An Algorithm to Compute the Nearest Point in the Lattice A_n^*

Robby G. McKilliam, I. Vaughan L. Clarkson and Barry G. Quinn

***Abstract*—The lattice A_n^* is an important lattice because of its covering properties in low dimensions. Clarkson [1] described an algorithm to compute the nearest lattice point in A_n^* that requires $O(n \log n)$ arithmetic operations. In this paper, we describe a new algorithm. While the complexity is still $O(n \log n)$, it is significantly simpler to describe and verify. In practice, we find that the new algorithm also runs faster.**

***Index Terms*—Lattice theory, nearest point algorithm, quantization, channel coding, frequency estimation, direction-of-arrival estimation, synchronization**

I. Introduction

THE study of point lattices is of great importance in several areas of number theory, particularly the studies of quadratic forms, the geometry of numbers and simultaneous Diophantine approximation, and also to the practical engineering problems of quantization and channel coding. They are also important in studying the sphere packing problem and the kissing number problem [1], [2].

A *lattice*, L, is a set of points in $\mathbb{R}^n$ such that

$$L = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} = \mathbf{B}\mathbf{w}, \mathbf{w} \in \mathbb{Z}^n\}$$

where $\mathbf{B}$ is termed the *generator matrix*.

The lattice A_n^* is an interesting lattice due to its covering properties in low dimensions. It gives the thinnest covering in all dimensions up to 8 [2]. A_n^* has also found application in a number of estimation problems including period estimation from sparse timing data [3], frequency estimation [4] and direction of arrival estimation [5].

The nearest lattice point problem is: Given $\mathbf{y} \in \mathbb{R}^n$ and some lattice L whose lattice points lie in $\mathbb{R}^n$, find the lattice point $\mathbf{x} \in L$ such that the Euclidean distance between $\mathbf{y}$ and $\mathbf{x}$ is minimized. If the lattice is used for vector quantization then the nearest lattice point corresponds to the minimum distortion point. If the lattice is used as a code for a Gaussian channel, then the nearest lattice point corresponds to maximum likelihood decoding [6].

Conway and Sloane [6] appear to have been the first to study the problem of computing the nearest lattice point in A_n^*. By decomposing A_n^* into a union of translations of its dual lattice A_n, they discovered an algorithm for computing the nearest lattice point to a given point in $O(n^2 \log n)$ arithmetic operations. Later [7], they were able to improve the execution time of the algorithm to $O(n^2)$ operations.

Clarkson [1] further improved upon the work of Conway and Sloane and described an algorithm to compute the nearest lattice point that requires only $O(n \log n)$ arithmetic operations. In this paper we describe an algorithm that is similar to Clarkson's algorithm. Like Clarkson's algorithm, our algorithm requires $O(n \log n)$ arithmetic operations. However, our algorithm and its derivation are simpler. The new algorithm, although of the same order of complexity, is computationally superior.

We now describe how the paper is organized. Section II introduces some preliminary results and definitions. In Section III we derive all results necessary to prove that the algorithm does find the nearest lattice point. Section IV describes the algorithm. A pseudocode implementation is provided. Is Section V, the arithmetic complexity of the algorithm is shown to be $O(n \log n)$. We also tabulate some practical computation times that show the new algorithm to be computationally superior to Clarkson's original algorithm.

II. Preliminary Theory

Vectors and matrices are written in bold. The ith element in a vector is denoted by a subscript: x_i. The transpose of a vector is indicated by superscript T: $\mathbf{x}^T$. We let $\mathbf{1}$ be a column vector of 1's and $\boldsymbol{e_i}$ be a column vector of zeros with a 1 in the ith position.

The *Voronoi region* or *nearest-neighbor region* $V(\mathbf{x})$ of a lattice point $\mathbf{x}$ is the subset of $\mathbb{R}^n$ such that, with respect to a given norm, all points in $V(\mathbf{x})$ are nearer to $\mathbf{x}$ than to any other point in the lattice. The Voronoi regions are n dimensional polytopes [2].

The cubic lattice $\mathbb{Z}^n$ is the set of n dimensional vectors with integer elements. The Voronoi regions of $\mathbb{Z}^n$ are hypercubes of side length 1.

The lattice A_n^* can be defined as the projection of the cubic lattice $\mathbb{Z}^{n+1}$ onto the hyperplane orthogonal to $\mathbf{1}$. This is,

$$A_n^* = \left\{\mathbf{Q}\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^{n+1}\right\} \tag{1}$$

where $\mathbf{Q}$ is the projection matrix

$$\mathbf{Q} = \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n+1}\right) \tag{2}$$

where $\mathbf{I}$ is the $(n+1) \times (n+1)$ identity matrix.

Let $\mathbf{\Pi}$ be a permutation matrix. Observe the following elementary properties:

1) $\mathbf{\Pi 1} = \mathbf{1}$,

Robby McKilliam is partly supported by a scholarship from the Wireless Technologies Laboratory, CSIRO ICT Centre, Sydney, Australia

Robby McKilliam and Vaughan Clarkson are with the School of Information Technology & Electrical Engineering, The University of Queensland, Qld., 4072, Australia

Barry Quinn is with the Department of Statistics, Macquarie University, Sydney, NSW, 2109, Australia

2) $\mathbf{1}^T\mathbf{\Pi} = \mathbf{1}^T$,
3) $\|\mathbf{\Pi x}\| = \|\mathbf{x}\|$.

Lemma 1: The matrices $\mathbf{\Pi}$ and $\mathbf{Q}$ commute, *i.e.*, $\mathbf{\Pi Q} = \mathbf{Q\Pi}$.

Proof: Using the properties of the permutation matrix, observe that

$$\begin{aligned}\mathbf{\Pi Q} &= \mathbf{\Pi}\big(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n+1}\big) = \mathbf{\Pi} - \frac{\mathbf{\Pi}\mathbf{1}\mathbf{1}^T}{n+1} \\ &= \mathbf{\Pi} - \frac{\mathbf{1}\mathbf{1}^T}{n+1} = \mathbf{\Pi} - \frac{\mathbf{1}\mathbf{1}^T\mathbf{\Pi}}{n+1} \\ &= \big(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n+1}\big)\mathbf{\Pi} = \mathbf{Q\Pi}.\end{aligned}$$

■

Corollary 1: For all $\mathbf{z} \in \mathbb{R}^{n+1}$, $\|\mathbf{Qz}\| = \|\mathbf{Q\Pi z}\|$.

Corollary 2: $\mathbf{x} \in A_n^*$ if and only if $\mathbf{\Pi x} \in A_n^*$.

Proof: Because the inverse of a permutation matrix is also a permutation matrix, we need only prove sufficiency. If $\mathbf{x} \in A_n^*$ then $\mathbf{x} = \mathbf{Qk}$ with $\mathbf{k} \in \mathbb{Z}^{n+1}$. Therefore, $\mathbf{\Pi x} = \mathbf{\Pi Q k} = \mathbf{Q\Pi k} = \mathbf{Qk'}$ where $\mathbf{k'} = \mathbf{\Pi k} \in \mathbb{Z}^{n+1}$ and so $\mathbf{\Pi x} \in A_n^*$. ■

Corollary 3: The lattice point $\mathbf{x}$ is a closest point in A_n^* to $\mathbf{y}$ if and only if $\mathbf{\Pi x}$ is a closest point in A_n^* to $\mathbf{\Pi y}$.

Proof: As for Corollary 2, we need only show sufficiency. We do this by contradiction. Suppose $\mathbf{\Pi x}$ is not closest to $\mathbf{\Pi y}$ but there is instead some $\mathbf{\Pi z} \in A_n^*$ such that

$$\|\mathbf{\Pi}(\mathbf{z} - \mathbf{y})\| < \|\mathbf{\Pi}(\mathbf{x} - \mathbf{y})\|$$

This implies that

$$\|\mathbf{z} - \mathbf{y}\| < \|\mathbf{x} - \mathbf{y}\|$$

which contradicts the assumption that $\mathbf{x}$ is a closest point to $\mathbf{y}$ in A_n^*. ■

Hence, in considering an algorithm to find a closest point in A_n^* to $\mathbf{y}$, it is sufficient to consider a canonical permutation of $\mathbf{y}$. We will see that it is very convenient to consider the permutation in which the *(centered) fractional parts of* $\mathbf{y}$, *i.e.*, $\{y_i\} = y_i - \lfloor y_i \rceil$, are sorted in descending order. That is, in the sequel, except where otherwise noted, we will assume that

$$\{y_1\} \geq \{y_2\} \geq \cdots \geq \{y_{n+1}\}. \tag{3}$$

In the case that two or more $\{y_i\}$ are equal then multiple orderings of $\mathbf{y}$ satisfy (3). The following arguments and the subsequent algorithm are valid for any ordering of $\mathbf{y}$ that satisfies (3).

III. Closest Point in A_n^*

Lemma 2: If $\mathbf{x} = \mathbf{Qk}$ is a closest point in A_n^* to $\mathbf{y} \in \mathbb{R}^{n+1}$ then there exists some $\lambda \in \mathbb{R}$ for which $\mathbf{k}$ is a closest point in $\mathbb{Z}^{n+1}$ to $\mathbf{y} + \lambda\mathbf{1}$.

Proof: Decompose $\mathbf{y}$ into orthogonal components $\mathbf{Qy}$ and $t\mathbf{1}$ for some $t \in \mathbb{R}$. Then

$$\|\mathbf{y} - \mathbf{x}\|^2 = \|\mathbf{Q}(\mathbf{y} - \mathbf{k})\|^2 + t^2(n+1). \tag{4}$$

Observe that

$$\mathbf{Q}(\mathbf{y} - \mathbf{k}) = \mathbf{y} + \lambda\mathbf{1} - \mathbf{k}$$

where we set

$$\lambda = \frac{\mathbf{1}^T(\mathbf{k} - \mathbf{y})}{n+1}.$$

Suppose $\mathbf{k}$ is not a closest point in $\mathbb{Z}^{n+1}$ to $\mathbf{y} + \lambda\mathbf{1}$. Suppose $\mathbf{k'}$ is closer. Let $\mathbf{x'} = \mathbf{Qk'}$. Then

$$\begin{aligned}\|\mathbf{y} - \mathbf{x'}\|^2 &= \|\mathbf{Q}(\mathbf{y} - \mathbf{k'})\|^2 + t^2(n+1) \\ &\leq \|\mathbf{y} + \lambda\mathbf{1} - \mathbf{k'}\|^2 + t^2(n+1) \\ &< \|\mathbf{y} + \lambda\mathbf{1} - \mathbf{k}\|^2 + t^2(n+1) = \|\mathbf{y} - \mathbf{x}\|^2,\end{aligned}$$

contradicting the assumption that $\mathbf{x}$ is a closest point in A_n^* to $\mathbf{y}$. ■

Now consider the function $\mathbf{f} : \mathbb{R} \mapsto \mathbb{Z}^{n+1}$ defined so that

$$\mathbf{f}(\lambda) = \lfloor \mathbf{y} + \lambda\mathbf{1} \rceil$$

where $\lfloor \cdot \rceil$ applied to a vector denotes the vector in which each element is rounded to a nearest integer[1]. That is, $\mathbf{f}(\lambda)$ gives a nearest point in $\mathbb{Z}^{n+1}$ to $\mathbf{y} + \lambda\mathbf{1}$ as a function of λ. Observe that $\mathbf{f}(\lambda + 1) = \mathbf{f}(\lambda) + \mathbf{1}$. Hence,

$$\mathbf{Qf}(\lambda + 1) = \mathbf{Qf}(\lambda). \tag{5}$$

Lemma 2 implies there exists some $\lambda \in \mathbb{R}$ such that $\mathbf{x} = \mathbf{Qf}(\lambda)$ is a closest point to $\mathbf{y}$. Furthermore, we see from (5) that λ can be found within an interval of length 1. Hence, if we define the set

$$\mathscr{S} = \{\mathbf{f}(\lambda) \mid \lambda \in [0, 1)\}$$

then $\mathbf{Q}\mathscr{S}$ contains a closest point in A_n^* to $\mathbf{y}$.

If the fractional parts of $\mathbf{y}$ are sorted as in (3), it is clear that $\mathscr{S}$ contains at most $n+2$ vectors, *i.e.*,

$$\mathscr{S} \subseteq \big\{\lfloor\mathbf{y}\rceil, \lfloor\mathbf{y}\rceil + \mathbf{e}_1, \lfloor\mathbf{y}\rceil + \mathbf{e}_1 + \mathbf{e}_2, \ldots, \lfloor\mathbf{y}\rceil + \mathbf{e}_1 + \cdots + \mathbf{e}_{n+1}\big\}. \tag{6}$$

It can be seen that the last vector listed in the set is simply $\lfloor\mathbf{y}\rceil + \mathbf{1}$ and so, once multiplied by $\mathbf{Q}$, the first and the last vector are identical.

An algorithm immediately suggests itself: test each of the $n+1$ distinct vectors and find the closest one to $\mathbf{y}$. Indeed, this is exactly the principle of the algorithm we propose here. It only remains to show that this can be done in $O(n \log n)$ arithmetic operations.

IV. Algorithm

We label the elements of $\mathscr{S}$ according to the order given in (6). That is, we set $\mathbf{u}_0 = \lfloor\mathbf{y}\rceil$ and, for $i = 1, \ldots, n$,

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \mathbf{e}_i. \tag{7}$$

Let $\mathbf{z}_i = \mathbf{y} - \mathbf{u}_i$. Clearly, $\mathbf{z}_0 = \{\mathbf{y}\}$. Following (4), the squared distance between $\mathbf{Qu}_i$ and $\mathbf{y}$ is

$$\|\mathbf{y} - \mathbf{Qu}_i\|^2 = d_i + t^2(n+1) \tag{8}$$

where we define d_i as

$$d_i = \|\mathbf{Qz}_i\|^2 = \left\|\mathbf{z}_i - \frac{\mathbf{z}_i^T\mathbf{1}}{n+1}\mathbf{1}\right\|^2 = \mathbf{z}_i^T\mathbf{z}_i - \frac{(\mathbf{z}_i^T\mathbf{1})^2}{n+1}. \tag{9}$$

[1]The direction of rounding for half-integers is not important. However, the authors have chosen to round up half-integers in their own implementation.

We know that the nearest point to $\mathbf{y}$ is that $\mathbf{Qu}_i$ which minimizes (8). Since the term $t^2(n+1)$ is independent of the index i, we can ignore it. That is, it is sufficient to minimize d_i, $i = 0, \ldots, n$.

We now show that d_i can be calculated inexpensively in a recursive fashion. We define two new quantities, $\alpha_i = \mathbf{z}_i^T\mathbf{1}$ and $\beta_i = \mathbf{z}_i^T\mathbf{z}_i$. From (7),

$$\alpha_i = \mathbf{z}_i^T\mathbf{1} = (\mathbf{z}_{i-1} - \mathbf{e}_i)^T\mathbf{1} = \alpha_{i-1} - 1 \tag{10}$$

and

$$\beta_i = \mathbf{z}_i^T\mathbf{z}_i = (\mathbf{z}_{i-1} - \mathbf{e}_i)^T(\mathbf{z}_{i-1} - \mathbf{e}_i) = \beta_{i-1} - 2\{y_i\} + 1. \tag{11}$$

Input: $\mathbf{y} \in \mathbb{R}^{n+1}$
1 $\mathbf{z} = \mathbf{y} - \lfloor \mathbf{y} \rceil$
2 $\alpha = \mathbf{z}^T\mathbf{1}$
3 $\beta = \mathbf{z}^T\mathbf{z}$
4 $\mathbf{s} = \mathit{dsortindices}(\mathbf{z})$
5 $D = \beta - \frac{\alpha^2}{n+1}$
6 $m = 0$
7 **for** $i = 1$ to n **do**
8 $\quad \alpha = \alpha - 1$
9 $\quad \beta = \beta - 2z_{s_i} + 1$
10 $\quad$ **if** $\beta - \frac{\alpha^2}{n+1} < D$ **then**
11 $\quad\quad D = \beta - \frac{\alpha^2}{n+1}$
12 $\quad\quad m = i$
13 $\mathbf{k} = \lfloor \mathbf{y} \rceil$
14 **for** $i = 1$ to m **do**
15 $\quad k_{s_i} = k_{s_i} + 1$
16 $\mathbf{x} = \mathbf{k} - \frac{\mathbf{1}^T\mathbf{k}}{n+1}\mathbf{1}$
17 **return** $\mathbf{x}$

Algorithm 1: Algorithm to find a nearest lattice point in A_n^* to $\mathbf{y} \in \mathbb{R}^{n+1}$

Algorithm 1 now follows. The main loop beginning at line 7 calculates the α_i and β_i recursively. There is no need to retain their previous values, so the subscripts are dropped. The variable D maintains the minimum value of the (implicitly calculated values of) d_i so far encountered, and m the corresponding index.

V. Computational Complexity

Each line of the main loop requires $O(1)$ arithmetic computations so the loop (and that on line 14) requires $O(n)$ in total. On line 4 the function *dsortindices*$(\mathbf{z})$ returns the vector $\mathbf{s}$ such that $z_{s_1} \geq z_{s_2} \geq \cdots \geq z_{s_{n+1}}$. This sorting operation requires $O(n \log n)$ arithmetic operations. The vector operations on lines 1–3, 13 and 16 all require $O(n)$ operations. It can be seen, then, that the computational cost of the algorithm is dominated by the sorting operation and is therefore $O(n \log n)$.

Clarkson's original algorithm required two sorts of $n+1$ elements. The new algorithm requires only a single sort. Seeing as the sort dominates the complexity of both algorithms, we might expect our algorithm to require approximately half the arithmetic operations of Clarkson's original algorithm. This appears to be the case for small n. Table I shows the practical computational performance of Clarkson's algorithm versus our new algorithm. It is evident that the new algorithm is computationally superior, particularly for small n. It appears that the computational performance of the algorithms converge for large n. The computer used for these trials is an Intel Core2 running at 2.13Ghz.

TABLE I
Computation time in seconds for 10^5 trials

Algorithm	n=20	n=50	n=100	n=500
Clarkson	4.57	6.97	11.11	47.81
New	2.05	3.86	7.125	35.44

As a final note, the algorithm proposed here can be extended to other lattices for which Lemmata 1 and 2 hold. Potential candidates are the Coxeter lattices [8], [9].

References

[1] I. V. L. Clarkson, "An algorithm to compute a nearest point in the lattice A_n^*," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, Marc Fossorier, Hideki Imai, Shu Lin, and Alain Poli, Eds., vol. 1719 of *Lecture Notes in Computer Science*, pp. 104–120. Springer, 1999.

[2] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*, Springer, 3rd edition, 1998.

[3] I. V. L. Clarkson, "Approximate maximum-likelihood period estimation from sparse, noisy timing data," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1779–1787, May 2008.

[4] I. V. L. Clarkson, "Frequency estimation, phase unwrapping and the nearest lattice point problem," *Proc. Internat. Conf. Acoust. Speech Signal Process.*, vol. 3, pp. 1609–1612, 1999.

[5] B. G. Quinn, "Estimating the mode of a phase distribution," *Asilomar Conference on Signals, Systems and Computers*, pp. 587–591, Nov 2007.

[6] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding and algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Th.*, vol. 28, no. 2, pp. 227–232, 1982.

[7] J. H. Conway and N. J. A. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. Inform. Th.*, vol. 32, no. 1, pp. 41–50, 1986.

[8] H.S.M. Coxeter, "Extreme forms," *Canad. J. Math.*, vol. 3, pp. 391–441, 1951.

[9] J. Martinet, *Perfect lattices in Euclidean spaces*, Springer, 2003.