. Choose a nonsingular $k \times k$ row submatrix of B and invert it to obtain L; determinant formulas bound its rational coefficients and the affine constant $-L\boldsymbol{v}$ in the same form. Clearing at most $n+1$ denominators for each coordinate preserves that bound for D_j.

At integer $\boldsymbol{x}$, $D_j\theta_j(\boldsymbol{x})$ and $D_j(1-\theta_j(\boldsymbol{x}))$ are integers. A positive slack is thus at least $1/D_j \geq 2^{1-\ell_{\max}} > 2^{-\ell_{\max}}$. The logarithmic definition of $\ell_{\max}$ proves the stated cutoff and its size. ☐

Definition 2.6 (Dyadic slack intervals and cells)**.** For the cutoff $\ell_{\max}$, use labels $0, 1, \ldots, \ell_{\max}$. Treat zero slack separately and use the dyadic slack intervals

$$I_0 = \{0\}, \qquad I_r = (2^{-r}, 2^{1-r}] \quad (1 \leq r \leq \ell_{\max}).$$

The label 0 denotes zero slack; the label 1 denotes the positive interval $(1/2, 1]$. A pair (r, s) is *compatible* if there are $u \in I_r$, $w \in I_s$ with $u+w = 1$. For a tuple of compatible pairs $\sigma = ((r_j, s_j))_{j=1}^{k}$, define the cell

$$C_\sigma = \{\boldsymbol{x} \in \pi : \theta_j(\boldsymbol{x}) \in I_{r_j},\ 1 - \theta_j(\boldsymbol{x}) \in I_{s_j}\ (1 \leq j \leq k)\}.$$

Each cell is specified by rational linear inequalities, which may be strict, and equalities. Empty cells may be retained. For $k = 0$, the empty tuple gives $C_\sigma = \pi$.

Lemma 2.7 (Compatible pairs and integer coverage)**.** *There are exactly* $2\ell_{\max} + 1$ *compatible pairs:*

$$(0,1), (1,0), (2,2), \quad (1,r), (r,1) \quad (2 \leq r \leq \ell_{\max}).$$

Consequently each piece has at most $(2\ell_{\max} + 1)^k$ *defined cells, and these cells cover its integer points.*

Proof. A zero slack forces its complement to equal one, giving the first two pairs. For two positive slacks, either one exceeds $1/2$ or both equal $1/2$. In the latter case both have label 2. In the former case the larger has label 1 and the smaller a label in $\{2, \ldots, \ell_{\max}\}$. Each listed pair can occur for real numbers summing to one. This proves the exact list. Lemma 2.5 ensures that the chosen intervals and the zero-slack case contain all integer-point slacks. Assigning each slack its label proves coverage. ☐

Definition 2.8 (Displacement set of a cell)**.** For a cell $C = C_\sigma$, set

$$a_j = \begin{cases} 0, & r_j = 0 \text{ or } s_j = 0, \\ 2^{-\max(r_j, s_j)}, & \text{otherwise.} \end{cases}$$

The *displacement set* of C is

$$D_C = \{\boldsymbol{d} \in V : |(L\boldsymbol{d})_j| \leq a_j \quad (1 \leq j \leq k)\}.$$

This is a rational polytope symmetric about the origin. Its coefficients depend only on the cell geometry. A zero width fixes the corresponding coordinate; a positive width bounds movement in both directions.

The next proposition verifies the two required inclusions and bounds the number of cells. Both inclusions hold for real points of the cells. When $\boldsymbol{x}$ and $\boldsymbol{d}$ are integral, the feasible points $\boldsymbol{x} + \boldsymbol{d}$ and $\boldsymbol{x} - \boldsymbol{d}$ are integral as well.

Proposition 2.9 (Refinement of the parallelepiped cover)**.** *Let P be bounded and let Π be the family from Theorem 2.4. Applying the affine-coordinate construction and Definitions 2.6 and 2.8 to its pieces constructs a family of pairs (C, D_C) such that*

1. *the cells cover $P \cap \mathbb{Z}^n$, and each cell lies in a parallelepiped $\pi \subseteq P$;*
2. *$C - C \subseteq D_C$ and $C + D_C, C - D_C \subseteq \pi$;*
3. *each D_C is defined in a rational subspace by at most n two-sided linear inequalities, whose normals have trivial common kernel in that subspace;*
4. *the number of pairs is at most $N[p(n)(1 + \langle A, \boldsymbol{b}\rangle_{\max})]^n$, and their descriptions have encoding length $p(n)(1 + \langle A, \boldsymbol{b}\rangle_{\max})$.*

Proof. Fix a cover piece $\pi = \boldsymbol{v} + B[0,1]^k$. Every integer point of π has each positive coordinate slack in one of the chosen intervals, by the choice of $\ell_{\max}$. Assigning labels to its $2k$ slacks therefore places it in a cell. Because the original pieces cover $P \cap \mathbb{Z}^n$, the cells do as well.

We verify the two geometric inclusions one coordinate at a time. Suppose θ_j and $1 - \theta_j$ have positive-slack labels r, s, and write $a_j = \min(2^{-r}, 2^{-s})$. For $\boldsymbol{x}, \boldsymbol{y} \in C$, both $\theta_j(\boldsymbol{x})$ and $\theta_j(\boldsymbol{y})$ lie in an interval of width 2^{-r}. Their complementary slacks lie in an interval of width 2^{-s}. Hence

$$|(L(\boldsymbol{y} - \boldsymbol{x}))_j| = |\theta_j(\boldsymbol{y}) - \theta_j(\boldsymbol{x})| \le a_j.$$

If a slack is zero throughout C, the corresponding coordinate is constant, so $(L(\boldsymbol{y} - \boldsymbol{x}))_j = 0$. Also $\boldsymbol{y} - \boldsymbol{x} \in V$, proving $C - C \subseteq D_C$.

For $\boldsymbol{x} \in C$ and $\boldsymbol{d} \in D_C$, the positive-slack case gives $\theta_j(\boldsymbol{x}) > 2^{-r} \ge a_j$ and $1 - \theta_j(\boldsymbol{x}) > 2^{-s} \ge a_j$. Therefore

$$0 \le \theta_j(\boldsymbol{x}) \pm (L\boldsymbol{d})_j \le 1.$$

A zero slack instead forces $(L\boldsymbol{d})_j = 0$, so the boundary coordinate stays fixed. Since $\boldsymbol{d} \in V$, the points $\boldsymbol{x} \pm \boldsymbol{d}$ belong to $\operatorname{aff} \pi$; the coordinate inequalities put both in π. This proves $C \pm D_C \subseteq \pi$.

There are at most $(2\ell_{\max} + 1)^k$ cells per piece. The rows of L have trivial common kernel on V, because $LB = I_k$. The constraints with $a_j = 0$ define a rational subspace of V. On that subspace, the remaining rows still have trivial common kernel, so their two-sided bounds define a bounded set. Finally, rational left inverses and the chosen interval endpoints have encoding length $p(n)(1+\langle A, \boldsymbol{b}\rangle_{\max})$, giving the stated count and description bounds. If $k = 0$, the piece is a singleton, the empty family of linear functionals has common kernel $\{\boldsymbol{0}\}$ in $V = \{\boldsymbol{0}\}$, and $D_C = \{\boldsymbol{0}\}$. ☐

Cost of producing the cell descriptions. Once the cover is available, computing left inverses and denominator cutoffs uses polynomial bit arithmetic per piece. Enumerate the explicit list of compatible pairs independently in each coordinate and write down the resulting inequalities and displacement constraints. This takes

$$N[p(n)(1 + \langle A, \boldsymbol{b}\rangle_{\max})]^n (1 + \langle A, \boldsymbol{b}\rangle)^{O(1)}$$

bit operations, after increasing p if necessary. Empty cells may be retained in the list. Later, the algorithm either discards a cell using a negative displacement or passes it to the cell optimizer, which tests integer feasibility.

Proof of Theorem 2.2. Apply Theorem 2.4 after clearing denominators, and refine its pieces using Proposition 2.9. The proposition gives the required geometry and descriptions. Multiplying its cell count by the cover bound (3) gives the stated bound on $|\mathcal{C}|$. The cover construction cost (4) and the cost of writing the cell descriptions give the claimed total time. ☐

Rational pieces also suffice. The refinement works for any explicitly given finite cover of $P \cap \mathbb{Z}^n$ by rational parallelepipeds contained in P, even when their vertices are not integral. Indeed, a rational representation $\boldsymbol{v} + B[0,1]^k$ still has a rational left inverse and affine-coordinate denominators. These denominators give a finite cutoff for positive slacks at ambient integer points, and the proofs of coverage and the displacement inclusions are unchanged. With entry encoding length at most E for $\boldsymbol{v}, B$, the coordinate data and cutoff have size $p(n)(1+E)$, giving at most $[p(n)(1+E)]^n$ cells per piece. Integral vertices are convenient because containment in P then bounds their encoding lengths from the input data, as in Lemma 2.5; for rational vertices, their denominator lengths must also be controlled. Thus integrality is a useful feature of the chosen construction, rather than a requirement of the geometric argument.

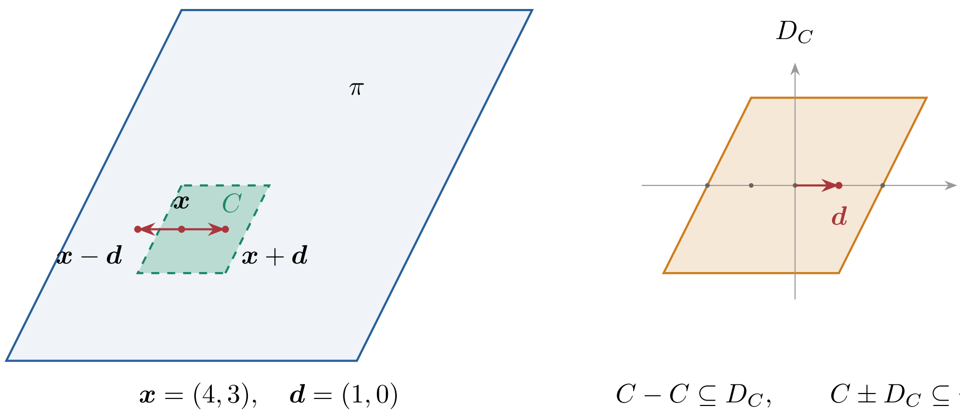


Figure 4: Every $\boldsymbol{d} \in D_C$ can be added to or subtracted from every $\boldsymbol{x} \in C$ while staying in π. Here $\pi = (8,0)[0,1] + (4,8)[0,1]$, the cell has $1/4 < \theta_1, \theta_2 < 1/2$, and $D_C = (8,0)[-1/4,1/4] + (4,8)[-1/4,1/4]$. All three marked points are integral and feasible. If $q(\boldsymbol{d}) < 0$, one of $\boldsymbol{x} + \boldsymbol{d}$ and $\boldsymbol{x} - \boldsymbol{d}$ has smaller objective value than $\boldsymbol{x}$. Dashed cell edges indicate strict inequalities.

2.3 Relation to other geometric covers

Dyadic reflection cells. Ari and Hildebrand [1, Lemma 7.2] use dyadic slack cells to isolate integer-hull vertices: if distinct integer points $\boldsymbol{v}, \boldsymbol{y}$ lie in one cell, their reflection $2\boldsymbol{v} - \boldsymbol{y}$ is feasible, so $\boldsymbol{v}$ cannot be an integer-hull vertex. This is closely related to our reflection geometry. Our cover requires the stronger, uniform conditions $C - C \subseteq D_C$ and $C \pm D_C \subseteq P$ for a computably described symmetric displacement set; that cited lemma does not supply these sets and their description bounds.

Triangulations. A rational triangulation of P supplies a cover with contained pieces, but a simplex taken as a single cell need not satisfy $C \pm (C - C) \subseteq P$. It can instead be refined by dyadic bands for its barycentric coordinates. On a k-simplex these are $k+1$ nonnegative affine functions summing to one. For each positive band, bound the corresponding displacement coordinate by the band's width; for a zero coordinate, require the displacement coordinate to vanish. Differences of cell points obey these bounds, while every cell point has enough slack to move in both directions. This proves the same two geometric inclusions, using at most $k+1$ two-sided inequalities. Rational denominators again give a finite cutoff at integer points. The displacement routine in Section 4 allows this larger number of inequalities. However, the total cost also depends on the triangulation size and the refined cell count; this observation alone does not give the quantitative bounds of Theorem 2.2.

Macbeath regions. There is a direct connection to the local symmetry underlying Macbeath regions [19]. For $\boldsymbol{z} \in P$, put

$$S_{\boldsymbol{z}} = (P - \boldsymbol{z}) \cap (\boldsymbol{z} - P), \qquad M_P^\lambda(\boldsymbol{z}) = \boldsymbol{z} + \lambda S_{\boldsymbol{z}}.$$

The set $S_{\boldsymbol{z}}$ is convex and symmetric about the origin, and $\boldsymbol{z} + S_{\boldsymbol{z}} \subseteq P$. For $0 < \lambda \leq 1/3$, let $C = M_P^\lambda(\boldsymbol{z})$ and $D = 2\lambda S_{\boldsymbol{z}}$. Then

$$C - C = 2\lambda S_{\boldsymbol{z}} = D, \qquad C \pm D = \boldsymbol{z} + 3\lambda S_{\boldsymbol{z}} \subseteq P.$$

Thus a cover of the integer points by sufficiently shrunken Macbeath regions has exactly the required geometric properties. For rational $\boldsymbol{z}$ and λ, these sets have rational linear descriptions. Such a construction would still need bounds on the number of regions, their encoding lengths, and the cost of finding them, including regions on proper faces. The dyadic construction above provides these controls explicitly for parallelepipeds.

***M*-ellipsoid coverings.** Dadush's thesis [7, Chapter IV, especially Sections 4.1.2 and 4.2] studies ellipsoids E for which both P can be covered by $2^{O(n)}$ translates of E and E by $2^{O(n)}$ translates of P, working in the affine hull when necessary. This suggests using a small cover by simple centrally symmetric shapes. The covering property alone, however, does not require the covering ellipsoids to lie in P. Intersecting them with P restores containment but need not preserve central symmetry or give uniform feasible displacements. To use this approach here, one would need an additional construction establishing the two displacement inclusions, exact coverage of boundary integer points, and suitable subproblem representations. We therefore use the connection as motivation for other possible constructions, without substituting an ellipsoid covering number for the cell count proved above.

Covering the integer hull. Only the integer points must be covered. In particular, one may work inside $P_{\mathbb{Z}} := \operatorname{conv}(P \cap \mathbb{Z}^n)$, since $P_{\mathbb{Z}} \cap \mathbb{Z}^n = P \cap \mathbb{Z}^n$ and $P_{\mathbb{Z}} \subseteq P$. Every integral parallelepiped contained in P already lies in $P_{\mathbb{Z}}$: it is the convex hull of its integral vertices. If the integer hull has a simpler available description or admits a smaller suitable cover than the linear relaxation, this may reduce the number of pieces and cells. The complexity of obtaining that description or cover must also be counted. Our construction does not require computing the integer hull, and its worst-case bound is expressed in the original input data.

3 Discarding cells and optimizing the rest

For each cell C, we search for an integer $\boldsymbol{d} \in D_C$ with $q(\boldsymbol{d}) < 0$. If one exists, both $\boldsymbol{x} + \boldsymbol{d}$ and $\boldsymbol{x} - \boldsymbol{d}$ are feasible for every integer $\boldsymbol{x} \in C$, and one has smaller objective value than $\boldsymbol{x}$. We can therefore discard C.

Otherwise, $q(\boldsymbol{y} - \boldsymbol{x}) \geq 0$ for all integer $\boldsymbol{x}, \boldsymbol{y} \in C$, because $C - C \subseteq D_C$. The quadratic identity then gives linear inequalities that separate an integer point of C from all integer points of C with smaller objective value. We use these inequalities with the integer feasibility algorithm of Hildebrand and Göß to minimize f on C.

3.1 Cell optimization and negative-displacement search

The algorithm uses two subroutines: Negative searches D_C for a negative integer displacement, and ConvicMin minimizes a quadratic whose quadratic part is nonnegative on integer differences.

We define their inputs and outputs here, prove the two possible outcomes for a cell, and then construct ConvicMin. Section 4 constructs Negative using ConvicMin.

Problem 1 (Cell optimization $\textsc{ConvicMin}(K, h)$)**.**

Data. A bounded rational polyhedron K and a rational quadratic

$$h(\boldsymbol{x}) = \boldsymbol{x}^\mathsf{T} Q_h \boldsymbol{x} + \boldsymbol{c}_h^\mathsf{T} \boldsymbol{x} + \gamma_h, \qquad Q_h = Q_h^\mathsf{T}.$$

Hypothesis. The quadratic part is nonnegative on integer differences:

$$(\boldsymbol{y} - \boldsymbol{x})^\mathsf{T} Q_h (\boldsymbol{y} - \boldsymbol{x}) \geq 0 \qquad (\boldsymbol{x}, \boldsymbol{y} \in K \cap \mathbb{Z}^n).$$

Output. An exact integer minimizer of h on K, or INFEASIBLE if $K \cap \mathbb{Z}^n$ is empty.

Proposition 3.7 implements ConvicMin using quadratic supporting inequalities and integer-query separation. For a cell from the cover, $h = f$. Section 4 will also use ConvicMin to minimize affine substitutions of the homogeneous form q.

Problem 2 (Negative-displacement search $\textsc{Negative}(D, Q)$)**.**

Data. A bounded rational polytope D containing $\mathbf{0}$ and symmetric about $\mathbf{0}$, and a rational symmetric matrix Q. In the cell-classification algorithm, $D = D_C$ from Definition 2.8.

Output. Either a witness $\boldsymbol{d} \in D \cap \mathbb{Z}^n$ with $\boldsymbol{d}^\mathsf{T} Q \boldsymbol{d} < 0$, or NONNEGATIVE, meaning $\boldsymbol{d}^\mathsf{T} Q \boldsymbol{d} \geq 0$ for every $\boldsymbol{d} \in D \cap \mathbb{Z}^n$.

Decision task. Determine whether

$$\min\{\boldsymbol{d}^\mathsf{T} Q \boldsymbol{d} : \boldsymbol{d} \in D \cap \mathbb{Z}^n\} < 0.$$

Any negative witness suffices; its value need not be the exact minimum. Since $\mathbf{0} \in D$, the outcome NONNEGATIVE means the minimum equals zero.

Strict inequalities on integer points. After clearing denominators, a strict inequality $\boldsymbol{a}^\mathsf{T} \boldsymbol{x} < b$ with integral $\boldsymbol{a}, b$ is equivalent on integer points to $\boldsymbol{a}^\mathsf{T} \boldsymbol{x} \leq b - 1$. Equalities can be written as two weak inequalities. We use this integer-equivalent weak description when passing a cell to the optimizer. It defines a closed rational polyhedron contained in the original cell and with exactly the same integer points. The geometric inclusions continue to hold for the original cell and for this smaller polyhedron.

3.2 The reflection identity and the two outcomes

If an integer displacement $\boldsymbol{d}$ is feasible in both directions and $q(\boldsymbol{d}) < 0$, one of the two directions improves the objective. Indeed, expanding the quadratic at $\boldsymbol{x}$ gives

$$f(\boldsymbol{x} \pm \boldsymbol{d}) = f(\boldsymbol{x}) \pm \nabla f(\boldsymbol{x})^\mathsf{T} \boldsymbol{d} + \boldsymbol{d}^\mathsf{T} Q \boldsymbol{d}.$$

Adding cancels the affine terms and yields the *reflection identity*

$$f(\boldsymbol{x} + \boldsymbol{d}) + f(\boldsymbol{x} - \boldsymbol{d}) = 2f(\boldsymbol{x}) + 2\boldsymbol{d}^\mathsf{T} Q \boldsymbol{d}. \tag{5}$$

When $q(\boldsymbol{d}) < 0$, the average of $f(\boldsymbol{x} + \boldsymbol{d})$ and $f(\boldsymbol{x} - \boldsymbol{d})$ is less than $f(\boldsymbol{x})$. At least one of the two values is therefore smaller than $f(\boldsymbol{x})$.

Proposition 3.1 (Negative displacements and cell optimization)**.** *For a cell and displacement set as above, the following alternatives hold.*

1. *If some $\boldsymbol{d} \in D_C \cap \mathbb{Z}^n$ satisfies $\boldsymbol{d}^\mathsf{T} Q \boldsymbol{d} < 0$, then every $\boldsymbol{x} \in C \cap \mathbb{Z}^n$ has an improving feasible integer reflection among $\boldsymbol{x} + \boldsymbol{d}$ and $\boldsymbol{x} - \boldsymbol{d}$. In particular, C contains no global integer minimizer on P.*

2. *If $\boldsymbol{d}^\mathsf{T} Q \boldsymbol{d} \geq 0$ for every $\boldsymbol{d} \in D_C \cap \mathbb{Z}^n$, then for all $\boldsymbol{x}, \boldsymbol{y} \in C \cap \mathbb{Z}^n$,*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\mathsf{T} (\boldsymbol{y} - \boldsymbol{x}). \tag{6}$$

Proof. In the first case, fix the negative direction $\boldsymbol{d}$ and take any integer $\boldsymbol{x} \in C$. Both $\boldsymbol{x} + \boldsymbol{d}$ and $\boldsymbol{x} - \boldsymbol{d}$ are integer points of P by the reflection inclusions. If both objective values were at least $f(\boldsymbol{x})$, their sum would be at least $2f(\boldsymbol{x})$, contradicting (5). Thus

$$\min\{f(\boldsymbol{x} + \boldsymbol{d}), f(\boldsymbol{x} - \boldsymbol{d})\} < f(\boldsymbol{x}).$$

The same $\boldsymbol{d}$ works for every integer point of C, so C contains no global integer minimizer. The improving sign may depend on $\boldsymbol{x}$.

In the second case, take any integer $\boldsymbol{x}, \boldsymbol{y} \in C$. Their difference $\boldsymbol{d} = \boldsymbol{y} - \boldsymbol{x}$ is integral and lies in D_C. By hypothesis $q(\boldsymbol{y} - \boldsymbol{x}) \geq 0$. The expansion

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\mathsf{T} (\boldsymbol{y} - \boldsymbol{x}) + q(\boldsymbol{y} - \boldsymbol{x})$$

then gives (6). □

Thus each cell requires one call to $\textsc{Negative}(D_C, Q)$. A negative displacement discards the cell. A NONNEGATIVE answer allows the call $\textsc{ConvicMin}(C, f)$, using the integer-equivalent weak description of C.

3.3 Discrete convicity from the supporting inequalities

The supporting inequalities imply discrete convicity, which is the hypothesis of the comparison optimizer in Appendix A. We prove this implication before constructing the separation oracle used by the main algorithm.

Definition 3.2 (Discrete convic function [22])**.** Let $E \subseteq \mathbb{Z}^n$. A function $h : E \to \mathbb{R}$ is *discrete convic* if, for every positive integer s, points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_s, \boldsymbol{y}, \boldsymbol{z} \in E$, and real numbers $\alpha_1, \ldots, \alpha_s \geq 0$ satisfying

$$h(\boldsymbol{x}_i) \leq h(\boldsymbol{y}) \quad (1 \leq i \leq s), \qquad \boldsymbol{z} = \boldsymbol{y} + \sum_{i=1}^{s} \alpha_i (\boldsymbol{y} - \boldsymbol{x}_i),$$

one has $h(\boldsymbol{z}) \geq h(\boldsymbol{y})$.

Lemma 3.3 (Quadratic support on integer differences)**.** *If $q(\boldsymbol{y} - \boldsymbol{x}) \geq 0$ for every $\boldsymbol{x}, \boldsymbol{y} \in E$, then $f|_E$ is discrete convic.*

Proof. Take a configuration from Definition 3.2 for $h = f|_E$, and expand at the common point $\boldsymbol{y}$. For each i,

$$f(\boldsymbol{x}_i) = f(\boldsymbol{y}) + \nabla f(\boldsymbol{y})^\mathsf{T} (\boldsymbol{x}_i - \boldsymbol{y}) + q(\boldsymbol{x}_i - \boldsymbol{y}).$$

Since $f(\boldsymbol{x}_i) \leq f(\boldsymbol{y})$ and $q(\boldsymbol{x}_i - \boldsymbol{y}) \geq 0$, rearranging yields

$$\nabla f(\boldsymbol{y})^\mathsf{T} (\boldsymbol{y} - \boldsymbol{x}_i) = f(\boldsymbol{y}) - f(\boldsymbol{x}_i) + q(\boldsymbol{x}_i - \boldsymbol{y}) \geq 0.$$

Multiply by $\alpha_i \geq 0$ and sum. The representation of $\boldsymbol{z}$ then gives $\nabla f(\boldsymbol{y})^\mathsf{T}(\boldsymbol{z}-\boldsymbol{y}) \geq 0$. Because $\boldsymbol{z}, \boldsymbol{y} \in E$, the hypothesis also gives $q(\boldsymbol{z}-\boldsymbol{y}) \geq 0$. A final expansion at $\boldsymbol{y}$ proves

$$f(\boldsymbol{z}) - f(\boldsymbol{y}) = \nabla f(\boldsymbol{y})^\mathsf{T}(\boldsymbol{z}-\boldsymbol{y}) + q(\boldsymbol{z}-\boldsymbol{y}) \geq 0,$$

which is precisely the defining implication. □

The supporting inequalities concern integer points; Q may be indefinite and f may be nonconvex on the real cell. The following separation oracle is for the convex hull of an integer objective sublevel set, and its query points belong to $\mathbb{Z}^n$.

3.4 The quadratic supplies a separation oracle

To minimize f, we test whether there is an integer point with value at most a given threshold τ. For a surviving cell, if an integer query point $\boldsymbol{x} \in C$ has $f(\boldsymbol{x}) > \tau$, then every integer $\boldsymbol{y} \in C$ with $f(\boldsymbol{y}) \leq \tau$ satisfies

$$\nabla f(\boldsymbol{x})^\mathsf{T}(\boldsymbol{y}-\boldsymbol{x}) \leq \tau - f(\boldsymbol{x}) < 0.$$

This gives a linear inequality satisfied by every such $\boldsymbol{y}$ and violated by $\boldsymbol{x}$. The following integer feasibility theorem uses this type of oracle.

Definition 3.4 (Integer-query separation oracle)**.** Let $S \subseteq \mathbb{R}^n$ be convex. An integer-query separation oracle for S takes $\boldsymbol{x} \in \mathbb{Z}^n$ and either confirms $\boldsymbol{x} \in S$, or returns a rational inequality $\boldsymbol{a}^\mathsf{T}\boldsymbol{z} \leq b$ that is valid for every $\boldsymbol{z} \in S$ and is violated by $\boldsymbol{x}$. The oracle is not required to accept queries at nonintegral points.

Theorem 3.5 (Convex integer feasibility from integer queries [2, Theorem 5.7 and Remark 5.11]; [14, Theorem 9])**.** *Let $R \geq 2$ be an integer and let $S \subseteq B_\infty(\mathbf{0}, R)$ be a closed convex set presented by an integer-query separation oracle. Let $\Phi_{\mathrm{sep}}(t)$ bound the bit cost and output encoding length of an oracle answer at any integer query of total encoding length at most t, with the fixed instance data included in this bound. Integer feasibility in $S \cap \mathbb{Z}^n$ can be solved in time*

$$2^{O(n\log(n+1))}\,\mathrm{poly}\big(n, 1+\langle R\rangle, \Phi_{\mathrm{sep}}(c_0 n(1+\langle R\rangle))\big),$$

where c_0 and the polynomial degree are absolute constants. Only integer points are submitted to the separation oracle.

The argument of Φ_{sep} accounts for the total encoding of an integer query in the containing box: each coordinate has $O(1+\log R)$ bits. Queries outside the box are separated by a box inequality without calling the supplied oracle. Under a recursive affine lattice parametrization, this test and the supplied oracle are applied to the corresponding point in the original coordinates.

The algorithmic bound follows from the pure-integer specialization of Basu's Theorem 5.7 and the exact-feasibility extension in Remark 5.11 [2]; Theorem 9 of Hildebrand and Göß [14] states the integer-query formulation explicitly. The centerpoint results of Basu and Oertel [3] provide information bounds; no integer-centerpoint counting algorithm is needed for the running time used here.

To see why only integer queries suffice, maintain a containing ellipsoid $\mathcal{E} = \{\boldsymbol{z} : (\boldsymbol{z}-\boldsymbol{c})^\mathsf{T}H(\boldsymbol{z}-\boldsymbol{c}) \leq 1\}$ with $H \succ 0$. CVP either finds an integer point within $1/(n+1)$ of its center in the H-norm, which can be queried and supplies a shallow cut if infeasible, or certifies that this central ellipsoid is lattice-free. In the latter case, flatness and dual SVP give a direction with $O(n^2)$ intersecting integer slices. Under this convention the width is

$$\mathrm{width}_{\boldsymbol{w}}(\mathcal{E}) = 2\sqrt{\boldsymbol{w}^\mathsf{T}H^{-1}\boldsymbol{w}}.$$

For exact termination, use the volume-below-one argument in Basu's Remark 5.11: a lattice-free translate exists, so translation-invariant width and SVP give at most $n+1$ integer slices to search. This also handles empty and lower-dimensional sets. Polynomially many volume reductions per node and polynomial branching per dimension drop give the stated dimension factor. We invoke Basu's bit-complexity result, including the rational rounding and lattice-coordinate bookkeeping delegated there to the standard ellipsoid literature. The cost of the supplied separation oracle is charged through Φ_{sep}.

Lemma 3.6 (Cell-sublevel separation). *Let $K \subseteq \mathbb{R}^n$ be a bounded rational polyhedron and suppose*

$$q(\boldsymbol{y}-\boldsymbol{x}) \geq 0 \qquad (\boldsymbol{x}, \boldsymbol{y} \in K \cap \mathbb{Z}^n). \tag{7}$$

For a rational threshold τ, put

$$S_\tau = \{\boldsymbol{y} \in K \cap \mathbb{Z}^n : f(\boldsymbol{y}) \leq \tau\}, \qquad \widehat{S}_\tau = \operatorname{conv}(S_\tau).$$

There is an integer-query separation oracle for $\widehat{S}_\tau$ with fixed-degree polynomial bit cost in the encodings of K, f, τ and the query point.

Proof. Query an integer point $\boldsymbol{x}$. If $\boldsymbol{x} \notin K$, return any violated defining inequality of K; it is valid for $\widehat{S}_\tau \subseteq K$. If $\boldsymbol{x} \in K$ and $f(\boldsymbol{x}) \leq \tau$, then $\boldsymbol{x} \in S_\tau \subseteq \widehat{S}_\tau$, so confirm membership.

It remains that $\boldsymbol{x} \in K$ and $f(\boldsymbol{x}) > \tau$. For every $\boldsymbol{y} \in S_\tau$, the quadratic identity and (7) give

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\mathsf{T}(\boldsymbol{y}-\boldsymbol{x}) + q(\boldsymbol{y}-\boldsymbol{x}) \quad \Longrightarrow \quad \nabla f(\boldsymbol{x})^\mathsf{T}(\boldsymbol{y}-\boldsymbol{x}) \leq \tau - f(\boldsymbol{x}) < 0.$$

Hence

$$\nabla f(\boldsymbol{x})^\mathsf{T}\boldsymbol{y} \leq \nabla f(\boldsymbol{x})^\mathsf{T}\boldsymbol{x} + \tau - f(\boldsymbol{x}) \tag{8}$$

is valid for S_τ, and therefore for its convex hull, while $\boldsymbol{x}$ violates it. All coefficients are obtained by rational arithmetic of fixed degree; multiplying by a common positive denominator gives an integral encoding if required. This also covers $\nabla f(\boldsymbol{x}) = \mathbf{0}$: then the displayed valid inequality has a negative right-hand side, which correctly certifies that S_τ is empty. □

Proposition 3.7 (Separation-oracle surviving-cell optimizer). *Under the hypotheses of Lemma 3.6, and given an integer $R \geq 2$ with $K \cap \mathbb{Z}^n \subseteq [-R, R]^n$, one can determine integer emptiness or return an exact minimizer of f on $K \cap \mathbb{Z}^n$ in time*

$$2^{O(n \log(n+1))}(1 + \varphi_{\mathrm{cell}})^{O(1)},$$

where φ_{cell} contains the encodings of K, f, R. The polynomial degree is absolute.

Proof. Clear the denominators of f by one positive multiplier d_f of polynomial encoding length. Thus $F = d_f f$ is integer-valued on $\mathbb{Z}^n$ and has exactly the same minimizers. The explicit bound

$$|f(\boldsymbol{x})| \leq |\gamma| + R\sum_i |c_i| + R^2 \sum_{ij} |Q_{ij}| \qquad (\boldsymbol{x} \in K \cap \mathbb{Z}^n)$$

gives integer bounds $\ell \leq F(\boldsymbol{x}) \leq u$ of polynomial encoding length. For an integer threshold $t \in [\ell, u]$, set $\tau = t/d_f$. Then S_τ consists exactly of the integer points of K with $F(\boldsymbol{x}) \leq t$. Since $S_\tau \subseteq [-R, R]^n$, its convex hull is also contained in this box.

Apply Theorem 3.5 to $\widehat{S}_\tau$, using the oracle from Lemma 3.6. The equality $\widehat{S}_\tau \cap \mathbb{Z}^n = S_\tau$ makes this the required integer sublevel feasibility test. To prove the equality, write an integer point of the hull as $\boldsymbol{x} = \sum_i \lambda_i \boldsymbol{y}_i$, where $\boldsymbol{y}_i \in S_\tau$, $\lambda_i \geq 0$, and $\sum_i \lambda_i = 1$. Convexity of K gives $\boldsymbol{x} \in K$. Expanding at $\boldsymbol{x}$ and using $\sum_i \lambda_i(\boldsymbol{y}_i - \boldsymbol{x}) = \mathbf{0}$ gives

$$\sum_i \lambda_i f(\boldsymbol{y}_i) = f(\boldsymbol{x}) + \sum_i \lambda_i q(\boldsymbol{y}_i - \boldsymbol{x}) \geq f(\boldsymbol{x}),$$

so $f(\boldsymbol{x}) \leq \tau$ and $\boldsymbol{x} \in S_\tau$.

Start with $t = u$. At this threshold, $S_{u/d_f} = K \cap \mathbb{Z}^n$, so an infeasible answer proves integer emptiness. Otherwise, binary-search the integer interval $[\ell, u]$ for the least feasible t. A point returned at that threshold is an exact minimizer. The interval has polynomial encoding length, so there are polynomially many feasibility calls. The cost of each oracle answer is polynomial in φ_{cell}, and the feasibility theorem gives the claimed total running time. ☐

Comparison-only implementation. Appendix A gives an alternative implementation of ConvicMin that supplies objective comparisons to a discrete-convic minimization algorithm. For a quadratic that is discrete convic on the integer domain, it returns an integer minimizer or determines integer emptiness in time $2^{O(n^2 \log(n+1))}(1 + \varphi_{\text{cell}})^{O(1)}$; see Proposition A.4. The explicitly given coefficients are used to construct the subproblems and answer the comparisons. The main algorithm uses Proposition 3.7, whose stronger bound follows from the explicit quadratic supporting inequalities.

4 Lattice refinement for negative-displacement search

This section implements the negative-displacement problem in boxed Problem 2. Given D, Q, it returns a negative integer witness or establishes that the minimum quadratic value is zero. For $D = D_C$, these outcomes respectively discard the cell C or permit cell optimization by ConvicMin from boxed Problem 1. Algorithm 1 gives the implementation.

Method. We keep D, Q fixed and pass from a coarse lattice to successively finer lattices. At each level, points in the same residue class modulo four have their half-difference in D and on the preceding, coarser lattice. Nonnegativity on that coarser lattice therefore gives the hypothesis of ConvicMin on each residue class. Optimizing over these classes either finds a negative displacement or proves nonnegativity at the finer level.

4.1 The displacement set and the lattice chain

Let $V \subseteq \mathbb{R}^n$ be a rational linear subspace, and let

$$D = \{\boldsymbol{d} \in V : |\boldsymbol{u}_i^\mathsf{T} \boldsymbol{d}| \leq \rho_i,\ 1 \leq i \leq t\} \tag{9}$$

be bounded, with rational $\boldsymbol{u}_i$ and positive rational ρ_i. Zero-width constraints are incorporated into V as equalities. The number t of two-sided inequalities is arbitrary. Let $Q = Q^\mathsf{T} \in \mathbb{Q}^{n \times n}$, and define

$$\Gamma_s = V \cap 2^s \mathbb{Z}^n = 2^s (V \cap \mathbb{Z}^n), \qquad s \geq 0.$$

Let φ_{neg} be the total encoding length of $V, (\boldsymbol{u}_i, \rho_i)_i, Q$ and the dimension. The case $n = 0$ is answered directly by $q(\mathbf{0}) = 0$; below $n \geq 1$.

Linearity of V gives the equality defining Γ_s. Residues are taken in the ambient integer coordinates. Every nonempty bounded rational polyhedron symmetric about zero has a description of this form: adjoin the reflections of its inequalities and incorporate zero-width constraints into V. The refinement lemma below requires only convexity and symmetry; the rational description is used to implement the algorithm.

Lemma 4.1 (An explicit terminal level)**.** *In polynomial time one can compute an integer $R \geq 1$ with $D \subseteq [-R, R]^n$ and a level $s_{\max}$ with $2^{s_{\max}} > R$. Both $\log(R+1)$ and $s_{\max}$ are polynomially bounded in φ_{neg}, and $D \cap \Gamma_{s_{\max}} = \{\mathbf{0}\}$.*

Proof. Write $V = \ker H$. The rows of H and the $\boldsymbol{u}_i^\top$ span $\mathbb{R}^n$; otherwise a common-kernel vector would generate a line in D. Choose n independent rows, forming an invertible matrix M. Assign a bound $a_j = 0$ to a selected equation row and $a_j = \rho_i$ to a selected row from a two-sided inequality. Since $|(M\boldsymbol{d})_j| \leq a_j$ for $\boldsymbol{d} \in D$, take

$$R = \max\left\{1, \left\lceil \max_k \sum_j |(M^{-1})_{kj}| a_j \right\rceil\right\}.$$

Rational elimination and determinant bounds give polynomial running time and output bit length. The matrix M supplies a bound in the original coordinates. Choose $s_{\max}$ from the binary length of R. Every nonzero point of $2^{s_{\max}}\mathbb{Z}^n$ has a coordinate of magnitude at least $2^{s_{\max}} > R$, proving the terminal assertion. ☐

Starting with $D \cap \Gamma_{s_{\max}} = \{\mathbf{0}\}$, process levels $s_{\max} - 1, \ldots, 0$. At the start of level s, the invariant is

$$q(\boldsymbol{d}) \geq 0 \qquad (\boldsymbol{d} \in D \cap \Gamma_{s+1}). \tag{10}$$

A negative vector found at any level solves the original problem, since $\Gamma_s \subseteq \mathbb{Z}^n$.

4.2 The modulo-four refinement lemma

For $\boldsymbol{\delta} \in \{0, 1, 2, 3\}^n$ and $s \geq 0$, set

$$E_{s,\boldsymbol{\delta}} = D \cap (2^s\boldsymbol{\delta} + 2^{s+2}\mathbb{Z}^n), \tag{11}$$

$$K_{s,\boldsymbol{\delta}} = \{\boldsymbol{z} \in \mathbb{R}^n : 2^s(\boldsymbol{\delta} + 4\boldsymbol{z}) \in D\}, \tag{12}$$

$$h_{\boldsymbol{\delta}}(\boldsymbol{z}) = q(\boldsymbol{\delta} + 4\boldsymbol{z}) = 16\boldsymbol{z}^\top Q\boldsymbol{z} + 8(Q\boldsymbol{\delta})^\top \boldsymbol{z} + q(\boldsymbol{\delta}). \tag{13}$$

The set $E_{s,\boldsymbol{\delta}}$ contains displacement vectors, while $K_{s,\boldsymbol{\delta}}$ expresses the same problem in the optimizer's integer variable $\boldsymbol{z}$. Every $\boldsymbol{d} \in D \cap \Gamma_s$ has a unique representation $\boldsymbol{d} = 2^s(\boldsymbol{\delta} + 4\boldsymbol{z})$, with $\boldsymbol{\delta} \in \{0, 1, 2, 3\}^n$ and $\boldsymbol{z} \in \mathbb{Z}^n$. Thus the sets $E_{s,\boldsymbol{\delta}}$ partition $D \cap \Gamma_s$, and the affine map is a bijection from $K_{s,\boldsymbol{\delta}} \cap \mathbb{Z}^n$ onto $E_{s,\boldsymbol{\delta}}$. The optimizer detects empty classes, including those that miss V.

Lemma 4.2 (Nonnegative differences within a residue class)**.** *Assume* (10)*. On each $E_{s,\boldsymbol{\delta}}$, q is nonnegative on integer differences. Consequently $h_{\boldsymbol{\delta}}$ meets the hypothesis of* ConvicMin *on $K_{s,\boldsymbol{\delta}} \cap \mathbb{Z}^n$. At most 4^n calls either find a negative point in $D \cap \Gamma_s$ or certify nonnegativity throughout that set.*

Proof. Take $\boldsymbol{x}, \boldsymbol{y} \in E_{s,\boldsymbol{\delta}}$. Convexity and symmetry give

$$\frac{\boldsymbol{x} - \boldsymbol{y}}{2} = \frac{1}{2}\boldsymbol{x} + \frac{1}{2}(-\boldsymbol{y}) \in D.$$

Their common residue gives $\boldsymbol{x} - \boldsymbol{y} \in 2^{s+2}\mathbb{Z}^n$, and hence $(\boldsymbol{x} - \boldsymbol{y})/2 \in \Gamma_{s+1}$. Thus

$$\tfrac{1}{2}(E_{s,\boldsymbol{\delta}} - E_{s,\boldsymbol{\delta}}) \subseteq D \cap \Gamma_{s+1}, \qquad q(\boldsymbol{x} - \boldsymbol{y}) = 4q((\boldsymbol{x} - \boldsymbol{y})/2) \geq 0. \tag{14}$$

If $\boldsymbol{z}, \boldsymbol{w} \in K_{s,\boldsymbol{\delta}} \cap \mathbb{Z}^n$, the corresponding movement difference is $2^{s+2}(\boldsymbol{w}-\boldsymbol{z})$. Dividing its nonnegative quadratic value by 4^s gives

$$(\boldsymbol{w}-\boldsymbol{z})^{\mathsf{T}}(16Q)(\boldsymbol{w}-\boldsymbol{z}) \geq 0.$$

This is the hypothesis of ConvicMin for $h_{\boldsymbol{\delta}}$. The quadratic identity supplies the supporting inequalities, and Lemma 3.3 gives discrete convicity. Moreover,

$$q\big(2^s(\boldsymbol{\delta}+4\boldsymbol{z})\big) = 4^s h_{\boldsymbol{\delta}}(\boldsymbol{z}).$$

The positive factor preserves minimizers and signs. A negative minimum gives a witness in $D \cap \Gamma_s$. If every nonempty class has nonnegative minimum, the partition proves nonnegativity throughout $D \cap \Gamma_s$. □

The modulus four ensures that division by two leaves the difference on the coarser lattice Γ_{s+1}. A common residue modulo two would give only a half-difference on Γ_s, where nonnegativity is still to be established.

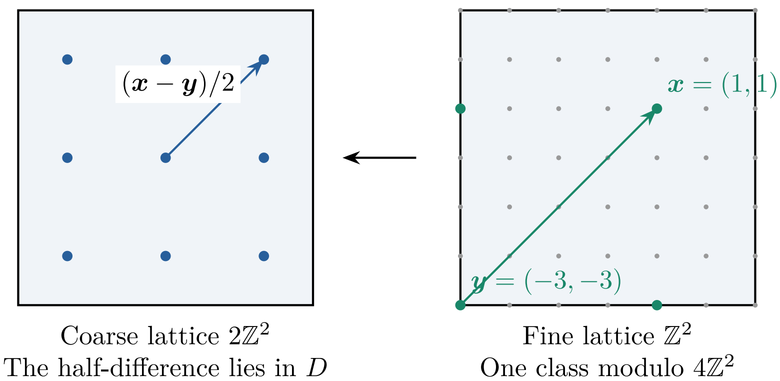


Figure 5: Refinement in $D = [-3,3]^2$. The green points have residue $(1,1)$ modulo four. Their half-difference $(2,2)$ lies in $D \cap 2\mathbb{Z}^2$, where nonnegativity has already been established.

4.3 Algorithm, example, and complexity

For $D = [-3,3]^2$ and $q(\boldsymbol{d}) = -d_1^2 + 2d_2^2$, choose $R = 3$, $s_{\max} = 2$. The terminal set $D \cap 4\mathbb{Z}^2$ contains only zero. At level $s = 1$, residue $\boldsymbol{\delta} = (1,0)$ gives the single feasible movement $(2,0)$. Its coordinate is $\boldsymbol{z} = (0,0)$, with $h_{(1,0)}(0) = -1$ and $q(2,0) = -4$. Processing this class gives the negative witness $(2,0)$. For $q(\boldsymbol{d}) = d_1^2 + d_2^2$, all residue-class minima are nonnegative, and the algorithm certifies $D \cap 2\mathbb{Z}^2$ and then $D \cap \mathbb{Z}^2$.

Proposition 4.3 (Negative-displacement procedure)**.** *Algorithm 1 is correct, uses at most $s_{\max}4^n$ optimizer calls, and runs in time*

$$2^{O(n\log(n+1))}(1+\varphi_{\text{neg}})^{O(1)}. \tag{15}$$

This bound holds for any number of two-sided inequalities in the explicit input.

Proof. Correctness. Lemma 4.1 initializes the invariant $q \geq 0$ on $D \cap \Gamma_{s+1}$. At each level, Lemma 4.2 establishes the hypothesis of every optimizer call. A negative result gives an integer witness in D. Otherwise, completing all residue classes proves nonnegativity on $D \cap \Gamma_s$ and supplies the invariant for the next level. Level zero proves the required conclusion.

Algorithm 1 $\textsc{Negative}(D, Q)$: lattice refinement with modulo-four classes

Input: Displacement data from Section 4.1.
Output: $\mathsf{NEG}(\boldsymbol{d})$ with $\boldsymbol{d} \in D \cap \mathbb{Z}^n$, $q(\boldsymbol{d}) < 0$, or $\mathsf{NONNEGATIVE}$.

if $n = 0$ **then**
  return $\mathsf{NONNEGATIVE}$
end if
Compute $R, s_{\max}$ by Lemma 4.1.
Certify $q \geq 0$ on $D \cap \Gamma_{s_{\max}} = \{\mathbf{0}\}$.
for $s = s_{\max} - 1, s_{\max} - 2, \ldots, 0$ **do**
  for each $\boldsymbol{\delta} \in \{0, 1, 2, 3\}^n$ **do**
    Form $K_{s,\boldsymbol{\delta}}, h_{\boldsymbol{\delta}}$ by (12)–(13).
    $\boldsymbol{z} \leftarrow \textsc{ConvicMin}(K_{s,\boldsymbol{\delta}}, h_{\boldsymbol{\delta}})$.
    if $\boldsymbol{z} \neq \mathsf{INFEASIBLE}$ and $h_{\boldsymbol{\delta}}(\boldsymbol{z}) < 0$ **then**
      return $\mathsf{NEG}\big(2^s(\boldsymbol{\delta} + 4\boldsymbol{z})\big)$
    end if
  end for
  Certify $q \geq 0$ on $D \cap \Gamma_s$.
end for
return $\mathsf{NONNEGATIVE}$

Encoding. Write $V = \ker H$. The coordinate problem has equations $H(\boldsymbol{\delta} + 4\boldsymbol{z}) = \mathbf{0}$ and inequalities

$$|\boldsymbol{u}_i^\mathsf{T}(\boldsymbol{\delta} + 4\boldsymbol{z})| \leq 2^{-s}\rho_i.$$

Since $s < s_{\max}$ is polynomially bounded in φ_{neg}, these descriptions have polynomial bit length: scaling a radius adds at most s bits, and each entry of $\boldsymbol{\delta}$ is at most three. The objective coefficients $16Q$, $8Q\boldsymbol{\delta}$, and $q(\boldsymbol{\delta})$ do not grow with s. For every real feasible $\boldsymbol{z}$,

$$\|\boldsymbol{z}\|_\infty \leq R/2^{s+2} + 3/4 \leq (R+3)/4.$$

Thus the integer box radius $R_{\text{cell}} = \lceil (R+3)/4 \rceil + 2$ suffices for every call and has polynomial encoding length. Proposition 3.7 bounds each call by $2^{O(n\log(n+1))}(1+\varphi_{\text{neg}})^{O(1)}$.

Total work. There are at most $s_{\max}4^n$ calls. The level count is polynomial in φ_{neg}, and 4^n is absorbed into the displayed dimension factor. Each call uses the optimizer of Proposition 3.7 directly, and mapping a witness back has polynomial bit cost. □

Relation to lattice scaling and reflection covers. Micciancio–Voulgaris [20, Section 3.3] refine scaled lattices using a Voronoi-cell procedure for closest vectors. This motivates the order of lattice levels here; the present refinement lemma and running-time bound follow from nonnegativity on integer differences and ConvicMin. The cover in Section 2 uses dyadic slack intervals to keep both reflections feasible. This geometry is related to the reflection sets of Cook–Hartmann–Kannan–McDiarmid [5], used by Dadush–Eisenbrand–Rothvoss [6, Section 5, Lemma 21]. The complete algorithm uses that cover and applies lattice refinement to each cell's negative-displacement problem.

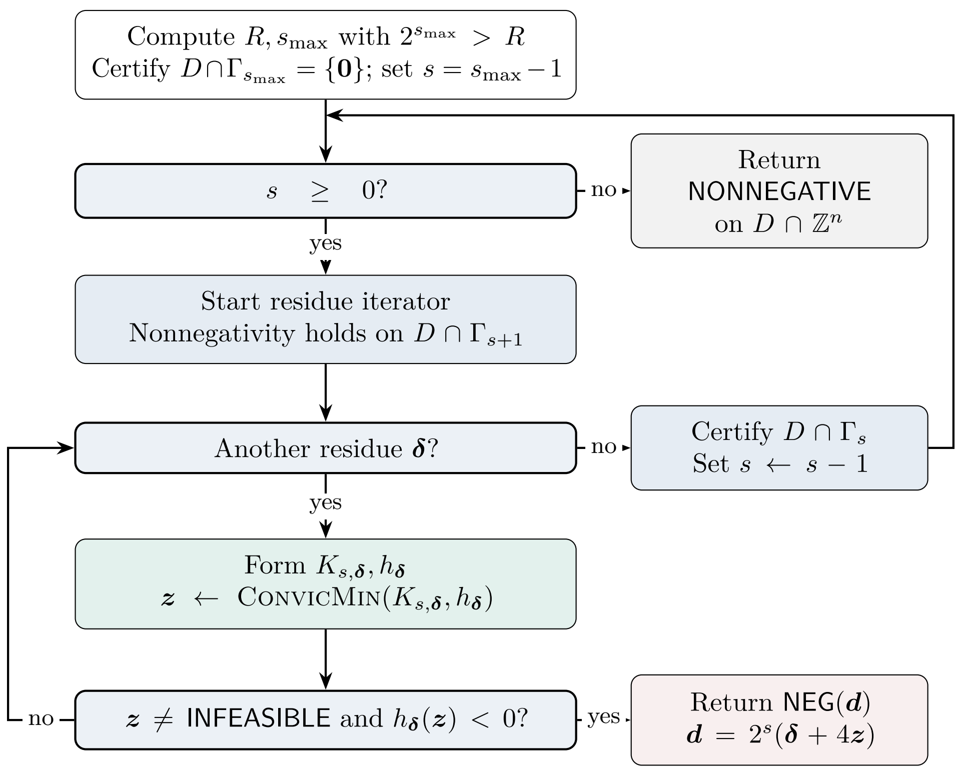


Figure 6: Algorithm 1 for $n \geq 1$. At each level, at most 4^n optimizer calls either return a negative integer displacement or establish nonnegativity on the finer lattice and advance to the next level.

4.4 Proof of the direct geometric solver

Algorithm 2 constructs the cover, tests each cell for a negative displacement, and optimizes over the surviving cells. Its control flow is shown in Figure 7.

Algorithm 2 $\textsc{DirectBounded}(K, h)$: cover, classify, and optimize

Input: A bounded rational polyhedron K and $h(\boldsymbol{x}) = \boldsymbol{x}^\top Q_h \boldsymbol{x} + \boldsymbol{c}_h^\top \boldsymbol{x} + \gamma_h$.
Output: An integer minimizer of h on K, or $\mathsf{INFEASIBLE}$.

Construct the Goemans–Rothvoß cover Π of $K \cap \mathbb{Z}^n$.
$\boldsymbol{x}_{\text{best}} \leftarrow \mathsf{INFEASIBLE}$; $v_{\text{best}} \leftarrow +\infty$.
for each $\pi \in \Pi$ **do**
  Construct its affine coordinates, cutoffs, and compatible cell tuples.
  for each tuple, with cell C and displacement set D_C **do**
    $a_C \leftarrow \textsc{Negative}(D_C, Q_h)$ {Refine the lattice chain for this displacement set.}
    if $a_C = \mathsf{NONNEGATIVE}$ **then**
      $\boldsymbol{x} \leftarrow \textsc{ConvicMin}(C, h)$.
      if $\boldsymbol{x} \neq \mathsf{INFEASIBLE}$ and $h(\boldsymbol{x}) < v_{\text{best}}$ **then**
        $\boldsymbol{x}_{\text{best}} \leftarrow \boldsymbol{x}$; $v_{\text{best}} \leftarrow h(\boldsymbol{x})$.
      end if
    else
      Discard C using the returned negative displacement.
    end if
  end for
end for
return $\boldsymbol{x}_{\text{best}}$

Each call to $\textsc{ConvicMin}$ uses the cell's integer-equivalent closed description and the sublevel separation implementation of Proposition 3.7. The bounded geometric data supply a containing ball.

Proof of Theorem 1.2. Algorithm 2 constructs the pairs (C, D_C) from Theorem 2.2. For each pair, apply Proposition 4.3 to D_C and q. Incorporating zero-width coordinates into the initial subspace leaves positive-width two-sided inequalities, as required by that proposition. A negative witness causes the algorithm to discard C; otherwise it calls $\textsc{ConvicMin}(C, f)$ with the closed integer-equivalent cell description. It returns the best point obtained, or integer infeasibility if no point is returned.

Every returned candidate is an integer point of P. If $P \cap \mathbb{Z}^n$ is nonempty, boundedness makes it finite, so it has a global minimizer $\boldsymbol{x}_{\text{opt}}$. Coverage places $\boldsymbol{x}_{\text{opt}}$ in some cell, and Proposition 3.1 excludes a negative displacement for that cell. Its integer differences therefore have nonnegative quadratic value, establishing the hypothesis of $\textsc{ConvicMin}$. The cell minimum is at most $f(\boldsymbol{x}_{\text{opt}})$ because it contains $\boldsymbol{x}_{\text{opt}}$, and at least $f(\boldsymbol{x}_{\text{opt}})$ by feasibility in P. The best candidate is consequently a global minimizer. If no candidate is returned, the same argument rules out a feasible point. If $P \cap \mathbb{Z}^n$ is empty, no cell can return an integer point. This proves both the optimization and infeasibility claims.

For the running time, each of the N parallelepipeds contributes at most $[p(n)(1 + \langle A, \boldsymbol{b}\rangle_{\max})]^n$ cells. Every displacement set is defined in a rational subspace by at most n two-sided inequalities, with entry encoding lengths $p(n)(1 + \langle A, \boldsymbol{b}\rangle_{\max})$. Its negative-displacement call costs at most $2^{O(n \log(n+1))}(1 + \varphi)^{O(1)}$, and the final cell-optimization call has no greater cost. Here the complete encoding of a displacement instance is polynomial in φ, by the geometric bounds in Theorem 2.2.

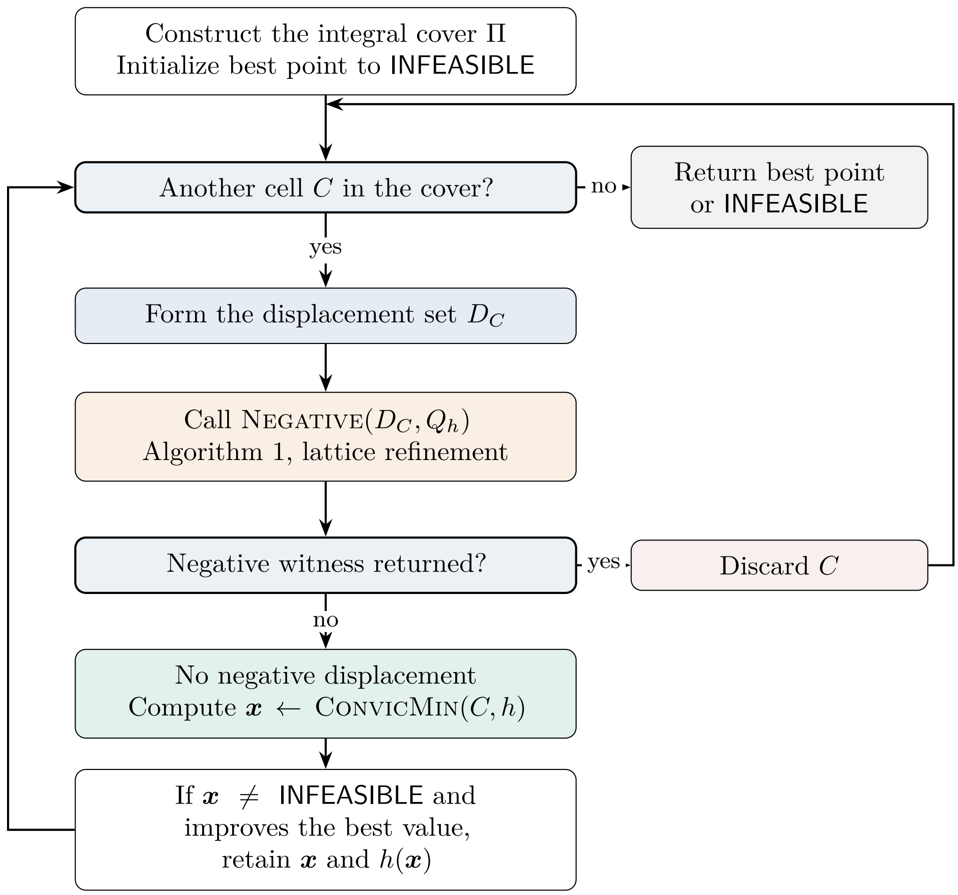


Figure 7: Algorithm 2 processes every compatible cell of every covering piece. A negative displacement excludes a cell from containing a global minimizer. Every other integer-nonempty cell contributes its exact minimum to the comparison.

Multiplying and absorbing $p(n)^{O(n)}$ into the dimension factor gives

$$T_{\mathrm{GR}} + 2^{O(n\log(n+1))} N(1+\langle A,\boldsymbol{b}\rangle_{\max})^n(1+\varphi)^{O(1)},$$

which is (2). Substituting the literature bounds for both the cover count and its construction time proves the first asserted bound. □

The work after cover construction is at most $2^{O(n\log(n+1))}(m+1)^n(1+\langle A,\boldsymbol{b}\rangle_{\max})^{2n}(1+\varphi)^{O(1)}$. The total also includes T_{GR}, whose cited bound uses an unspecified constant power of the cover bound. The resulting total encoding exponent is therefore stated as $O(n)$.

5 Localization and dependence on coefficient encoding

The direct solver's bound contains $(1+\langle A,\boldsymbol{b}\rangle_{\max})^{O(n)}$. To prove Theorem 1.1, we construct finitely many bounded translated problems whose geometry has encoding length controlled by A, Q, and show that one preserves the global minimum. The construction has three steps:

1. Construct regions R_I covering all global integer minimizers.
2. For each region containing an integer point $\boldsymbol{x}_0$, show that every value of f on $R_I \cap \mathbb{Z}^n$ is attained near $\boldsymbol{x}_0$.
3. Translate by $\boldsymbol{x}_0$ and remove inequalities that are redundant on the resulting bounded box.

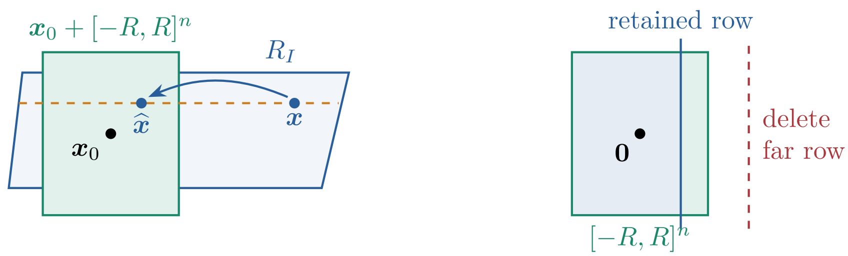
Throughout this section, a *structural encoding bound* is $p(n)\varphi_{A,Q}$ for a fixed polynomial p, where $\varphi_{A,Q} = 1 + \langle A, Q\rangle_{\max}$. It bounds bit length independently of $\boldsymbol{b}, \boldsymbol{c}, \gamma$. The reference point may depend on these data; the radius about it will have a structural encoding bound.

Where the cover theorem is used. The proof of Theorem 1.2 invokes Theorem 2.2 directly. Here we apply that same solver to each translated region after removing large offsets. Thus the order is localization, translation and row deletion, then cover construction and cell optimization. We never need to construct the cover of the original P to obtain Theorem 1.1. Figure 8 illustrates the reduction for one integer-nonempty region.

Normalization. Normalize each row of A by denominators in A alone and floor its scaled right-hand side, preserving integer feasibility. Scale the objective by a positive common denominator of Q alone, making Q integral. The resulting entries of A, Q have encoding length $p(n)\varphi_{A,Q}$. Right-hand sides and affine objective coefficients are charged to the full input length φ.

Lemma 5.1 (Small integral vectors spanning a kernel)**.** *Let $T \in \mathbb{Z}^{a\times n}$ have rank r, and suppose $|T_{ij}| \le U$ with $U \ge 1$. One can compute a real basis of* $\ker T$ *consisting of integer vectors whose coordinates have absolute value at most $r!U^r$. The computation has polynomial bit cost in the input length. In particular, each output coordinate has $O(1 + n\log(n+1) + n\log U)$ bits.*

Proof. For $r = 0$, take the standard coordinate vectors; for $r = n$, the kernel basis is empty. Otherwise select r independent rows of T, which have the same kernel, and apply the adjugate construction of Ari and Hildebrand [1, Lemma 2.1]. Its coordinates are signed minors of the selected matrix. Every minor has order at most r and magnitude at most $r!U^r$. The explicit construction uses rational elimination and determinant computations, giving polynomial bit cost and the stated encoding bound. This is a basis over $\mathbb{R}$, not necessarily a basis of the integer kernel lattice. □



Preserve values near a reference point
$F\widehat{\boldsymbol{x}} = F\boldsymbol{x}, \quad f(\widehat{\boldsymbol{x}}) = f(\boldsymbol{x})$

Translate and bound the remaining data
$\boldsymbol{w} = \boldsymbol{x} - \boldsymbol{x}_0$

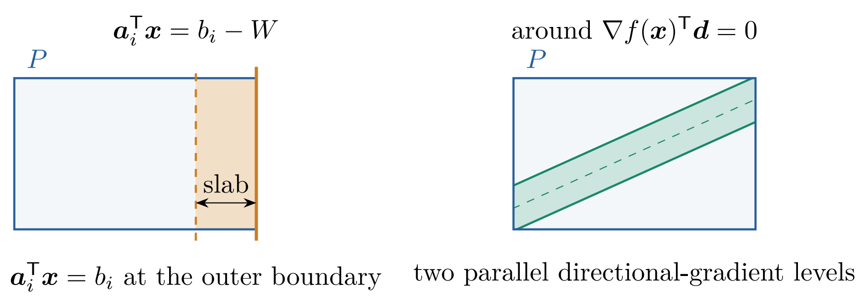
Figure 8: Localization for one region R_I (schematic). Left: every integer point $\boldsymbol{x}$ in the region has an integer representative $\widehat{\boldsymbol{x}}$ in the displayed box on the same F-fiber (dashed), with the same objective value. The box need not contain the original point or the entire region. Right: after translation, a row $\boldsymbol{M}_i\boldsymbol{w} \le \eta_i$ is redundant on the box if $\eta_i \ge R\|\boldsymbol{M}_i\|_1$. Retained rows have $0 \le \eta_i < R\|\boldsymbol{M}_i\|_1$, so their encoding is controlled by A, Q. The cover theorem is applied to the resulting bounded polyhedron. The radius has structurally bounded *bit length*, not necessarily small numerical value.

5.1 Regions containing all global minimizers

The geometry follows the boundary/gradient principle of Lokshtanov [18, Lemmas 2–5], developed through curvature batching by Ari and Hildebrand [1, Lemma 2.3 and Section 3]. For an integer direction $\boldsymbol{d}$, optimality and feasibility of both $\boldsymbol{x} \pm \boldsymbol{d}$ imply

$$|\nabla f(\boldsymbol{x})^{\mathsf{T}}\boldsymbol{d}| \le q(\boldsymbol{d}).$$

If this inequality fails at an optimum, an improving move must be blocked by a constraint, placing the point in a boundary slab. We retain whole slabs instead of branching over all integer values within them; their numerical widths may be large even when their encoding lengths are small. Figure 9 illustrates both types. The later Cook–Gerards–Schrijver–Tardos sensitivity argument then gives nearby representatives without enumerating those levels.



Boundary slab
$0 \le b_i - \boldsymbol{a}_i^{\mathsf{T}}\boldsymbol{x} \le W$

Directional-gradient slab
$|\nabla f(\boldsymbol{x})^{\mathsf{T}}\boldsymbol{d}| \le q(\boldsymbol{d})$

Figure 9: The two kinds of slabs used to construct R_I (schematic). Left: a constraint slack is bounded by W. Right: for $q(\boldsymbol{d}) > 0$, a directional derivative lies between $-q(\boldsymbol{d})$ and $q(\boldsymbol{d})$; the dashed line is its zero level. This is a slab because $\nabla f(\boldsymbol{x})^{\mathsf{T}}\boldsymbol{d}$ is affine in $\boldsymbol{x}$. For zero curvature it becomes an equality (possibly vacuous); for negative curvature the condition is impossible. A region intersects P with several such conditions. The widths are measured in slack or derivative values, not Euclidean distance, and we keep the full slabs rather than enumerate their integer levels.

Write $\boldsymbol{a}_i^\mathsf{T}$ for row i of A. For each independent row subset I, put $V = \ker A_I$. We will test a finite family of integer directions in V. An improving move from a global minimizer must be blocked by another constraint; adding that constraint will increase the rank of A_I. This proves coverage by the regions below.

Choose integral columns Y forming a real basis of V, and integral columns Z forming a real basis of

$$N = \{\boldsymbol{z} \in V : Y^\mathsf{T} Q \boldsymbol{z} = \boldsymbol{0}\}.$$

The columns of Y, Z are bases over $\mathbb{R}$; they need not generate the corresponding integer lattices. Apply Lemma 5.1 first to A_I to obtain Y, and then to $(A_I; Y^\mathsf{T} Q)$ to obtain Z. Let U be the maximum of one and the absolute entries of the normalized A, Q. The entries of Y have magnitude at most $B_Y = n!U^n$. The second matrix has entries of magnitude at most nUB_Y, so those of Z have magnitude at most $B_Z = n!(nUB_Y)^n$. Since $\log U \le p(n)\varphi_{A,Q}$, both bases have entry encoding length $p(n)\varphi_{A,Q}$. These bounds are uniform over I and independent of $\boldsymbol{b}, \boldsymbol{c}, \gamma$. The defining condition for N is $\boldsymbol{y}^\mathsf{T} Q \boldsymbol{z} = 0$ for every $\boldsymbol{y} \in V$. In particular, if $\boldsymbol{z} = Y\boldsymbol{\lambda} \in N$, then $q(\boldsymbol{z}) = \boldsymbol{\lambda}^\mathsf{T} Y^\mathsf{T} Q \boldsymbol{z} = \boldsymbol{0}$; the converse can fail for indefinite forms. These directions will preserve objective values in the region. Choose an integer $W \ge 1$ strictly greater than $\|A\boldsymbol{d}\|_\infty$ for every column $\boldsymbol{d}$ of every such Y, Z. This maximum is computable over at most $\sum_{j=0}^{n} \binom{m}{j}$ subsets and has encoding length $p(n)\varphi_{A,Q}$. Alternatively, the single explicit choice $W = 1 + nU \max(B_Y, B_Z)$ satisfies the same requirement without computing that maximum.

For each independent row subset I, let $\mathcal{D}_I$ be the finite family of columns of Y and Z. With the preceding choice of W, define the localization region R_I by the inequalities of P together with

$$\begin{aligned} 0 \le b_i - \boldsymbol{a}_i^\mathsf{T} \boldsymbol{x} &\le W && (i \in I), \\ |\nabla f(\boldsymbol{x})^\mathsf{T} \boldsymbol{d}| &\le q(\boldsymbol{d}) && (\boldsymbol{d} \in \mathcal{D}_I). \end{aligned}$$

The normals depend only on A, Q and have entry encoding length $p(n)\varphi_{A,Q}$; $\boldsymbol{b}, \boldsymbol{c}$ occur only on the right. Floor weak right-hand sides to obtain integral inequalities with the same integer points. If an equality has an integral normal and a nonintegral right-hand side, encode the empty integer region by adding $0 \le -1$ to the constraints of P. Otherwise replace the equality by two inequalities. These operations preserve the integer points and can only restrict the real region.

The first family bounds the selected-row slacks by W. The second rules out improvement along either sign of each prescribed integer direction, since

$$\min\{f(\boldsymbol{x} + \boldsymbol{d}) - f(\boldsymbol{x}), f(\boldsymbol{x} - \boldsymbol{d}) - f(\boldsymbol{x})\} = q(\boldsymbol{d}) - |\nabla f(\boldsymbol{x})^\mathsf{T} \boldsymbol{d}|.$$

For each column $\boldsymbol{z}$ of Z, $q(\boldsymbol{z}) = 0$, so the directional inequality gives $\nabla f(\boldsymbol{x})^\mathsf{T} \boldsymbol{z} = 0$. By linearity, the derivative vanishes in every direction in N.

The next proof adapts the blocking-row argument of Lokshtanov [18, Lemma 2] and Ari and Hildebrand [1, Lemma 2.3] to the family $\mathcal{D}_I$ and the uniform slack bound W. We include the adaptation because we retain slabs and claim coverage of every global minimizer.

Lemma 5.2 (Coverage by localization regions)**.** *Every global integer minimizer belongs to some R_I.*

Proof. Fix a global integer minimizer $\boldsymbol{x}$. Starting with $I = \varnothing$, we enlarge I while maintaining independence and $0 \le b_i - \boldsymbol{a}_i^\mathsf{T} \boldsymbol{x} \le W$ for every $i \in I$. Construct Y, Z for the current I. If all directional inequalities hold, then $\boldsymbol{x} \in R_I$.

If a directional inequality fails for $\boldsymbol{v} \in \mathcal{D}_I$, then $q(\boldsymbol{v}) - |\nabla f(\boldsymbol{x})^\mathsf{T} \boldsymbol{v}| < 0$, so one of the integer directions $\boldsymbol{d} = \boldsymbol{v}$ or $\boldsymbol{d} = -\boldsymbol{v}$ strictly improves f at $\boldsymbol{x}$. This also covers $q(\boldsymbol{v}) < 0$ and a nonzero derivative along a column of Z. Every such $\boldsymbol{d}$ lies in $\ker A_I$ and satisfies $\|A\boldsymbol{d}\|_\infty < W$.

Since $\boldsymbol{x}$ is globally optimal, the improving integer point $\boldsymbol{x}+\boldsymbol{d}$ cannot be feasible. Some constraint row j therefore satisfies

$$\boldsymbol{a}_j^\top(\boldsymbol{x}+\boldsymbol{d}) > b_j \geq \boldsymbol{a}_j^\top \boldsymbol{x}, \qquad 0 \leq b_j - \boldsymbol{a}_j^\top \boldsymbol{x} < \boldsymbol{a}_j^\top \boldsymbol{d} \leq \|A\boldsymbol{d}\|_\infty < W.$$

Every row in the span of A_I annihilates $\boldsymbol{d} \in \ker A_I$, whereas $\boldsymbol{a}_j^\top \boldsymbol{d} > 0$. Adding row j therefore preserves independence and increases rank. Its slack satisfies the required bound, and the previous slack bounds remain valid at the fixed point $\boldsymbol{x}$.

Recompute the bases and repeat. There are at most n additions. At rank n, the kernel and both direction families are empty, so the directional conditions hold. The process therefore terminates with $\boldsymbol{x} \in R_I$, and the integer-equivalent rounding preserves $\boldsymbol{x}$. $\square$

5.2 A bounded representative of every region value

For each region containing an integer point, choose an integer reference point $\boldsymbol{x}_0$. We will show that every value of f on $R_I \cap \mathbb{Z}^n$ is attained at an integer point within ℓ_∞ distance R of $\boldsymbol{x}_0$, for a radius R with encoding length bounded by $p(n)\varphi_{A,Q}$.

Finding a reference point. The region $R_I \subseteq P$ is bounded. Call $\textsc{ConvicMin}(R_I, 0)$: the zero objective meets the optimizer's hypothesis on every integer domain. The call either returns $\boldsymbol{x}_0 \in R_I \cap \mathbb{Z}^n$ or certifies integer emptiness. A containing ball is obtained from the bounded rational description by vertex-coordinate determinant bounds, with polynomial full-input encoding length, as in Proposition 3.7. This preliminary call is charged to the full input length. Fix a returned reference point $\boldsymbol{x}_0$.

The map below collects the selected boundary coordinates and all directional-gradient coordinates. Its kernel is the flat subspace from the batching argument of Ari and Hildebrand [1, Lemma 3.4 and Theorem 3.5]: fixing these coordinates removes the quadratic part along each fiber. Corollary 2.4 of that paper gives an affine residual objective. Here the additional directional conditions for Z make that affine objective constant on each fiber within R_I. We prove this extra assertion and the required width bounds explicitly.

Lemma 5.3 (Objective values determined by linear coordinates)**.** *Set*

$$F = \begin{pmatrix} A_I \\ 2Y^\top Q \end{pmatrix}, \qquad W_F = \max(\{W\} \cup \{2q(\boldsymbol{y}) : \boldsymbol{y} \textit{ a column of } Y\}).$$

Then $\ker F = N$*, and* W_F *has a structural encoding bound. For all real* $\boldsymbol{x} \in R_I$*,*

$$\|F(\boldsymbol{x} - \boldsymbol{x}_0)\|_\infty \leq W_F.$$

If $\boldsymbol{x}, \widehat{\boldsymbol{x}} \in R_I$ *and* $F\widehat{\boldsymbol{x}} = F\boldsymbol{x}$*, then* $f(\widehat{\boldsymbol{x}}) = f(\boldsymbol{x})$*.*

Proof. Kernel. The definition gives directly

$$\ker F = \{\boldsymbol{z} : A_I \boldsymbol{z} = 0,\ Y^\top Q\boldsymbol{z} = \boldsymbol{0}\} = N.$$

The rows of F may be dependent. *Widths.* For a row in A_I, the defining slack lies in $[0, W]$, so its value at $\boldsymbol{x}$ differs from its value at $\boldsymbol{x}_0$ by at most W. For a column $\boldsymbol{y}$ of Y, the directional inequality is equivalent to

$$-q(\boldsymbol{y}) - \boldsymbol{c}^\top \boldsymbol{y} \leq 2\boldsymbol{y}^\top Q\boldsymbol{x} \leq q(\boldsymbol{y}) - \boldsymbol{c}^\top \boldsymbol{y}.$$

Nonemptiness forces $q(\boldsymbol{y}) \geq 0$, and the interval has width $2q(\boldsymbol{y})$, independently of $\boldsymbol{c}^\mathsf{T}\boldsymbol{y}$. Each row of F therefore varies by at most W_F. The construction of Y and the bound on Q give the structural encoding bound for W_F.

Value preservation. Put $\boldsymbol{z} = \widehat{\boldsymbol{x}} - \boldsymbol{x}$. The equality $F\widehat{\boldsymbol{x}} = F\boldsymbol{x}$ puts $\boldsymbol{z}$ in N, so $q(\boldsymbol{z}) = 0$. At $\boldsymbol{x}$, the directional derivative vanishes along every column of Z, and hence along every vector in their real span N. Thus $\nabla f(\boldsymbol{x})^\mathsf{T}\boldsymbol{z} = 0$, and expansion gives

$$f(\widehat{\boldsymbol{x}}) - f(\boldsymbol{x}) = \nabla f(\boldsymbol{x})^\mathsf{T}\boldsymbol{z} + q(\boldsymbol{z}) = 0.$$

□

Thus fixing $F\boldsymbol{x}$ within R_I fixes the objective value. The next lemma shows that each value of $F\boldsymbol{x}$ attained by an integer point is also attained by an integer point near $\boldsymbol{x}_0$.

Lemma 5.4 (Localization from integer sensitivity)**.** *Let $S = \{\boldsymbol{x} : M\boldsymbol{x} \leq \boldsymbol{r}\}$, with integral $M, \boldsymbol{r}$, and let F be integral. Suppose $\boldsymbol{x}_0 \in S \cap \mathbb{Z}^n$ and $\|F(\boldsymbol{x} - \boldsymbol{x}_0)\|_\infty \leq W_F$ for all $\boldsymbol{x} \in S$. Let G be the matrix obtained by stacking $M, F, -F$. Every $\boldsymbol{x} \in S \cap \mathbb{Z}^n$ has a representative $\widehat{\boldsymbol{x}} \in S \cap \mathbb{Z}^n$ such that*

$$F\widehat{\boldsymbol{x}} = F\boldsymbol{x}, \qquad \|\widehat{\boldsymbol{x}} - \boldsymbol{x}_0\|_\infty \leq n\Delta(G)(W_F + 2).$$

Here $\Delta(G)$ is the maximum absolute square subdeterminant, including one. The set S need not be bounded.

Proof. Fix $\boldsymbol{x} \in S \cap \mathbb{Z}^n$ and compare the two integer systems

$$\begin{aligned} M\boldsymbol{u} \leq \boldsymbol{r}, \quad F\boldsymbol{u} = F\boldsymbol{x}_0, \quad \boldsymbol{u} \in \mathbb{Z}^n, \\ M\boldsymbol{u} \leq \boldsymbol{r}, \quad F\boldsymbol{u} = F\boldsymbol{x}, \quad \boldsymbol{u} \in \mathbb{Z}^n. \end{aligned} \tag{16}$$

Their common inequality matrix is $G = (M; F; -F)$, and their right-hand sides are respectively

$$\boldsymbol{h}_0 = \begin{pmatrix} \boldsymbol{r} \\ F\boldsymbol{x}_0 \\ -F\boldsymbol{x}_0 \end{pmatrix}, \qquad \boldsymbol{h}_{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{r} \\ F\boldsymbol{x} \\ -F\boldsymbol{x} \end{pmatrix}.$$

Thus $\|\boldsymbol{h}_{\boldsymbol{x}} - \boldsymbol{h}_0\|_\infty = \|F(\boldsymbol{x} - \boldsymbol{x}_0)\|_\infty \leq W_F$. The points $\boldsymbol{x}_0$ and $\boldsymbol{x}$ establish integer feasibility of the two systems. Give both the zero objective. Every feasible point is optimal, so both continuous and integer optima are attained even when these sets are unbounded. Cook–Gerards–Schrijver–Tardos [4, Theorem 5(ii)] now gives an integer point $\widehat{\boldsymbol{x}}$ in the second system within $n\Delta(G)(W_F + 2)$ of the specified optimum $\boldsymbol{x}_0$ of the first. This is the required representative. The common block $\boldsymbol{r}$ cancels from the perturbation bound. □

Apply the lemma to $S = R_I$ and the matrix F of Lemma 5.3. Since $F\widehat{\boldsymbol{x}} = F\boldsymbol{x}$, the representative has the same objective value as $\boldsymbol{x}$. Set $H = \max\{1, \max_{i,j} |(G_I)_{ij}|\}$ for $G_I = (M; F; -F)$. Every square minor has order at most n, so $\Delta(G_I) \leq n!H^n$. We may therefore use

$$R = n\, n! H^n (W_F + 2).$$

Its encoding length is $p(n)\varphi_{A,Q}$, since the entries of G_I and W_F have structural encoding bounds. Hence every value of f on $R_I \cap \mathbb{Z}^n$ is attained on $R_I \cap (\boldsymbol{x}_0 + [-R, R]^n) \cap \mathbb{Z}^n$. The construction computes R directly from these bounds.

5.3 Removing the large offsets

The deletion step is also used in the integer-hull candidate construction of Ari and Hildebrand [1, Proposition 7.3], where proximity localizes integer-hull vertices near vertices of the continuous relaxation. For a general indefinite quadratic, an optimum need not be an integer-hull vertex, so we instead use the value-preserving region localization above. We record the translated version of the deletion argument to make its encoding consequence explicit.

Lemma 5.5 (Translation and deletion of irrelevant rows)**.** *Translate $\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{w}$ and restrict $\boldsymbol{w}$ to $[-R, R]^n$, with the radius R above. Write $\boldsymbol{M}_i$ for row i of M. Delete each row whose translated right-hand side is at least $R\|\boldsymbol{M}_i\|_1$, considering only rows inherited from M. Retain all bounding-box inequalities $-R \leq w_j \leq R$. The resulting bounded polyhedron has the same feasible points in this box, and all its defining coefficients have encoding length $p(n)\varphi_{A,Q}$. The translated objective is*

$$\boldsymbol{w}^\mathsf{T} Q \boldsymbol{w} + \nabla f(\boldsymbol{x}_0)^\mathsf{T} \boldsymbol{w} + f(\boldsymbol{x}_0).$$

Its quadratic matrix is unchanged; its affine coefficients have polynomial full-input encoding length.

Proof. The translated right-hand side $\eta_i = r_i - \boldsymbol{M}_i \boldsymbol{x}_0$ is a nonnegative integer. For every $\boldsymbol{w} \in [-R, R]^n$, the triangle inequality gives

$$\boldsymbol{M}_i \boldsymbol{w} \leq |\boldsymbol{M}_i \boldsymbol{w}| \leq R\|\boldsymbol{M}_i\|_1.$$

Each row with $\eta_i \geq R\|\boldsymbol{M}_i\|_1$ therefore holds throughout the box and may be deleted. Every retained row satisfies $0 \leq \eta_i < R\|\boldsymbol{M}_i\|_1$. Its right-hand side therefore has a structural encoding bound, as do its matrix entries and the added box inequalities. Expansion of $f(\boldsymbol{x}_0 + \boldsymbol{w})$ gives the displayed objective. The point $\boldsymbol{x}_0$ has polynomial full-input encoding length, so exact arithmetic gives the same bound for the affine coefficients. The bounded solver's dependence on these coefficients is polynomial in their encoding length. $\square$

For example, on $|w| \leq R$, the row $2w \leq 10^{100}$ is redundant if $10^{100} \geq 2R$. Otherwise its right-hand side is less than $2R$, giving the required encoding bound.

Algorithm 3 compares candidates from different regions by translating them back and evaluating the original objective f.

5.4 The complete bounded-IQP algorithm

Algorithm 3 constructs and localizes a region for each independent constraint-row subset, then applies Algorithm 2 to its translated problem. The bounded solver in turn processes covering pieces and cells. Figure 10 gives the outer algorithm's control flow.

The feasibility call meets the hypothesis of ConvicMin because the zero objective is nonnegative on every integer difference. Cell-optimization calls obtain this hypothesis from a preceding NONNEGATIVE result, and calls within lattice refinement obtain it from the coarser lattice's nonnegativity invariant. The preliminary feasibility call uses the full input encoding; the radius R and the translated geometry have structural encoding bounds.

Algorithm 3 $\textsc{BoundedIQP}(P, f)$: bounded IQP with separated encoding

Input: Bounded $P = \{\boldsymbol{x} : A\boldsymbol{x} \le \boldsymbol{b}\}$ and rational $f(\boldsymbol{x}) = \boldsymbol{x}^\mathsf{T} Q \boldsymbol{x} + \boldsymbol{c}^\mathsf{T}\boldsymbol{x} + \gamma$.
Output: An exact integer minimizer of the original f, or $\mathsf{INFEASIBLE}$.

Normalize constraint rows and scale the objective positively to obtain $\widetilde{P}, \widetilde{f}$.
Enumerate independent row subsets I of $\widetilde{A}$.
Construct real bases Y_I, Z_I consisting of integer columns, and the common structural bound W.
$\boldsymbol{x}_{\text{best}} \leftarrow \mathsf{INFEASIBLE}$; $v_{\text{best}} \leftarrow +\infty$.
for each independent subset I **do**
  Form the integer-equivalent localization region $R_I = \{\boldsymbol{x} : M_I\boldsymbol{x} \le \boldsymbol{r}_I\}$.
  $\boldsymbol{x}_0 \leftarrow \textsc{ConvicMin}(R_I, 0)$, using its bounded rational description.
  if $\boldsymbol{x}_0 \ne \mathsf{INFEASIBLE}$ **then**
    Set $F = (\widetilde{A}_I; 2Y_I^\mathsf{T}\widetilde{Q})$.
    Compute the structural bounds W_F, H of the localization construction.
    $R \leftarrow n\, n! H^n (W_F + 2)$.
    $K_I \leftarrow \{\boldsymbol{w} : M_I\boldsymbol{w} \le \boldsymbol{r}_I - M_I\boldsymbol{x}_0,\ \|\boldsymbol{w}\|_\infty \le R\}$.
    Delete only inherited M_I rows with right-hand side at least $R\sum_j |(M_I)_{ij}|$; retain the box.
    $g_I(\boldsymbol{w}) \leftarrow \widetilde{f}(\boldsymbol{x}_0 + \boldsymbol{w})$, retaining its constant term.
    $\boldsymbol{w} \leftarrow \textsc{DirectBounded}(K_I, g_I)$.
    if $\boldsymbol{w} \ne \mathsf{INFEASIBLE}$ **then**
      $\boldsymbol{x} \leftarrow \boldsymbol{x}_0 + \boldsymbol{w}$.
      if $f(\boldsymbol{x}) < v_{\text{best}}$ **then**
        $\boldsymbol{x}_{\text{best}} \leftarrow \boldsymbol{x}$; $v_{\text{best}} \leftarrow f(\boldsymbol{x})$.
      end if
    end if
  end if
end for
return $\boldsymbol{x}_{\text{best}}$

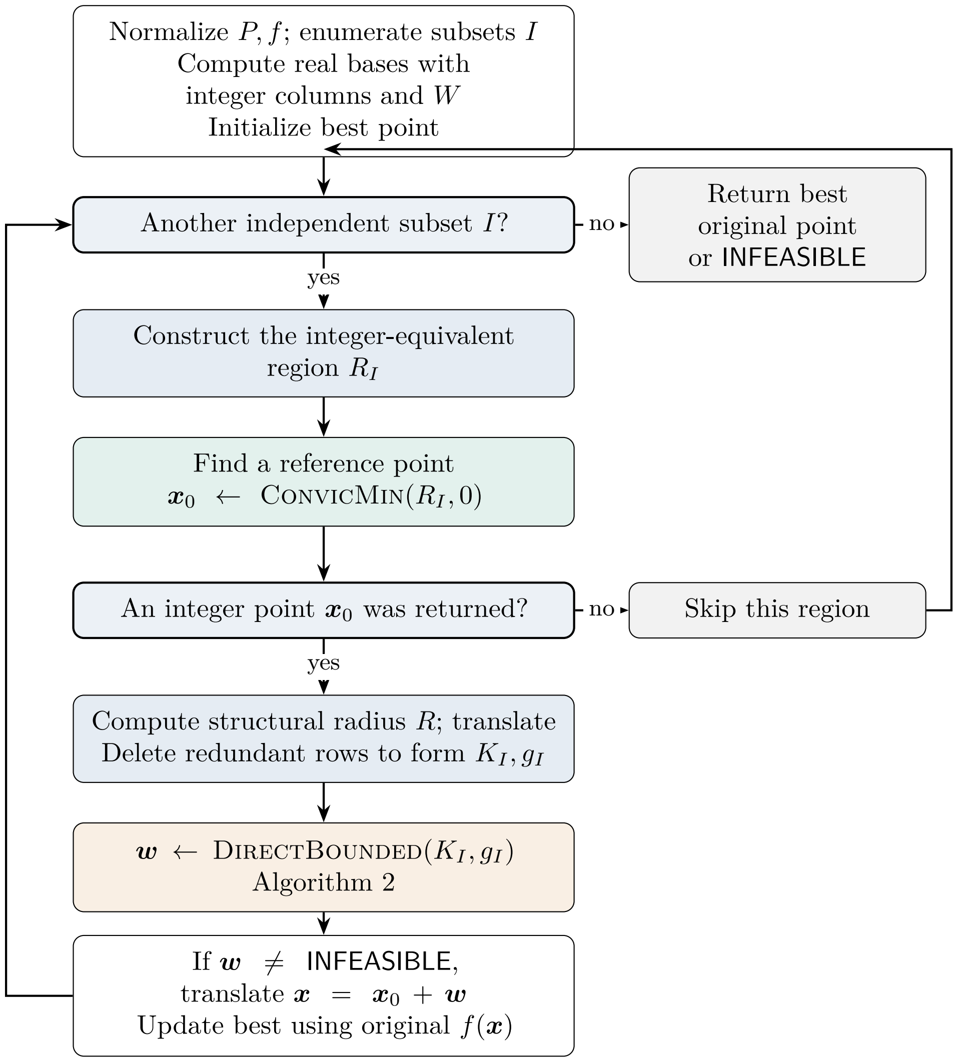


Figure 10: Algorithm 3 constructs one region per independent row subset. Each integer-nonempty region supplies a reference point and a bounded translated problem. Returned points are translated back and compared using the original objective, including its constant term.

5.5 Proof of the main theorem

We now combine coverage, preservation of objective values, and the bounded solver to prove Theorem 1.1.

Proof of Theorem 1.1 (main theorem). Construction. Algorithm 3 first normalizes the rows and objective as described above. It enumerates all independent subsets I of the normalized rows and constructs R_I. The zero-objective call either proves integer emptiness or returns $\boldsymbol{x}_0 \in R_I \cap \mathbb{Z}^n$. For each nonempty region, construct R, translate by $\boldsymbol{x}_0$, delete inherited rows redundant on the box, and apply Theorem 1.2. Translate the returned points back and evaluate them using the original objective f. Return a candidate of least value, or integer infeasibility if no candidate was obtained.

Preservation of an optimum. If the original integer feasible set is nonempty, boundedness of P makes it finite. Let $\boldsymbol{x}_{\text{opt}}$ be one of its minimizers. Lemma 5.2 places $\boldsymbol{x}_{\text{opt}}$ in some R_I. That region passes the integer-feasibility test. Lemmas 5.4 and 5.3 give $\widehat{\boldsymbol{x}} \in R_I \cap (\boldsymbol{x}_0 + [-R, R]^n) \cap \mathbb{Z}^n$ with $f(\widehat{\boldsymbol{x}}) = f(\boldsymbol{x}_{\text{opt}})$. Translation is a bijection between the integer points of this localized set and its translated formulation; deleting rows that are redundant on the box changes neither set, by Lemma 5.5. Hence the corresponding bounded problem contains a point with the global optimum value. Its solver returns a point of no greater value. Every candidate maps back to a feasible point of the original problem, so no candidate can have smaller value than $f(\boldsymbol{x}_{\text{opt}})$. Taking the best candidate therefore returns an exact minimizer.

Infeasibility. If the original instance is integer-infeasible, every $R_I \subseteq P$ is integer-empty and no candidate is returned. In the other direction, a nonempty original integer feasible set has a minimizer, and the preceding argument produces a candidate. Therefore the algorithm's infeasibility answer is correct in both directions.

Separated running time. There are at most $\sum_{j=0}^{n} \binom{m}{j} \leq (n+1)(m+1)^n$ subsets. The basis constructions, structural constants, feasibility calls, and translations cost at most $2^{O(n \log(n+1))}(m + 1)^{O(n)}(1+\varphi)^{O(1)}$ in total. Each localized instance has $m + O(n)$ inequalities with maximum entry encoding length at most $p(n)\varphi_{A,Q}$ and full input length polynomial in φ and n. Theorem 1.2 therefore solves it within

$$2^{O(n \log(n+1))}(m+1)^{O(n)}\varphi_{A,Q}^{O(n)}(1+\varphi)^{O(1)}.$$

Multiplication by the number of regions preserves this form. The constants hidden in the input polynomial remain independent of n. Thus $\boldsymbol{b}, \boldsymbol{c}, \gamma$ affect feasibility, translations, and objective arithmetic only through the full-input polynomial; the dimension-dependent encoding factor uses $\varphi_{A,Q} = 1 + \langle A, Q\rangle_{\max}$. □

6 Oracle variants and the unbounded reduction

Separation versus comparison. The quadratic support identity implements the integer-query separation oracle of Theorem 3.5, giving the main theorem's dimension factor $2^{O(n \log(n+1))}$. Replacing the cell optimizer by Proposition A.4 gives the same IQP algorithm and encoding separation with dimension factor $2^{O(n^2 \log(n+1))}$. The input still consists of explicitly given rational coefficients $A, \boldsymbol{b}, Q, \boldsymbol{c}, \gamma$. The algorithm uses these coefficients to construct the displacement and localization problems and to answer the optimizer's comparisons. The comparison restriction applies to the discrete-convic minimization routine. Both bounds allow f to be nonconvex over $\mathbb{R}^n$.

Unbounded feasible regions. Ari and Hildebrand [1, Theorem 5.1] give an $(m+n)^{O(n)}\,\mathrm{poly}(\varphi)$ procedure distinguishing integer infeasibility, unboundedness below, and a finite optimal value. In the unbounded case it supplies an integer basepoint and recession ray. Its finite-value outcome reduces the remaining task to optimization with a finite optimum. The recession analysis and certificates are supplied by their work on integer and mixed-integer quadratic programming.

When P is unbounded but a global minimizer exists, the region and localization arguments in Section 5 still apply and reduce optimization to bounded translated regions. To obtain a reference point, first restrict an unbounded R_I to a finite box. Cook–Gerards–Schrijver–Tardos [4, Corollary 3] supplies the radius $(n+1)\Delta((M_I \mid \boldsymbol{r}_I))$ whenever the integral system is feasible. It suffices to use the computable upper bound $(n+1)(n+1)!U_0^{n+1}$, where U_0 is the maximum of one and the absolute entries of $(M_I \mid \boldsymbol{r}_I)$. This radius has polynomial full-input encoding length and is used only for the reference-point call; the later structural radius is unchanged. Rational quadratic values on integer points belong to $(1/D)\mathbb{Z}$ for a fixed positive integer D, so a finite infimum on a nonempty integer feasible set is attained. The cited classification, followed by localization and the bounded solver, therefore extends the algorithm to arbitrary rational polyhedra. This extension uses the recession result of Ari and Hildebrand. The bounded main theorem is independent of that result.

A Comparison-only implementation on an integer ball

The algorithm of Veselov, Gribanov, Zolotykh, and Chirkov [22] minimizes a discrete convic function on an integer ball using comparisons of objective values. To apply it to a bounded rational polyhedron, we extend the objective to a containing integer ball. The extension assigns larger values to infeasible points and preserves discrete convicity. For the cell problems of Problem 1, Lemma 3.3 proves the required discrete convicity from nonnegativity on integer differences. This implementation applies whenever the objective is discrete convic on the integer domain.

Lemma A.1 (Explicit extension to an integer ball)**.** *Let $K = \{\boldsymbol{x} : L\boldsymbol{x} \le \boldsymbol{r}\}$ be bounded and rational, and suppose f is discrete convic on $E = K \cap \mathbb{Z}^n$. Write $\boldsymbol{L}_i$ for row i of L. Choose rational $\rho \ge 2$ with E in the Euclidean ball $\mathcal{B}_\rho$, and an explicitly computed M satisfying $f(\boldsymbol{x}) \le M$ for every $\boldsymbol{x} \in E$ (vacuously if $E = \varnothing$). On $\mathcal{B}_\rho \cap \mathbb{Z}^n$ define*

$$v(\boldsymbol{x}) = \max(0, \max_i(\boldsymbol{L}_i\boldsymbol{x} - r_i)), \qquad H(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & v(\boldsymbol{x}) = 0, \\ M + v(\boldsymbol{x}) & v(\boldsymbol{x}) > 0. \end{cases}$$

Then H is discrete convic. Its minimizer yields a minimizer on E if E is nonempty, and otherwise certifies that E is empty.

Proof. Consider the defining configuration for discrete convicity. If $\boldsymbol{y}$ is feasible, every $\boldsymbol{x}_i$ with $H(\boldsymbol{x}_i) \le H(\boldsymbol{y})$ is feasible. If $\boldsymbol{z}$ is feasible, use convicity on E; otherwise $H(\boldsymbol{z}) > M \ge H(\boldsymbol{y})$. If $\boldsymbol{y}$ is infeasible, select a row a attaining $v(\boldsymbol{y}) > 0$. The inequalities $H(\boldsymbol{x}_i) \le H(\boldsymbol{y})$ imply $v(\boldsymbol{x}_i) \le v(\boldsymbol{y})$, whether or not $\boldsymbol{x}_i$ is feasible. For a feasible $\boldsymbol{x}_i$, this follows from $v(\boldsymbol{x}_i) = 0$; for an infeasible $\boldsymbol{x}_i$, subtract M from $H(\boldsymbol{x}_i) \le H(\boldsymbol{y})$. In either case

$$\boldsymbol{L}_a\boldsymbol{x}_i - r_a \le v(\boldsymbol{x}_i) \le v(\boldsymbol{y}) = \boldsymbol{L}_a\boldsymbol{y} - r_a,$$

so $\boldsymbol{L}_a(\boldsymbol{y} - \boldsymbol{x}_i) \ge 0$. Using the representation of $\boldsymbol{z}$ gives

$$\boldsymbol{L}_a\boldsymbol{z} - r_a = \boldsymbol{L}_a\boldsymbol{y} - r_a + \sum_i \alpha_i \boldsymbol{L}_a(\boldsymbol{y} - \boldsymbol{x}_i) \ge v(\boldsymbol{y}) > 0.$$

In particular $\boldsymbol{z}$ is infeasible. It follows that $H(\boldsymbol{z}) \ge M + v(\boldsymbol{y}) = H(\boldsymbol{y})$. Finally, the integer ball is finite and nonempty, so H has a minimizer. All feasible values are at most M and all infeasible values exceed M. If the minimizer is feasible it minimizes f on E; if it is infeasible, there can be no feasible point in the ball. Since the ball contains E, this certifies $E = \varnothing$. $\square$

One may take $M = |\gamma| + \rho \sum_i |c_i| + \rho^2 \sum_{ij} |Q_{ij}|$. Each oracle comparison is exact and has fixed-degree polynomial bit cost in the description and $\log \rho$. For cells with strict inequalities, use the integer-equivalent weak description from Section 3.1.

Definition A.2 (Comparison oracle)**.** For $h : E \to \mathbb{R}$, a comparison oracle takes $\boldsymbol{x}, \boldsymbol{y} \in E$ and decides whether $h(\boldsymbol{x}) \le h(\boldsymbol{y})$. It need not return either value. A bound on the number of oracle calls counts each such answer as one call; evaluating that answer must be charged separately in an explicit bit implementation.

Theorem A.3 (Discrete convic minimization [22, Theorem 1])**.** *Let $\rho \ge 2$ be an integer radius and $\mathcal{B}_\rho = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 \le \rho\}$. Given a comparison oracle for a discrete convic function $h : \mathcal{B}_\rho \cap \mathbb{Z}^n \to \mathbb{R}$, one can find an integer point minimizing h on this domain using*

$$2^{O(n^2 \log(n+1))} \log \rho$$

comparison calls. The algorithm has bit complexity

$$2^{O(n^2 \log(n+1))} \operatorname{poly}(\log \rho)$$

in the oracle model. The polynomial degree is absolute; calls are made only at integer points of the specified ball.

We round a rational containing radius upward to an integer; this changes $\log \rho$ by at most a constant. We write $\log(n+1)$ in place of the source's $\log n$ to include $n = 1$. In the following application, the running time includes both the oracle algorithm and the computation of each comparison.

Proposition A.4 (Comparison-only optimizer)**.** *Let K be a bounded rational polyhedron, and suppose f is discrete convic on $K \cap \mathbb{Z}^n$. Given an integer $\rho \geq 2$ containing these integer points, let φ_{cell} include the encodings of K, f, ρ. Then the comparison-only implementation either returns an exact integer minimizer or determines integer emptiness in time $2^{O(n^2 \log(n+1))}(1 + \varphi_{\text{cell}})^{O(1)}$.*

Proof. Construct H by Lemma A.1, using the explicit quadratic bound M above. To compare $H(\boldsymbol{x})$ and $H(\boldsymbol{y})$, evaluate the constraint rows at both integer query points, compute their maximum violations, and evaluate the quadratic only where feasible. All operations are rational comparisons, additions, and multiplications. Each queried coordinate has $O(1 + \log \rho)$ bits, so one oracle answer costs $T_{\text{cmp}} \leq (1 + \varphi_{\text{cell}})^{O(1)}$.

Theorem A.3 gives the total bound

$$\begin{aligned} T_{\textsc{ConvicMin}} &\leq 2^{O(n^2 \log(n+1))} \left[(1 + \log \rho)^{O(1)} + (1 + \log \rho) T_{\text{cmp}} \right] \\ &\leq 2^{O(n^2 \log(n+1))} (1 + \varphi_{\text{cell}})^{O(1)}. \end{aligned}$$

The first term pays for the oracle algorithm's own bit operations; the second pays for answering its comparisons. Lemma A.1 converts its minimizer to the required result on K. The bound also applies to cells with strict inequalities after the integer-equivalent conversion in Section 3.1. $\square$

The comparison-only implementation therefore gives dimension factor $2^{O(n^2 \log(n+1))}$. The explicit separation inequalities in Lemma 3.6 allow the main algorithm to use the $2^{O(n \log(n+1))}$ bound of Proposition 3.7.

Statements and declarations

AI assistance. ChatGPT was used to assist with drafting, developing and checking mathematical arguments, and preparing figures and LaTeX source. Responsibility for the mathematical claims and final manuscript rests with the authors.

Data and materials availability. This theoretical study reports no experimental datasets. The algorithms and mathematical constructions are described in the manuscript.

References

[1] Cinar Ari and Robert Hildebrand. Curvature batching for integer and mixed-integer quadratic programming. arXiv preprint arXiv:2604.04851, 2026. URL `https://arxiv.org/abs/2604.04851`.

[2] Amitabh Basu. Complexity of optimizing over the integers. *Mathematical Programming*, 200 (2):739–780, 2023. https://doi.org/10.1007/s10107-022-01862-z. URL https://arxiv.org/abs/2110.06172v6. Theorem and remark numbering as in arXiv:2110.06172v6 (2022).

[3] Amitabh Basu and Timm Oertel. Centerpoints: A link between optimization and convex geometry. *SIAM Journal on Optimization*, 27(2):866–889, 2017. https://doi.org/10.1137/16M1092908.

[4] W. Cook, A. M. H. Gerards, A. Schrijver, and É. Tardos. Sensitivity theorems in integer linear programming. *Mathematical Programming*, 34(3):251–264, 1986. https://doi.org/10.1007/BF01582230.

[5] W. Cook, M. Hartmann, R. Kannan, and C. McDiarmid. On integer points in polyhedra. *Combinatorica*, 12(1):27–37, 1992. https://doi.org/10.1007/BF01191202.

[6] Daniel Dadush, Friedrich Eisenbrand, and Thomas Rothvoss. From approximate to exact integer programming. *Mathematical Programming*, 210(1–2):223–241, 2025. https://doi.org/10.1007/s10107-024-02084-1. See Section 5 and Lemma 21, also present in arXiv:2211.03859v4 (2024).

[7] Daniel Nicolas Dadush. *Integer Programming, Lattice Algorithms, and Deterministic Volume Estimation*. PhD thesis, Georgia Institute of Technology, August 2012. URL https://homepages.cwi.nl/~dadush/papers/dadush-thesis.pdf.

[8] Alberto Del Pia and Mingchen Ma. Proximity in concave integer quadratic programming. *Mathematical Programming*, 194(1–2):871–900, 2022. https://doi.org/10.1007/s10107-021-01655-w.

[9] Alberto Del Pia and Robert Weismantel. Integer quadratic programming in the plane. In *Proceedings of the Twenty-Fifth Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 840–846. Society for Industrial and Applied Mathematics, 2014. https://doi.org/10.1137/1.9781611973402.62.

[10] Alberto Del Pia, Robert Hildebrand, Robert Weismantel, and Kevin Zemmer. Minimizing cubic and homogeneous polynomials over integers in the plane. *Mathematics of Operations Research*, 41(2):511–530, 2016. https://doi.org/10.1287/moor.2015.0738.

[11] Michel X. Goemans and Thomas Rothvoss. Polynomiality for bin packing with a constant number of item types. *Journal of the ACM*, 67(6):38:1–38:21, 2020. https://doi.org/10.1145/3421750.

[12] M. Hartmann. Cutting planes and the complexity of the integer hull. Technical Report 819, Cornell University, School of Operations Research and Industrial Engineering, September 1988. URL https://hdl.handle.net/1813/8702.

[13] Anton Herrmann. Integer quadratic programming is W[1]-hard parameterized by the number of variables. arXiv preprint arXiv:2608.17818, 2026. URL https://arxiv.org/abs/2608.17818.

[14] Robert Hildebrand and Adrian Göß. Complexity of integer programming in reverse convex sets via boundary hyperplane cover. Theorem 9 and Appendix B refer to the revised manuscript of 13 May 2026; the linked 2024 version lacks this appendix, May 2026. URL https://arxiv.org/abs/2409.05308v1.

[15] Robert Hildebrand and Matthias Köppe. A new Lenstra-type algorithm for quasiconvex polynomial integer minimization with complexity $2^{O(n \log n)}$. *Discrete Optimization*, 10(1):69–84, 2013. https://doi.org/10.1016/j.disopt.2012.11.003.

[16] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987. https://doi.org/10.1287/moor.12.3.415.

[17] H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983. https://doi.org/10.1287/moor.8.4.538.

[18] Daniel Lokshtanov. Parameterized integer quadratic programming: Variables and coefficients. arXiv preprint arXiv:1511.00310, April 2017. URL https://arxiv.org/abs/1511.00310v2. Version 2, 10 April 2017.

[19] A. M. Macbeath. A theorem on non-homogeneous lattices. *Annals of Mathematics*, 56(2):269–293, 1952. https://doi.org/10.2307/1969800. URL https://doi.org/10.2307/1969800.

[20] 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. https://doi.org/10.1137/100811970. See Section 3.3.

[21] Victor Reis and Thomas Rothvoss. The subspace flatness conjecture and faster integer programming. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 974–988. IEEE, 2023. https://doi.org/10.1109/FOCS57990.2023.00060.

[22] S. I. Veselov, D. V. Gribanov, N. Yu. Zolotykh, and A. Yu. Chirkov. A polynomial algorithm for minimizing discrete convic functions in fixed dimension. *Discrete Applied Mathematics*, 283: 11–19, 2020. https://doi.org/10.1016/j.dam.2019.10.006. See Theorem 1.

[23] Kevin Zemmer. *Integer Polynomial Optimization in Fixed Dimension*. Doctoral dissertation, ETH Zurich, 2017. URL https://doi.org/10.3929/ethz-b-000241796.