SIAM J. DISCRETE MATH.
Vol. 28, No. 3, pp. 1405–1422

FINDING A CLOSEST POINT IN A LATTICE OF VORONOI'S FIRST KIND*

ROBBY G. MCKILLIAM†, ALEX GRANT†, AND I. VAUGHAN L. CLARKSON‡

Abstract. We show that for those lattices of Voronoi's first kind with known obtuse superbasis, a closest lattice point can be computed in $O(n^4)$ operations, where n is the dimension of the lattice. To achieve this a series of relevant lattice vectors that converges to a closest lattice point is found. We show that the series converges after at most n terms. Each vector in the series can be efficiently computed in $O(n^3)$ operations using an algorithm to compute a minimum cut in an undirected flow network.

Key words. lattices, closest point algorithm, closest vector problem

AMS subject classification. 11Y16

DOI. 10.1137/140952806

1. Introduction.

An n-dimensional *lattice* Λ is a discrete set of vectors from $\mathbb{R}^m$, $m \geq n$, formed by the integer linear combinations of a set of linearly independent basis vectors $b_1, \dots, b_n$ from $\mathbb{R}^m$ [11]. That is, Λ consists of all those vectors, or *lattice points*, $x \in \mathbb{R}^m$, satisfying

$$x = b_1 u_1 + b_2 u_2 + \cdots + b_n u_n \qquad u_1, \dots, u_n \in \mathbb{Z}.$$

Given a lattice Λ in $\mathbb{R}^m$ and a vector $y \in \mathbb{R}^m$, a problem of interest is to find a lattice point $x \in \Lambda$ such that the squared Euclidean norm

$$\|y - x\|^2 = \sum_{i=1}^{m} (y_i - x_i)^2$$

is minimized. This is called the *closest lattice point problem* (or *closest vector problem*) and a solution is called a *closest lattice point* (or simply *closest point*) to y. A related problem is to find a lattice point of minimum nonzero Euclidean length, that is, a lattice point of length

$$\min_{x \in \Lambda \backslash \{\mathbf{0}\}} \|x\|^2,$$

where $\Lambda \backslash \{\mathbf{0}\}$ denotes the set of lattice points not equal to the origin $\mathbf{0}$. This is called the *shortest vector problem* and a solution is called a *short vector*.

The closest lattice point problem and the shortest vector problem have interested mathematicians and computer scientists due to their relationship with integer programming [29, 26, 4], the factoring of polynomials [28], and cryptanalysis [25, 43, 41]. Solutions of the closest lattice point problem have engineering applications. For example, if a lattice is used as a vector quantizer then the closest lattice point corresponds

*Received by the editors January 15, 2014; accepted for publication (in revised form) July 30, 2014; published electronically September 9, 2014.
http://www.siam.org/journals/sidma/28-3/95280.html

†Institute for Telecommunications Research, University of South Australia, 5006 Adelaide, Australia (robby.mckilliam@unisa.edu.au, alex.grant@unisa.edu.au).
‡School of Information Technology and Electrical Engineering, University of Queensland, 4072 Brisbane, Australia (v.clarkson@uq.edu.au).

to the minimum distortion point [9, 8, 7]. If the lattice is used as a code, then the closest lattice point corresponds to what is called *lattice decoding* and has been shown to yield arbitrarily good codes [18, 17]. The closest lattice point problem occurs in communications systems involving multiple antennas [48, 56]. The unwrapping of phase data for location estimation can be posed as a closest lattice point problem and this has been applied to the global positioning system [52, 22]. The problem has also found application to circular statistics [36], single frequency estimation [37], and related signal processing problems [32, 6, 33, 46, 38].

The closest lattice point problem is known to be NP-hard [40, 15, 47, 20, 24]. Nevertheless, algorithms exist that can compute a closest lattice point in reasonable time if the dimension is small [45, 26, 1]. These algorithms require a number of operations that grows as $O(n^{O(n)})$ or $O(n^{O(n^2)})$, where n is the dimension of the lattice. Recently, Micciancio and Voulgaris [42] described a solution for the closest lattice point problem that requires a number of operations that grows as $O(2^{2n})$. This single exponential growth in complexity is the best known.

Although the problem is NP-hard in general, fast algorithms are known for specific highly regular lattices, such as the integer lattice $\mathbb{Z}^n$, the root lattices A_n and D_n, their dual lattices A_n^* and D_n^*, and the related Coxeter lattices [11, Chap. 4], [7, 34, 39]. In this paper we consider a particular class of lattices, those of *Voronoi's first kind* [10, 54, 55]. Each lattice of Voronoi's first kind has what is called an *obtuse superbasis*. We show that if the obtuse superbasis is known, then a closest lattice point can be computed in $O(n^4)$ operations. This is achieved by enumerating a series of *relevant vectors* of the lattice. Each relevant vector in the series can be computed in $O(n^3)$ operations using an algorithm for computing a minimum cut in an undirected flow network [44, 49, 53, 12]. We show that the series converges to a closest lattice point after at most n terms, resulting in $O(n^4)$ operations in total. This result extends upon a recent result by some of the authors showing that a short vector in a lattice of Voronoi's first kind can be found by computing a minimum cut in a weighted graph [35].

Our results can be placed in the context of a modification of the closest lattice point problem called the *closest vector problem with preprocessing* [40, 20, 47, 3, 27, 14]. In this problem some "advice" about the lattice is assumed to be given. The advice might come in the form of a particular basis for the lattice or might take other forms. The advice may be used to compute a closest lattice point, hopefully with reduced complexity. Our algorithm can be viewed as an efficient solution for the closest vector problem with preprocessing for the lattices of Voronoi's first kind. The advice given is the obtuse superbasis.

The paper is structured as follows. Section 2 describes the relevant vectors and the *Voronoi cell* of a lattice. Section 3 describes a procedure to find a closest lattice point by enumerating a series of relevant vectors. The series is guaranteed to converge to a closest point after a finite number of terms. In general the procedure might be computationally expensive because the number of terms required might be large and because computation of each relevant vector in the series might be expensive. Section 4 describes lattices of Voronoi's first kind and their obtuse superbasis. In section 5 it is shown that for these lattices the series of relevant vectors results in a closest lattice point after at most n terms. Section 6 shows that each relevant vector in the series can be computed in $O(n^3)$ operations by computing a minimum cut in an undirected flow network. Section 7 discusses some potential applications of this algorithm and poses some interesting questions for future research.

2. Voronoi cells and relevant vectors. The (closed) *Voronoi cell*, denoted $\mathrm{Vor}(\Lambda)$, of a lattice Λ in $\mathbb{R}^m$ is the subset of $\mathbb{R}^m$ containing all points closer or of equal distance (here with respect to the Euclidean norm) to the lattice point at the origin than to any other lattice point. The Voronoi cell is an m-dimensional convex polytope that is symmetric about the origin.

Equivalently, the Voronoi cell can be defined as the intersection of the half-spaces

$$H_v = \{x \in \mathbb{R}^n \mid \|x\| \leq \|x - v\|\}$$
$$= \left\{x \in \mathbb{R}^n \mid x \cdot v \leq \frac{1}{2} v \cdot v\right\}$$

for all $v \in \Lambda\backslash\{\mathbf{0}\}$. We denote by $x \cdot v$ the inner product of vectors x and v. It is not necessary to consider all $v \in \Lambda\backslash\{\mathbf{0}\}$ to define the Voronoi cell. The *relevant vectors* are those lattice points $v \in \Lambda\backslash\{\mathbf{0}\}$ for which

$$v \cdot x < x \cdot x \qquad \text{for all } x \in \Lambda\backslash\{\mathbf{0}\}.$$

Denote by $\mathrm{Rel}(\Lambda)$ the set of relevant vectors of the lattice Λ. The Voronoi cell is the intersection of the half-spaces corresponding with the relevant vectors, that is,

$$\mathrm{Vor}(\Lambda) = \cap_{v \in \mathrm{Rel}(\Lambda)} H_v.$$

The closest lattice point problem and the Voronoi cell are related in that $x \in \Lambda$ is a closest lattice point to y if and only $y - x \in \mathrm{Vor}(\Lambda)$, that is, if and only if

$$\|y - x\| \leq \|y - x - v\| \qquad \text{for all } v \in \mathrm{Rel}(\Lambda). \tag{2.1}$$

If s is a short vector in a lattice Λ then

$$\rho = \frac{\|s\|}{2} = \frac{1}{2} \min_{x \in \Lambda\backslash\{\mathbf{0}\}} \|x\|$$

is called the *packing radius* (or *inradius*) of Λ [11]. The packing radius is the minimum distance between the boundary of the Voronoi cell and the origin. It is also the radius of the largest sphere that can be placed at every lattice point such that no two spheres intersect (see Figure 1). The following well-known results will be useful.

PROPOSITION 2.1. *Let $\Lambda \subset \mathbb{R}^m$ be an n-dimensional lattice. For $r \in \mathbb{R}$ let $\lfloor r \rfloor$ denote the largest integer less than or equal to r. Let $t \in \mathbb{R}^m$. The number of lattice points inside the scaled and translated Voronoi cell $r\,\mathrm{Vor}(\Lambda) + t$ is at most $(\lfloor r \rfloor + 1)^n$.*

Proof. Let $V \subset \mathrm{Vor}(\Lambda)$ contain all those points from the interior of the closed Voronoi cell $\mathrm{Vor}(\Lambda)$, but with boundaries defined so that V tessellates $\mathbb{R}^m$ under translations by Λ. That is, $\mathbb{R}^m = \cup_{x \in \Lambda}(V + x)$ and the intersection $(V + x) \cap (V + y)$ is empty for distinct lattice points x and y. For positive integer k, the scaled and translated cell $kV + t$ contains precisely one coset representative for each element of the quotient group $\Lambda/k\Lambda$ [31, section 2.4]. There are k^n coset representatives. Thus, the number of lattice points inside $r\,\mathrm{Vor}(\Lambda) + t \subset (\lfloor r \rfloor + 1)V + t$ is at most $(\lfloor r \rfloor + 1)^n$. ∎

PROPOSITION 2.2. *Let $\Lambda \subset \mathbb{R}^m$ be an n-dimensional lattice with packing radius ρ. Let S be an m-dimensional hypersphere of radius r centered at $t \in \mathbb{R}^m$. The number of lattice points from Λ in the sphere S is at most $(\lfloor r/\rho \rfloor + 1)^n$.*

Proof. The packing radius ρ is the Euclidean length of a point on the boundary of the Voronoi cell $\mathrm{Vor}(\Lambda)$ that is closest to the origin. Therefore, the sphere S is a subset of $\mathrm{Vor}(\Lambda)$ scaled by r/ρ and translated by t. That is, $S \subset r/\rho\,\mathrm{Vor}(\Lambda) + t$. The proof follows because the number of lattice points in $r/\rho\,\mathrm{Vor}(\Lambda) + t$ is at most $(\lfloor r/\rho \rfloor + 1)^n$ by Proposition 2.1. ∎

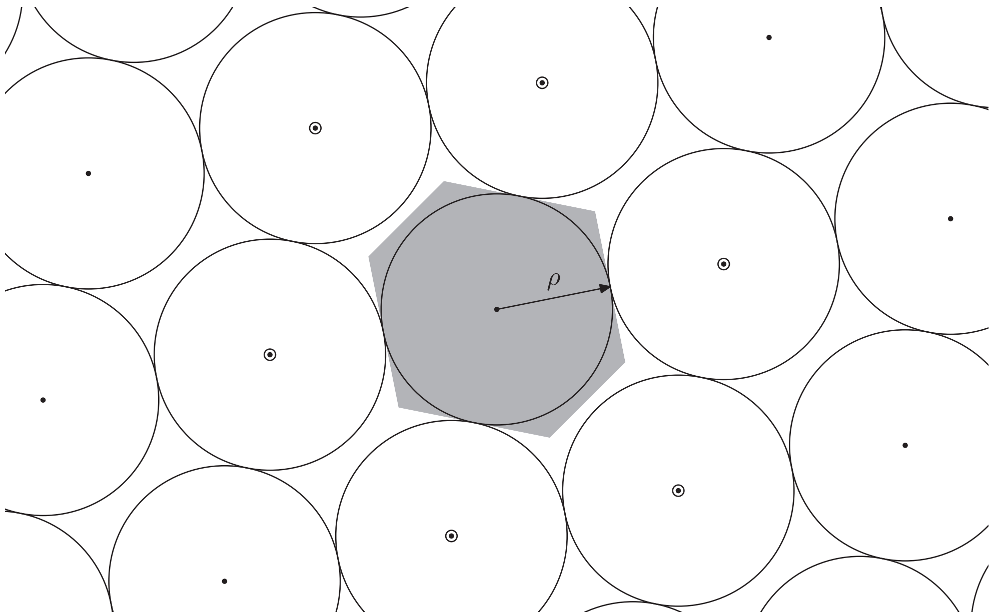

FIG. 1. *The 2-dimensional lattice with basis vectors* $(3, 0.6)$ *and* $(0.6, 3)$. *The lattice points are represented by dots and the relevant vectors are circled. The Voronoi cell* $\mathrm{Vor}(\Lambda)$ *is the shaded region and the packing radius* ρ *and corresponding sphere packing (circles) are depicted.*

3. Finding a closest lattice point by a series of relevant vectors. Let Λ be a lattice in $\mathbb{R}^m$ and let $y \in \mathbb{R}^m$. A simple method to compute a lattice point $x \in \Lambda$ closest to y is as follows. Let x_0 be some lattice point from Λ, for example, the origin. Motivated by Sommer, Feder, and Shalvi [51] and Micciancio and Voulgaris [42], we consider the following iteration,

$$\begin{aligned} x_{k+1} &= x_k + v_k, \\ v_k &= \arg \min_{v \in \mathrm{Rel}(\Lambda) \cup \{\mathbf{0}\}} \|y - x_k - v\|, \end{aligned} \tag{3.1}$$

where $\mathrm{Rel}(\Lambda) \cup \{\mathbf{0}\}$ is the set of relevant vectors of Λ including the origin. This iterative procedure is depicted in Figure 2. The minimum over $\mathrm{Rel}(\Lambda) \cup \{\mathbf{0}\}$ may not be unique, that is, there may be multiple vectors from $\mathrm{Rel}(\Lambda) \cup \{\mathbf{0}\}$ that are closest to $y - x_k$. In this case, any one of the minimizers may be chosen. The results that we will describe do not depend on this choice. We make the following straightforward propositions.

PROPOSITION 3.1. *At the kth iteration either* x_k *is a closest lattice point to* y *or* $\|y - x_k\| > \|y - x_{k+1}\|$.

Proof. If x_k is a closest lattice point to y then $\|y - x_k\| \le \|y - x_{k+1}\|$ by definition. On the other hand if x_k is not a closest lattice point to y we have $y - x_k \notin \mathrm{Vor}(\Lambda)$ and from (2.1) there exists a relevant vector v such that

$$\begin{aligned} \|y - x_k\| &> \|y - x_k - v\| \\ &\ge \arg \min_{v \in \mathrm{Rel}(\Lambda) \cup \{\mathbf{0}\}} \|y - x_k - v\| \\ &= \|y - x_k - v_k\| \\ &= \|y - x_{k+1}\|. \qquad \square \end{aligned}$$

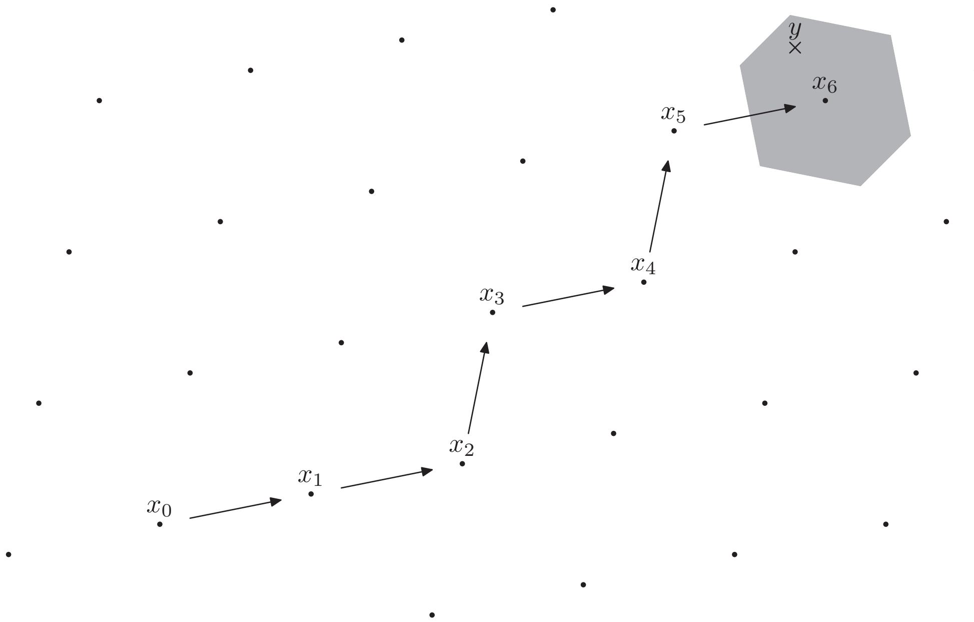

FIG. 2. *Example of the iterative procedure described in* (3.1) *to compute a closest lattice point to* $y = (4, 3.5)$ *(marked with a cross) in the* 2*-dimensional lattice generated by basis vectors* $(2, 0.4)$ *and* $(0.4, 2)$*. The initial lattice point for the iteration is* $x_0 = (-4.4, -2.8)$*. The shaded region is the Voronoi cell surrounding the closest lattice point* $x_6 = (4.4, 2.8)$.

PROPOSITION 3.2. *There is a finite number* K *such that* $x_K, x_{K+1}, x_{K+2}, \dots$ *are all closest points to* y.

Proof. Suppose no such finite K exists, then

$$\|y - x_0\| > \|y - x_1\| > \|y - x_2\| > \cdots$$

and so $x_0, x_1, \dots$ is an infinite sequence of distinct (due to the strict inequality) lattice points all contained inside an n-dimensional hypersphere of radius $r = \|y - x_0\|$ centered at y. This is a contradiction because, if ρ is the packing radius of the lattice, then less than $(\lfloor r/\rho \rfloor + 1)^n$ lattice points lie inside this sphere by Proposition 2.2. ∎

Proposition 3.2 asserts that after some finite number K of iterations the procedure arrives at x_K, a closest lattice point to y. Using Proposition 3.1 we can detect that x_K is a closest lattice point by checking whether $\|y - x_K\| \leq \|y - x_{K+1}\|$. This simple iterative approach to compute a closest lattice point is related to what is called the *iterative slicer* [51]. Micciancio and Voulgaris [42] describe a related, but more sophisticated, iterative algorithm that can compute a closest point in a number of operations that grows exponentially as $O(2^{2n})$. This single exponential growth in complexity is the best known.

Two factors contribute to the computational complexity of this iterative approach to compute a closest lattice point. The first factor is computing the minimum over the set $\text{Rel}(\Lambda) \cup \{\mathbf{0}\}$ in (3.1). In general a lattice can have as many as $2^{n+1} - 2$ relevant vectors so computing a minimizer directly can require a number of operations that grows exponentially with n. To add to this it is often the case that the set of relevant vectors $\text{Rel}(\Lambda)$ must be stored in memory so the algorithm can require an amount of memory that grows exponentially with n [42, 51]. We will show that for a lattice of Voronoi's first kind the set of relevant vectors has a compact representation in terms of what is called its *obtuse superbasis*. To store the obtuse superbasis requires an amount of memory of order $O(n^2)$ in the worst case. We also show that for a lattice of Voronoi's first kind the minimization over $\text{Rel}(\Lambda) \cup \{\mathbf{0}\}$ in (3.1) can be solved efficiently by computing a minimum cut in an undirected flow network. Using known algorithms a minimizer can be computed in $O(n^3)$ operations [21, 16, 12].

The other factor affecting the complexity is the number of iterations required before the algorithm arrives at a closest lattice point, that is, the size of K. Proposition 2.2 suggests that this number might be as large as $(\lfloor r/\rho \rfloor + 1)^n$, where $r = \|y - x_0\|^2$ and ρ is the packing radius of the lattice. Thus, the number of iterations required might grow exponentially with n. The number of iterations required depends on the lattice point that starts the iteration x_0. It is helpful for x_0 to be, in some sense, a close approximation of the closest point x_K. Unfortunately, computing close approximations of a closest lattice point is known to be computationally difficult [20, 47, 2, 3, 14]. We will show that for a lattice of Voronoi's first kind a simple and easy to compute choice for x_0 ensures that a closest lattice point is reached in at most n iterations and so $K \leq n$. Combining this with the fact that each iteration of the algorithm requires $O(n^3)$ operations results in an algorithm that requires $O(n^4)$ operations to compute a closest point in a lattice of Voronoi's first kind.

4. Lattices of Voronoi's first kind. An n-dimensional lattice Λ is said to be of *Voronoi's first kind* if it has what is called an *obtuse superbasis* [10]. That is, there exists a set of $n+1$ vectors $b_1, \dots, b_{n+1}$ such that $b_1, \dots, b_n$ are a basis for Λ,

$$b_1 + b_2 \cdots + b_{n+1} = \mathbf{0} \tag{4.1}$$

(the *superbasis* condition), and the inner products satisfy

$$q_{ij} = b_i \cdot b_j \leq 0 \qquad \text{for} \qquad i, j = 1, \dots, n+1, i \neq j \tag{4.2}$$

(the *obtuse* condition). The q_{ij} are called the *Selling parameters* [50]. It is known that all lattices of dimension less than or equal to 3 are of Voronoi's first kind [10]. An interesting property of lattices of Voronoi's first kind is that their relevant vectors have a straightforward description.

THEOREM 4.1 (Conway and Sloane [10, Theorem 3]). *Let Λ be a lattice of Voronoi's first kind with obtuse superbasis $b_1, \dots, b_{n+1}$. The relevant vectors of Λ are of the form*

$$\sum_{i \in I} b_i$$

where I is a strict subset of $\{1, 2, \dots, n+1\}$ that is not empty, i.e., $I \subset \{1, 2, \dots, n+1\}$ and $I \neq \emptyset$.

Classical examples of lattices of Voronoi's first kind are the n-dimensional root lattice A_n and its dual lattice A_n^* [11]. For A_n and A_n^* there exist efficient algorithms that can compute a closest lattice point in $O(n)$ operations [39, 7]. For this reason we do not recommend using the algorithm described in this paper for A_n and A_n^*. The fast algorithms for A_n and A_n^* rely of the special structure of these lattices and are not applicable to other lattices. In contrast, the algorithm we describe here works for all lattices of Voronoi's first kind. Questions that arise are how "large" (in some sense) is the set of lattices of Voronoi's first kind? Are there lattices of Voronoi's first kind that are useful in applications such as coding, quantization, or signal processing? These questions are discussed in section 7. We now focus on the problem of computing a closest point in a lattice of Voronoi's first kind.

5. A series of relevant vectors from a lattice of Voronoi's first kind. Let $\Lambda \subset \mathbb{R}^m$ be an n-dimensional lattice of Voronoi's first kind with obtuse superbasis $b_1, \dots, b_{n+1}$ and let $y \in \mathbb{R}^m$. We want to find n integers $w_1, \dots, w_n$ that minimize

$$\left\| y - \sum_{i=1}^{n} b_i w_i \right\|^2 .$$

We can equivalently find $n+1$ integers $w_1, \dots, w_{n+1}$ that minimize

$$\left\| y - \sum_{i=1}^{n+1} b_i w_i \right\|^2 .$$

The iterative procedure described in (3.1) will be used to do this. In what follows it is assumed that y lies in the space spanned by the basis vectors $b_1, \dots, b_n$. This assumption is without loss of generality because x is a closest lattice point to y if and only if x is a closest lattice point to the orthogonal projection of y into the space spanned by $b_1, \dots, b_n$. Let

$$B = (b_1 \; b_2 \; \dots \; b_{n+1})$$

be the m by $n+1$ matrix with columns given by $b_1, \dots, b_{n+1}$ and let $z \in \mathbb{R}^{n+1}$ be a column vector such that $y = Bz$. We now want to find a column vector $w = (w_1, \dots, w_{n+1})'$ of integers such that

$$\|B(z - w)\|^2 \tag{5.1}$$

is minimized. Define the column vector $u_0 = \lfloor z \rceil$, where $\lfloor \cdot \rceil$ operates on vectors elementwise. In view of Theorem 4.1 the iterative procedure (3.1) to compute a closest lattice point can be written in the form

$$x_{k+1} = Bu_{k+1}, \tag{5.2}$$

$$u_{k+1} = u_k + t_k,$$

$$t_k = \arg \min_{t \in \{0,1\}^{n+1}} \|B(z - u_k - t)\|, \tag{5.3}$$

where $\{0,1\}^{n+1}$ denotes the set of column vectors of length $n+1$ with elements equal to zero or one. The procedure is initialized at the lattice point $x_0 = Bu_0 = B\lfloor z \rceil$. This choice of initial lattice point is important. In section 6 we show how minimization over $\{0,1\}^{n+1}$ in (5.3) can be computed efficiently in $O(n^3)$ operations by computing a minimum cut in an undirected flow network. The minimizer may not be unique corresponding with the existence of multiple minimum cuts. In this case any one of the minimizers may be chosen. Our results do not depend on this choice. In the remainder of this section we prove that this iterative procedure results in a closest lattice point after at most n iterations. That is, we show that there exists a positive integer $K \leq n$ such that x_K is a closest lattice point to $y = Bz$.

We first provide an intuitive description of our proof technique. Denote by $\min(p)$ and $\max(p)$ the minimum and maximum values obtained by the elements of the vector $p \in \mathbb{R}^{n+1}$ and define the function

$$\operatorname{rng}(p) = \max(p) - \min(p).$$

Observe that $\operatorname{rng}(p)$ cannot be negative and that if $\operatorname{rng}(p) = 0$ then all of the elements of p are equal. Let ℓ be a nonegative integer. Say that a lattice point x is ℓ-*close* to y if there exists a $v \in \mathbb{Z}^{n+1}$ with $\operatorname{rng}(v) = \ell$ such that $x + Bv$ is a closest point to y. We will show (Lemma 5.6) that the lattice point $x_0 = B\lfloor z \rceil$ that initializes the iterative procedure (5.2) is K-close to y where $K \leq n$. We then prove (Lemma 5.8) that if the lattice point x_k obtained on the kth iteration of the procedure is ℓ-close, then the lattice point x_{k+1} obtained on the next iteration is $(\ell - 1)$-close. Since x_0 is K-close it follows that after $K \leq n$ iterations the lattice point x_K is 0-close. At this stage it is guaranteed that x_K itself is a closest lattice point to y. This is shown in the following lemma.

LEMMA 5.1. *If x is a lattice point that is* 0*-close to y then x is a closest lattice point to y.*

Proof. Because x is 0-close there exists a $v \in \mathbb{Z}^{n+1}$ with $\operatorname{rng}(v) = 0$ such that $x + Bv$ is a closest point to y. Because $\operatorname{rng}(v) = 0$ all elements from v are identical, that is, $v_1 = v_2 = \cdots = v_{n+1}$. In this case $Bv = \sum_{i=1}^{n+1} v_n b_n = v_1 \sum_{i=1}^{n+1} b_n = \mathbf{0}$ as a result of the superbasis condition (4.1). Thus $x = x + Bv$ is a closest point to y. ∎

We now proceed with a formal proof culminating in Theorem 5.9 stated at the end of this section. We first introduce some notation. For S a subset of indices $\{1, \dots, n+1\}$, let $\mathbf{1}_S$ denote the column vector of length $n+1$ with ith element equal to one if $i \in S$ and zero otherwise. For $S \subseteq \{1, \dots, n+1\}$ and $p \in \mathbb{R}^{n+1}$ define the function

$$\Phi(S, p) = \sum_{i \in S} \sum_{j \notin S} q_{ij}(1 + 2p_i - 2p_j),$$

where $q_{ij} = b_i \cdot b_j$ are the Selling parameters from (4.2). Denote by $\bar{S}$ the complement of the set of indices S, that is $\bar{S} = \{i \in \{1, \dots, n+1\} \mid i \notin S\}$.

LEMMA 5.2. *Let $p \in \mathbb{R}^{n+1}$ and let S and T be subsets of the indices of p. The following equalities hold:*
1. $\|Bp\|^2 - \|B(p + \mathbf{1}_S)\|^2 = \Phi(S, p)$;
2. $\|Bp\|^2 - \|B(p - \mathbf{1}_S)\|^2 = \Phi(\bar{S}, p)$;
3. $\|Bp\|^2 - \|B(p + \mathbf{1}_S - \mathbf{1}_T)\|^2 = \Phi(S, p) + \Phi(\bar{T}, p) + 2\sum_{i \in S}\sum_{j \in T} q_{ij}$.

Proof. Part 3 follows immediately from parts 1 and 2 because

$$\|Bp\|^2 - \|B(p + \mathbf{1}_S - \mathbf{1}_T)\|^2 = \|Bp\|^2 - \|B(p + \mathbf{1}_S)\|^2 + \|Bp\|^2 - \|B(p - \mathbf{1}_T)\|^2 + 2\sum_{i \in S}\sum_{j \in T} q_{ij}.$$

We give a proof for part 1. The proof for part 2 is similar. Put $Q = B'B$ where superscript $'$ indicates the vector or matrix transpose. The $n+1$ by $n+1$ matrix Q has elements given by the Selling parameters, that is, $Q_{ij} = q_{ij} = b_i \cdot b_j$. Denote by $\mathbf{1}$ the column vector of length $n+1$ containing all ones. Now $B\mathbf{1} = \sum_{i=1}^{n+1} b_i = \mathbf{0}$ as a result of the superbasis condition (4.1) and so $Q\mathbf{1} = \mathbf{0}$. Since $\mathbf{1}_S = \mathbf{1} - \mathbf{1}_{\bar{S}}$ it follows that $Q\mathbf{1}_S = -Q\mathbf{1}_{\bar{S}}$. With $\circ$ the elementwise vector product, i.e., the Schur or Hadamard product, we have

$$\begin{aligned}\|Bp\|^2 - \|B(p+\mathbf{1}_S)\|^2 &= -\mathbf{1}_S'Q\mathbf{1}_S - 2p'Q\mathbf{1}_S\\ &= \mathbf{1}_S'Q\mathbf{1}_{\bar{S}} - 2p'Q\mathbf{1}_S\\ &= \mathbf{1}_S'Q\mathbf{1}_{\bar{S}} - 2(p\circ\mathbf{1}_{\bar{S}})'Q\mathbf{1}_S - 2(p\circ\mathbf{1}_S)'Q\mathbf{1}_S\\ &= \mathbf{1}_S'Q\mathbf{1}_{\bar{S}} - 2(p\circ\mathbf{1}_{\bar{S}})'Q\mathbf{1}_S + 2(p\circ\mathbf{1}_S)'Q\mathbf{1}_{\bar{S}}\end{aligned}$$

which is precisely $\Phi(S,p)$. ◻

Define the function $\operatorname{subr}(p)$ to return the largest subset, say S, of the indices of p such that $\min\{p_i, i\in S\} - \max\{p_i, i\notin S\} \geq 2$. If no such subset exists then $\operatorname{subr}(p)$ is the empty set $\emptyset$. For example,

$$\operatorname{subr}(2,-1,4) = \{1,3\}, \qquad \operatorname{subr}(2,1,3)=\emptyset, \quad \operatorname{subr}(1,3,1)=\{2\}.$$

To make the definition of subr clear we give the following alternative and equivalent definition. Let $p\in\mathbb{R}^n$ and let σ be the permutation of the indices $\{1,\dots,n\}$ that puts the elements of p in ascending order, that is,

$$p_{\sigma(1)} \leq p_{\sigma(2)} \leq \cdots \leq p_{\sigma(n)}.$$

Let T be the smallest integer from $\{2,\dots,n\}$ such that $p_{\sigma(T)} - p_{\sigma(T-1)} \geq 2$. If no such integer T exists then $\operatorname{subr}(p)=\emptyset$. Otherwise

$$\operatorname{subr}(p) = \{\sigma(T), \sigma(T+1), \dots, \sigma(n)\}.$$

The following straightforward property of subr will be useful.

PROPOSITION 5.3. *Let $p\in\mathbb{Z}^{n+1}$. If $\operatorname{subr}(p)=\emptyset$ then $\operatorname{rng}(p)\leq n$.*

Proof. Let σ be the permutation of the indices $\{1,\dots,n+1\}$ that puts the elements of p in ascending order. Because $\operatorname{subr}(p)=\emptyset$ and because the elements of p are integers we have $p_{\sigma(i+1)} \leq p_{\sigma(i)}+1$ for all $i=1,\dots,n$. It follows that

$$p_{\sigma(n+1)} \leq p_{\sigma(n)}+1 \leq p_{\sigma(n-1)}+2 \leq \cdots \leq p_{\sigma(1)}+n$$

and so $\operatorname{rng}(p) = p_{\sigma(n+1)} - p_{\sigma(1)} \leq n$. ◻

Finally, define the function

$$\operatorname{decrng}(p) = p - \mathbf{1}_{\operatorname{subr}(p)}$$

that decrements those elements from p with indices from $\operatorname{subr}(p)$. If $\operatorname{subr}(p)=\emptyset$, then $\operatorname{decrng}(p)=p$, that is, decrng does not modify p. On the other hand, if $\operatorname{subr}(p)\neq\emptyset$ then

$$\operatorname{rng}\big(\operatorname{decrng}(p)\big) = \operatorname{rng}(p)-1$$

because $\operatorname{subr}(p)$ contains all those indices i such that $p_i = \max(p)$. By repeatedly applying decrng to a vector one eventually obtains a vector for which further application of decrng has no effect. For example,

$$\begin{aligned}&\operatorname{decrng}(2,-1,4) = (2,-1,4) - \mathbf{1}_{\operatorname{subr}(2,-1,4)} = (2,-1,4) - \mathbf{1}_{\{1,3\}} = (1,-1,3),\\ &\operatorname{decrng}(1,-1,3) = (1,-1,3) - \mathbf{1}_{\{1,3\}} = (0,-1,2),\\ &\operatorname{decrng}(0,-1,2) = (0,-1,2) - \mathbf{1}_{\{3\}} = (0,-1,1),\\ &\operatorname{decrng}(0,-1,1) = (0,-1,1) - \mathbf{1}_{\emptyset} = (0,-1,1).\end{aligned}$$

This will be a useful property so we state it formally in the following proposition.

PROPOSITION 5.4. *Let $p \in \mathbb{R}^{n+1}$ and define the infinite sequence $d_0, d_1, d_2, \dots$ of vectors according to $d_0 = p$ and $d_{k+1} = \operatorname{decrng}(d_k)$. There is a finite integer T such that $d_T = d_{T+1} = d_{T+2} = \cdots$.*

Proof. Assume that no such T exists. In this case $\operatorname{decrng}(d_k) \neq d_k$ for all positive integers k and so

$$\operatorname{rng}(d_k) = \operatorname{rng}(d_{k-1}) - 1 = \operatorname{rng}(d_{k-2}) - 2 = \cdots = \operatorname{rng}(p) - k.$$

Choosing $k > \operatorname{rng}(p)$ we have $\operatorname{rng}(d_k) < 0$ contradicting that $\operatorname{rng}(d_k)$ is nonnegative. ∎

We are now ready to study properties of the lattice point $x_0 = B\lfloor z \rceil$ that initializes the iterative procedure (5.2).

LEMMA 5.5. *If $v \in \mathbb{Z}^{n+1}$ is such that $B(\lfloor z \rceil + v)$ is a closest lattice point to $y = Bz$, then $B\big(\lfloor z \rceil + \operatorname{decrng}(v)\big)$ is also a closest lattice point to y.*

Proof. The lemma is trivial if $\operatorname{subr}(v) = \emptyset$ so that $\operatorname{decrng}(v) = v$. It remains to prove the lemma when $\operatorname{subr}(v) \neq \emptyset$. In this case put $S = \operatorname{subr}(v)$ and put

$$u = \operatorname{decrng}(v) = v - \mathbf{1}_S.$$

Let $\zeta = z - \lfloor z \rceil$ be the column vector containing the fractional parts of the elements of z. We have $\zeta - u = \zeta - v + \mathbf{1}_S$. Applying part 1 of Lemma 5.2 with $p = \zeta - v$ we obtain

$$\begin{aligned} \|B(\zeta - v)\|^2 - \|B(\zeta - u)\|^2 &= \Phi(S, \zeta - v) \\ &= \sum_{i \in S} \sum_{j \notin S} q_{ij} \big(1 + 2(\zeta_i - \zeta_j) - 2(v_i - v_j)\big). \end{aligned} \tag{5.4}$$

Observe that $\zeta_i = z_i - \lfloor z_i \rceil \in [0, 1)$ for all $i = 1, \dots, n+1$ and so $-1 < \zeta_i - \zeta_j < 1$ for all $i, j = 1, \dots, n+1$. Also, for $i \in S$ and $j \notin S$ we have

$$v_i - v_j \geq \min\{v_i, i \in S\} - \max\{v_j, j \notin S\} \geq 2$$

by the definition of $\operatorname{subr}(v) = S$. Thus,

$$1 + 2(\zeta_i - \zeta_j) - 2(v_i - v_j) < 1 + 2 - 4 = -1 < 0 \qquad \text{for } i \in S \text{ and } j \notin S.$$

Substituting this inequality into (5.4) and using that $q_{ij} \leq 0$ for $i \neq j$ (the obtuse condition (4.2)) we find that

$$\|B(z - \lfloor z \rceil - v)\|^2 - \|B(z - \lfloor z \rceil - u)\|^2 \geq 0.$$

It follows that $B(\lfloor z \rceil + u) = B\big(\lfloor z \rceil + \operatorname{decrng}(v)\big)$ is a closest lattice point to $y = Bz$ whenever $B(\lfloor z \rceil + v)$ is. ∎

LEMMA 5.6. *There exists a closest lattice point to $y = Bz$ in the form $B(\lfloor z \rceil + v)$, where $v \in \mathbb{Z}^{n+1}$ with $\operatorname{rng}(v) \leq n$.*

Proof. Let $d_0 \in \mathbb{Z}^{n+1}$ be such that $B(\lfloor z \rceil + d_0)$ is a closest lattice point to y. Define the sequence of vectors $d_0, d_1, \dots$ from $\mathbb{Z}^{n+1}$ according to the recursion $d_{k+1} = \operatorname{decrng}(d_k)$. It follows from Lemma 5.5 that $B(\lfloor z \rceil + d_k)$ is a closest lattice point for all positive integers k. By Proposition 5.4 there is a finite T such that

$$d_{T+1} = d_T = \operatorname{decrng}(d_T).$$

Thus $\text{subr}(d_T) = \emptyset$ and $\text{rng}(d_T) \leq n$ by Proposition 5.3. The proof follows with $v = d_T$. $\square$

Recall that a lattice point x is said to be ℓ-close to y if there exists a $v \in \mathbb{Z}^{n+1}$ with $\text{rng}(v) = \ell$ such that $x + Bv$ is a closest point to y. Lemma 5.6 above asserts that the lattice point $x_0 = B\lfloor z \rceil$ that initializes the iterative procedure is K-close to y, where $K \leq n$. We now prove Lemma 5.8 from which it will follow that if the lattice point x_k obtained on the kth iteration of the procedure is ℓ-close, then the lattice point x_{k+1} obtained on the next iteration is $(\ell - 1)$-close. Before giving the proof of Lemma 5.8 we require the following simple result.

LEMMA 5.7. *Let $h \in \{0,1\}^{n+1}$ and $v \in \mathbb{Z}^{n+1}$ with $\text{rng}(v) \geq 1$. Suppose that $h_i = 0$ whenever $v_i = \min(v)$ and that $h_i = 1$ whenever $v_i = \max(v)$. Then*

$$h_i - h_j \leq v_i - v_j$$

when either $v_i = \max(v)$ or $v_j = \min(v)$.

Proof. If $v_i = \max(v)$ then $h_i = 1$ and we need only show that $1 - h_j \leq \max(v) - v_j$ for all j. If $v_j = \max(v)$ then $h_j = 1$ and the results hold since $1 - h_j = 0 = \max(v) - v_j$. Otherwise if $v_j < \max(v)$ then $1 - h_j \leq 1 \leq \max(v) - v_j$ because $\max(v)$ and v_j are integers.

Now, if $v_j = \min(v)$ then $h_j = 0$ and we need only show that $h_i \leq v_i - \min(v)$ for all i. If $v_i = \min(v)$ then $h_i = 0 = v_i - \min(v)$ and the results hold. Otherwise if $v_i > \min(v)$ then $h_i \leq 1 \leq v_i - \min(v)$ because $\min(v)$ and v_i are integers. $\square$

LEMMA 5.8. *Let Bu with $u \in \mathbb{Z}^{n+1}$ be a lattice point that is ℓ-close to $y = Bz$, where $\ell > 0$. Let $g \in \{0,1\}^{n+1}$ be such that*

$$\|B(z - u - g)\|^2 = \min_{t \in \{0,1\}^{n+1}} \|B(z - u - t)\|^2. \tag{5.5}$$

The lattice point $B(u + g)$ is $(\ell - 1)$-close to y.

Proof. Because Bu is ℓ-close to y there exists $v \in \mathbb{Z}^{n+1}$ with $\text{rng}(v) = \ell$ such that $B(u + v)$ is a closest lattice point to y. Define subsets of indices

$$S = \{i \mid g_i = 0, v_i = \max(v)\}, \qquad T = \{i \mid g_i = 1, v_i = \min(v)\},$$

and put $h = g + \mathbf{1}_S - \mathbf{1}_T \in \{0,1\}^{n+1}$ and $w = v - h$. Observe that $h_i = 0$ whenever $v_i = \min(v)$ and so $\min(w) = \min(v - h) = \min(v)$. Also, $h_i = 1$ whenever $v_i = \max(v)$ and so $\max(w) = \max(v - h) = \max(v) - 1$. Thus,

$$\text{rng}(w) = \max(v) - 1 - \min(v) = \text{rng}(v) - 1 = \ell - 1.$$

The lemma will follow if we show that $B(u+g+w)$ is a closest lattice point to y since then the lattice point $B(u+g)$ will be $(\ell-1)$-close to y. The proof is by contradiction. Suppose $B(u + g + w)$ is not a closest point to y, that is, suppose

$$\|B(z - u - g - w)\|^2 > \|B(z - u - v)\|^2.$$

Putting $p = z - u - v$ we have

$$\|B(p + \mathbf{1}_S - \mathbf{1}_T)\|^2 > \|Bp\|^2.$$

By part 3 of Lemma 5.2 we obtain

$$\|Bp\|^2 - \|B(p + \mathbf{1}_S - \mathbf{1}_T)\|^2 = \Phi(S, p) + \Phi(\bar{T}, p) + 2 \sum_{i \in S} \sum_{j \in T} q_{ij} < 0. \tag{5.6}$$

As stated already $h_i = 0$ whenever $v_i = \min(v)$ and $h_i = 1$ whenever $v_i = \max(v)$. It follows from Lemma 5.7 that

$$h_i - h_j \leq v_i - v_j \tag{5.7}$$

when either $v_i = \max(v)$ or $v_j = \min(v)$. Since $v_i = \max(v)$ for $i \in S$ and $v_j = \min(v)$ for $j \in T$ the inequality (5.7) holds when either $i \in S$ or $j \in T$.

Put $r = z - u - h$. By (5.7) we have $r_i - r_j \geq p_i - p_j$ when either $i \in S$ or $j \in T$. Now, since $q_{ij} \leq 0$ for $i \neq j$,

$$\Phi(S, r) = \sum_{i \in S} \sum_{j \notin S} q_{ij}(1 + 2r_i - 2r_j) \leq \sum_{i \in S} \sum_{j \notin S} q_{ij}(1 + 2p_i - 2p_j) = \Phi(S, p)$$

and

$$\Phi(\bar{T}, r) = \sum_{i \notin T} \sum_{j \in T} q_{ij}(1 + 2r_i - 2r_j) \leq \sum_{i \notin T} \sum_{j \in T} q_{ij}(1 + 2p_i - 2p_j) = \Phi(\bar{T}, p).$$

Using part 3 of Lemma 5.2 again,

$$\begin{aligned} \|B(z - u - h)\|^2 - \|B(z - u - g)\|^2 &= \|Br\|^2 - \|B(r + \mathbf{1}_S - \mathbf{1}_T)\|^2 \\ &= \Phi(S, r) + \Phi(\bar{T}, r) + 2 \sum_{i \in S} \sum_{j \in T} q_{ij} \\ &\leq \Phi(S, p) + \Phi(\bar{T}, p) + 2 \sum_{i \in S} \sum_{j \in T} q_{ij} < 0 \end{aligned}$$

as a result of (5.6). However, $h \in \{0, 1\}^{n+1}$ and so this implies

$$\|B(z - u - g)\| > \|B(z - u - h)\| \geq \min_{t \in \{0,1\}^{n+1}} \|B(z - u - t)\|$$

contradicting (5.5). Thus, our original supposition is false and $B(u+g+w)$ is a closest lattice point to y. Because $\operatorname{rng}(w) = \ell - 1$ the lattice point $B(u + g)$ is $(\ell - 1)$-close to y. ∎

We are now ready to prove our main theorem asserting that the iterative procedure (5.2) converges to a closest lattice point in $K \leq n$ iterations. This is the primary result of this section.

THEOREM 5.9. *Let $\Lambda \subset \mathbb{R}^m$ be a lattice of Voronoi's first kind with obtuse superbasis $b_1, \dots, b_{n+1}$. Let $z \in \mathbb{R}^{n+1}$, let $y = Bz$, and let $x_0, x_1, \dots$ be a sequence of lattice points given by the iterative procedure* (5.2). *There exists $K \leq n$ such that x_K is a closest lattice point to y.*

Proof. Let $x_k = Bu_k$ be the lattice point obtained on the kth iteration of the procedure. Suppose that x_k is ℓ-close to y with $\ell > 0$. The procedure computes $t_k \in \{0, 1\}^{n+1}$ satisfying

$$\|B(z - u_k - t_k)\| = \min_{t \in \{0,1\}^{n+1}} \|B(z - u_k - t)\|$$

and puts $x_{k+1} = B(u_k + t_k)$. It follows from Lemma 5.8 that x_{k+1} is $(\ell - 1)$-close to y. By Lemma 5.6 the lattice point that initializes the procedure $x_0 = B\lfloor z \rceil$ is K-close to y where $K \leq n$. Thus, x_1 is $(K - 1)$-close, x_2 is $(K - 2)$-close, and so on until x_K is 0-close. That x_K is a closest lattice point to y follows from Lemma 5.1. ∎

6. Computing a closest relevant vector. In the previous section we showed that the iterative procedure (5.2) results in a closest lattice point in at most n iterations. It remains to show that each iteration of the procedure can be computed efficiently. Specifically, it remains to show that the minimization over the set of binary vectors $\{0,1\}^{n+1}$ described in (5.3) can be computed efficiently. Putting $p = z - u_k$ in (5.3) we require an efficient method to compute a $t \in \{0,1\}^{n+1}$ such that the binary quadratic form

$$\|B(p-t)\|^2 = \left\| \sum_{i=1}^{n+1} b_i(p_i - t_i) \right\|^2$$

is minimized. Expanding this quadratic form gives

$$\left\| \sum_{i=1}^{n+1} b_i(p_i - t_i) \right\|^2 = \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} q_{ij}p_ip_j - 2\sum_{i=1}^{n+1}\sum_{j=1}^{n+1} q_{ij}p_jt_i + \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} q_{ij}t_it_j.$$

The first sum above is independent of t and can be ignored for the purpose of minimization. Letting $s_i = -2\sum_{j=1}^{n+1} q_{ij}p_j$, we can equivalently minimize the binary quadratic form

$$Q(t) = \sum_{i=1}^{n+1} s_it_i + \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} q_{ij}t_it_j. \tag{6.1}$$

We will show that a minimizer of $Q(t)$ can be found efficiently by computing a minimum cut in an undirected flow network. This technique has appeared previously [44, 49, 53, 12] but we include the derivation here so that this paper is self-contained. At the core of this technique is the fact that a one-to-one correspondence exists between the obtuse superbasis of a lattice of Voronoi's first kind and the *Laplacian matrix* [5, 13] of a simple weighted graph with $n+1$ vertices and positive edge weights equal to the negated Selling parameters $-q_{ij}$.

Let G be an undirected graph with $n+3$ vertices $v_0, \dots, v_{n+2}$ contained in the set V and edges e_{ij} connecting v_i to v_j. To each edge we assign a *weight* $w_{ij} \in \mathbb{R}$. The graph is undirected so the weights are symmetric, that is, $w_{ij} = w_{ji}$. By calling the vertex v_0 the *source* and the vertex v_{n+2} the *sink* the graph G is what is called a *flow network*. The flow network is undirected because the weights assigned to each edge are undirected. A *cut* in the flow network G is a subset $C \subset V$ of vertices with its complement $\bar{C} \subset V$ such that the source vertex $v_0 \in C$ and the sink vertex $v_{n+2} \in \bar{C}$.

The weight of a cut is

$$W(C,\bar{C}) = \sum_{i\in I}\sum_{j \in J} w_{ij},$$

where $I = \{i \mid v_i \in C\}$ and $J = \{j \mid v_j \in \bar{C}\}$. That is, $W(C,\bar{C})$ is the sum of the weights on the edges crossing from the vertices in C to the vertices in $\bar{C}$. In what follows we often drop the argument and write W rather than $W(C,\bar{C})$. A *minimum cut* is a C and $\bar{C}$ that minimize the weight W. If all of the edge weights w_{ij} for $i \neq j$ are nonnegative, a minimum cut can be computed in order $O(n^3)$ arithmetic operations [12, 19].

We require some properties of the weights w_{ij} in relation to W. If the graph is allowed to contain loops, that is, edges from a vertex to itself, then the weight of these edges w_{ii} has no effect on the weight of any cut. We may choose any values for the w_{ii} without affecting W. It will be convenient to set $w_{0,0} = w_{n+2,n+2} = 0$. The remaining w_{ii} shall be specified shortly. The edge $e_{0,n+2}$ is in every cut. If a constant is added to the weight of this edge, that is, $w_{0,n+2}$ is replaced by $w_{0,n+2} + c$ then W is replaced by $W + c$ for every C and $\bar{C}$. In particular, the subsets C and $\bar{C}$ corresponding to a minimum cut are not changed. It will be convenient to choose $w_{0,n+2} = w_{n+2,0} = 0$.

If vertex v_i is in C then edge $e_{i,n+2}$ contributes to the weight of the cut. If $v_i \notin C$, i.e., $v_i \in \bar{C}$, then edge $e_{0,i}$ contributes to the weight of the cut. So, either $e_{0,i}$ or $e_{i,n+2}$, *but not both*, contribute to every cut. If a constant, say c, is added to the weights of these edges, that is, $w_{0,i}$ and $w_{i,n+2}$ are replaced by $w_{0,i} + c$ and $w_{i,n+2} + c$, then W is replaced by $W + c$ for every C and $\bar{C}$. The C and $\bar{C}$ corresponding to a minimum cut are unchanged. In this way, the minimum cut is only affected by the differences

$$d_i = w_{i,n+2} - w_{0,i}$$

for each i and not the specific values of the weights $w_{i,n+2}$ and $w_{0,i}$.

We now show how $W(C, \bar{C})$ can be represented as a binary quadratic form. Put $t_0 = 1$ and $t_{n+2} = 0$ and

$$t_i = \begin{cases} 1, & i \in C, \\ 0, & i \in \bar{C} \end{cases}$$

for $i = 1, 2, \dots, n+1$. Observe that

$$t_i(1 - t_j) = \begin{cases} 1, & i \in C, j \in \bar{C}, \\ 0, & \text{otherwise.} \end{cases}$$

The weight can now be written as

$$W(C, \bar{C}) = \sum_{i \in C} \sum_{j \in \bar{C}} w_{ij} = \sum_{i=0}^{n+2} \sum_{j=0}^{n+2} w_{ij} t_i (1 - t_j) = F(t).$$

Finding a minimum cut is equivalent to finding the binary vector $t = (t_1, \dots, t_{n+1})$ that minimizes $F(t)$. Write

$$F(t) = \sum_{i=0}^{n+2} \sum_{j=0}^{n+2} w_{ij} t_i - \sum_{i=0}^{n+2} \sum_{j=0}^{n+2} w_{ij} t_i t_j.$$

Letting $k_i = \sum_{j=0}^{n+2} w_{ij}$, and using that $t_0 = 1$ and $t_{n+2} = 0$,

$$F(t) = \sum_{i=0}^{n+1} k_i t_i - w_{00} - \sum_{i=1}^{n+1} w_{i0} t_i - \sum_{j=1}^{n+1} w_{0j} t_j - \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} w_{ij} t_i t_j.$$

Because $w_{00} = 0$ and $w_{ij} = w_{ji}$ we have

$$F(t) = k_0 + \sum_{i=1}^{n+1} (k_i - 2w_{i0}) t_i - \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} w_{ij} t_i t_j.$$

The constant term k_0 is unimportant for the purpose of minimization so finding a minimum cut is equivalent to minimizing the binary quadratic form

$$\sum_{i=1}^{n+1} g_i t_i - \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} w_{ij} t_i t_j,$$

where $g_i = k_i - 2w_{i0} = d_i + \sum_{j=1}^{n+1} w_{ij}$. It only remains to observe the equivalence of this quadratic form and $Q(t)$ from (6.1) when the weights are assigned to satisfy

$$q_{ij} = -w_{ij} \qquad i,j = 1,\dots,n+1,$$
$$s_i = g_i = d_i + \sum_{j=1}^{n+1} w_{ij}.$$

Because the q_{ij} are nonpositive for $i \neq j$ the weights w_{ij} are nonnegative for all $i \neq j$ with $i,j = 1,\dots,n+1$. As discussed the value of the weights w_{ii} has no effect on the weight of any cut W so setting $q_{ii} = -w_{ii}$ for $i = 1,\dots,n+1$ is of no consequence. Finally the weights $w_{i,n+2}$ and $w_{0,i}$ can be chosen so that both are nonnegative and

$$w_{i,n+2} - w_{0,i} = d_i = s_i + \sum_{j=1}^{n+1} q_{ij} = s_i$$

because $\sum_{j=1}^{n+1} q_{ij} = 0$ due to the superbasis condition (4.1); that is, we choose $w_{i,n+2} = s_i$ and $w_{0,i} = 0$ when $s_i \geq 0$ and $w_{i,n+2} = 0$ and $w_{0,i} = -s_i$ when $s_i < 0$. With these choices, all the weights w_{ij} for $i \neq j$ are nonnegative. A minimizer of $Q(t)$, and, correspondingly, a solution of (5.3) can be computed in $O(n^3)$ operations by computing a minimum cut in the undirected flow network G assigned with these nonnegative weights [44, 49, 53, 12].

7. Discussion. The closest lattice point problem has a number of applications, for example, channel coding and data quantization [9, 8, 7, 18, 17]. A significant hurdle in the practical application of lattices as codes or as quantizers is that computing a closest lattice point is computationally difficult in general [40]. The best known general purpose algorithms require a number of operations of order $O(2^{2n})$ [42]. In this paper we have focused on the class of lattices of Voronoi's first kind. We have shown that computing a closest point in a lattice of Voronoi's first kind can be achieved in a comparatively modest number of operations of order $O(n^4)$ under the assumption that the obtuse superbasis is known. Besides being of theoretical interest, the algorithm has potential for practical application.

A question of immediate interest to communications engineers is, do there exist lattices of Voronoi's first kind that produce good codes or good quantizers? Since lattices that produce good codes and quantizers often also describe dense *sphere packings* [11], a related question is, do there exist lattices of Voronoi's first kind that produce dense sphere packings? These questions do not appear to have trivial answers. The questions have heightened importance due to the algorithm described in this paper.

It is straightforward to construct an "arbitrary" lattice of Voronoi's first kind. One approach is to construct the $n+1$ by $n+1$ symmetric matrix $Q = B'B$ with elements $Q_{ij} = q_{ij} = b_i \cdot b_j$ given by the Selling parameters. Choose the off-diagonal entries of Q to be nonpositive with $q_{ij} = q_{ji}$ and set the diagonal elements $q_{ii} = -\sum_{j \neq i} q_{ij}$.

The matrix Q is diagonally dominant, that is, $|q_{ii}| \geq \sum_{j \neq i} |q_{ij}|$, and so Q is positive semidefinite. A rank deficient Cholesky decomposition [23] can now be used to recover a matrix B such that $B'B = Q$. The columns of B are vectors of the obtuse superbasis.

A number applications such as phase unwrapping [52, 22], single frequency estimation [37], and related signal processing problems [32, 6, 33, 46] also require computing a closest lattice point. In these applications the particular lattice arises from the signal processing problem under consideration. If that lattice happens to be of Voronoi's first kind then our algorithm can be used. An example where this occurs is the problem of computing the *sample intrinsic mean* in circular statistics [36]. In this particular problem the lattice A_n^* is involved. A fast closest point algorithm requiring only $O(n)$ operations exists for A_n^* [39, 34] and so the algorithm described in this paper is not needed in this particular case. However, there may exist other signal processing problems where lattices of Voronoi's first kind arise.

Another interesting question is, are there subfamilies of Voronoi's first kind that admit even faster algorithms? Both A_n and A_n^* are examples of this, but there might exist other subfamilies with algorithms faster than $O(n^4)$. A related question is, can the techniques developed in this paper be applied to other families of lattices, i.e., beyond just those of Voronoi's first kind?

A final remark is that our algorithm assumes that the obtuse superbasis is known in advance. It is known that all lattices of dimension less than or equal to 3 are of Voronoi's first kind and an algorithm exists to recover the obtuse superbasis in this case [11]. Lattices of dimension larger than 3 need not be of Voronoi's first kind. Given a lattice, is it possible to efficiently decide whether it is of Voronoi's first kind? Is it possible to efficiently find an obtuse superbasis if it exists? It is suspected that the answer to this second question is no because an efficient solution would yield a solution to a known problem, that of determining whether a lattice is rectangular (has a basis consisting of pairwise orthogonal vectors) given an arbitrary basis [30].

8. Conclusion.

The paper describes an algorithm to compute a closest lattice point in a lattice of Voronoi's first kind when the obtuse superbasis is known [10]. The algorithm requires $O(n^4)$ operations where n is the dimension of the lattice. The algorithm iteratively computes a series of relevant vectors that converges to a closest lattice point after at most n terms. Each relevant vector in the series can be efficiently computed in $O(n^3)$ operations by computing a minimum cut in an undirected flow network. The algorithm has potential application in communications engineering problems such as coding and quantization. An interesting problem for future research is to find lattices of Voronoi's first kind that produce good codes, good quantizers, or dense sphere packings [11, 8].

REFERENCES

[1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, *Closest point search in lattices*, IEEE Trans. Inform. Theory, 48 (2002), pp. 2201–2214.
[2] D. Aharonov and O. Regev, *Lattice problems in np co-np*, J. ACM, 52 (2005), pp. 749–765.
[3] M. Alekhnovich, S. Khot, G. Kindler, and N. Vishnoi, *Hardness of approximating the closest vector problem with pre-processing*, Comput. Complexity, 20 (2011), pp. 741–753.
[4] L. Babai, *On Lovász lattice reduction and the nearest lattice point problem*, Combinatorica, 6 (1986), pp. 1–13.
[5] F. R. K Chung, *Spectral Graph Theory*, American Mathematical Society, Providence, RI, 1997.
[6] I. V. L. Clarkson, *Approximate maximum-likelihood period estimation from sparse, noisy timing data*, IEEE Trans. Signal Process., 56 (2008), pp. 1779–1787.

[7] J. H. Conway and N. J. A. Sloane, *Fast quantizing and decoding and algorithms for lattice quantizers and codes*, IEEE Trans. Inform. Theory, 28 (1982), pp. 227–232.
[8] J. H. Conway and N. J. A. Sloane, *Voronoi regions of lattices, second moments of polytopes, and quantization*, IEEE Trans. Inform. Theory, 28 (1982), pp. 211–226.
[9] J. H. Conway and N. J. A. Sloane, *A fast encoding method for lattice codes and quantizers*, IEEE Trans. Inform. Theory, 29 (1983), pp. 820–824.
[10] J. H. Conway and N. J. A. Sloane, *Low-dimensional lattices.* VI. *Voronoi reduction of three-dimensional lattices*, Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci., 436 (1992), pp. 55–68.
[11] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed., Springer, New York, 1998.
[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001.
[13] D. M. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Applications*, 3rd ed., John Wiley and Sons, New York, 1998.
[14] D. Dadush, O. Regev, and N. Stephens-Davidowitz, *On the closest vector problem with a distance guarantee*, in IEEE Computational Complexity Conference, Vancouver, Canada, 2014.
[15] I. Dinur, G. Kindler, R. Raz, and S. Safras, *Approximating CVP to within almost-polynomial factors in NP-hard*, Combinatorica, 23 (2003), pp. 205–243.
[16] J. Edmonds and R. M. Karp, *Theoretical improvements in algorithmic efficiency for network flow problems*, J. ACM, 19 (1972), pp. 248–264.
[17] U. Erez, S. Litsyn, and R. Zamir, *Lattices which are good for (almost) everything*, IEEE Trans. Inform. Theory, 51 (2005), pp. 3401–3416.
[18] U. Erez and R. Zamir, *Achieving* $1/2 \log(1 + SNR)$ *on the AWGN channel with lattice encoding and decoding*, IEEE Trans. Inform. Theory, 50 (2004), pp. 2293–2314.
[19] S. Even, *Graph Algorithms*, Computer Science Press, Rockville, MD, 1979.
[20] U. Feige and D. Micciancio, *The inapproximability of lattice and coding problems with preprocessing*, J. Comput. System Sci., 69 (2004), pp. 45–67.
[21] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, in Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing, STOC '86, New York, NY, 1986, ACM, New York, pp. 136–146.
[22] A. Hassibi and S. P. Boyd, *Integer parameter estimation in linear models with applications to GPS*, IEEE Trans. Signal Process., 46 (1998), pp. 2938–2952.
[23] N. J. Higham, *Analysis of the Cholesky decomposition of a semi-definite matrix*, in Reliable Numerical Computation, Oxford University Press, Oxford, 1990, pp. 161–185.
[24] J. Jalden and B. Ottersten, *On the complexity of sphere decoding in digital communications*, IEEE Trans. Signal Process., 53 (2005), pp. 1474–1484.
[25] A. Joux and J. Stern, *Lattice reduction: A toolbox for the cryptanalyst*, J. Cryptology, 11 (1998), pp. 161–185.
[26] R. Kannan, *Minkowski's convex body theorem and integer programming*, Math. Oper. Res., 12 (1987), pp. 415–440.
[27] S. A. Khot, P. Popat, and N. K. Vishnoi, *Almost polynomical factor hardness for the closest vector problem with preprocessing*, SIAM J. Comput., 43 (2014), pp. 1184–1205.
[28] A. K. Lenstra, H. W. Lenstra, and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann., 261 (1982), pp. 515–534.
[29] H. W. Lenstra, *Integer programming with a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548.
[30] H. W. Lenstra and A. Silverberg, *Revisiting the Gentry-Szydlo Algorithm*, in Advances in Crypto 2014, Santa Barbara, CA, Springer, New York, 2014.
[31] R. G. McKilliam, *Lattice Theory, Circular Statistics and Polynomial Phase Signals*, PhD thesis, University of Queensland, Queensland, Australia, 2010.
[32] R. G. McKilliam and I. V. L. Clarkson, *Maximum-likelihood period estimation from sparse, noisy timing data*, in IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, IEEE, Piscataway, NJ, 2008, pp. 3697–3700.
[33] R. G. McKilliam and I. V. L. Clarkson, *Identifiability and aliasing in polynomial-phase signals*, IEEE Trans. Signal Process., 57 (2009), pp. 4554–4557.
[34] R. G. McKilliam, I. V. L. Clarkson, and B. G. Quinn, *An algorithm to compute the nearest point in the lattice* A_n^*, IEEE Trans. Inform. Theory, 54 (2008), pp. 4378–4381.
[35] R. G. McKilliam and A. Grant, *Finding short vectors in a lattice of Voronoi's first kind*, in IEEE International Symposium on Information Theory Proceedings (ISIT), IEEE, Piscataway, NJ, 2012, pp. 2157–2160.

[36] R. G. McKilliam, B. G. Quinn, and I. V. L. Clarkson, *Direction estimation by minimum squared arc length*, IEEE Trans. Signal Process., 60 (2012), pp. 2115–2124.
[37] R. G. McKilliam, B. G. Quinn, I. V. L. Clarkson, and B. Moran, *Frequency estimation by phase unwrapping*, IEEE Trans. Signal Process., 58 (2010), pp. 2953–2963.
[38] R. G. McKilliam, B. G. Quinn, I. V. L. Clarkson, B. Moran, and B. N. Vellambi, *Polynomial phase estimation by least squares phase unwrapping*, IEEE Trans. Signal Process., 62 (2014), pp. 1962–1975.
[39] R. G. McKilliam, W. D. Smith, and I. V. L. Clarkson, *Linear-time nearest point algorithms for Coxeter lattices*, IEEE Trans. Inform. Theory, 56 (2010), pp. 1015–1022.
[40] D. Micciancio, *The hardness of the closest vector problem with preprocessing*, IEEE Trans. Inform. Theory, 47 (2001), pp. 1212–1215.
[41] D. Micciancio and O. Regev, *Lattice based cryptography*, in Post Quantum Cryptography, D .J. Bernstein, J. Buchmann, and E. Dahmen, eds., Springer, Berlin, 2009.
[42] D. Micciancio and P. Voulgaris, *A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations*, SIAM J. Comput., 42 (2013), pp. 1364–1391.
[43] P. Nguyen and J. Stern, *The two faces of lattices in cryptology*, in Cryptography and Lattices, Lecture Notes in Comput. Sci. 2146, Springer-Verlag, Berlin, 2001, pp. 146–180.
[44] J. C. Picard and H. D. Ratliff, *Minimum cuts and related problems*, Networks, 5 (1974), pp. 357–370.
[45] M. Pohst, *On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications*, ACM SIGSAM Bull., 15 (1981), pp. 37–44.
[46] B. G. Quinn, I. V. L. Clarkson, and R. G. McKilliam, *Estimating period from sparse, noisy timing data*, in IEEE Statistical Signal Processing Workshop (SSP), IEEE, Piscataway, NJ, 2012, pp. 193–196.
[47] O. Regev, *Improved inapproximability of lattice and coding problems with preprocessing*, IEEE Trans. Inform. Theory, 50 (2004), pp. 2031–2037.
[48] D. J. Ryan, I. B. Collings, I. V. L. Clarkson, and R. W. Heath, *Performance of vector perturbation multiuser MIMO systems with limited feedback*, IEEE Trans. Commun., 57 (2009), pp. 2633–2644.
[49] C. Sankaran and A. Ephremides, *Solving a class of optimum multiuser detection problems with polynomial complexity*, IEEE Trans. Inform. Theory, 44 (1998), pp. 1958–1961.
[50] E. Selling, *Ueber die binären und ternären quadratischen formen*, J. Reine Angew. Math., 77 (1874), pp. 143–229.
[51] N. Sommer, M. Feder, and O. Shalvi, *Finding the closest lattice point by iterative slicing*, SIAM J. Discrete Math., 23 (2009), pp. 715–731.
[52] P. J. G. Teunissen, *The least-squares ambiguity decorrelation adjustment: A method for fast GPS integer ambiguity estimation*, J. Geodesy, 70 (1995), pp. 65–82.
[53] S. Ulukus and R. D. Yates, *Optimum multiuser detection is tractable for synchronous CDMA systems using m-sequences*, IEEE Commun. Lett., 2 (1998), pp. 89–91.
[54] F. Vallentin, *Sphere Coverings, Lattices, and Tilings (in Low Dimensions)*, PhD thesis, Zentrum Mathematik, Technische Universität München, Munich, Germany, 2003.
[55] G. F. Voronoi, *Nouvelles applications des paramètres continus à la théorie des formes quadratiques*, J. Reine Angew. Math., 133 (1908), pp. 97–178.
[56] D. Wubben, D. Seethaler, J. Jalden, and G. Matz, *Lattice reduction*, IEEE Signal Process. Mag., 28 (2011), pp. 70–91.