

# On the Sphere Decoding Algorithm. I. Expected Complexity

BABAK HASSIBI    HARIS VIKALO

Department of Electrical Engineering  
California Institute of Technology, Pasadena, CA 91125  
e-mail: hassibi, hvikalo@systems.caltech.edu

## Abstract

The problem of finding the least-squares solution to a system of linear equations where the unknown vector is comprised of integers, but the matrix coefficient and given vector are comprised of real numbers, arises in many applications: communications, cryptography, GPS, to name a few. The problem is equivalent to finding the closest lattice point to a given point and is known to be NP-hard. In communications applications, however, the given vector is not arbitrary, but rather is an unknown lattice point that has been perturbed by an additive noise vector whose statistical properties are known. Therefore in this paper, rather than dwell on the worst-case complexity of the integer-least-squares problem, we study its expected complexity, averaged over the noise and over the lattice. For the “sphere decoding” algorithm of Fincke and Pohst we find a closed-form expression for the expected complexity, both for the infinite and finite lattice. It is demonstrated in the second part of this paper that, for a wide range of signal-to-noise ratios (SNR) and numbers of antennas, the expected complexity is polynomial, in fact often roughly cubic. Since many communications systems operate at noise levels for which the expected complexity turns out to be polynomial, this suggests that maximum-likelihood decoding, which was hitherto thought to be computationally intractable, can in fact be implemented in real-time—a result with many practical implications.

*Index Terms*—Integer least-squares problem, sphere decoding, wireless communications, multiple-antenna systems, lattice problems, NP hard, expected complexity.

## 1 Introduction and Problem Statement

In this paper we shall be concerned with the following so-called *integer least-squares problem*

$$\min_{s \in \mathcal{Z}^m} \|x - Hs\|^2, \quad (1)$$

where  $x \in \mathcal{R}^{n \times 1}$ ,  $H \in \mathcal{R}^{n \times m}$ , and  $\mathcal{Z}^m$  denotes the  $m$ -dimensional integer lattice, i.e.,  $s$  is an  $m$ -dimensional vector with integer entries. Often, the search space is a (finite) subset of the infinite lattice,

$\mathcal{D} \subset \mathcal{Z}^m$ , in which case we have

$$\min_{s \in \mathcal{D} \subset \mathcal{Z}^m} \|x - Hs\|^2. \quad (2)$$

The integer least-squares problem has a simple geometric interpretation. As the entries of  $s$  run over the integers,  $s$  spans the “rectangular”  $m$ -dimensional lattice,  $\mathcal{Z}^m$ . However, for any given *lattice-generating matrix*  $H$ , the  $n$ -dimensional vector  $Hs$  spans a “skewed” lattice. (When  $n > m$ , this skewed lattice lives in an  $m$ -dimensional subspace of  $\mathcal{R}^{n \times 1}$ .) Therefore, given the skewed lattice  $Hs$ , and given a vector  $x \in \mathcal{R}^{n \times 1}$ , the integer least-squares problem is to find the “closest” lattice point (in a Euclidean sense) to  $x$ —see Figure 1.

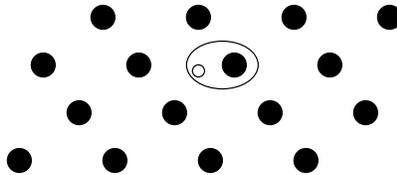


Figure 1: *Geometrical interpretation of the integer least-squares problem.*

Compared to the standard least-squares problem where the unknown vector  $s$  is an arbitrary vector in  $\mathcal{R}^{m \times 1}$ , and the solution is obtained via a simple pseudo-inverse, it is much more difficult to find the solution to (1) or (2). The main reason is that the search space is discrete (whether it is finite or infinite). In fact, it is well known that problems (1) and (2) are, for a general  $H$ , NP hard, both in a worst-case sense [1], as well as in an average sense [2].

The remainder of this paper is organized as follows. In section 2 we give an overview of some heuristic and exact methods to solve the integer least-squares problem. We show that the exact methods can offer substantial gains over the heuristic ones. However, they generally require an exponential worst-case complexity, whereas the heuristic methods require only cubic,  $O(m^3)$ , computations. Section 3 introduces the sphere decoding algorithm of Fincke and Pohst and argues that, if the received point is arbitrary, then the expected complexity of the algorithm is exponential. Section 4 introduces a random model for the integer least-squares problem, where the received point is assumed to be a lattice point perturbed by an additive Gaussian noise vector with known statistical properties. It then proceeds to compute the expected complexity of the sphere decoding algorithm, averaged over both the noise and the lattice, as a function of the noise variance, or SNR. This is done both for the infinite lattice, as well as some finite lattices encountered in communications problems. Section 5 provides the conclusion. The appendices give some mathematical background for the problems encountered in this paper.

## 2 Overview of Methods

Since the integer least-squares problem arises in many applications and finding the exact solution is, in general, NP hard, all practical systems employ some approximations, heuristics or combinations thereof. In communications applications, these approximations can be broadly categorized into three classes.

1. Solve the unconstrained least-squares problem to obtain  $\hat{s} = H^\dagger x$ , where  $H^\dagger$  denotes the pseudo-inverse of  $H$ . Since the entries of  $\hat{s}$  will not necessarily be integers, round them off to the closest integer (a process referred to as slicing) to obtain

$$\hat{s}_B = \left[ H^\dagger x \right]_{\mathcal{Z}^m}. \quad (3)$$

The above  $\hat{s}_B$  is often called the Babai estimate [1]. In communications parlance, this procedure is referred to as *zero-forcing equalization*.

2. *Nulling and cancelling*. In this method, the Babai estimate is used for only one of the entries of  $s$ , say the first entry  $s_1$ .  $s_1$  is then assumed to be known and its effect is cancelled out to obtain a reduced-order integer least-squares problem with  $m - 1$  unknowns. The process is then repeated to find  $s_2$ , etc. In communications parlance this is known as *decision-feedback equalization*.
3. *Nulling and cancelling with optimal ordering*. Nulling and cancelling can suffer from “error-propagation”: if  $s_1$  is estimated incorrectly it can have an adverse effect on the estimation of the remaining unknowns  $s_2, s_3$ , etc. To minimize the effect of error propagation, it is advantageous to perform nulling and cancelling from the “strongest” to the “weakest” signal. This is the method proposed for V-BLAST [3]—see also [4].

The above heuristic methods all require  $O(mn^2)$  computations, essentially because they all first solve the unconstrained least-squares problem.

### 2.1 Lattice Reduction

The aforementioned heuristic methods are exact only if the columns of  $H$  are orthogonal. In this case  $H$  can be diagonalized by a unitary transformation on the left, and so slicing the unconstrained least-squares solution yields the exact solution.

In practice, however, the columns of  $H$  are rarely orthogonal. Orthogonalizing the columns of  $H$  via a QR decomposition, or otherwise, generally destroys the lattice structure. (The reason being that, if  $s$  has integer entries,  $Rs$  need not have integer entries.) One method that attempts to alleviate this is *lattice reduction*. In these methods, one attempts to find an invertible  $m \times m$  matrix  $T$ , such that  $T$  and  $T^{-1}$  have integer entries (thereby preserving the lattice structure), and such that the matrix  $G = HT$  is as “orthogonal as possible”. Having found such a  $T$ , rather than solve (1), one can solve the integer least-squares problem

$$\min_{t \in \mathcal{Z}^m} \|x - Gt\|^2, \quad (4)$$

using the earlier mentioned heuristics and set  $s = Tt$ . Of course, lattice reduction is itself NP-hard. A common heuristic is the LLL (Lenstra, Lenstra and Lovász [5]) algorithm which, permitting a gross oversimplification, can be regarded as Gram-Schmidt over integers.

While lattice reduction may lead to some improvement in the solution of (1), the integer least-squares problem over the infinite lattice, it is not useful for (2), which is over a subset of the lattice. The reason is that the lattice transforming matrix  $T$  often destroys the properties of the subset  $\mathcal{D} \subset \mathcal{Z}^m$ . Since in communications applications, we are always interested in a *finite* subset of the integer lattice, we shall therefore not consider lattice reduction methods in this paper.

## 2.2 Exact Methods

With the abundance of heuristic methods, it is natural to ask what their performance is, and how close they come to the optimal solution. In [6] this question is studied in the context of V-BLAST where it is shown that the exact solution significantly outperforms even the best heuristics. We also give an example here in the context of space-time codes from [7], which is shown in Figure 2. The example is a space-time code for a system with  $M = 8$  transmit and  $N = 4$  receive antennas which spreads  $\min(M, N)T = 32$  complex symbols from a 16QAM constellation over  $T = 8$  channel uses. [The rate of the code is  $R = 32/8 \times \log_2 16 = 16$ .] The equivalent channel matrix (see [7]) has  $m = 2 \min(M, N)T = 64$  rows and  $n = 2NT = 64$  columns. Therefore, the resulting integer least-squares problem corresponds to dimension  $m = 64$  and the entries of  $s$  each take on 4 integer values, say  $\{-3, -1, 1, 3\}$ . The number of lattice points in  $\mathcal{D}$  is  $4^{64} = 2^{128} \approx 3.4 \times 10^{38}$ . As can be seen from Figure 2, the bit-error rate (BER) performance of the exact integer least-squares solution is far superior to that of the best heuristic, which in this case is nulling

and cancelling with optimal ordering.<sup>1</sup>

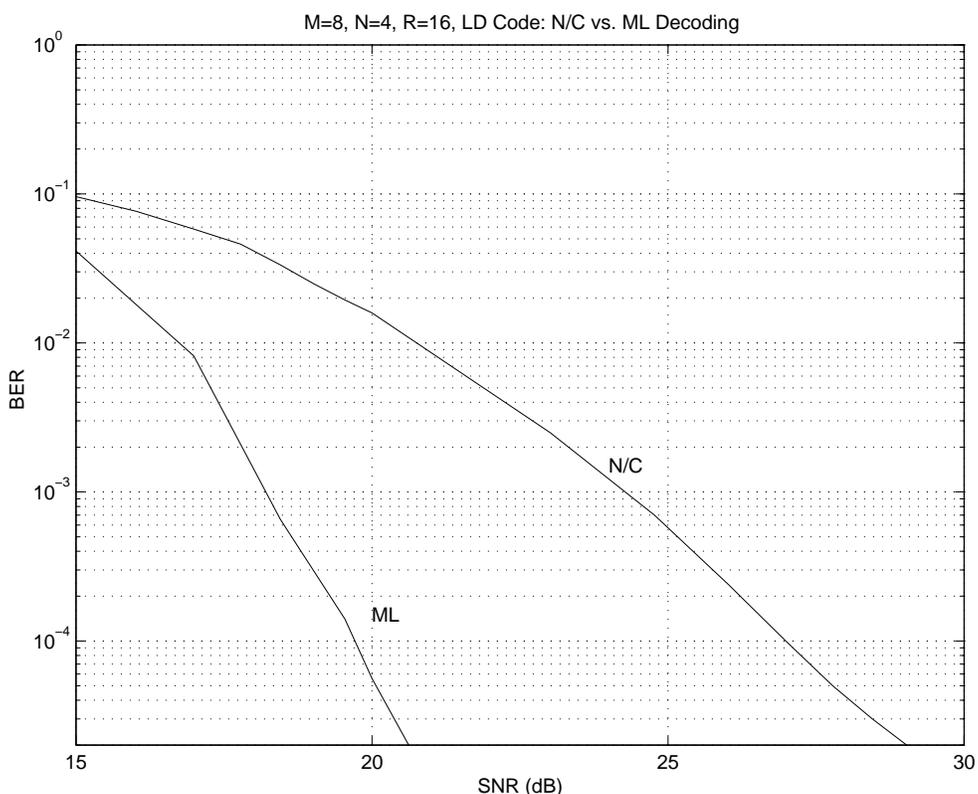


Figure 2: *Bit error performance of a rate  $R = 16$  linear space-time code, corresponding to  $m = 64$ . Exact ML solution vs. nulling/cancelling with optimal ordering. (No. of lattice points =  $2^{128} \approx 3.4 \times 10^{38}$ ).*

The above discussion shows that there is merit in studying exact solutions. The most obvious one is to form a search over the entire lattice which, although theoretically feasible for finite lattices, invariably requires an exponential search. There do, however, exist exact methods that are a bit more sophisticated than the above full search. These include Kannan’s algorithm [8] (which searches only over restricted parallelograms), the KZ algorithm [9] (based on the Korkin-Zolotarev reduced basis [10]) and the sphere decoding algorithm of Fincke and Pohst [11, 12]. Since then the work of Fincke and Pohst the sphere decoding algorithm has been rediscovered in several contexts (see, e.g., [13] in the context of GPS systems) and is the algorithm we will be considering in this paper. Many of the results of this paper and various extensions can be found in the second author’s PhD dissertation [14].

---

<sup>1</sup>Of course, at this point it may appear surprising that one can even generate the ML curve in Figure 2, since it requires finding the exact solution among a set of size  $10^{38}$ —more on this later.

### 3 Sphere Decoding

The basic premise in sphere decoding is rather simple: we attempt to search over only lattice points  $s \in \mathcal{Z}^m$  that lie in a certain sphere of radius  $d$  around the given vector  $x$ , thereby reducing the search space and hence the required computations (see Figure 3). Clearly, the closest lattice point inside the sphere will also be the closest lattice point for the whole lattice. However, close scrutiny of this basic idea leads to two key questions.

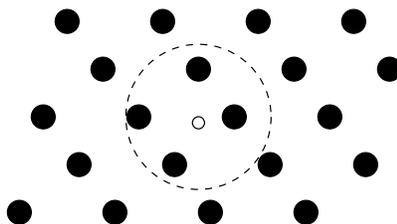


Figure 3: *Idea behind the sphere decoder*

1. *How to choose  $d$ ?* Clearly, if  $d$  is too large, we obtain too many points and the search remains exponential in size, whereas if  $d$  is too small, we obtain no points inside the sphere.

A natural candidate for  $d$  is the *covering radius* of the lattice, defined to be the smallest radius of spheres centered at the lattice points that cover the entire space. This is clearly the smallest radius that guarantees the existence of a point inside the sphere for any vector  $x$ . The problem with this choice of  $d$  is that determining the covering radius for a given lattice is itself NP hard [15].

Another choice is to use  $d$  as the distance between the Babai estimate and the vector  $x$ , i.e.,  $d = \|x - H\hat{s}_B\|$ , since this radius guarantees the existence of at least one lattice point (here the Babai estimate) inside the sphere. However, it is again not clear in general whether this choice of radius leads to too many lattice points lying inside the sphere.

2. *How can we tell which lattice points are inside the sphere?* If this requires testing the distance of each lattice point from  $x$  (to determine whether it is less than  $d$ ), then there is no point in sphere decoding as we will still need an exhaustive search.

Sphere decoding does not really address the first question. However, it does propose an efficient way to answer the second, and more pressing, one. The basic observation is the following. Although it is difficult

to determine the lattice points inside a general  $m$ -dimensional sphere, it is trivial to do so in the (one-dimensional) case of  $m = 1$ . The reason is that a one-dimensional sphere is simply an interval and so the desired lattice points will be the integer values that lie in this interval. We can use this observation to go from dimension  $k$  to dimension  $k + 1$ . Suppose we have determined all  $k$ -dimensional lattice points that lie in a sphere of radius  $d$ . Then for any such  $k$ -dimensional point, the set of admissible values of the  $k + 1$ -th dimensional coordinate that lie in the higher dimensional sphere of the *same* radius  $d$  forms an interval.

The above means that we can determine all lattice points in a sphere of dimension  $m$  and radius  $d$  by successively determining all lattice points in spheres of lower dimensions  $1, 2, \dots, m$  and the same radius  $d$ . Such an algorithm for determining the lattice points in an  $m$ -dimensional sphere essentially constructs a tree where the branches in the  $k$ -th level of the tree correspond to the lattice points inside the sphere of radius  $d$  and dimension  $k$ —see Figure 4. Moreover, the complexity of such an algorithm will depend on the *size* of the tree, i.e., on the number of lattice points visited by the algorithm in different dimensions.

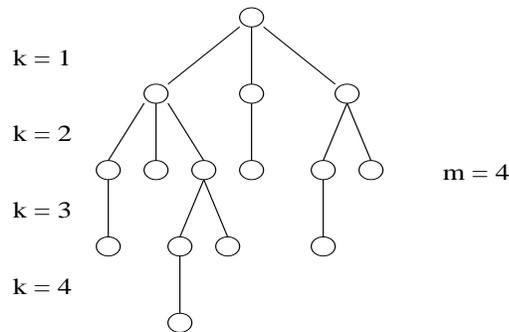


Figure 4: *Sample tree generated to determine lattice points in a 4-dimensional sphere.*

With this brief discussion we can now be more specific about the problem at hand. To this end, we shall assume that  $n \geq m$ , i.e., that there are at least as many equations as unknowns in  $x \approx Hs$  (the case  $n < m$  is considered in Part II). Note that the lattice point  $Hs$  lies in a sphere of radius  $d$  if, and only if,

$$d^2 \geq \|x - Hs\|^2. \quad (5)$$

In order to break the problem into the subproblems described above, it is useful to consider the QR factorization of the matrix  $H$

$$H = Q \begin{bmatrix} R \\ 0_{(n-m) \times m} \end{bmatrix}, \quad (6)$$

where  $R$  is an  $m \times m$  upper triangular matrix and  $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$  is an  $n \times n$  orthogonal matrix. The matrices  $Q_1$  and  $Q_2$  represent the first  $m$  and last  $n - m$  orthonormal columns of  $Q$ , respectively. The condition (5) can therefore be written as

$$d^2 \geq \left\| x - \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 = \left\| \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} x - \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 = \|Q_1^* x - Rs\|^2 + \|Q_2^* x\|^2,$$

where  $(\cdot)^*$  here denotes Hermitian matrix transposition. Or in other words,

$$d^2 - \|Q_2^* x\|^2 \geq \|Q_1^* x - Rs\|^2. \quad (7)$$

Defining  $y = Q_1^* x$  and  $d'^2 = d^2 - \|Q_2^* x\|^2$  allows us to rewrite this as

$$d'^2 \geq \sum_{i=1}^m \left( y_i - \sum_{j=i}^m r_{i,j} s_j \right)^2. \quad (8)$$

where  $r_{i,j}$  denotes an  $(i, j)$  entry of  $R$ . Here is where the upper triangular property of  $R$  comes in handy.

The right-hand side (RHS) of the above inequality can be expanded as

$$d'^2 \geq (y_m - r_{m,m} s_m)^2 + (y_{m-1} - r_{m-1,m} s_m - r_{m-1,m-1} s_{m-1})^2 + \dots \quad (9)$$

where the first term depends only on  $s_m$ , the second term on  $\{s_m, s_{m-1}\}$  and so on. Therefore a necessary condition for  $Rs$  to lie inside the sphere is that  $d'^2 \geq (y_m - r_{m,m} s_m)^2$ . This condition is equivalent to  $s_m$  belonging to the interval

$$\left[ \frac{-d' + y_m}{r_{m,m}} \right] \leq s_m \leq \left[ \frac{d' + y_m}{r_{m,m}} \right], \quad (10)$$

where  $\lceil \cdot \rceil$  denotes rounding to the nearest larger element in the set of numbers which spans the lattice.<sup>2</sup> Similarly,  $\lfloor \cdot \rfloor$  denotes rounding to the nearest smaller element in the set of numbers which spans the lattice.

Of course, (10) is by no means sufficient. For every  $s_m$  satisfying (10), defining  $d'_{m-1} = d'^2 - (y_m - r_{m,m} s_m)^2$  and  $y_{m-1|m} = y_{m-1} - r_{m-1,m} s_m$ , a stronger necessary condition can be found by looking

---

<sup>2</sup>Clearly, for the case of infinite lattice  $\mathcal{Z}^m$ , this set is the set of integer numbers  $\mathcal{Z}$ . The set which spans a finite subset of  $\mathcal{Z}^m$  is a finite subset of  $\mathcal{Z}$ , possibly shifted.

at the first two terms in (9), which leads to  $s_{m-1}$  belonging to the interval

$$\left\lceil \frac{-d'_{m-1} + y_{m-1|m}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \frac{d'_{m-1} + y_{m-1|m}}{r_{m-1,m-1}} \right\rfloor. \quad (11)$$

One can continue in a similar fashion for  $s_{m-2}$ , and so on until  $s_1$ , thereby obtaining all lattice points belonging to (5).

### 3.1 The Sphere Decoding Algorithm

We can now formalize the algorithm.

$$\text{Input: } Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}, R, x, y = Q_1^* x, d.$$

1. Set  $k = m$ ,  $d'_m = d^2 - \|Q_2^* x\|^2$ ,  $y_{m|m+1} = y_m$
2. (Bounds for  $s_k$ ) Set  $UB(s_k) = \lfloor \frac{d'_k + y_{k|k+1}}{r_{k,k}} \rfloor$ ,  $s_k = \lceil \frac{-d'_k + y_{k|k+1}}{r_{k,k}} \rceil - 1$
3. (Increase  $s_k$ )  $s_k = s_k + 1$ . If  $s_k \leq UB(s_k)$  go to 5, else go to 4.
4. (Increase  $k$ )  $k = k + 1$ ; if  $k = m + 1$  terminate algorithm, else go to 3.
5. (Decrease  $k$ ) If  $k = 1$  go to 6. Else  $k = k - 1$ ,  $y_{k|k+1} = y_k - \sum_{j=k+1}^m r_{k,j} s_j$ ,  $d'_k = d'^2_{k+1} - (y_{k+1|k+2} - r_{k+1,k+1} s_{k+1})^2$ , and go to 2.
6. Solution found. Save  $s$  and its distance from  $x$ ,  $d'^2_m - d'^2_1 + (y_1 - r_{1,1} s_1)^2$ , and go to 3.

Note that the subscript  $k|k+1$  in  $y_{k|k+1}$  above is used to denote received signal  $y_k$  adjusted with the already estimated symbol components  $s_{k+1}, \dots, s_m$ . Furthermore, note that in step 2 and step 3 of the code, we assumed unit spacing between any two nearest elements of the set spanning the lattice. If the lattice is scaled, i.e., if the spacing between two nearest neighbors in the set spanning the lattice is different from 1, those algorithm steps need to be adjusted accordingly.

We should mention that the original paper of Fincke and Pohst [12] used slightly different notation to the one we have used. For completeness, we shall include it here. The paper [12] makes use of the unconstrained least-squares solution  $\hat{s} = H^\dagger x = R^{-1} Q_1^* x$ . In this case, it follows that  $\|Q_2^* x\|^2 = \|x\|^2 - \|H\hat{s}\|^2$  and so inequality (7) becomes

$$d^2 - \|x\|^2 + \|H\hat{s}\|^2 \geq \|R(\hat{s} - s)\|^2. \quad (12)$$

Expansion (9) becomes

$$d'^2 \geq r_{m,m}^2 (s_m - \hat{s}_m)^2 + r_{m-1,m-1}^2 \left( s_{m-1} - \hat{s}_{m-1} + \frac{r_{m-1,m}}{r_{m-1,m-1}} (s_m - \hat{s}_m) \right)^2 + \dots \quad (13)$$

and the intervals (10) and (11)

$$\left[ \hat{s}_m - \frac{d'}{r_{m,m}} \right] \leq s_m \leq \left[ \hat{s}_m + \frac{d'}{r_{m,m}} \right] \quad (14)$$

and

$$\left[ \hat{s}_{m-1|m} - \frac{d'_{m-1}}{r_{m-1,m-1}} \right] \leq s_{m-1} \leq \left[ \hat{s}_{m-1|m} + \frac{d'_{m-1}}{r_{m-1,m-1}} \right] \quad (15)$$

respectively, where we have defined  $\hat{s}_{m-1|m} = \hat{s}_{m-1} - \frac{r_{m-1,m}}{r_{m-1,m-1}} (s_m - \hat{s}_m)$ . We can now alternatively write the algorithm as

*Input:*  $R, x, \hat{s}, d$ .

- 1a. Set  $k = m$ ,  $d'_m = d^2 - \|x\|^2 + \|H\hat{s}\|^2$ ,  $\hat{s}_{m|m+1} = \hat{s}_m$
- 2a. (Bounds for  $s_k$ ) Set  $z = \frac{d'_k}{r_{k,k}}$ ,  $UB(s_k) = \lfloor z + \hat{s}_{k|k+1} \rfloor$ ,  $s_k = \lceil -z + \hat{s}_{k|k+1} \rceil - 1$
- 3a. (Increase  $s_k$ )  $s_k = s_k + 1$ . If  $s_k \leq UB(s_k)$  go to 5a, else to 4a.
- 4a. (Increase  $k$ )  $k = k + 1$ ; if  $k = m + 1$  terminate algorithm, else go to 3a.
- 5a. (Decrease  $k$ ) If  $k = 1$  go to 6a. Else  $k = k - 1$ ,  $\hat{s}_{k|k+1} = \hat{s}_k - \sum_{j=k+1}^m \frac{r_{k,j}}{r_{k,k}} (s_j - \hat{s}_j)$ ,  $d'_k = d'_{k+1} - r_{k+1,k+1}^2 (s_{k+1} - \hat{s}_{k+1|k+2})^2$ , and go to 2.
- 6a. Solution found. Save  $s$  and its distance from  $x$ ,  $d'_m - d_1'^2 + r_{1,1}^2 (s_1 - \hat{s}_{1|2})^2$ , and go to 3a.

### 3.2 A First Look at Complexity

The paper [12] gives a complexity analysis of the above algorithm. The main result is that the number of arithmetic operations of the aforementioned algorithms (excluding Steps 1, 2, 3) is at most

$$\frac{1}{6}(2m^3 + 3m^2 - 5m) + \frac{1}{2}(m^2 + 12m - 7) \left( (2\lfloor \sqrt{d^2 t} \rfloor + 1) \binom{\lfloor 4d^2 t \rfloor + m - 1}{\lfloor 4d^2 t \rfloor} + 1 \right), \quad (16)$$

where  $t = \max(r_{1,1}^2, \dots, r_{m,m}^2)$ . In practice  $t$  grows proportional to  $n$  ( $r_{1,1}^2$ , for example, is simply the squared norm of the first column of  $H$ , which has  $n$  entries) and  $d^2$  grows proportional to  $m$  (for more on this see below) and so the upper bound on the number of computations in (16) can be quite large. Our experience with numerical implementations of the algorithm shows that the bound is extremely loose. Moreover, although it does depend on the lattice-generating matrix  $H$  (through the quantity  $t$ ), it offers little insight into the complexity of the algorithm. We will therefore not further consider it.

In this paper we propose to study the complexity of the sphere decoding algorithm using the geometric interpretation we have developed so far. As mentioned earlier, the complexity of the sphere decoding algorithm depends on the size of the generated tree in Fig. 4, which is equal to the sum of the number of lattice points in spheres of radius  $d$  and dimensions  $k = 1, \dots, m$ . The size of this tree depends on the matrix  $H$ , as well as on the vector  $x$ . Therefore, unlike the complexity of solving the unconstrained least-squares problem which only depends on  $m$  and  $n$  and *not* on the specific  $H$  and  $x$ , the complexity of the sphere decoding algorithm is *data-dependent*.

### 3.2.1 Expected Complexity

Of course since the integer least-squares problem is NP hard, the worst-case complexity of sphere decoding is exponential. However, if we assume that the matrix  $H$  and vector  $x$  are generated randomly (according to some known distributions), then the complexity of the algorithm will itself be a random variable. In this case, it is meaningful to study the expected (or average) complexity of sphere decoding, and perhaps even some of its higher order moments.<sup>3</sup>

In what follows we will give a rough argument for the expected complexity of sphere decoding being exponential, although it is not too difficult to make it rigorous. (For a rigorous treatment, albeit using a different approach, see [2].) For an arbitrary point  $x$ , and an arbitrary lattice  $H$ , it is not too difficult to show that the expected number of lattice points inside the  $k$ -dimensional sphere of radius  $d$  is proportional to its volume given by (see, e.g., [15])

$$\frac{\pi^{k/2}}{\Gamma(k/2 + 1)} d^k.$$

Therefore the expected total number of points visited by the sphere decoding is proportional to the total

---

<sup>3</sup>In passing, we should mention that there is recent interest in studying the expected, rather than worst-case, complexity of various algorithms in the computer science literature. The reader may wish to refer to the survey paper [16] and the references therein, as well as the influential papers [17, 18]. In these works, a uniform distribution on the underlying problem is often (artificially) assumed and complexity issues such as NP-completeness, etc., are re-visited. However, as we shall see below, our problem allows for a very natural stochastic model.

number of lattice points inside spheres of dimension  $k = 1, \dots, m$ :

$$P = \sum_{k=1}^m \frac{\pi^{k/2}}{\Gamma(k/2 + 1)} d^k.$$

A simple lower bound on  $P$  can be obtained by considering only the volume of an arbitrary intermediate dimension, say  $\bar{k}$ ,

$$P \geq \frac{\pi^{\bar{k}/2}}{\Gamma(\bar{k}/2 + 1)} d^{\bar{k}} \approx \left( \frac{2e\pi d^2}{\bar{k}} \right)^{\bar{k}/2} \frac{1}{\sqrt{\pi \bar{k}}},$$

where we have assumed  $m \geq \bar{k} \gg 1$  and have used Stirling's formula for the Gamma function. Clearly,  $P$ , and its lower bound, depend on the radius  $d^2$ . This must be chosen in such a way that the probability of the sphere decoder finding a lattice point does not vanish to zero. This clearly requires the volume of the  $m$ -dimensional sphere to not tend to zero, i.e.,

$$\left( \frac{2e\pi d^2}{m} \right)^{m/2} \frac{1}{\sqrt{\pi m}} = O(1),$$

which for large  $m$  implies that  $2e\pi d^2 = m^{1+\frac{1}{m}}$ . Plugging this into the lower bound for  $P$  yields

$$P \geq \left( \frac{m^{1+\frac{1}{m}}}{\bar{k}} \right)^{\bar{k}/2} \frac{1}{\sqrt{\pi \bar{k}}} = \frac{1}{\sqrt{\pi}} \alpha^{\frac{m}{2\alpha} + \frac{1}{2}} m^{\frac{1}{2\alpha} - \frac{1}{2}},$$

where we have defined  $\alpha = m/\bar{k} > 1$ . This last expression clearly shows that the expected number of points  $P$ , and hence the complexity of the algorithm, grows exponentially in  $m$ . (Take, e.g.,  $\alpha = 2$ .)

## 4 A Random Model

Although not unexpected, the above is a discouraging result. In communications applications, however, the vector  $x$  is not arbitrary, but rather is a lattice point perturbed by additive noise with known statistical properties. Thus, we will assume

$$x = Hs + v, \tag{17}$$

where the entries of  $v$  are independent  $\mathcal{N}(0, \sigma^2)$  random variables with known variance and the entries of  $H$  are independent  $\mathcal{N}(0, 1)$  random variables. Furthermore,  $H$  and  $v$  are mutually independent.

## 4.1 Choice of the Radius

The first by-product of this assumption is a method to determine the desired radius  $d$ . Note that  $\frac{1}{2\sigma^2} \cdot \|v\|^2 = \frac{1}{2\sigma^2} \cdot \|x - Hs\|^2$  is a  $\chi^2$  random variable with  $n$  degrees of freedom. Thus we may choose the radius to be a scaled variance of the noise,

$$d^2 = \alpha n \sigma^2,$$

in such a way that with a high probability we find a lattice point inside the sphere,

$$\int_0^{\alpha n/2} \frac{\lambda^{n/2-1}}{\Gamma(n/2)} e^{-\lambda} d\lambda = 1 - \epsilon,$$

where the integrand is the probability density function of the  $\chi^2$  random variable with  $n$  degrees of freedom, and where  $1 - \epsilon$  is set to a value close to 1, say,  $1 - \epsilon = 0.99$ . [If the point is not found, we can increase the probability  $1 - \epsilon$ , adjust the radius, and search again.]

The important point is that the radius  $d$  is chosen based on the statistics of the noise, and *not* based on the lattice  $H$ . Making the choice based on  $H$  quickly leads us to NP hard problems (such as determining the covering radius). Moreover, choosing the radius based on the noise has a beneficial effect on the computational complexity.

## 4.2 Implications for Complexity

Clearly, when  $\sigma^2 = 0$ , i.e., when there is no noise, the exact solution can be found in  $O(mn^2)$  time. (The pseudo-inverse does the trick). On the other hand, when  $\sigma^2 \rightarrow \infty$ , the received vector  $x$  becomes arbitrary, for which we argued in section 3.2 that the expected complexity is exponential. What we are interested in is what happens at intermediate noise levels. In other words, how do we transition from cubic-time to exponential complexity?

In our analysis we shall compute the expected complexity averaged over both the noise  $v$ , as well as over the lattice-generating matrix  $H$ . Thus, we need a random model for  $H$  and will assume that it is comprised of independent  $\mathcal{N}(0, 1)$  entries. This assumption is made for two reasons:

1. It makes the problem analytically tractable.
2. It is also a very reasonable assumption for large, unstructured, matrices  $H$ . (There exist many results in random matrix theory, such as Wigner's semi-circle law, mixing conditions, etc. that are not very sensitive to Gaussian assumptions—see e.g., [19].)

Of course, if  $H$  possesses special structure, such as Toeplitz structure, then this is not a reasonable assumption and the structure must be explicitly taken into account. However, this merits a separate analysis and will be considered in the second part of this paper.

Now, as argued in the previous section, the complexity of the sphere decoding algorithm is proportional to the number of nodes visited on the tree in Figure 4 and, consequently, to the number of points visited in the spheres of radius  $d$  and dimensions  $k = 1, 2, \dots, m$ . Hence the expected complexity is proportional to the number of points in such spheres that the algorithm visits on average. Thus the expected complexity of sphere decoding algorithm is given by

$$C(m, \sigma^2, d^2) = \sum_{k=1}^m \underbrace{(\text{expected \# of points in } k\text{-sphere of radius } d)}_{\triangleq E_p(k, d^2 = \alpha n \sigma^2)} \cdot \underbrace{(\text{flops/point})}_{\triangleq f_p(k) = 2k + 11}. \quad (18)$$

The summation in (18) goes over the dimensions  $k = 1$  through  $k = m$ . The coefficient

$$f_p(k) = 2k + 11$$

is the number of elementary operations (additions, subtractions, and multiplications) that the Fincke-Pohst algorithm performs per each visited point in dimension  $k$ .

We need to compute  $E_p(k, d^2)$ , the expected number of points inside the  $k$ -dimensional sphere of radius  $d$ . Let us first begin with the highest dimension, i.e.,  $k = m$ .

#### 4.2.1 $k = m$

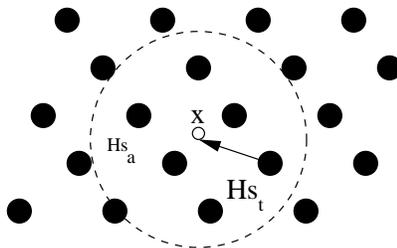


Figure 5:  $s_t$  transmitted and  $x$  received. We are interested whether an arbitrary point  $Hs_a$  lies in a sphere of radius  $d$  centered around  $x$ .

If the lattice point  $s_t$  was transmitted and the vector  $x = Hs_t + v$  received, we are interested in the

number of arbitrary lattice points  $s_a$  such that

$$\|x - Hs_a\|^2 \leq d^2.$$

(See Figure 5.) But, since  $x = Hs_t + v$ , this is just

$$\|v + H(s_t - s_a)\|^2 \leq d^2. \quad (19)$$

Now the vector  $w = v + H(s_t - s_a)$  is clearly a zero-mean Gaussian random vector, since its entries are the sums of zero-mean Gaussian random variables. Now the covariance matrix has  $(i, j)$  entry

$$\begin{aligned} Ew_i w_j &= E \left\{ v_i + \sum_{k=1}^m h_{ik}(s_{t,k} - s_{a,k}) \right\} \left\{ v_j + \sum_{l=1}^m h_{jl}(s_{t,l} - s_{a,l}) \right\} \\ &= \sigma^2 \delta_{ij} + \sum_{k=1}^m \sum_{l=1}^m \delta_{ij} \delta_{kl} (s_{t,k} - s_{a,k})(s_{t,l} - s_{a,l}) \\ &= \delta_{ij} (\sigma^2 + \|s_t - s_a\|^2), \quad \text{where } \|s\|^2 = \sum_{k=1}^m s_k^2. \end{aligned}$$

Thus,  $w$  is an  $n$ -dimensional vector of zero-mean iid Gaussian random variables with variance  $\sigma^2 + \|s_t - s_a\|^2$ . This implies that  $\frac{\|w\|^2}{2(\sigma^2 + \|s_t - s_a\|^2)} = \frac{\|v + H(s_t - s_a)\|^2}{2(\sigma^2 + \|s_t - s_a\|^2)}$  is a  $\chi^2$  random variable with  $n$  degrees of freedom. Thus, the probability that the lattice point  $s_a$  lies in a sphere of radius  $d$  around  $x$  is given by the normalized incomplete gamma function

$$\gamma \left( \frac{d^2}{2(\sigma^2 + \|s_a - s_t\|^2)}, \frac{n}{2} \right) = \int_0^{\frac{d^2}{2(\sigma^2 + \|s_a - s_t\|^2)}} \frac{\lambda^{n/2-1}}{\Gamma(n/2)} e^{-\lambda} d\lambda. \quad (20)$$

Now that we have computed this probability, the expected number of points in the  $m$ -dimensional sphere can be evaluated. However, before doing so, let us turn to the  $k < m$  case.

#### 4.2.2 $k < m$

Referring back to (9), we are interested in all  $k$ -dimensional lattice points  $s$  such that

$$d'^2 \geq \sum_{i=m-k+1}^m \left( y_i - \sum_{j=i}^m r_{i,j} s_j \right)^2. \quad (21)$$

To better understand this set, let us again consider the QR decomposition of (6) to write

$$\begin{aligned} \|x - Hs_a\|^2 = \|v + H(s_t - s_a)\|^2 &= \left\| v + Q \begin{bmatrix} R \\ 0_{(n-m) \times m} \end{bmatrix} (s_t - s_a) \right\|^2 \\ &= \left\| Q^*v + \begin{bmatrix} R \\ 0_{(n-m) \times m} \end{bmatrix} (s_t - s_a) \right\|^2. \end{aligned}$$

[We shall henceforth suppress the subscripts of the all-zero submatrices, as long as their dimensions are self-evident.] Now if we partition the upper triangular matrix  $R$  and the vector  $u \triangleq Q^*v$  as

$$R = \begin{bmatrix} R_{m-k, m-k} & R_{m-k, k} \\ 0_{k \times (m-k)} & R_{k, k} \end{bmatrix} \quad \text{and} \quad u = \begin{bmatrix} u^{m-k} \\ u^k \\ u^{n-m} \end{bmatrix} \quad (22)$$

where the block matrices  $R_{m-k, m-k}$ ,  $R_{m-k, k}$  and  $R_{k, k}$  are  $(m-k) \times (m-k)$ ,  $(m-k) \times k$  and  $k \times k$ , respectively, and the vectors  $u^{m-k}$ ,  $u^k$  and  $u^{n-m}$  are  $m-k$ ,  $k$  and  $n-m$  dimensional, respectively, then we can write

$$\begin{aligned} \|x - Hs_a\|^2 &= \|u^{m-k} + R_{m-k, m-k}(s_t^{m-k} - s_a^{m-k}) + R_{m-k, k}(s_t^k - s_a^k)\|^2 + \\ &\quad \|u^k + R_{k, k}(s_t^k - s_a^k)\|^2 + \|u^{n-m}\|^2, \end{aligned}$$

where the vectors  $s_t^{m-k}$ ,  $s_a^{m-k}$ , and  $s_t^k$ ,  $s_a^k$ , are  $m-k$  and  $k$  dimensional, respectively, and such that

$$s_t = \begin{bmatrix} s_t^{m-k} \\ s_t^k \end{bmatrix}, \quad s_a = \begin{bmatrix} s_a^{m-k} \\ s_a^k \end{bmatrix}.$$

It is now straightforward to see that  $d'^2 = d^2 - \|u^{n-m}\|^2$  and that  $\|u^k + R_{k, k}(s_t^k - s_a^k)\|^2$  is simply the sum of the last  $k$  terms in the sum (8). Thus, we may rewrite the inequality (21) as

$$d^2 \geq \|u^k + R_{k, k}(s_t^k - s_a^k)\|^2 + \|u^{n-m}\|^2 = \left\| \begin{bmatrix} u^k \\ u^{n-m} \end{bmatrix} + \begin{bmatrix} R_{k, k} \\ 0 \end{bmatrix} (s_t^k - s_a^k) \right\|^2. \quad (23)$$

Thus, to compute the expected number of  $k$ -dimensional lattice points that satisfy (21), we need to determine the probability distribution of the RHS of (23). For this we need the following result.

**Lemma 1.** Let  $H$  be an  $n \times m$  (with  $n \geq m$ ) random matrix with iid columns such that each column has a distribution that is rotationally-invariant from the left. In other words, for any  $n \times n$  unitary matrix  $\Theta$ , the distribution of  $\underline{h}_i$ , the  $i$ -th column of  $H$ , satisfies

$$p_{\underline{h}}(\underline{h}_i) = p_{\underline{h}}(\Theta \underline{h}_i).$$

Consider now the QR decomposition

$$H = Q \begin{bmatrix} R \\ 0 \end{bmatrix},$$

where  $Q$  is  $n \times n$  and unitary, and  $R$  is  $m \times m$  and upper triangular with non-negative diagonal entries. Then  $Q$  and  $R$  are independent random matrices, where

1.  $Q$  has an isotropic distribution, i.e., one that is invariant under pre-multiplication by any  $n \times n$  unitary matrix:

$$p_{\mathbf{Q}}(Q) = p_{\mathbf{Q}}(\Theta Q), \quad \forall \Theta \Theta^* = \Theta^* \Theta = I.$$

2. Consider the partitioning of  $R$  according to (22) and further partition  $H$  as

$$H = \begin{bmatrix} H_{m-k, m-k} & H_{m-k, k} \\ H_{n-m+k, m-k} & H_{n-m+k, k} \end{bmatrix},$$

where the subscripts indicate the dimensions of the sub-matrices. Then  $R_{k,k}$  has the same distribution as the  $R$  obtained from the QR decomposition of the  $(n - m + k) \times k$  matrix  $H_{n-m+k, k}$ .

**Proof:** See Appendix A. □

**Remarks:**

1. What is interesting about the above Lemma is that even though the  $(n - m + k) \times k$  submatrix  $\begin{bmatrix} R_{k,k} \\ 0 \end{bmatrix}$  is *not* the  $R$  of the QR decomposition of the  $(n - m + k) \times k$  submatrix  $H_{n-m+k, k}$  it has the *same* distribution.
2. Lemma 1 clearly holds for an  $H$  with iid zero-mean unit-variance Gaussian entries. In this case, one can be explicit about the distribution of  $R$ : the entries are all independent with the  $i$ -th diagonal term

having a  $\chi^2$  distribution with  $n - i + 1$  degrees of freedom and the strictly upper triangular entries having iid zero-mean unit-variance Gaussian distributions.

Let us now apply Lemma 1 to the problem at hand. First, since  $v$  has iid zero-mean  $\sigma^2$ -variance Gaussian entries and  $Q$  is unitary, the same is true of  $u = Q^*v$  and also of the sub-vectors  $u^{m-k}$ ,  $u^k$  and  $u^{n-m}$ . Moreover, since  $Q$  is independent of  $R$ , the same is true of  $u$ . Returning to the inequality (23) let us multiply the vector inside the norm by an isotropically-random unitary matrix  $\Theta$ . Since this does not change norms, we have

$$d^2 \geq \left\| \Theta \begin{bmatrix} u^k \\ u^{n-m} \end{bmatrix} + \Theta \begin{bmatrix} R_{k,k} \\ 0 \end{bmatrix} (s_t^k - s_a^k) \right\|^2.$$

Now clearly, the  $n - m + k$ -dimensional vector  $\bar{v} = \Theta \begin{bmatrix} u^k \\ u^{n-m} \end{bmatrix}$  has iid zero-mean  $\sigma^2$ -variance Gaussian entries. Also, from Lemma 1 part 2, the  $(n - m + k) \times k$  matrix  $\bar{H} = \Theta \begin{bmatrix} R_{k,k} \\ 0 \end{bmatrix}$  has iid zero-mean unit-variance Gaussian entries. Thus, we may write (23) as

$$d^2 \geq \|\bar{v} + \bar{H}(s_t^k - s_a^k)\|^2, \quad (24)$$

which is precisely (19), except that the dimensions have changed from  $n$  and  $m$  to  $n - m + k$  and  $k$ . Thus, using the same argument as presented after (19), we conclude that the probability that the  $k$ -dimensional lattice point  $s_a^k$  lies in a sphere of radius  $d$  around  $x$  is

$$\gamma \left( \frac{d^2}{2(\sigma^2 + \|s_a^k - s_t^k\|^2)}, \frac{n - m + k}{2} \right) = \int_0^{\frac{d^2}{2(\sigma^2 + \|s_a^k - s_t^k\|^2)}} \frac{\lambda^{(n-m+k)/2-1}}{\Gamma((n-m+k)/2)} e^{-\lambda} d\lambda. \quad (25)$$

Given this probability and the one in (20), one could in principle proceed by finding the argument of the gamma function in (20) and (25) for each pair of points  $(s_a, s_t)$ , and sum their contributions; however, even for a finite lattice this would clearly be a computationally formidable task (and not doable at all in the infinite lattice case). Therefore, we shall find it useful to enumerate the lattice, i.e., count the number of points with the same argument of the gamma function in (20) and (25). Enumeration of infinite and finite lattices is treated separately.

### 4.3 The Infinite Lattice Case

The above probability (25) depends only on the Euclidean distance between the points  $s_t^k$  and  $s_a^k$ , that is, on  $\|s_a^k - s_t^k\|^2 = \|s^k\|^2$ . Now, since in an infinite integer lattice  $s_a - s_t = s$  is just another lattice point, we conclude that the probability in (25) depends only on the squared norm of an arbitrary lattice point in the  $k$ -dimensional lattice. It is thus straightforward to see that the expected number of of lattice points inside a  $k$ -dimensional sphere of radius  $d$  is given by:

$$E_p(k, d^2) = \sum_{l=0}^{\infty} \gamma \left( \frac{d^2}{2(\sigma^2 + l)}, \frac{n - m + k}{2} \right) \cdot (\# \text{ of } k\text{-dimensional lattice points with } \|s^k\|^2 = l). \quad (26)$$

Since  $\|s\|^2 = s_1^2 + \dots + s_k^2$ , we basically need to figure out how many ways a non-negative integer  $l$  can be represented as the sum of  $k$  squared integers. This is a classic problem in number theory and the solution is denoted by  $r_k(l)$  [20]. There exist a plethora of results on how to compute  $r_k(l)$ . We only mention one here due to Euler:  $r_k(l)$  is given by the coefficient of  $x^l$  in the expansion

$$\left( 1 + 2 \sum_{m=1}^{\infty} x^{m^2} \right)^k = 1 + \sum_{l=1}^{\infty} r_k(l) x^l. \quad (27)$$

(For more on the problem of representing integers as the sum of squares see Appendix B.)

The above arguments lead to the following result.

**Theorem 1 (Expected complexity of sphere decoding over infinite lattice).** *Consider the model*

$$x = Hs + v,$$

where  $v \in \mathcal{R}^{n \times 1}$  is comprised of iid  $\mathcal{N}(0, \sigma^2)$  entries,  $H \in \mathcal{R}^{n \times m}$  is comprised of iid  $\mathcal{N}(0, 1)$  entries, and  $s \in \mathcal{Z}^m$  is an  $m$ -dimensional vector whose entries are integer numbers. Then the expected complexity of the sphere decoding algorithm of section 3.1 with a search radius  $d$  for solving the integer least-squares problem,

$$\min_{s \in \mathcal{Z}^m} \|x - Hs\|^2,$$

is given by

$$C(m, \sigma^2, d^2) = \sum_{k=1}^m f_p(k) \sum_{l=0}^{\infty} \gamma \left( \frac{d^2}{2(\sigma^2 + l)}, \frac{n - m + k}{2} \right) r_k(l). \quad (28)$$

**Proof:** Follows from the earlier discussions. □

We should remark that, for any given search radius  $d$ , there always exists a probability that no lattice point is found. Therefore, to obtain the optimal solution, it is necessary to increase the search radius. One plausible way of doing this is to start with a radius for which the probability of finding the *transmitted* point is  $1 - \epsilon$ , then if no point is found to increase the search radius to a value such that the probability of not finding the transmitted point is  $1 - \epsilon^2$ , and so on. For such a strategy, we have the following result.

**Corollary 1 (Expected complexity for finding the optimal solution).** *Consider the setting of Theorem 1. Given any  $0 < \epsilon \ll 1$ , consider a strategy where we first choose a radius such that we find the transmitted lattice point with probability  $1 - \epsilon$ , and then increase it to a probability of  $1 - \epsilon^2$ , and so on, if no point is found. Then the expected complexity of the sphere decoding algorithm to find the optimal solution is bounded by*

$$C(m, \sigma^2, \epsilon) \leq \sum_{i=1}^{\infty} (1 - \epsilon)\epsilon^{i-1} \sum_{k=1}^m f_p(k) \sum_{l=0}^{\infty} \gamma\left(\frac{\alpha_i n \sigma^2}{2(\sigma^2 + l)}, \frac{n - m + k}{2}\right) r_k(l), \quad (29)$$

where  $\alpha_i$  is chosen such that

$$\gamma\left(\frac{\alpha_i n}{2}, \frac{n}{2}\right) = 1 - \epsilon^i, \quad i = 1, 2, \dots \quad (30)$$

Note that the probability of having to perform  $i$  decoding steps in order to find the transmitted point can be calculated to be  $(1 - \epsilon)\epsilon^{i-1}$  as follows

$$p(d(i)^2 > \|v\|^2 > d(i-1)^2) = p(\|v\|^2 < d(i)^2) - p(\|v\|^2 < d(i-1)^2) = 1 - \epsilon^i - 1 + \epsilon^{i-1} = (1 - \epsilon)\epsilon^{i-1},$$

where  $d(i)$  denotes sphere radius at the  $i^{\text{th}}$  decoding step. We remark that this is different from the probability of having to perform  $i$  decoding steps in (29) since there is always the *error probability* of the sphere decoder finding a lattice point even when the transmitted lattice point is not inside the sphere. This explains why we have only an upper bound in the above corollary.

#### 4.4 Finite Lattice Case

In communications problems, rather than being unbounded integers, the entries in the unknown  $m$ -dimensional vector  $s$  are often points that belong to an  $L$ -PAM constellations,

$$\mathcal{D}_L = \left\{ -\frac{L-1}{2}, -\frac{L-3}{2}, \dots, \frac{L-3}{2}, \frac{L-1}{2} \right\}. \quad (31)$$

In fact,  $L$  is often taken as a power of 2. We say that the point  $s$  then belongs to the lattice constellation  $\mathcal{D}_L^m$ ,

$$\mathcal{D}_L^m = \underbrace{\mathcal{D}_L \times \mathcal{D}_L \times \dots \times \mathcal{D}_L}_{m\text{-times}},$$

where  $\times$ -operation denotes the Cartesian product.

Furthermore, in this case, rather than the noise variance  $\sigma^2$ , one is interested in the signal-to-noise ratio  $\rho$ ,

$$\rho = \frac{m(L^2 - 1)}{12\sigma^2}. \quad (32)$$

The probability expression (25) for finding an arbitrary lattice point  $s_a^k$  inside a sphere around the given point  $x$  when the point lattice  $s_t^k$  was transmitted, holds for the finite lattice case as well. However, counting the lattice points which have the same argument of the incomplete gamma function in (25) is not as easy. The reason is that unlike in the infinite lattice case, the difference between two lattice points in  $\mathcal{D}_L^k$ ,  $s_a^k - s_t^k$ , is not necessarily another lattice point in  $\mathcal{D}_L^k$ . Thus, the lattice enumeration that we used in the previous section needs to be performed over pairs of points,  $(s_a^k, s_t^k)$ .

More formally, the number of subset lattice points in the  $k$ -dimensional sphere is given by

$$\frac{1}{L^k} \sum_l \sum_{s_t^k, s_a^k \in \mathcal{D}_L^k, \|s_t^k - s_a^k\|^2 = l} \gamma \left( \frac{\alpha n \sigma^2}{2(\sigma^2 + l)}, \frac{n - m + k}{2} \right),$$

and enumerating the set

$$\left\{ (s_t^k, s_a^k) \mid s_t^k, s_a^k \in \mathcal{D}_L^k, \|s_t^k - s_a^k\|^2 = l \right\},$$

appears to be complicated.

For this, we propose a modification of Euler's generating function technique. In particular, for various finite lattices we will define generating polynomials that, when combined appropriately, perform the counting operations for us.

Let us do the case study for various values of  $L$ :

1.  $\mathcal{D}_2^k$ : The constellation  $\mathcal{D}_2^k$  consists of the corners of a  $k$ -dim hypercube, as illustrated in Figure 6.

Due to symmetry, all points in the hypercube are essentially equivalent. Therefore, without loss

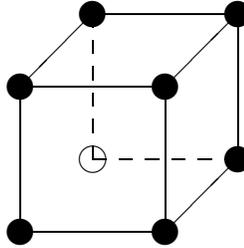


Figure 6: *Counting for  $\mathcal{D}_2^k$*

of generality, we can assume that the “lower-left-corner” point  $s_t^k = [-\frac{1}{2}, -\frac{1}{2}, \dots, -\frac{1}{2}]^T$  has been transmitted. Then, depending on whether the corresponding entry of  $s_a^k$  is  $-\frac{1}{2}$  or  $\frac{1}{2}$ , the vector  $s_a^k - s_t^k$  is comprised of zero and one entries. The number of such vectors whose squared norm is  $l$  is clearly given by the number of vectors that have  $l$  entries at one, which is  $\binom{k}{l}$ . This gives the number of points in  $\mathcal{D}_2^k$  at distance  $l$  from  $s_t^k$ .

2.  $\mathcal{D}_4^k$ : In this case, all points in the constellation are *not* the same. For each entry of  $s_t^k$ , we can distinguish between the “corner” and the “center” points, as illustrated in Figure 7. Extending

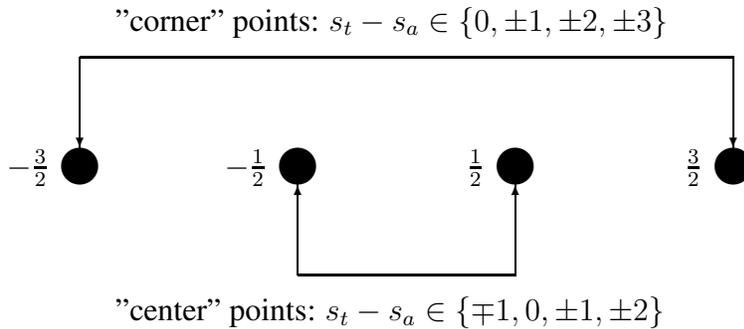


Figure 7: *Counting for  $\mathcal{D}_4^k$*

Euler's idea, for the corner points we identify the generating polynomial

$$\theta_0(x) = 1 + x + x^4 + x^9, \quad (33)$$

and for the center points the polynomial

$$\theta_1(x) = 1 + 2x + x^4. \quad (34)$$

Essentially, the powers in the polynomials  $\theta_0(x)$  and  $\theta_1(x)$  contain information about possible squared distances between an arbitrary point  $s_a^k$  and the transmitted point  $s_t^k$ . For instance, if an entry in the transmitted vector  $s_t^k$ , say  $s_{t,1}$ , is a corner point, then  $s_{a,1} - s_{t,1} \in \{0, \pm 1, \pm 2, \pm 3\}$ , depending on  $s_{a,1} \in \mathcal{D}_4$ . Thus the squared norm of their difference  $|s_{a,1} - s_{t,1}|^2$  can be either 0, 1, 4, or 9, as described by the powers of  $\theta_0(x)$ . On the other hand, if  $s_{t,1}$  is a center point, then  $s_{a,1} - s_{t,1} \in \{0, \pm 1, \mp 1, \pm 2\}$  (which explains coefficient 2 in front of term  $x$  in  $\theta_1(x)$ ). Now, if among the  $k$  entries of  $s_t^k$ , we choose a corner point  $j$  times, the number of ways  $\|s_t^k - s_a^k\|^2$  can add up to  $l$  is given by the coefficient of  $x^l$  in the polynomial

$$\binom{k}{j} \theta_0^j(x) \theta_1^{k-j}(x). \quad (35)$$

3.  $\mathcal{D}_8^k$ : Note that

$$\mathcal{D}_8^k = \left\{ -\frac{7}{2}, -\frac{5}{2}, -\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2} \right\}^k.$$

Let us define the following subsets of  $\mathcal{D}_8$ :

$$\begin{aligned} \mathcal{S}_0 &= \left\{ -\frac{7}{2}, \frac{7}{2} \right\} & \mathcal{S}_1 &= \left\{ -\frac{5}{2}, \frac{5}{2} \right\} \\ \mathcal{S}_2 &= \left\{ -\frac{3}{2}, \frac{3}{2} \right\} & \mathcal{S}_3 &= \left\{ -\frac{1}{2}, \frac{1}{2} \right\} \end{aligned} \quad (36)$$

Similar to the  $L = 4$  case, we can identify the following polynomials for counting  $s_a - s_t$  in  $\mathcal{D}_8^k$

lattice:

$$\begin{aligned}
\psi_0(x) &= 1 + x + x^4 + x^9 + x^{16} + x^{25} + x^{36} + x^{49}, \\
\psi_1(x) &= 1 + 2x + x^4 + x^9 + x^{16} + x^{25} + x^{36}, \\
\psi_2(x) &= 1 + 2x + 2x^4 + x^9 + x^{16} + x^{25}, \\
\psi_3(x) &= 1 + 2x + 2x^4 + 2x^9 + x^{16}.
\end{aligned} \tag{37}$$

Therefore, if among  $k$  entries of  $s_t^k$ , we choose  $j_i$  points from  $\mathcal{S}_i$ ,  $i \in \{0, 1, 2, 3\}$ , then the number of ways  $\|s_t^k - s_a^k\|^2$  can add up to  $l$  is given by the coefficient of  $x^l$  in the polynomial

$$\binom{k}{j_0, j_1, j_2, j_3} \psi_0^{j_0}(x) \psi_1^{j_1}(x) \psi_2^{j_2}(x) \psi_3^{j_3}(x), \tag{38}$$

where  $j_0 + j_1 + j_2 + j_3 = k$  and  $\binom{k}{j_0, j_1, j_2, j_3} = \frac{k!}{j_0! j_1! j_2! j_3!}$ .

4. Counting for  $\mathcal{D}_{16}^k$  and higher order lattices is done similarly.

We can now summarize the above results for the expected computational complexity of the Fincke-Pohst algorithm for finite lattices in the following theorem.

**Theorem 2. [Expected complexity of the sphere decoding over a finite lattice]** *Consider the model*

$$x = Hs + v,$$

where  $v \in \mathcal{R}^{n \times 1}$  is comprised of iid  $\mathcal{N}(0, 1)$  entries,  $H \in \mathcal{R}^{n \times m}$  is comprised of iid  $\mathcal{N}(0, \rho/m)$  entries, and  $s \in \mathcal{D}_L^m$  is an  $m$ -dimensional vector whose entries are elements of an  $L$ -PAM constellation. Then the expected complexity of the sphere decoding algorithm of section (3.1) with a search radius of  $d$  for solving the integer least-squares problem

$$\min_{s \in \mathcal{D}_L^m} \|x - Hs\|^2,$$

1. for a 2-PAM constellation is

$$C(m, \rho, d^2) = \sum_{k=1}^m f_p(k) \sum_{l=0}^k \binom{k}{l} \gamma \left( \frac{\alpha n}{2(1 + \frac{12\rho l}{m(L^2-1)})}, \frac{n - m + k}{2} \right) \tag{39}$$

2. for a 4-PAM constellation is

$$C(m, \rho, d^2) = \sum_{k=1}^m f_p(k) \sum_q \frac{1}{2^k} \sum_{l=0}^k \binom{k}{l} g_{kl}(q) \gamma \left( \frac{\alpha n}{2(1 + \frac{12\rho q}{m(L^2-1)})}, \frac{n-m+k}{2} \right), \quad (40)$$

where  $g_{kl}(q)$  is the coefficient of  $x^q$  in the polynomial

$$(1 + x + x^4 + x^9)^l (1 + 2x + x^4)^{k-l}$$

3. for a 8-PAM constellation is

$$C(m, \rho, d^2) = \sum_{k=1}^m f_p(k) \sum_q \frac{1}{4^k} \sum_{j_0+j_1+j_2+j_3=k} g_{kj_0j_1j_2j_3}(q) \gamma \left( \frac{\alpha n}{2(1 + \frac{12\rho q}{m(L^2-1)})}, \frac{n-m+k}{2} \right), \quad (41)$$

where  $g_{kj_0j_1j_2j_3}(q)$  is the coefficient of  $x^q$  in the polynomial (38).

4. similar expressions can be obtained for 16-PAM, etc., constellations.

The number of elementary operations per visited point in (39)-(41) is  $f_p(k) = 2k + 9 + 2L$ .<sup>4</sup>

**Proof:** Follows from the above discussions. □

We remark that to obtain the optimal solution to the integer least-squares problem we will occasionally need to increase the search radius  $d$ , and so we can obtain a result similar to that of Corollary 1, which we omit for brevity.

## 5 Conclusion

In many communication problems, maximum-likelihood detection reduces to solving an integer least-squares problem, i.e., a search for the closest integer lattice point to the given vector. In such applications ML detection is rarely performed, on the grounds that it requires exponential complexity and is therefore computationally intractable. However, what distinguishes the communications applications from many other instances of integer-least-squares problems is that the given vector is not arbitrary, but rather is an unknown lattice

---

<sup>4</sup>Since  $\mathcal{D}_L^m$  is *shifted* integer lattice, we assume that each rounding in step 2 of the algorithm in Section 3.1 takes  $L-1$  operations. Hence  $f_p(k)$  slightly differs from the one used to find expected complexity of sphere decoding in the infinite lattice  $\mathcal{Z}^m$ .

point that has been perturbed by an additive noise vector with known statistical properties. Therefore, the complexity of finding the closest lattice point for any given algorithm should, in fact, be viewed as a random variable. This is the approach taken in this paper and, for the “sphere decoding” algorithm of Fincke and Pohst, we obtained a closed-form expression for the mean, averaged over both the noise and the lattice, of the complexity. This was done for both finite and infinite lattices.

Based on these closed-form expressions, in the second part of this paper, we will demonstrate that over a wide range of SNRs, rates and dimensions, the expected complexity of sphere decoding is polynomial (often roughly cubic), which implies the practical feasibility of sphere decoding in many applications. The second part of this paper will deal with various generalizations of this result and will also address the computation of the variance of the sphere decoding algorithm.

We should also mention that there are many variants of the sphere decoder algorithm, some of which are mentioned in Sec. 7 of second part of this paper. While these algorithms generally outperform the standard sphere decoder mentioned here, the computation of their expected complexity appears to be a formidable task. Our results may therefore be viewed as upper bounds for these more powerful algorithms.

## A Proof of Lemma 1

Let us start with part 1. Since  $H$  is rotationally-invariant from the left, the matrix  $\Theta H$  has the same distribution as  $H$ , for any unitary matrix  $\Theta$ . The following simple calculation shows that the same is true for any random unitary  $\Theta$  that is independent of  $H$ :

$$p(\Theta H) = \int p(\Theta H|\Theta)p(\Theta)d\Theta = \int p(H)p(\Theta)d\Theta = p(H),$$

where in the second step we used the fact that  $\Theta$  that is independent of  $H$  to conclude  $p(\Theta H|\Theta) = p(H)$ . In particular, the above equation is true for an independent isotropically random unitary  $\Theta$ .<sup>5</sup> Now, for such a  $\Theta$ , we have

$$\Theta H = \Theta Q R,$$

from which, due to the uniqueness of the QR decomposition when  $R$  has positive diagonal entries (see, e.g., [23, 24]), we conclude that  $\Theta Q$  is the Q and  $R$  remains the R in the QR decomposition of  $\Theta H$ . Now, since  $\Theta$  is independent isotropically random,  $\Theta Q$  is also isotropically random and, moreover, it is independent of  $Q$ . Therefore  $\Theta Q$  must be independent of  $R$ , as well. Since  $\Theta H$  and  $H$  have the same distribution, the Q's in their QR decompositions must have the same distribution, from which we conclude that  $Q$  must be an isotropically random unitary matrix, independent of  $R$ .

This concludes the proof of part 1. [We remark that the proof of part 1 only required that  $H$  be rotationally-invariant. We did not require the independence of the columns of  $H$ . This independence is required for the proof of part 2, to which we now turn our attention.]

Consider the partitioning

$$H = \begin{bmatrix} H_{m-k,m-k} & H_{m-k,k} \\ H_{n-m+k,m-k} & H_{n-m+k,k} \end{bmatrix},$$

where the subscripts indicate the dimensions of the sub-matrices. Now consider the QR decomposition of the leading  $m - k$  columns of  $H$ , i.e.,

$$\begin{bmatrix} H_{m-k,m-k} \\ H_{n-m+k,m-k} \end{bmatrix} = Q_1 \begin{bmatrix} P_{m-k,m-k} \\ 0 \end{bmatrix},$$

where  $Q_1$  is  $n \times (m - k)$  unitary and  $P_{m-k,m-k}$  is upper triangular with non-negative diagonal entries. Now

---

<sup>5</sup>For some of the properties of isotropically random unitary matrices the reader may refer to [21, 22].

since  $P_{m-k, m-k}^* P_{m-k, m-k} = H_{m-k, m-k}^* H_{m-k, m-k} + H_{n-m+k, m-k}^* H_{n-m+k, m-k}$ , which is the leading  $(m-k) \times (m-k)$  submatrix of  $H^* H$ , we conclude that  $P_{m-k, m-k}$  is indeed the leading  $(m-k) \times (m-k)$  submatrix of  $R$  in the partitioning (22), i.e.,  $P_{m-k, m-k} = R_{m-k, m-k}$ .

Applying the unitary matrix  $Q_1^*$  to the full  $H$  (and not just its leading  $m-k$  columns) we have

$$Q_1^* H = \begin{bmatrix} R_{m-k, m-k} & \bar{H}_{m-k, k} \\ 0 & \bar{H}_{n-m+k, k} \end{bmatrix}. \quad (\text{A.1})$$

Now, since  $Q_1$  depends only on the first  $m-k$  columns of  $H$  and these are independent of the remaining  $k$  columns, by the rotational-invariance of the columns of  $H$ , we conclude that  $\bar{H}_{n-m+k, k}$  has the *same* distribution as  $H_{n-m+k, k}$ .<sup>6</sup> Now if we consider the QR decomposition of  $\bar{H}_{n-m+k, k}$ :

$$\bar{H}_{n-m+k, k} = Q_2 \begin{bmatrix} R_{k, k} \\ 0 \end{bmatrix},$$

combining this with (A.1), we have

$$Q_1^* H = \begin{bmatrix} I & 0 \\ 0 & Q_2 \end{bmatrix} \begin{bmatrix} R_{m-k, m-k} & \bar{H}_{m-k, k} \\ 0 & R_{k, k} \\ 0 & 0 \end{bmatrix},$$

and so

$$H = Q_1 \begin{bmatrix} I & 0 \\ 0 & Q_2 \end{bmatrix} \begin{bmatrix} R_{m-k, m-k} & \bar{H}_{m-k, k} \\ 0 & R_{k, k} \\ 0 & 0 \end{bmatrix}. \quad (\text{A.2})$$

Since  $Q_1 \begin{bmatrix} I & 0 \\ 0 & Q_2 \end{bmatrix}$  is unitary and since the diagonal entries of  $R_{m-k, m-k}$  and  $R_{k, k}$  are non-negative, we conclude that this is indeed the QR decomposition of  $H$  (which justifies our use of the notation  $R_{k, k}$  for the  $R$  in the QR of  $\bar{H}_{k, k}$ ).<sup>7</sup> Since  $\bar{H}_{n-m+k, k}$  and  $H_{n-m+k, k}$  have the same distribution, we conclude that  $R_{k, k}$  has the *same* distribution as the  $R$  obtained from the QR decomposition of  $H_{n-m+k, k}$ .

This concludes the proof of part 2.

---

<sup>6</sup>This is also true of  $\bar{H}_{m-k, k}$  and  $H_{m-k, k}$ , though we shall not need this fact.

<sup>7</sup>Thus, for example,  $\bar{H}_{m-k, k} = R_{m-k, k}$ .

## B Representing Integers as a Sum of Squares

The quest of representing a positive integer as the sum of squares has a long history in mathematics and number theory. The problem of determining the number of ways that a non-negative integer  $l$  can be represented as  $k$  squares was first posed by Waring in 1770 and is denoted by  $r_k(l)$ .<sup>8</sup> The first known result in this direction is due to Diophantus of Alexandria (325-409 A.D.) who showed that no integer of the form  $4m + 3$  can be represented as the sum of two squares. In other words,  $r_2(4m + 3) = 0$ . In 1632, Girard conjectured that  $l$  is the sum of two squares if the prime divisors of  $l$  of the form  $4m + 3$  occur in  $l$  in an even power. (For example,  $l = 3^2 \cdot 5 = 3^2 + 6^2$ , while  $l = 3^3 \cdot 5$  cannot be represented as the sum of two squares.) Euler proved this conjecture in 1749. However, he did not give an explicit formula for  $r_2(l)$ . This was done by Legendre in 1798 and Gauss in 1801, who showed that

$$r_2(l) = 4(d_1(l) - d_3(l)), \tag{B.1}$$

where  $d_1(l)$  and  $d_3(l)$  are the number of divisors of  $l$  congruent to 1 and 3 mod 4, respectively.

In 1770, Lagrange proved his famous Four Squares Theorem, which states that every positive integer can be represented as the sum of four squares. This essentially establishes that  $r_4(l) > 0$  for all positive integers  $l$ ; however, Lagrange did not give an explicit formula for  $r_4(l)$ .

In terms of computing the value of  $r_k(l)$ , the first result is due to Euler who introduced (what is now known as) the Jacobi theta function

$$\theta(x) = \sum_{m=-\infty}^{\infty} x^{m^2} = 1 + 2 \sum_{m=1}^{\infty} x^{m^2} \tag{B.2}$$

and established the following.

**Theorem 3.** *Let  $\theta(x)$  be given by (B.2). Then*

$$\theta^k(x) = 1 + \sum_{l=1}^{\infty} r_k(l)x^l. \tag{B.3}$$

*In other words, the number of ways a non-negative integer  $l$  can be represented as the sum of  $k$  squares is*

---

<sup>8</sup>In fact, Waring considered the much more general problem of determining the number of ways an integer can be represented as the sum of  $k$  integers raised to the power  $q$ . In this sense, the number of ways an integer can be represented as the sum of  $k$  squares is essentially the  $q = 2$  Waring problem.

given by the coefficient of  $x^l$  in the expansion of  $\theta^k(x)$ .

This can be illustrated as follows:  $\theta^k(x)$  is clearly a series in which each term has an exponent that is obtained as the sum of  $k$  squares; since the summation in (B.2) goes over all integers, the coefficients in front of each term in the series expansion  $\theta^k(x)$  must be equal to the number of ways that the exponent in that same term can be represented as the sum of  $k$  squares.

Using the connection between the above theta function and elliptic functions, Jacobi in 1829 obtained closed-form expressions for  $r_k(l)$  when  $k = 2, 4, 6, 8$  (see [20], chapter 9). His formula for  $k = 4$  immediately yields Lagrange's Four Squares Theorem. Solutions for  $k = 10$  and  $k = 12$  were found by Liouville and Eisenstein. Later, Ramanujan, Hardy, and Littlewood obtained formulas for even  $k \leq 24$ . For odd  $k$ , the only results are due to Dirichlet, who found  $r_3(l)$ , and Eisenstein, Smith, and Minkowski, who found  $r_5(l)$  and  $r_7(l)$ .

For a long time, these were the only known explicit formulas for  $r_k(l)$ . Indeed, results by Glaisher, and by R. Rankin (1965), using the theory of modular forms, discouraged many researchers from obtaining further closed-form expressions. The subject was therefore presumed to be "dead" until very recently. In 1994, as a consequence of their study of certain affine super-algebras, V. Kac and M. Wakimoto conjectured formulas for  $\theta^k(x)$  when  $k = 4m^2$  and  $k = 4m(m + 1)$  [25]. In 1996, these conjectures were proved by S. Milne using Jacobi's elliptic functions, Hankel determinants and continued fractions [26]. For an expository review of this, and subsequent results, the interested reader is referred to [27].

This exhausts known closed-form solutions for  $r_k(l)$ . There exist many asymptotic results (in both  $k$  and  $l$ )—see e.g., [28], chapter 5. In any case, for any given  $k$  and  $l$ , the value of  $r_k(l)$  can be numerically computed using Euler's formula (B.3). Moreover,  $r_k(l)$  is also a built-in function in *Mathematica*, `SumOfSquaresR[k, l]` [29].

## References

- [1] M. Grotschel, L. Lovász, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag, 2nd ed., 1993.
- [2] M. Ajtai, “The shortest vector problem in  $L_2$  is NP-hard for randomized reductions,” *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 10–19, 1998.
- [3] G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” *Bell Labs. Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.
- [4] B. Hassibi, “An efficient square-root algorithm for BLAST,” *submitted to IEEE Trans. Sig. Proc.*, 2000. Download available at <http://mars.bell-labs.com>.
- [5] A. K. Lenstra, H. W. Lenstra, and L. Lovász, “Factoring polynomials with rational coefficients,” *Math. Ann.*, pp. 515–534, 1982.
- [6] M. O. Damen, A. Chkeif, and J.-C. Belfiore, “Lattice code decoder for space-time codes,” *IEEE Comm. Let.*, pp. 161–163, May 2000.
- [7] B. Hassibi and B. Hochwald, “High-rate codes that are linear in space and time,” *IEEE Trans. Info. Theory*, vol. 48, pp. 1804–1824, July 2002.
- [8] R. Kannan, “Improved algorithms on integer programming and related lattice problems,” *Proc. 15th Annu. ACM Symp. on Theory of Computing*, pp. 193–206, 1983.
- [9] J. Lagarias, H. Lenstra, and C. Schnorr, “Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal,” *Combinatorica*, vol. 10, pp. 333–348, 1990.
- [10] A. Korkin and G. Zolotarev, “Sur les formes quadratiques,” *Math. Ann.*, vol. 6, pp. 366–389, 1873.
- [11] M. Pohst, “On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications,” *ACM SIGSAM Bull.*, vol. 15, pp. 37–44, 1981.
- [12] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Mathematics of Computation*, vol. 44, pp. 463–471, April 1985.
- [13] A. Hassibi and S. Boyd, “Integer parameter estimation in linear models with applications to GPS,” *IEEE Transactions on Signal Processing*, vol. 46, pp. 2938–52, November 1998.

- [14] H. Vikalo, *Sphere Decoding Algorithms for Digital Communications*. PhD Thesis, Stanford University, 2003.
- [15] J. Conway and N. Sloane, *Sphere Packings, Lattices and Graphs*. Springer-Verlag, 1993.
- [16] J. Wang, “Average-case computational complexity theory,” *Complexity Theory Retrospective II*, pp. 295–328, 1997.
- [17] L. Levin, “Average case complete problems,” *SIAM Journal on Computing*, vol. 15, pp. 285–86, 1986.
- [18] Y. Gurevich, “Average case completeness,” *Journal of Computer and System Sciences*, vol. 42, no. 3, pp. 346–398, 1991.
- [19] M. L. Mehta, *Random Matrices*. Academic Press, 2nd ed., 1991.
- [20] G. Hardy, *Ramanujan: Twelve Lectures*. Chelsea Publishing, 1940.
- [21] B. M. Hochwald and T. L. Marzetta, “Unitary space-time modulation for multiple-antenna communication in Rayleigh flat-fading,” *IEEE Trans. Info. Theory*, vol. 46, pp. 543–564, Mar. 2000.
- [22] B. Hassibi and T. L. Marzetta, “Block-fading channels and isotropically-random unitary inputs: The received signal density in closed-form,” *IEEE Trans. on Info. Thy.*, vol. 48, pp. 1473–84, June 2002.
- [23] R. Horn and C. Johnson, *Topics in Matrix Analysis*. Cambridge: University Press, 1991.
- [24] G. Golub and C. V. Loan, *Matrix Computations*. Balitmore: Johns Hopkins University Press, 2nd ed., 1989.
- [25] V. Kac and M. Wakimoto, “Integrable highest weight modules over affine super-algebras and Appell’s function,” *Comm. Math. Phys.*, vol. 215, no. 3, pp. 631–682, 2001.
- [26] S. Milne, “Infinite families of exact sums of squares formulas, Jacobi elliptic functions, continued fractions, and Schur functions,” *Ramanujan*, vol. 6, no. 1, pp. 7–149, 2002.
- [27] I. Peterson, “Surprisingly square,” *Science News*, vol. 159, June 2001.
- [28] M. Knopp, *Modular Functions in Analytic Number Theory*. Chicago, IL: Markham Publishing Company, 1970.
- [29] S. Wolfram, *The Mathematica Book*. Cambridge University Press, 4th ed., 1999.