A Faster Algorithm for Quasi-convex Integer Polynomial Optimization

Robert Hildebrand and Matthias Köppe

June 23, 2010

Abstract

We present a faster exponential-time algorithm for integer optimization over quasi-convex polynomials. We study the minimization of a quasi-convex polynomial subject to s quasi-convex polynomial constraints and integrality constraints for all variables. The new algorithm is an improvement upon the best known algorithm due to Heinz (*Journal of Complexity*, 2005). A lower time complexity is reached through applying a stronger ellipsoid rounding method and applying a recent advancement in the shortest vector problem to give a smaller exponential-time complexity of a Lenstra-type algorithm. For the bounded case, our algorithm attains a time-complexity of $s(rlMd)^{O(1)}2^{2n\log_2(n)+O(n)}$ when M is a bound on the number of monomials in each polynomial and r is the binary encoding length of a bound on the feasible region. In the general case, $sl^{O(1)}d^{O(n)}2^{2n\log_2(n)}$. In each we assume $d \geq 2$ is a bound on the total degree of the polynomials and l bounds the maximum binary encoding size of the input.

1 Introduction

We study the integer minimization problem over quasi-convex polynomials. That is, given $\hat{F}, F_1, \ldots, F_s \in \mathbb{Z}[\mathbf{x}] = \mathbb{Z}[x_1, \ldots, x_n]$ quasi-convex polynomials with integer coefficients, we wish to solve the following problem

$$\begin{array}{ll} \min & \hat{F}(\mathbf{x}) \\ \text{subject to} & F_i(\mathbf{x}) < 0 \text{ for all } i = 1, \ldots, s \\ & \mathbf{x} \in \mathbb{Z}^n. \end{array} \tag{1}$$

We approach the optimization problem by setting $F_0 = \hat{F} - z^*$ and solving the feasibility problem over $Y \cap \mathbb{Z}^n$, where

$$Y := \{\mathbf{x} \in \mathbb{R}^n : F_i(\mathbf{x}) < 0, i = 0, 1, \ldots, s+1\}, \tag{2}$$

and applying binary search. Strict inequalities are used to ensure that if Y is non-empty, then it is full dimensional in $\mathbb{R}^n$. Since $F_i(\mathbf{x}) \in \mathbb{Z}$ for all $\mathbf{x} \in \mathbb{Z}^n$, problem (1) can be easily formulated by weak inequalities. This follows from the observation that the inequalities $z < 0$ and $z + 1 \leq 0$ are equivalent for $z \in \mathbb{Z}$.

The purpose of this paper is to prove a faster exponential-time algorithm for quasi-convex integer polynomial optimization. Integer optimization is solvable in polynomial time when the dimension n is fixed using Lenstra's algorithm for reduction in dimension. Khachiyan and Porkolab [20] showed that Lenstra's algorithm could be generalized to operate on convex semialgebraic sets. For the specific case of quasi-convex minimization, the currently best algorithm is due to Heinz and has time-complexity of $sl^{O(1)}d^{O(n)}2^{O(n^3)}$, where $d \geq 2$ is an upper bound on the total degree of the polynomials and l is the maximum binary encoding size of all coefficients. We improve this complexity through stronger ellipsoid roundings, utilizing new results on the shortest vector problem and applying those results to Lenstra's algorithm.

Theorem 1.1. *Let $\hat{F}, F_1, \ldots, F_s \in \mathbb{Z}[\mathbf{x}]$ be sparsely encoded quasi-convex polynomials. Let $d \geq 2$ be an upper bound for the degree of the polynomials $F_0, \ldots, F_s$, let M be the maximum number of monomials in each, and let the binary length of the coefficients be bounded by l. Then there exists an algorithm for the minimization problem* (1) *which computes a minimum point or confirms that such a point does not exist.*

(a) *If the continuous relaxation of the feasible region is bounded such that r is the binary encoding length of a bound on that region with $r \leq ld^{O(n)}$, then the algorithm has time-complexity of $s(rlMd)^{O(1)}2^{2n\log_2(n)+O(n)}$ and output-complexity of $(l + r)(dn)^{O(1)}$.*

(b) *Otherwise, the algorithm has time-complexity of $sl^{O(1)}d^{O(n)}2^{2n\log_2(n)}$ and output-complexity of $ld^{O(n)}$.*

For $d = O(1)$, this complexity is $sl^{O(1)}2^{2n\log_2(n)+O(n)}$.
If $d = O(n^k)$ for some $k > 0$, then the complexity becomes $sl^{O(1)}2^{O(n\log(n))}$.

Kannan improved Lenstra's algorithm for linear integer optimization by writing a shortest vector algorithm to determine an optimal flatness direction and hence reducing the number subcases [15]. Using an exact shortest vector method, the number of subcases is reduced from $2^{O(n^3)}$ to $2^{O(n\log(n))}$, an exact number of subcases dependent on the width of a convex set containing no integer points. Kannan and Lovász presented a $O(n^2)$ bound on the width of a convex set that contains no integer points and mention that any stronger width bound would improve the complexity of Lenstra's algorithm further [18]. Banaszczyk gives us such a bound of $O(n)$ [4] for ellipsoids. We follow Kannan and Lovász's presentation of Lenstra's algorithm and show how the cost of each step effects the overall time-complexity. We comment on current best complexities for each piece and apply this approach to quasi-convex integer polynomial optimization.

2 Background

2.1 Lattices

Given m linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m \in \mathbb{R}^n$, the *lattice* Λ generated by these vectors is defined as

$$\Lambda = \Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_m) = \left\{\textstyle\sum_{i=1}^{m} \lambda_i \mathbf{b}_i : \lambda_i \in \mathbb{Z}\right\}.$$

The vectors $\mathbf{b}_1, \ldots, \mathbf{b}_m$, or similarly $B = [\mathbf{b}_1, \ldots, \mathbf{b}_m]$, is called a *basis* for the lattice Λ.
The *dual lattice* Λ^* is given by

$$\Lambda^* = \{\mathbf{v} \in \operatorname{span}(B) : \mathbf{v}^T \mathbf{b}_i \in \mathbb{Z} \ \forall \ i = 1, \ldots, m\}.$$

The *covering radius* $\mu(\Lambda)$ is the smallest number α such that the closed balls of radius α centered at the lattice points cover all of $\mathbb{R}^n$. Closely related is the *packing radius* $\rho(\Lambda)$, which is the largest number β such that the open balls of radius β around the lattice points do not intersect. Lastly, the shortest vector problem (SVP) is to find a non-zero lattice vector $\mathbf{x} \in \Lambda \setminus \{\mathbf{0}\}$ with minimal length $SV(\Lambda)$.

Unless otherwise stated, we will assume to be working in the Euclidean norm for SVP. For further review on lattice, see, for instance [8].

We will need the following simple lemma, which can for instance be found in [13]. We indicate it here with a proof to give a precise complexity.

Lemma 2.1. *Suppose $\Lambda \subset \mathbb{Z}^n$ is a lattice with basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$ and $\dim \Lambda = m \geq 2$. If $\mathbf{d} \in \Lambda \setminus \{\mathbf{0}\}$ is* primitive *(i.e., $\alpha\mathbf{d} \notin \Lambda$ for all $0 < \alpha < 1$). Suppose $\mathbf{d} = \lambda_1 \mathbf{b}_1 + \cdots + \lambda_m \mathbf{b}_m$ with $\lambda_i \in \mathbb{Z}$, for $i = 1, \ldots, m$. Then there exists an algorithm that computes vectors $\bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_m$ such that $\{\mathbf{d}, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_m\}$ is a basis for Λ. This algorithm has time-complexity $(n \log M)^{O(1)}$ where M is the largest of the λ_i's and the entries of the $\mathbf{b}_i$'s in absolute value.*

Proof. Let $B = [\mathbf{b}_1, \ldots, \mathbf{b}_m]$ and let $B' = [\mathbf{d}, \mathbf{b}_2, \ldots, \mathbf{b}_m]$. Without loss of generality, we assume B' has rank m, otherwise we can simply reorder the basis vectors. Let $A \in \mathbb{Z}^{n \times n}$ such that $B' = BA$. We now decompose A into Hermite normal form, which can be done in polynomial time in the input size and the dimension [17]. That is, we find a unimodular matrix $U \in \mathbb{Z}^{n \times n}$ and an upper triangular matrix $T \in \mathbb{Z}^{n \times m}$ such that $A = UT$; therefore, $B' = (BU)T$. There are several algorithms to compute Hermite normal form. For a worst case complexity bound, we use Storjohann and Labahn [31] with run time $O(n^3 T(nM))$ where M is a bound on the maximum binary encoding length of each entry of A, and $T(t)$ is the time required multiply two numbers of size $\lceil t \rceil$. The entries of A are all 1's, 0's, and λ_i's, thus $M = \max_{1 \leq i \leq n} |\lambda_i|$. The unimodularity of U implies that $BU = [\bar{\mathbf{b}}_1, \ldots, \bar{\mathbf{b}}_m]$ is a basis for Λ. Since T is upper triangular, we find that $T_{11}\bar{\mathbf{b}}_1 = \mathbf{d}$, and because $\mathbf{d}$ is primitive, we have $\bar{\mathbf{b}}_1 = \mathbf{d}$. Thus $\{\mathbf{d}, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_m\}$ is a basis for Λ. □

2.2 Polynomial Encoding

In this paper, we allow our complexity results to vary based on the encoding scheme chosen for the polynomials. Multi-variable polynomials can be presented in a list of the coefficients of all the monomials up to degree d, requiring a large storage space. This is typically referred to as a *dense encoding*. Under this scheme, the following remark holds.

Remark 2.2 (Remark 2.1 in [12])**.** Let $F \in \mathbb{Z}[\mathbf{x}]$ be a polynomial of total degree d at most with integer coefficients of binary length bounded by l. Moreover, let $\hat{\mathbf{x}} \in \mathbb{Q}^n$ be a fixed point with $\langle \hat{\mathbf{x}} \rangle < r$. Then there is an algorithm with time complexity $(lrn)^{O(1)} d^{O(n)}$ and output-complexity $(l + r)(dn)^{O(1)}$ which computes the value of the function F and the gradient ∇F at the point $\hat{\mathbf{x}} \in \mathbb{Q}^n$.

This time-complexity however, is too pessimistic; for example, it seems to require $n^{O(n)}$ time to evaluate a monomial of degree $d = n$.

An alternative is *sparse encoding*, where monomials are listed with their nonzero exponents and their coefficients, allowing for a much shorter representation for short polynomials and a more refined time-complexity analysis. Polynomials and their gradients can then be evaluated in $(lrdMn)^{O(1)}$ time, where M is a bound on the number of monomials in the polynomial. This scheme is potentially problematic in Lenstra's algorithm because in each iteration we reduce dimensions by intersecting our region with a hyperplane, which would lead to a loss of sparsity (fill-in). For instance, the given polynomial is x_n^d and our hyperplane is $x_n = x_1 + \cdots + x_{n-1} + 1$, then in the reduced dimension becomes $(x_1 + \cdots + x_{n-1} + 1)^d$. We note that in the algorithm, we never expand these expressions, allowing sparse encoding to continue to be useful. We instead leave the polynomials alone and store coordinate transformation matrices at each step and then compute the coordinates in the original space to input into the polynomials. Gradients are computed via the chain rule. This is discussed in more detail in Remark 2.7.

2.3 Quasi-convex Polynomials

A function $F \colon \mathbb{R}^n \to \mathbb{R}$ is called *quasi-convex* if all the lower level sets $\{\mathbf{x} \in \mathbb{R}^n \colon F(\mathbf{x}) \leq \alpha\}, \alpha \in \mathbb{R}$ are convex subsets of $\mathbb{R}^n$. Although quasi-convex functions are not necessarily convex, all convex functions are quasi-convex. We follow [12] for a review on quasi-convex polynomials.

Lemma 2.3 (Section 4.1, Remark 1 in [7])**.** *Let $F \in \mathbb{R}[\mathbf{x}]$ be a quasi-convex polynomial, $\hat{\mathbf{x}} \in \mathbb{R}^n$ a fixed point and $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{a} \neq \mathbf{0}$ a fixed vector. If the polynomial $F(\hat{\mathbf{x}} + \lambda \mathbf{a})$ in $\lambda \in \mathbb{R}$ is strongly decreasing (or constant, respectively), $F(\mathbf{x} + \lambda \mathbf{a})$ is strongly decreasing (or constant, respectively) for all $\mathbf{x} \in \mathbb{R}^n$.*

This lemma does not necessarily hold if the function is not a polynomial. For example, $f(x, y) = e^{|xy|}$ is quasi-convex where $f(x, 0) = 1$ is constant, but

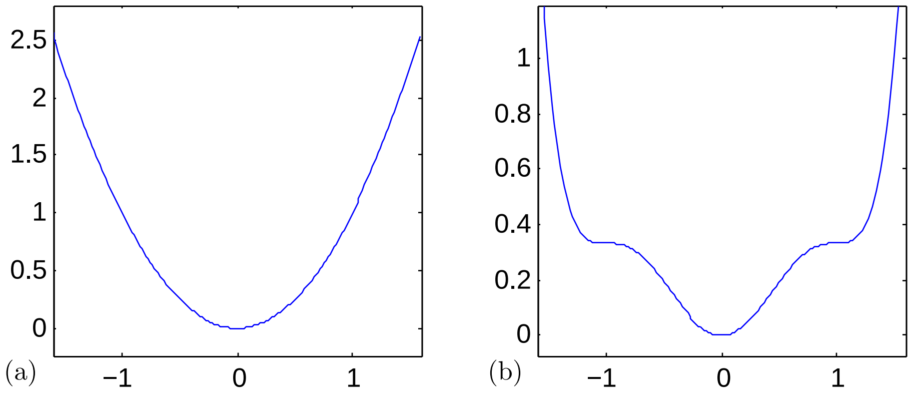

Figure 1: (a) Convex polynomial $f(x) = x^2$, (b) Quasi-convex polynomial $g(x) = x^2 - x^4 + \frac{x^6}{3}$

$f(x, 1) = e^{|x|}$ varies with x. Lemma 2.3 can be used as a quick check for whether a quasi-convex polynomial is constant.

Corollary 2.4 (Corollary 2.3 in [12])**.** *Let $F \in \mathbb{R}[\mathbf{x}]$ be a quasi-convex polynomial of degree d at most, $\hat{\mathbf{x}} \in \mathbb{R}^n$ a point, and let the set $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$ be a basis of $\mathbb{R}^n$. If for every $i = 1, \ldots, n$, there are pairwise distinct real numbers $\lambda_{i1}, \ldots, \lambda_{id} \in \mathbb{R}$ satisfying $\nabla F(\hat{\mathbf{x}} + \lambda_{ij}\mathbf{b}_i) = \mathbf{0}$ for all $j = 1, \ldots, d$, then the polynomial F is constant.*

Lemma 2.5 (Lemma 2.4 in [12])**.** *Let $F \in \mathbb{R}[\mathbf{x}]$ be a quasi-convex polynomial and let $\hat{\mathbf{x}} \in \mathbb{R}^n$ be a fixed point. If $F(\hat{\mathbf{x}}) \geq 0$ and $\nabla F(\hat{\mathbf{x}}) \neq \mathbf{0}$, for every other $\mathbf{x} \in \mathbb{R}^n$ that satisfies $F(\mathbf{x}) < 0$, we have that*

$$\nabla F(\hat{\mathbf{x}}) \cdot \mathbf{x} \leq \nabla F(\hat{\mathbf{x}}) \cdot \hat{\mathbf{x}}.$$

For Lenstra's algorithm in this setting, we need quasi-convex polynomials to remain quasi-convex polynomials when we fix variables and reduce dimensions. We also need that the initial ellipsoid bound $\mathbf{x}^T A_0 \mathbf{x} < R$ reduces to similar ellipsoid bound.

Remark 2.6 (Within the proof of Theorem 4.2 in [12])**.** Let $F_0, \ldots, F_s \in \mathbb{Z}[\mathbf{x}]$ be quasi-convex polynomials, $R \in \mathbb{Z}$, $A_0 \in \mathbb{Z}^{n \times n}$ a positive definite matrix, and $F_{s+1} \in \mathbb{Z}[\mathbf{x}]$ a polynomial defined by $F_{s+1}(\mathbf{x}) := -R + \mathbf{x}^T A_0 \mathbf{x}$, for $\mathbf{x} \in \mathbb{R}^n$. Moreover, let the binary length of the coefficients be bounded by l, let d be an upper bound for the degree of the polynomials. Let $B \in \mathbb{Z}^{n \times n}$ be nonsingular, $t \in \mathbb{Z}$, with entries of B and t of binary length at most $l(dn)^{O(1)}$. Let

$$Y = \{\mathbf{x} \in \mathbb{R}^n : F_i(\mathbf{x}) < 0, i = 1, \ldots, s+1\}$$

and let

$$Y_t := \{\tilde{\mathbf{x}} \in \mathbb{R}^{n-1} : B[\tilde{\mathbf{x}}, t] \in Y\}.$$

Consider the set Y_t and the new coordinates $\tilde{x}_1, \ldots, \tilde{x}_{n-1}$ induced by $\mathbf{x} = B[\tilde{\mathbf{x}}, t]$, fixing the last coordinate $\tilde{\mathbf{x}}_n = t$ and rewrite the quasi-convex polynomials in terms of the new coordinates. The maximum binary length of all coefficients belonging to the new polynomials $\tilde{F}_0, \ldots, \tilde{F}_{s+1} \in \mathbb{Z}[\tilde{x}_1, \ldots, \tilde{x}_{n-1}]$, is $l(dn)^{O(1)}$. Furthermore, all new polynomials are quasi-convex since the transformation is linear and F_{s+1} preserves its form for a new suitable $\tilde{A}_0$. The degree bound d and the number s of polynomials remain unchanged, but the number of coordinates reduces by one.

Remark 2.7. Following the notation of Remark 2.6, we will explain here how we evaluate the polynomials and their gradients under the sparse encoding scheme. Suppose $k+1$ of such coordinate transformations $B^n, \ldots, B^{n-k} \in \mathbb{Z}^{n \times n}$ are done to produce the variable $\tilde{\mathbf{x}}^{n-k} \in \mathbb{Z}^{n-k-1}$. Each B^{n-i} is a block diagonal matrix where the last block is an identity matrix of size i. In each transformation B^{n-i}, we are restricting the last variable to be t_i. A polynomial F transformed into the new coordinates we will denote as $\tilde{F}^{n-k}$. For a given $\tilde{\mathbf{x}}^{n-k} \in \mathbb{Z}^{n-k}$, we can compute $\tilde{F}^{n-k}(\tilde{\mathbf{x}}^{n-k})$ as

$$\tilde{F}^{n-k}(\tilde{\mathbf{x}}^{n-k}) = F(\mathbf{x})$$

where

$$\mathbf{x} = B^n B^{n-1} \cdots B^{n-k} \begin{bmatrix} \tilde{\mathbf{x}}^{n-k} \\ t_{n-k} \\ \vdots \\ t_n \end{bmatrix}$$

The product $C^{n-k} = B^n \cdots B^{n-k}$ is computed ahead of time and a depth first search when reducing dimensions allows us to store at most n of these products at any given time. The partial derivatives of $\tilde{F}^{n-k}$ then have a simple representation as

$$\frac{\partial \tilde{F}^{n-k}}{\partial \tilde{x}_i^{n-k}} = \nabla F(\mathbf{x}) \cdot \frac{\partial \mathbf{x}}{\partial \tilde{x}_i^{n-k}} = \nabla F(\mathbf{x}) \cdot C_i^{n-k}$$

where C_i^{n-k} is the i^{th} column of C^{n-k}.

2.4 Lattice Widths and the Shortest Vector Problem

Finding flatness directions for branching on hyperplanes is the key technique of Lenstra's algorithm. The *width* of a set is determined by minimizing a linear function.

Let $Y \subset \mathbb{R}^n$ be a non-empty closed subset of $\mathbb{R}^n$ and let $\mathbf{d} \in \mathbb{R}^n$ be a vector. The *width of* Y *along* $\mathbf{d}$ is the number

$$w_{\mathbf{d}}(Y) = \max\{\mathbf{d}^T\mathbf{x} : \mathbf{x} \in Y\} - \min\{\mathbf{d}^T\mathbf{x} : \mathbf{x} \in Y\}.$$

The *lattice width* of Y is defined as

$$w(Y) = \min_{\mathbf{d} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}} w_{\mathbf{d}}(Y),$$

and any $\mathbf{d}$ that minimizes $w_{\mathbf{d}}(Y)$ is called a *flatness direction* of Y.

Theorem 2.8 (Khinchin's flatness theorem [21])**.** *Let $K \subset \mathbb{R}^n$ be a convex body. Either K contains an integer point, or $w(K) \leq \omega(n)$, where $\omega(n)$ is a constant depending on the dimension only.*

The currently best known bound for $\omega(n)$ is $O(n^{3/2})$ and it is conjectured that $\omega(n) = \Theta(n)$ [5]. We will see in the next subsection that, for the specific case of ellipsoids, we can obtain this bound.

Flatness directions are invariant under dilations. In particular, we write ellipsoids in the form $E(A, \mathbf{a}) = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x} - \mathbf{a}||_{A^{-1}} \leq 1\}$ where $||\mathbf{v}||_B := \sqrt{\mathbf{v}^T B \mathbf{v}}$, $A \in \mathbb{R}^{n \times n}$ is a positive definite matrix and $\mathbf{a} \in \mathbb{R}^n$.

Lemma 2.9. *Let $\mathbf{d} \in \mathbb{Z}^n$ be a flatness direction for $E(A, \mathbf{a})$. Then for any $\beta \in \mathbb{R}$, $\mathbf{d}$ is a flatness direction for $E(\frac{1}{\beta^2}A, \mathbf{a})$ with*

$$\tfrac{1}{\beta} w(E(A, \mathbf{a})) = w(E(\tfrac{1}{\beta^2}A, \mathbf{a})).$$

Kannan first observed that SVP could be used to minimize the number of branching directions in Lenstra's algorithm [15]. We follow Eisenbrand in presenting this in the context of flatness directions [10].

Remark 2.10. For an ellipsoid, a flatness direction can be computed by solving the shortest vector problem in the lattice $\Lambda((A^{1/2})^T)$. To see this, consider the width along a direction $\mathbf{d}$ of the ellipsoid $E(A, \mathbf{0})$,

$$\begin{aligned} w_{\mathbf{d}}(E(A, \mathbf{a})) &= \max\{\mathbf{d}^T \mathbf{x} : \mathbf{x} \in E(A, \mathbf{0})\} - \min\{\mathbf{d}^T \mathbf{x} : \mathbf{x} \in E(A, \mathbf{0})\} \\ &= \max_{\mathbf{x}_1, \mathbf{x}_2 \in E(A, \mathbf{0})} \mathbf{d}^T(\mathbf{x}_1 - \mathbf{x}_2). \end{aligned}$$

We have $\mathbf{d}^T(\mathbf{x}_1 - \mathbf{x}_2) = \mathbf{d}^T A^{1/2}(A^{-1/2}\mathbf{x}_1 - A^{-1/2}\mathbf{x}_2)$ where $A^{-1/2}\mathbf{x}_1$ and $A^{-1/2}\mathbf{x}_2$ are contained in the unit ball if and only if $\mathbf{x}_1, \mathbf{x}_2 \in E(A, \mathbf{0})$. Thus properly choosing $\mathbf{x}_1$ and $\mathbf{x}_2$ on the boundary of $E(A, \mathbf{0})$, we see that

$$w_{\mathbf{d}}(E(A, \mathbf{0})) = 2||\mathbf{d}^T A^{1/2}||_2.$$

Finding the minimum lattice width is then a shortest vector problem over the lattice $\Lambda((A^{1/2})^T)$.

2.5 Results from the Geometry of Numbers

The geometry of numbers produces a small bound on the lattice width of an ellipsoid not containing an integer point. By considering the specific case of ellipsoids, we can produce an $O(n)$ bound. Using properties of LLL reduced bases, Lenstra originally observed that this value did not exceed $2^{O(n^2)}$ [25]. For an arbitrary lattice, the product of the length of a shortest vector in a lattice and the covering radius of the dual lattice is bounded by a constant $f(n)$ dependent only on dimension. Using the Fourier transform applied to a probability measure on a lattice, Banaszczyk showed that this function is bounded by a linear factor in the dimension n.

Theorem 2.11 (Theorem 2.2 in [4])**.** *Let $\Lambda \subset \mathbb{R}^n$ be a lattice with $n \geq 1$. Then $SV(\Lambda)\mu(\Lambda^*) \leq f(n) \leq \frac{1}{2}n$.*

If we assume that a specific ellipsoid does not contain a lattice point, then the covering radius of the associated lattice is greater than one. Since the lattice width of an ellipsoid is simply twice the length of a shortest vector, we obtain the following inequality for ellipsoids.

Theorem 2.12 (Theorem 14.26 in [10])**.** *If $E(A, \mathbf{a}) \subset \mathbb{R}^n$ is an ellipsoid that does not contain an integer point, then $w(E(A, \mathbf{a})) \leq 2f(n)$.*

Thus a convenient bound follows directly from Theorems 2.11 and 2.12.

Corollary 2.13. *Let $E(A, \mathbf{a}) \subset \mathbb{R}^n$ be an ellipsoid not containing an integer point, then $w(E(A, \mathbf{a})) \leq n$.*

2.6 Complexity of the Shortest and Closest Vector Problems

The shortest vector problem as been shown to be NP-hard, even to approximate it within a constant factor [26]. Until recently, the best known deterministic solution to SVP was given by Kannan with time-complexity $n!$ [16]. The well known Ajtai, Kumar, and Sivakumar [1] sieving method is a probabilistic method that solves SVP with very high probability and achieved the first singly exponential time-complexity, which was shown by [30] to be $2^{5.9n}$. Micciancio and Voulgaris improved this type of method to achieve a run time of $2^{3.199n}$ [28]. Micciancio and Voulgaris have announced in an extended abstract [27] deterministic, singly exponential time algorithms for SVP.

Theorem 2.14 (Corollary 3.2 [27])**.** *There is a deterministic single exponential time algorithm to solve SVP.*

3 Ellipsoid Rounding

Let Y be a convex set. The ellipsoid $E(A, \mathbf{a})$ is a *β-rounding* of Y if

$$E(\tfrac{1}{\beta^2}A, \mathbf{a}) \subseteq Y \subseteq E(A, \mathbf{a})$$

where β is called the *radius* of the rounding [29]. John [14] showed there there exists a n-rounding for any convex set. The shallow cut ellipsoid method is one way to obtain an ellipsoid rounding. One transforms the problem to be contained in the unit ball and then chooses points in the ball of radius $\frac{1}{n+1}$. Taking the convex hull of these points yields a polytope where the rounding radius is

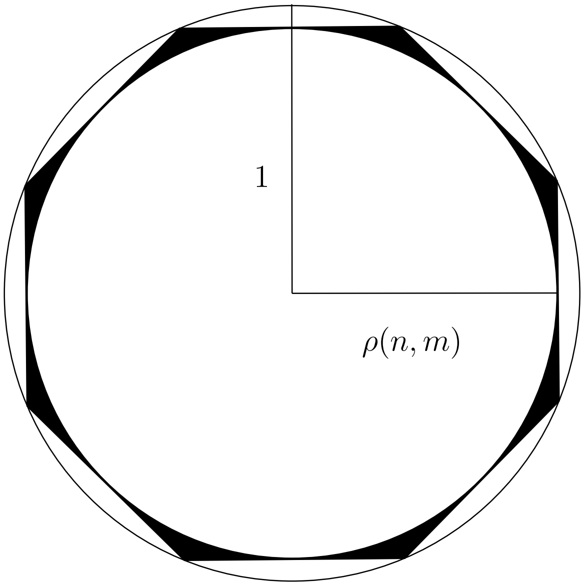

Figure 2: The approximating radius $\rho(n,m)$ for a polytope in $\mathbb{R}^2$.

then dependent on the maximum inscribed sphere in the polytope. Using a cross-polytope ($\text{conv}(\{\pm\mathbf{e}_i : i = 1, \ldots, n\})$ where $\mathbf{e}_i$ is the i^{th} unit vector), [11] obtains a $O(n^{3/2})$-rounding of a polytope. Heinz used this idea to obtain a rounding of a convex region given by quasi-convex polynomials [12]. We generalize and improve Heinz's method by applying sphere approximating polytopes of Kochol [22] that attain an optimal bound within a constant factor. A later note by Kochol, modified to give more detail, shows the following result.

Theorem 3.1 (Theorem 3 in [23])**.** *Let n, m be positive integers, $2n \le m \le c^n$, where $c > 1$ is a constant. Let $\rho(n,m)$ as the maximal radius of a ball (with center at the origin) contained in the convex hull of m points chosen from the n-dimensional sphere of radius 1. Then there exist constants c_1 and c_2 such that*

$$c_1\sqrt{\frac{\log(m/n)}{n}} \le \rho(n,m) \le c_2\sqrt{\frac{\log(m/n)}{n}}.$$

Furthermore, there exists a polynomial time algorithm in n and m to construct a set of vectors $V \subset \mathbb{Z}^n$ with $|V| \le m$ such that the polytope with extreme points $\{\mathbf{v}/||\mathbf{v}||_2 : \mathbf{v} \in V\}$ is symmetric across all axes and attains such bounds.

Kochol notes that choosing $m := n^2$ points improves the $O(n^{3/2})$-rounding to $O(n^{3/2}/\sqrt{\log n})$ and still allows a polynomial time rounding, and improves upon the complexity for Lenstra's algorithm given in [11]. Theorem 3.1 also shows that an exponential number of points is necessary to obtain an $O(n)$-rounding via the shallow cut ellipsoid method. For our purposes, a better rounding is advantageous, thus in the final result we will choose a singly exponential number of points, $m := n2^n$ will suffice, to obtain an $O(n)$-rounding while submitting to an exponential number of evaluations. This will give us a slightly better complexity result.

We remark that other methods for computing ellipsoid roundings are available. Nesterov describes an algorithm to obtain a γn-rounding, $\gamma > 1$ for an arbitrary convex set and also how to obtain a $\gamma\sqrt{n}$-rounding for centrally symmetric convex sets [29], although each is based on the assumption that a difficult optimization problem can be solved. For this, Nesterov uses linear programming, whereas we would need to maximize over nonlinear polynomials. In our model, no supplementary optimization problem need be solved. Ellipsoid roundings have also been studied recently by Khachiyan [19], which has been improved by [24] and [32]. Some other methods use a volumetric barrier [2, 3].

We denote the sphere of radius r as $S^{n-1}(r) = \{\mathbf{x} \in \mathbb{R}^n \colon ||\mathbf{x}||_2 = r\}$. A 1-*net* of $S^{n-1}(r)$ is a set of points $N \subset S^{n-1}(r)$ such that for any point $\mathbf{v} \in S^{n-1}(r)$, there exists a point $\tilde{\mathbf{v}}$ such that $||\mathbf{v} - \tilde{\mathbf{v}}||_2 \leq 1$.

Lemma 3.2. *Let N be a 1-net of $S^{n-1}(1)$ and let $0 < \epsilon < \frac{1}{2}$. Suppose that $\tilde{N}$ is an* ϵ-approximation *of N, that is to say that for all $\mathbf{v} \in N$ there exists a $\tilde{\mathbf{v}} \in \tilde{N}$ such that*

$$||\mathbf{v} - \tilde{\mathbf{v}}||_2 \leq \epsilon,$$

then for all $0 \leq \alpha < 1/2$ we have that $\mathrm{conv}(\tilde{N}) \supseteq S^{n-1}(\alpha)$.

Proof. Suppose there exists a point $\mathbf{z} \in S^{n-1}(\alpha)$ that does not belong to $\mathrm{conv}(\tilde{N})$, then separating $\mathbf{z}$ from $\mathrm{conv}(\tilde{N})$ by a hyperplane $p_{\mathbf{z}}$ we get a cap of $S^{n-1}(1)$ which is disjoint from $\tilde{N}$ and its top $\mathbf{t}$ where $\mathbf{t}$ is perpendicular to $p_{\mathbf{z}}$. Since $\mathbf{t}$ is in $S^{n-1}(1)$, there exists a point $\mathbf{v} \in N$ from the 1-net that satisfies $||\mathbf{t}-\mathbf{v}||_2 \leq 1$. See Figure 3 for the geometry. Letting d be the minimum distance between $\mathbf{v}$ and the hyperplane $p_{\mathbf{z}}$, we can see that $d > \epsilon$, which is a contradiction since $\tilde{N}$ is an ϵ-approximation of N. □

Corollary 3.3. *Let N be the set of points given by Kochol's construction for an approximation of the unit sphere and let $\tilde{N}$ be an ϵ-approximation of N with $0 < \epsilon < 1/2$. Then there exist a constant $c_1(\epsilon)$ such that the ball of radius $c_1\sqrt{\frac{\log(m/n)}{n}}$ is contained in* $\mathrm{conv}(\tilde{N})$.

Proof. Using Lemma 3.2, the proof is very similar to Theorem 1 in [22]. □

We will now follow similarly to Heinz to define our separation oracle for ellipsoid rounding. Since ellipsoids are just an affine transformation of the unit ball, the goal here is to describe a polytope encompassing a ball of radius less than or equal to $\frac{1}{n+1}$. Under the affine transformation, if the polytope is a subset

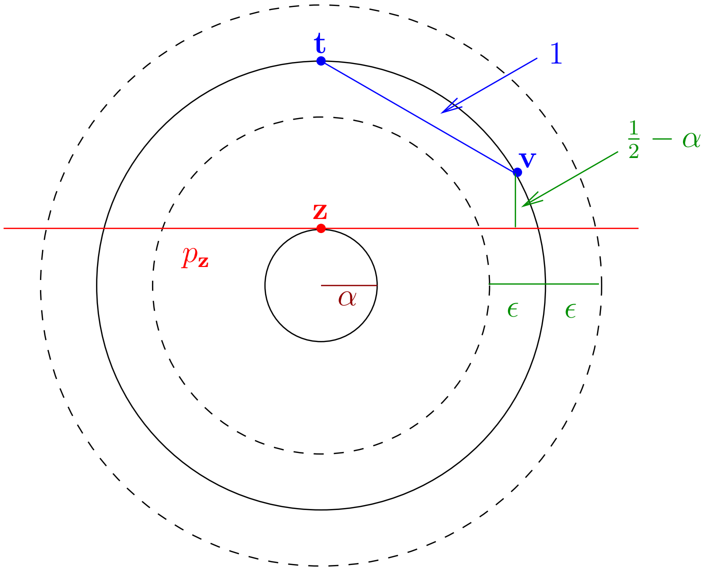

Figure 3: Geometry of the proof for Lemma 3.2

of Y, then the transformed smaller ball is also a subset of Y. The transformed ball is then the desired inscribed ellipsoid. Heinz used the cross-polytope as his rounding. This polytope was inscribed in the ball of radius $\frac{1}{n+3/2}$ and rational points were chosen with lengths between $\frac{1}{n+3/2}$ and $\frac{1}{n+1}$. If the closest points to the origin are contained in the set Y, then Y contains the cross-polytope and hence Y also contains the ball of radius $\frac{1}{(n+1)^{3/2}}$. If not, then the other points are utilized to find a separating hyperplane.

We will use a similar idea, except we will have more points to allow inscribing a larger ball.

Recall that we are solving the feasibility problem over the set

$$Y = \{\mathbf{x} \in \mathbb{R}^n : F_i(\mathbf{x}) < 0 \text{ for } i = 0, 1, \ldots, s+1\}$$

where all the $F_i's$ are quasi-convex polynomials. Consider an ellipsoid $E(A, \hat{\mathbf{x}})$ and let $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$ be an orthogonal basis of $\mathbb{R}^n$ according to the matrix A (where the inner product is given by $\langle \mathbf{x}, \mathbf{y} \rangle_A = \mathbf{x}^T A \mathbf{y}$). Define the affine map $\tau : \mathbb{R}^n \to \mathbb{R}^n$ such that

$$\tau(\mathbf{x}) := B^T(\mathbf{x} - \hat{\mathbf{x}}) \tag{3}$$

where

$$B := \left(\frac{\mathbf{b}_1}{||\mathbf{b}_1||_A}, \ldots, \frac{\mathbf{b}_n}{||\mathbf{b}_n||_A} \right) \in \mathbb{R}^{n \times n}. \tag{4}$$

Thus $\mathbf{x}^T \mathbf{x} \leq 1$ if and only if $\tau^{-1}(\mathbf{x}) \in E(A, \hat{\mathbf{x}})$.

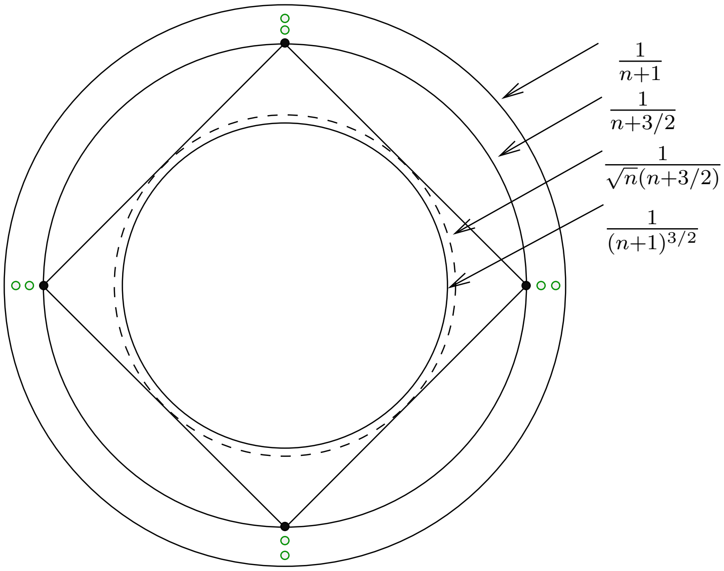

Figure 4: Under the affine transformation, the open dots are the test points for Heinz's separation oracle

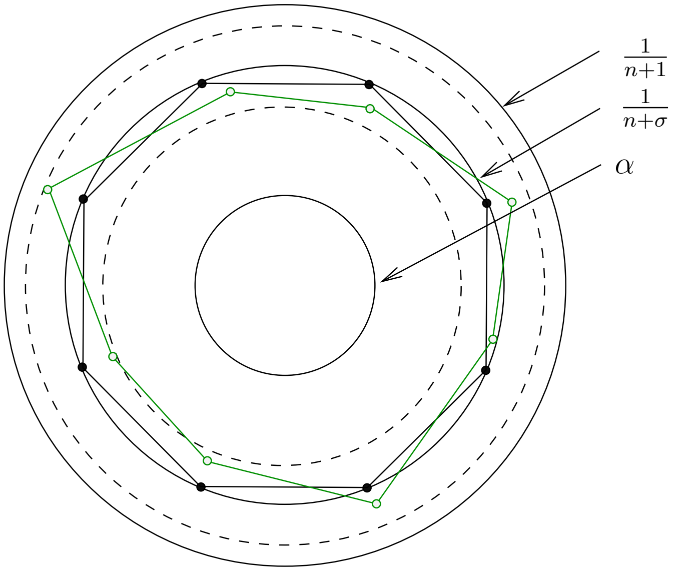

Figure 5: An inscribed polytope and a rational approximation with vertices, respectively, as filled in dots and open dots. The innermost circle is the ball that we can guarantee will remain inside the convex hull of the approximated points.

Theorem 3.4. *Let $\hat{c} > 1$ and let $m\colon \mathbb{N} \to \mathbb{N}$, such that $2n \leq m(n) \leq \hat{c}^n$. Then there exists a function $\beta\colon \mathbb{N} \to \mathbb{R}$ where $\beta(n) = O(\frac{n^{3/2}}{\log(m/n)})$ and an algorithm with the following input:*

(I_1) sparsely encoded quasi-convex polynomials $F_0, \ldots, F_{s+1} \in \mathbb{Z}[\mathbf{x}]$ of total degree d, at most M monomials in each, and whose coefficients' binary encoding lengths are bounded by l,

(I_2) an ellipsoid $E(A, \hat{\mathbf{x}})$ containing Y as defined in (2), where the binary encoding length of the columns of A and of $\hat{\mathbf{x}}$ are bounded by r,
and outputs one of the following answers:

1. *confirmation that the ellipsoid $E(A, \hat{\mathbf{x}})$ is a β-rounding of Y, or*
2. *a vector $\mathbf{c} \in \mathbb{Q}^n, \mathbf{c} \neq \mathbf{0}$, with the property*

$$Y \subset \left\{ \mathbf{x} \in \mathbb{R}^n : \mathbf{c}^T \hat{\mathbf{x}} + \frac{1}{n+1} ||\mathbf{c}||_A \right\}. \tag{5}$$

This algorithm runs in time-complexity $s(lnrmM)^{O(1)}$ and with output-complexity $(l+r)(dn)^{O(1)}$.

Proof. First compute an orthogonal basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ according to A. Let $\sigma > 1$. Next construct a polytope approximating $S^{n-1}(1)$ according to Theorem 3.1 using m vertices and let $V \subset \mathbb{Z}^n$ denote the set of vertices.

Then for every $\mathbf{v} \in V$, define

$$\mathbf{b}_\mathbf{v} := \sum_{i=1}^{n} \frac{v_i \mathbf{b}_i}{||\mathbf{v}||_2 ||\mathbf{b}_i||_A},$$

$$\mathbf{x}_\mathbf{v} := \hat{\mathbf{x}} + \frac{1}{(n+\sigma)} \mathbf{b}_\mathbf{v}.$$

Since we cannot compute $\mathbf{b}_\mathbf{v}$ and $\mathbf{x}_\mathbf{v}$ exactly due to square roots in the norms, we will approximate the square roots. This can be done with any root finding technique using fixed point arithmetic, for example, the Newton-Raphson method has quadratic convergence and will find the desired approximation within polynomial time. For a reference on numerical methods, see [9]. We note that

$$||\mathbf{v}||_2 ||\mathbf{b}_i||_A = \sqrt{\mathbf{v}^T \mathbf{v}} \sqrt{\mathbf{b}_i^T A \mathbf{b}_i} = \sqrt{\mathbf{v}^T \mathbf{v} \mathbf{b}_i^T A \mathbf{b}_i}.$$

Thus we will approximate the reciprocal of that square root within an accuracy of $\delta = \epsilon / (||A^{1/2}||_\infty ||\mathbf{b}_{\max}||_2 \sqrt{n})$ where $||A||_\infty$ is the maximum row sum of A and $\mathbf{b}_{\max} = \operatorname{argmax}\{||\mathbf{b}_i||_2 : i = 1, \ldots, n\}$. Let $0 < \delta_{\mathbf{v},i} < \delta$ be the exact error on each approximation. Let $\tilde{\mathbf{b}}_\mathbf{v} \in \mathbb{Q}^n$ and $\tilde{\mathbf{x}}_\mathbf{v} \in \mathbb{Q}^n$ denote the rational approximations of $\mathbf{b}_\mathbf{v}$ and $\mathbf{x}_\mathbf{v}$ respectively. We note here that since V is symmetric across all axes, it suffices to store only the vertices in the first orthant. This exponentially reduces the number of root approximations necessary.

Case 1: Suppose $\tilde{\mathbf{x}}_\mathbf{v} \in Y$ for all $\mathbf{v} \in V$. We will show that $E(A, \hat{\mathbf{x}})$ is a β-rounding of Y.

We will first note that for two vectors in $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, if $||\mathbf{x} - \mathbf{y}||_\infty \leq \epsilon/\sqrt{n}$, then we know that $||\mathbf{x} - \mathbf{y}||_2 \leq \epsilon$.

Then

$$B^T A \tilde{\mathbf{b}}_\mathbf{v} = \sum_{i=1}^n \mathbf{v}_i \frac{\mathbf{b}_i^T A \mathbf{b}_i}{||\mathbf{b}_i||_A} \left(\frac{1}{||\mathbf{v}||_2 ||\mathbf{b}_i||_A} + \delta_{\mathbf{v},i} \right) = \sum_{i=1}^n \mathbf{e}_i \mathbf{v}_i (\frac{1}{||\mathbf{v}||_2} + \delta_{\mathbf{v},i} ||\mathbf{b}_i||_A), \tag{6}$$

where $\mathbf{e}_i \in \mathbb{R}^n$ is the i^{th} unit vector. Hence, component-wise we have

$$\left| \frac{\mathbf{v}_i}{||\mathbf{v}||_2} - \tau(\tilde{\mathbf{x}}_\mathbf{v}) \right| = \mathbf{v}_i \delta_{\mathbf{v},i} ||\mathbf{b}_i||_A \leq \delta_{\mathbf{v},i} ||\mathbf{b}_i||_A \leq \delta ||A^{1/2}||_\infty ||\mathbf{b}_{\max}||_2 \leq \epsilon/\sqrt{n}.$$

Choosing our accuracy $\delta = \epsilon/(||A^{1/2}||_\infty ||\mathbf{b}_{\max}||_2 \sqrt{n})$ meets the necessary conditions.

Hence $\{B^T A \tilde{\mathbf{b}}_\mathbf{v} : \mathbf{v} \in V\}$ is an ϵ-approximation of V. Therefore by Corollary 3.3, there exists a $\hat{\beta} = O(\sqrt{n/\log(m/n)})$ such that $\text{conv}(K) \subset E(\frac{1}{\hat{\beta}^2} I, \mathbf{0})$. Hence, letting $\beta = \hat{\beta}(n + \sigma)$, we have that

$$E(\tfrac{1}{\beta^2} A, \hat{\mathbf{x}}) = \tau^{-1}(E(\tfrac{1}{\hat{\beta}^2} I, \mathbf{0})) \subset \text{conv}(\{\tilde{\mathbf{x}}_\mathbf{v} : \mathbf{v} \in V\}) \subset Y \subset E(A, \hat{\mathbf{x}}).$$

Case 2: $\tilde{\mathbf{x}}_{\bar{\mathbf{v}}} \notin Y$ for some $\bar{\mathbf{v}} \in V$. We will show that there exists the desired hyperplane.

Then for some $F \in \{F_0, \ldots, F_{s+1}\}$, we know that $F(\tilde{\mathbf{x}}_{\bar{\mathbf{v}}}) \geq 0$.

Case 2.1: $F(\hat{\mathbf{x}}) < 0$.
Pick scalars $\lambda_1, \ldots, \lambda_d$, such that

$$\frac{n+1}{n+\sigma} < \lambda_1 < \cdots < \lambda_d < \frac{1}{||\tilde{\mathbf{b}}_{\bar{\mathbf{v}}}||_A}, \tag{7}$$

and define

$$\mathbf{x}(i) = \hat{\mathbf{x}} + \frac{1}{n+1} \lambda_i A \tilde{\mathbf{b}}_{\bar{\mathbf{v}}}. \tag{8}$$

Since the inequalities $F(\hat{\mathbf{x}}) < 0$ and $F(\tilde{\mathbf{x}}_{\bar{\mathbf{v}}}) \geq 0$ are valid, the polynomial $F(\hat{\mathbf{x}} + \frac{1}{n+1} \lambda \tilde{\mathbf{b}}_{\bar{\mathbf{v}}})$ is of degree d at most and not constant with respect to λ. Therefore, we may choose a point $\hat{\mathbf{y}} \in \mathbb{R}^n$ satisfying

$$\hat{\mathbf{y}} = \hat{\mathbf{x}} + \frac{1}{n+1} \lambda_k A \tilde{\mathbf{b}}_{\bar{\mathbf{v}}} \text{ and } \nabla F(\hat{\mathbf{y}}) \neq 0.$$

Define $\mathbf{c} := \nabla F(\hat{\mathbf{y}})^T$. Note that $\hat{\mathbf{y}} \notin Y$ since $\tilde{\mathbf{x}}_\mathbf{v}$ is a convex combination of $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$. Thus we have that for all $\mathbf{x} \in Y$,

$$\mathbf{c}^T \mathbf{x} \leq \mathbf{c}^T \hat{\mathbf{y}} = \mathbf{c}^T \hat{\mathbf{x}} + \frac{1}{n+1} \lambda_k \mathbf{c}^T A \tilde{\mathbf{b}}_{\bar{\mathbf{v}}} \leq \mathbf{c}^T \hat{\mathbf{x}} + \frac{1}{n+1} \frac{\mathbf{c}^T A \tilde{\mathbf{b}}_{\bar{\mathbf{v}}}}{||\tilde{\mathbf{b}}_{\bar{\mathbf{v}}}||_A} \leq \mathbf{c}^T \hat{\mathbf{x}} + \frac{1}{n+1} ||\mathbf{c}||_A.$$

The last inequality comes from the Cauchy-Schwarz inequality for the scalar product generated by the matrix A.

Case 2.2: $F(\hat{\mathbf{x}}) \geq 0$ (i.e. $\hat{\mathbf{x}} \notin Y$).
Pick scalars $\lambda_{i1}, \dots, \lambda_{id}$ for each $i = 1, \dots, n$ such that

$$\frac{n+1}{n+\sigma} \frac{1}{||\mathbf{b}_i||_A} < \lambda_{i1} < \cdots < \lambda_{id} < \frac{1}{||\mathbf{b}_i||_A}$$

once again using the Newton-Raphson method to approximate the roots, but this time the same precision is not required. Let

$$C = \left\{ \hat{\mathbf{x}} \pm \frac{1}{n+1} \lambda_{ij} A \mathbf{b}_i \colon j = 1, \dots, n \right\}.$$

By Lemma 2.4, if $\nabla F(\mathbf{y}) = \mathbf{0}$ for all $\mathbf{y} \in C$, then F is constant. If so, then any point $\mathbf{c} \in \mathbb{Q}^n$ will suffice to output.

Otherwise, for every $i = 1, \dots, n$, define the finite subsets

$$C_i^+ := \left\{ \hat{\mathbf{x}} + \frac{1}{n+1} \lambda_{ij} A \mathbf{b}_i \colon j = 1, \dots, n \right\},$$
$$C_i^- := \left\{ \hat{\mathbf{x}} - \frac{1}{n+1} \lambda_{ij} A \mathbf{b}_i \colon j = 1, \dots, n \right\}$$

For every $i = 1, \dots, n$, since $\hat{\mathbf{x}}$ is a convex combination of points in C_i^+ and C_i^-, at least one of the sets $C_i^+ \cap Y$ or $C_i^- \cap Y$ is empty. Thus there exists a point $\hat{\mathbf{y}} \in C$ such that, similar to Case 2.1,

$$F(\hat{\mathbf{y}}) \geq 0 \text{ and } \nabla F(\hat{\mathbf{y}}) \neq \mathbf{0}.$$

Define $\mathbf{c} = \nabla F(\hat{\mathbf{y}})$. The remaining calculation is the same as above. □

Corollary 3.5. *Let $\hat{c} > 1$ and let $m \colon \mathbb{N} \to \mathbb{N}$, such that $2n \leq m(n) \leq \hat{c}^n$. Then there exists a function $\beta \colon \mathbb{N} \to \mathbb{R}$ where $\beta(n) = O(\frac{n^{3/2}}{\log(m/n)})$ and an algorithm with the following input:*

(I$_1$) *sparsely encoded quasi-convex polynomials $F_0, \dots, F_s \in \mathbb{Z}[\mathbf{x}]$ of total degree d and of at most M monomials,*

(I$_2$) *an integer R and a positive definite matrix $A_0 \in \mathbb{Z}^{n \times n}$ and define $F_{s+1}(\mathbf{x}) := -R + \mathbf{x}^T A_0 \mathbf{x}$. Let l be a bound on the maximum binary encoding length of the coefficients of $F_0, \dots, F_{s+1}$.*

(I$_3$) *a positive number $\epsilon \in \mathbb{Q}$,*

and outputs a positive definite matrix $A \in \mathbb{Q}^{n \times n}$ and a point $\hat{\mathbf{x}} \in \mathbb{Q}^n$ such that one of the following holds:

1. *$Y \subseteq E(A, \hat{\mathbf{x}})$ and $\operatorname{vol}(E(A, \hat{\mathbf{x}})) \leq \epsilon$, or*
2. *the ellipsoid $E(A, \hat{\mathbf{x}})$ is a β-rounding of Y.*

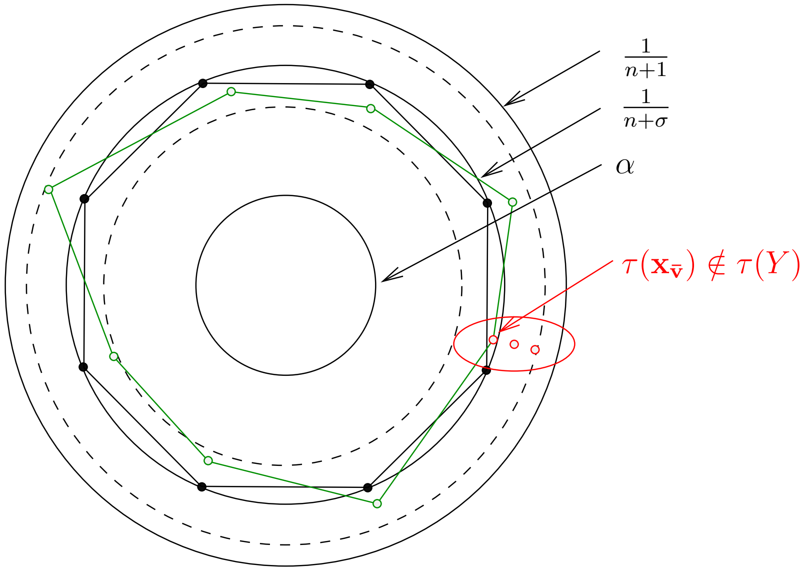

Figure 6: Case 2.1: If $F(\hat{\mathbf{x}}) < 0$ and $F(\mathbf{x}_{\bar{\mathbf{v}}}) \geq 0$ then F restricted to the line through those two points is not constant. One of the points on that line must have a non-zero gradient and not lie within Y.

This algorithm runs in time-complexity $s(lnmdM\langle\epsilon\rangle)^{O(1)}$ and with output-complexity $(l + \langle\epsilon\rangle)(dn)^{O(1)}$.

Proof. The proof is similar to [12, Corollary 3.4]. □

4 Lenstra-type Algorithm

Here we state a modern version of Lenstra's algorithm for a class $\mathcal{C}$ of convex sets. The algorithm is left as an outline, allowing each step to be filled in with the best known procedure for that step. Any filled in steps involving the feasible region must be compatible with the class $\mathcal{C}$. The projection of a set in $\mathcal{C}$ must also be shown to be in $\mathcal{C}$. Potential sources of improvement in the algorithm are the ellipsoid rounding procedure, the upper bound $\tilde{\omega}(n)$ for the lattice width of an ellipsoid not containing lattice points, and the complexity of SVP.

Input: A convex set $Y \subset \mathbb{R}^n$ in class $\mathcal{C}$.
Output: A point $\mathbf{x}^* \in Y \cap \mathbb{Z}^n$ or confirmation that no such point exists.
PROCEDURE:

1. **Declare Minimum Volume:** Assume that the set $Y \cap \mathbb{Z}^n$ is not empty. Based on the input data, find an $\epsilon > 0$ such that $\epsilon < \mathrm{vol}(Y)$ holds.
2. **Ellipsoid Rounding:** Compute an ellipsoid $E(A, \hat{\mathbf{x}})$ for such an ϵ such that we have either

(a) $Y \subseteq E(A, \hat{\mathbf{x}})$ and $\mathrm{vol}(E(A, \hat{\mathbf{x}})) \leq \epsilon$, or

(b) $E(A, \hat{\mathbf{x}})$ is a β-rounding of Y.

If we are in case (a), then no such point exists.
Otherwise we proceed as we are in case (b).

3. **Compute a Flatness Direction via Shortest Vector Problem:** Compute a flatness direction $\mathbf{c} \in \mathbb{Z}^n$ of $E(A, \hat{\mathbf{x}})$ following Remark 2.10 by solving the shortest vector problem on $\Lambda((A^{1/2})^T)$. If $w(E(\frac{1}{\beta^2}A, \hat{\mathbf{x}})) > \tilde{\omega}(n)$, that is that if $w(E(A, \hat{\mathbf{x}})) > \tilde{\omega}(n)\beta$, then by Theorem 2.13 there exists an integer point $\mathbf{x}^* \in E(\frac{1}{\beta^2}A, \hat{\mathbf{x}}) \subset Y$.
   Otherwise, proceed knowing that $w(E(A, \hat{\mathbf{x}})) \leq \tilde{\omega}(n)\beta$.

4. **Compute Sublattice:** Compute vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1} \in \mathbb{Z}^n$ such that $\mathbb{Z}^n = \mathbf{c}\mathbb{Z} \oplus \Lambda$ where $\Lambda = \Lambda(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$. This may be done according to Lemma 2.1.

5. **Recurse:** Note that $|\mathbf{c}^T\mathbf{x}| \leq \tilde{\omega}(n)\beta$ for every $\mathbf{x} \in E(A, \mathbf{0})$. Now we define the set

   $$M := \{z + \lfloor \mathbf{c}^T\hat{\mathbf{x}} \rfloor : |z| \leq \tilde{\omega}(n)\beta + 1, z \in \mathbb{Z}\},$$

   and define $B = [\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}, \mathbf{c}]$. Thus B is a basis for the lattice $\mathbb{Z}^n$. Also now define the set

   $$Y_t = \{\tilde{\mathbf{x}} \in \mathbb{R}^{n-1} : B[\tilde{\mathbf{x}}, t] \in Y\}.$$

   This step must ensure the set $\{\tilde{\mathbf{x}} \in \mathbb{R}^{n-1} : [\tilde{\mathbf{x}}, t] \in Y_t\}$ is also a convex set in class $\mathcal{C}$ to allow for a recursion.
   The number of elements in M is $2\tilde{\omega}(n)\beta + 3$. This means that if the algorithm runs to its full extent, the total number of subcases it will have to evaluate is

   $$\prod_{i=1}^{n}(2\tilde{\omega}(i)\beta(i) + 3).$$

   Here we have $\beta = \beta(i)$ because the rounding factor is a function of dimension as well.

We will follow this procedure to improve the complexity bound for solving integer optimization over quasiconvex polynomials. To do this, we will need the following lemma.

Lemma 4.1 (Lemma 4.1 in [12]). *Let $F_0, \ldots, F_{s+1} \in \mathbb{Z}[\mathbf{x}]$ be polynomials, $R \in \mathbb{Z}$ an integer, $A_0 \in \mathbb{Z}^{n \times n}$ a positive definite matrix, and $F_{s+1} \in \mathbb{Z}[\mathbf{x}]$ a polynomial defined by $F_{s+1}(\mathbf{x}) := -R + \mathbf{x}^T A_0 \mathbf{x}$ for $\mathbf{x} \in \mathbb{R}^n$. Moreover, let the binary encoding length of the coefficients bounded above by l, and let d be an upper bound for the degree of the polynomials and let the set Y contain an integer point $\hat{\mathbf{x}} \in \mathbb{Z}^n$. Then there is a positive rational number $\epsilon \in \mathbb{Q}$ which bounds the volume $0 < \epsilon < \mathrm{vol}(Y)$ such that its binary length $\langle\epsilon\rangle$ is in the class $l(dn)^{O(1)}$.*

We now will prove the main theorem that solves the feasibility problem.

Theorem 4.2. *Let $F_0, \ldots, F_s \in \mathbb{Z}[\mathbf{x}]$ be quasi-convex polynomials, $R > 0$ an integer, $A_0 \in \mathbb{Z}^{n\times n}$ a positive definite matrix, and $F_{s+1} \in \mathbb{Z}[\mathbf{x}]$ a polynomial defined by $F_{s+1}(\mathbf{x}) = -R + \mathbf{x}^T A_0 \mathbf{x}$ for $\mathbf{x} \in \mathbb{R}^n$. Let d be an upper bound for the degree of the polynomials $F_0, \ldots, F_{s+1}$, presented as a sparse list of monomials with at most M monomials, and let the binary length of the coefficients be bounded by l. Moreover, consider the set*

$$Y := \{\mathbf{x} \in \mathbb{R}^n : F_i(\mathbf{x}) < 0, i = 0, 1, \ldots, s+1\} \tag{9}$$

Then there is an algorithm with time-complexity $s(dMl)^{O(1)}2^{2n\log_2(n)+O(n)}$ which computes a point $\mathbf{x}^* \in Y \cap \mathbb{Z}^n$ or confirms that no such point exists.

Proof. We just need to fill in the steps of the Lenstra-type algorithm above.

1. **Declare Minimum Volume:** Based on the input data, following Lemma 4.1, we have such an $\epsilon > 0$ such that $\langle\epsilon\rangle$ is $l(dn)^{O(1)}$.
2. **Ellipsoid Rounding:** According to Corollary 3.4, choosing $m := n2^n$, we obtain an $O(n)$-rounding, and this is done in time-complexity $s(ln2^n dM\langle\epsilon\rangle)^{O(1)}$. Here we note that $r \leq l$ due to how l is defined here.
3. **Compute a Flatness Direction via Shortest Vector Problem:** This may be done in according to Remark 2.10 and Lemma 2.14 in $2^{O(n)}$ time.
4. **Compute Sublattice:** This may be done according to Lemma 2.1. Since the input vector here is of length $||d||_2 \leq n$, this implies that $||d||_\infty \leq n$. Thus, this step is computable in $n^{O(1)}$ time. Because the original lattice is $\mathbb{Z}^n$, no matrix multiplication is necessary in the Lemma since the original basis matrix is the $n \times n$ identity matrix.
5. **Recurse:** This is possible according to Remark 2.6.

This results in an overall time-complexity of

$$\begin{aligned} s(ln2^n dM)^{O(1)}2^{O(n)}\prod_{i=1}^{n}&(2(i)(O(i))+3) \\ &= s(dMl)^{O(1)}2^{O(n)}(n!)^2 \leq s(dMl)^{O(1)}2^{2n\log_2(n)+O(n)}. \end{aligned}$$

□

From the approach of this method, due to the recursion, we will unlikely be able to obtain better than an $2^{O(n\log(n))}$ time-complexity in terms of the dimension. An open problem is then to find an algorithm that is singly exponential in dimension.

We now provide an outline for the proof of Theorem 1.1, which follows from the same reasoning as [12, Theorem 5.1].

Proof of Theorem 1.1. Bank shows in [6, p. 27] that if a minimum point exists, then there exists a ball of radius $R_* \in \mathbb{Z}$ containing such a point, where the binary length of R_* is $ld^{O(n)}$. (If a smaller bound on this number is known, either because of previous information about the feasible region or a tighter bound is derived, then we will use that and obtain the first complexity given. Otherwise, we use Bank's bound.) We define the polynomial $F_{s+1} \in \mathbb{Z}[\mathbf{x}]$ by $F_{s+1}(\mathbf{x}) \colon = -R_*^2 + \mathbf{x}^T\mathbf{x}$ for $\mathbf{x} \in \mathbb{R}^n$. To solve the optimization problem, we compute the smallest integer $z^* \in \mathbb{Z}$ such that

$$\{\mathbf{x} \in \mathbb{Z}^n \colon F_0(\mathbf{x}) - z^* < 0, \text{ and } F_i(\mathbf{x}) < 0 \text{ for } i = 1, \dots, s+1\} \neq \emptyset.$$

We then apply binary search, testing for integer points using Theorem 4.2, to find an optimal z^* and obtain the desired time-complexity because of the bound given by [6]. □

Acknowledgments

The first author was supported by grant DMS-0636297, and both authors were supported by grant DMS-0914873 of the National Science Foundation.

References

[1] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC '01: Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, pages 601–610, New York, NY, USA, 2001. ACM.

[2] K. M. Anstreicher. Ellipsoidal approximations of convex sets based on the volumetric barrier. *Mathematics of Operations Research*, 24(1):193–203, 1999.

[3] K. M. Anstreicher. Improved complexity for maximum volume inscribed ellipsoids. *SIAM Journal on Optimization*, 13(2):309–320, 2002.

[4] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993.

[5] W. Banaszczyk, A. E. Litvak, A. Pajor, and S. J. Szarek. The flatness theorem for nonsymmetric convex bodies via the local theory of Banach spaces. *Mathematics of Operations Research*, 24(3):728–750, 1999.

[6] B. Bank. Optimization and real equation solving. In *II Escuela de Matemática Aplicada (25 al 29 de agosto de 1997): Notas de los Cursos.* Universidad de Buenos Aires, 1997.

[7] B. Bank and R. Mandel. *Parametric Integer Optimization.* Akademie-Verlag, Berlin, 1988.

[8] D. Bertsimas and R. Weismantel. *Optimization over Integers.* Dynamic Ideas, Belmont, MA, May 2005.

[9] R. L. Burden and J. D. Faires. *Numerical Analysis.* Thomson Brooks/Cole, 8th edition, 2005.

[10] F. Eisenbrand. Integer programming and algorithmic geometry of numbers. In M. Jünger, T. Liebling, D. Naddef, W. Pulleyblank, G. Reinelt, G. Rinaldi, and L. Wolsey, editors, *50 Years of Integer Programming 1958–2008.* Springer-Verlag, 2010.

[11] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag Berlin Heidelberg, 1988.

[12] S. Heinz. Complexity of integer quasiconvex polynomial optimization. *Journal of Complexity*, 21(4):543–556, 2005.

[13] M. Henk. Note on shortest and nearest lattice vectors. *Information Processing Letters*, 61:183–188, 1997.

[14] F. John. Extremum problems with inequalities as subsidiary conditions. *Studies and Essays*, Courant Anniversary Volume:187–204, 1948.

[15] R. Kannan. Improved algorithms for integer programming and related lattice problems. In *STOC '83: Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 193–206, New York, NY, USA, 1983. ACM.

[16] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

[17] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979.

[18] R. Kannan and L. Lovász. Covering minima and lattice point free convex bodies. In *Proc. of the Sixth Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 193–213, New York, NY, USA, 1986. Springer-Verlag New York, Inc.

[19] L. Khachiyan. Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research*, 21(2):307–320, 1996.

[20] L. Khachiyan and L. Porkolab. Integer optimization on convex semialgebraic sets. *Discrete & Computational Geometry*, 23(2):207–224, 2000.

[21] A. Khinchin. A quantitative formulation of Kronecker's theory of approximation (in russian). *Izvestiya Akademii Nauk SSR Seriya Matematika*, 12:113–122, 1948.

[22] M. Kochol. Constructive approximation of a ball by polytopes. *Mathematica Slovaca*, 44(1):99–105, 1994.

[23] M. Kochol. A note on approximation of a ball by polytopes. *Discrete Optimization*, 1(2):229–231, 2004.

[24] P. Kumar and E. A. Yıldırım. Minimum-volume enclosing ellipsoids and core sets. *Journal of Optimization Theory and Applications*, 126:1–21, 2005.

[25] H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[26] D. Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, Mar. 2001. Preliminary version in FOCS 1998.

[27] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations (extended abstract), 2009.

[28] D. Micciancio and P. Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *Proceedings of SODA*. ACM/SIAM, Jan 2010.

[29] Yu. Nesterov. Rounding of convex sets and efficient gradient methods for linear programming problems. *Optimization Methods Software*, 23(1):109–128, 2008.

[30] P. Q. Nguyen and T. Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2(2), 2008.

[31] A. Storjohann and G. Labahn. Asymptotically fast computation of Hermite normal forms of integer matrices. In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, pages 259–266. ACM Press, 1996.

[32] M. J. Todd and E. A. Yıldırım. On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathematics*, 155(13):1731–1744, 2007.

Robert Hildebrand: Department of Mathematics, University of California, Davis, One Shields Avenue, Davis, CA, 95616, USA
E-mail address: `rhildebrand@math.ucdavis.edu`

Matthias Köppe: Department of Mathematics, University of California, Davis, One Shields Avenue, Davis, CA, 95616, USA
E-mail address: `mkoeppe@math.ucdavis.edu`