On the Sphere-Decoding Algorithm I. Expected Complexity

Babak Hassibi and Haris Vikalo

***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 (SNRs) 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*—Expected complexity, integer least-squares problem, lattice problems, multiple-antenna systems, NP hard, sphere decoding, wireless communications.**

I. Introduction and Problem Statement

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

$$\min_{s\in\mathcal{Z}^m} \|x - Hs\|^2 \tag{1}$$

where $x \in \mathcal{R}^{n\times 1}$, $H \in \mathcal{R}^{n\times m}$, and $\mathcal{Z}^m$ denote 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

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

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 Fig. 1).

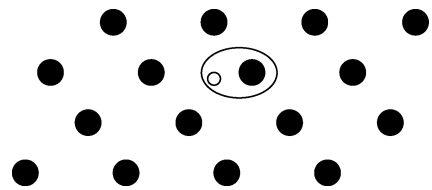

Fig. 1. Geometrical interpretation of the integer least-squares problem.

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 II, 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 III 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 IV 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 signal-to-noise ratio (SNR). This is done both for the infinite lattice as well as some finite lattices encountered in communications problems. Section V provides the conclusion. The Appendixes give some mathematical background for the problems encountered in this paper.

II. 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,

Manuscript received June 25, 2003; revised September 19, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhi Ding.

The authors are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: hassibi@systems.caltech.edu; hvikalo@systems.caltech.edu).

Digital Object Identifier 10.1109/TSP.2005.850352

1053-587X/$20.00 © 2005 IEEE

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

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

$$\hat{s}_B = [H^{\dagger}x]_{\mathcal{Z}^m}. \tag{3}$$

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.

A. 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

$$\min_{t \in \mathcal{Z}^m} \|x - Gt\|^2 \tag{4}$$

using the earlier mentioned heuristics and set $s = Tt$. Of course, lattice reduction is itself NP-hard. A common heuristic is the Lenstra, Lenstra, and Lovász (LLL) [5] algorithm that, 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, therefore, shall not consider lattice reduction methods in this paper.

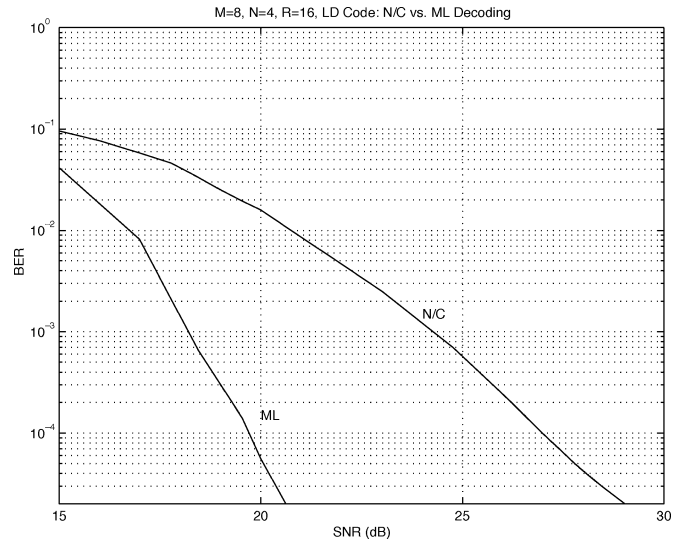

Fig. 2. Bit-error performance of a rate $R = 16$ linear space–time code, corresponding to $m = 64$. Exact ML solution versus nulling/cancelling with optimal ordering. (No. of lattice points $= 2^{128} \approx 3.4 \times 10^{38}$).

B. 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 Fig. 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 16-QAM 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 four 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 Fig. 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.[1]

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 that, 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

[1]Of course, at this point, it may appear surprising that one can even generate the maximum-likelihood (ML) curve in Fig. 2, since it requires finding the exact solution among a set of size 10^{38}—more on this later.

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

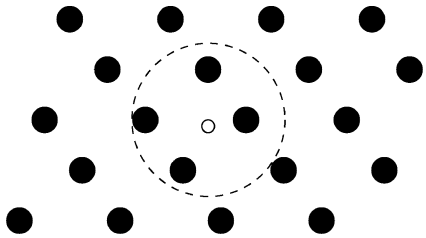

Fig. 3. Idea behind the sphere decoder.

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's sphere-decoding algorithm has been rediscovered in several contexts (see, e.g., [13] in the context of GPS systems) and is the algorithm that 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 Ph.D. dissertation [14].

III. 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 Fig. 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.

1) *How do you 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 reduces to the endpoints of 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 that 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.

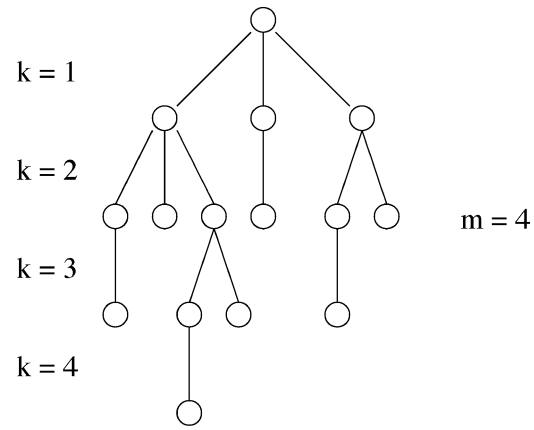

Fig. 4. Sample tree generated to determine lattice points in a four-dimensional sphere.

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 $1, 2, \ldots, m$ 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 kth level of the tree correspond to the lattice points inside the sphere of radius d and dimension k (see Fig. 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.

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 inside a sphere of radius d centered at x if and only if

$$d^2 \geq \|x - Hs\|^2. \tag{5}$$

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

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

where R is an $m \times m$ upper triangular matrix, and $Q = [Q_1 \quad Q_2]$ is an $n \times n$ orthogonal matrix. The matrices Q_1 and Q_2 represent the first m and last $n - m$ orthonormal

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

columns of Q, respectively. The condition (5) can, therefore, be written as

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

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

$$d^2 - \|Q_2^* x\|^2 \geq \|Q_1^* x - Rs\|^2 . \tag{7}$$

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

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

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

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

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 Hs 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

$$\left\lceil \frac{-d' + y_m}{r_{m,m}} \right\rceil \leq s_m \leq \left\lfloor \frac{d' + y_m}{r_{m,m}} \right\rfloor \tag{10}$$

where $\lceil \cdot \rceil$ denotes rounding to the nearest larger element in the set of numbers that spans the lattice.[2] Similarly, $\lfloor \cdot \rfloor$ denotes rounding to the nearest smaller element in the set of numbers that spans the lattice.

Of course, (10) is by no means sufficient. For every s_m satisfying (10), defining $d'^2_{m-1} = d'^2 - (y_m - r_{m,m} s_m)^2$ and $y_{m-1|m} = y_{m-1} - r_{m-1,m} s_m$, a stronger necessary condition can be found by looking at the first two terms in (9), which leads to s_{m-1} belonging to the interval

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

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).

A. Sphere-Decoding Algorithm

We can now formalize the algorithm.

[2]Clearly, for the case of infinite lattice $\mathcal{Z}^m$, this set is the set of integer numbers $\mathcal{Z}$. The set that spans a finite subset of $\mathcal{Z}^m$ is a finite subset of $\mathcal{Z}$, possibly shifted.

Input: $Q = [Q_1 \quad Q_2]$, R, x, $y = Q_1^* x$, d.
1. Set $k = m$, $d'^2_m = d^2 - \|Q_2^* x\|^2$, $y_{m|m+1} = y_m$
2. (Bounds for s_k) Set $UB(s_k) = \lfloor (d'_k + y_{k|k+1})/r_{k,k} \rfloor$, $s_k = \lceil (-d'_k + y_{k|k+1})/r_{k,k} \rceil - 1$.
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 $k = k - 1$, $y_{k|k+1} = y_k - \sum_{j=k+1}^{m} r_{k,j} s_j$, $d'^2_k = d'^2_{k+1} - (y_{k+1|k+2} - r_{k+1,k+1} s_{k+1})^2$, 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 steps 2 and 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

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

Expansion (9) becomes

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

and the intervals (10) and (11)

$$\left\lceil \hat{s}_m - \frac{d'}{r_{m,m}} \right\rceil \leq s_m \leq \left\lfloor \hat{s}_m + \frac{d'}{r_{m,m}} \right\rfloor \tag{14}$$

and

$$\left\lceil \hat{s}_{m-1|m} - \frac{d'_{m-1}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \hat{s}_{m-1|m} + \frac{d'_{m-1}}{r_{m-1,m-1}} \right\rfloor \tag{15}$$

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

Input: R, x, $\hat{s}$, d.
1a. Set $k = m$, $d'^2_m = d^2 - \|x\|^2 + \|H\hat{s}\|^2$, $\hat{s}_{m|m+1} = \hat{s}_m$
2a. (Bounds for s_k) Set $z = d'_k / r_{k,k}$, $UB(s_k) = \lfloor z + \hat{s}_{k|k+1} \rfloor$, $s_k = \lceil -z + \hat{s}_{k|k+1} \rceil - 1$

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

3a. (Increase s_k) $s_k = s_k + 1$. If $s_k \leq UB(s_k)$, go to 5a; else, go 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, $k = k - 1$, $\hat{s}_{k|k+1} = \hat{s}_k - \sum_{j=k+1}^{m}(r_{k,j}/r_{k,k})(s_j - \hat{s}_j)$, $d_k'^2 = d_{k+1}'^2 - r_{k+1,k+1}^2(s_{k+1} - \hat{s}_{k+1|k+2})^2$, and go to 2.
6a. Solution found. Save s and its distance from x, $d_m'^2 - d_1'^2 + r_{1,1}^2(s_1 - \hat{s}_{1|2})^2$, and go to 3a.

B. 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 1–3) is at most

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

where $t = \max(r_{1,1}^2, \ldots, r_{m,m}^2)$. In practice, t grows proportionally 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 proportionally 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, therefore, will 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 $k = 1, \ldots, m$. 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*.

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.[3]

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])

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

Therefore, the expected total number of points visited by the sphere decoding is proportional to the total number of lattice points inside spheres of dimension $k = 1, \ldots, m$

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

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

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

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 not to tend to zero, i.e.,

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

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

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

where we have defined $\delta = m/\bar{k} > 1$. 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., $\delta = 2$).

IV. Random Model

Although not unexpected, the above is a discouraging result. In communications applications, however, the vector x is not

[3]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] and [18]. In these works, a uniform distribution on the underlying problem is often (artificially) assumed, and complexity issues, such as NP-completeness, etc., are revisited. However, as we shall see below, our problem allows for a very natural stochastic model.

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

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.

A. Choice of the Radius

The first by-product of this assumption is a method to determine the desired radius d. Note that $(1/\sigma^2) \cdot \|v\|^2 = (1/\sigma^2) \cdot \|x - Hs\|^2$ is a χ^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

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

where the integrand is the probability density function of the χ^2 random variable with n degrees of freedom, and $1 - \epsilon$ 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, as noted in [19] and [20], choosing the radius based on the noise has a beneficial effect on the computational complexity.

B. 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 $\sigma^2 \to \infty$, the received vector x becomes arbitrary, for which we argued in Section III-B 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 semicircle law, mixing conditions, etc., that are not very sensitive to Gaussian assumptions (see, e.g., [21]).

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 Fig. 4 and, consequently, to the number of points visited in the spheres of radius d and dimensions $k = 1, 2, \ldots, m$. 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 the sphere-decoding algorithm is given by

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

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., $k = m$.

1) $k = m$: 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

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

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

$$\|v + H(s_t - s_a)\|^2 \le d^2. \tag{19}$$

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

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

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

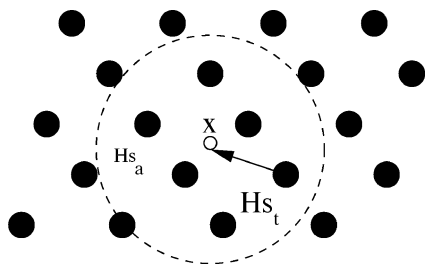

Fig. 5. s_t transmitted and x received. We are interested whether an arbitrary point Hs_a lies in a sphere of radius d centered around x.

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 $\|w\|^2/2(\sigma^2+\|s_t-s_a\|^2) = \|v+H(s_t-s_a)\|^2/2(\sigma^2+\|s_t-s_a\|^2)$ is a χ^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

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

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.

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

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

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

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

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

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

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

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

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 give the following result.[4]

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 Θ, the distribution of $\underline{h}_i$, which is the ith column of H, satisfies

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

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 premultiplication 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 submatrices. 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 $\left[\begin{smallmatrix} R_{k,k} \\ 0\end{smallmatrix}\right]$ 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

[4]The result as stated is more general than we need, which is for the case of an H with iid Gaussian entries. We give it for the case of a left rotationally invariant H with independent columns, because the result may be of interest itself.

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

about the distribution of R: The entries are all independent with the ith diagonal term having a χ^2 distribution with $n-i+1$ degrees of freedom and the strictly upper triangular entries having iid zero-mean unit-variance Gaussian distributions [22], [23]. In fact, in the Gaussian case, part 2 of Lemma 1 can be inferred from [22, Th. 3.2.14].

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

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

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

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

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

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

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.

C. 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 lattice points inside a k-dimensional sphere of radius d is given by

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

Since $\|s^k\|^2 = s_1^2 + \ldots + s_k^2$, 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 $r_k(l)$ [24]. 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

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

(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 III-A with a search radius d for solving the integer least-squares problem

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

is given by

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

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

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

point is found. Then, the expected complexity of the sphere-decoding algorithm to find the optimal solution is bounded by

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

where α_i is chosen such that

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

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:

$$\begin{aligned} &p\left(d(i)^2 > \|v\|^2 > d(i-1)^2\right) \\ &\quad = p\left(\|v\|^2 < d(i)^2\right) - p\left(\|v\|^2 < d(i-1)^2\right) \\ &\quad = 1-\epsilon^i - 1 + \epsilon^{i-1} = (1-\epsilon)\epsilon^{i-1} \end{aligned}$$

where $d(i)$ denotes sphere radius at the ith 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.

D. 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

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

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

$$\mathcal{D}_L^m = \underbrace{\mathcal{D}_L \times \mathcal{D}_L \times \ldots \times \mathcal{D}_L}_{m\text{–times}}$$

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

Furthermore, in this case, rather than the noise variance σ^2, one is interested in the signal-to-noise ratio ρ

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

The probability expression (25) for finding an arbitrary lattice point Hs_a^k inside a sphere around the given point x when the point s_t^k was transmitted holds for the finite lattice case as well. However, counting the lattice points that 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).

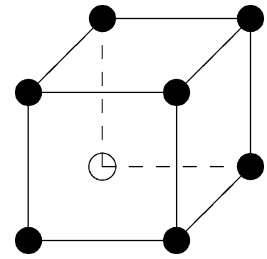

Fig. 6. Counting for $\mathcal{D}_2^k$.

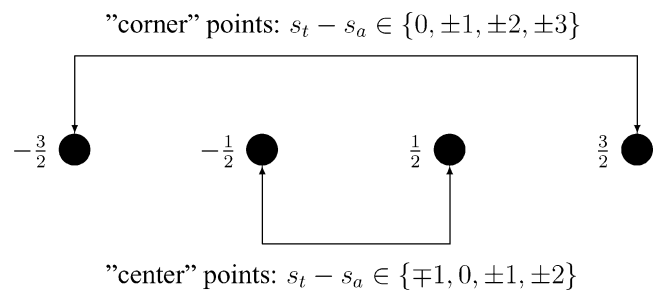

Fig. 7. Counting for $\mathcal{D}_4^k$.

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

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

and enumerating the set

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

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 Fig. 6. Due to symmetry, all points in the hypercube are essentially equivalent. Therefore, without loss of generality, we can assume that the "lower-left-corner" point $s_t^k = [-(1/2), -(1/2), \ldots, -(1/2)]^T$ has been transmitted. Then, depending on whether the corresponding entry of s_a^k is $-(1/2)$ or 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 Fig. 7. Extending Euler's idea, for the corner points, we identify the generating polynomial

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

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

and, for the center points, the polynomial

$$\theta_1(x) = 1 + 2x + x^4. \tag{34}$$

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

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

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 $\mathcal{D}_8$:

$$\mathcal{S}_0 = \left\{ -\frac{7}{2}, \frac{7}{2} \right\}, \quad \mathcal{S}_1 = \left\{ -\frac{5}{2}, \frac{5}{2} \right\}$$
$$\mathcal{S}_2 = \left\{ -\frac{3}{2}, \frac{3}{2} \right\}, \quad \mathcal{S}_3 = \left\{ -\frac{1}{2}, \frac{1}{2} \right\}. \tag{36}$$

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 $\binom{k}{j_0, j_1, j_2, j_3} = k!/j_0!j_1!j_2!j_3!$.

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 III-A 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

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

2) for a 4-PAM constellation is

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

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 an 8-PAM constellation is

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

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 $f_p(k) = 2k + 9 + 2L$.[5]

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]Since $\mathcal{D}_L^m$ is a *shifted* integer lattice, we assume that each rounding in step 2 of the algorithm in Section III-A 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 $\mathcal{Z}^m$.

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

V. Conclusion

In many communication problems, ML 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 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-decoding algorithm, some of which are mentioned in Section VII of the 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.

Appendix

A. Proof of Lemma 1

Let us start with part 1. Since H is rotationally invariant from the left, for any unitary matrix Θ, we may write

$$p_H(H) = p_H(\Theta H).$$

If we consider the QR decomposition $H = QR$, the above equation implies that

$$p_H(H) = p_H(QR) = p_H(R) \tag{A.1}$$

since Q is unitary. In other words, for random matrices that are rotationally invariant from the left, the distribution only depends on the upper triangular matrix R and not on the unitary Q. For example, for a complex Gaussian matrix with iid $\mathcal{CN}(0,1)$ entries, we have $p_H(H) = (1/\pi^{N^2})e^{-\mathrm{tr}H^*H} = (1/\pi^{N^2})e^{-\mathrm{tr}R^*R}$.) We now show that (A.1) implies that Q and R are independent and Q is isotropically random. Indeed, the joint distribution of Q and R is given by

$$p_{Q,R}(Q,R) = p_H(QR) \cdot \frac{\partial H}{\partial(Q,R)} \tag{A.2}$$

where the Jacobian $\partial H/\partial(Q,R)$ can be computed to be (see, e.g., [23, Th. 3.1])

$$\frac{\partial H}{\partial(Q,R)} = K \prod_{i=1}^{m} r_{ii}^{2(n-i)+1} \delta(QQ^* - I)$$

where K is a constant independent of R and Q (that need not concern us here), and the delta function is defined as $\delta(W) = [\prod_{i=1}^{m} \delta(w_{ii})] \prod_{i>j} \delta(\Re(w_{ij}))\delta(\Im(w_{ij}))$ for Hermitian matrices $W = W^*$. Using (A.1) and (A.2), we have

$$p_{Q,R}(Q,R) = K \cdot p_H(R) \prod_{i=1}^{m} r_{ii}^{2(n-i)+1} \delta(QQ^* - I)$$

which, due to the separability of the function, implies

$$p_{Q,R}(Q,R) = p_Q(Q) p_R(R)$$

where

$$p_Q(Q) = K_1 \delta(QQ^* - I), \text{ and } p_R(R) = K_2 p_H(R) \prod_{i=1}^{m} r_{ii}^{2(n-i)+1} \tag{A.3}$$

for suitable constants K_1 and K_2. Thus, Q and R are independent. The distribution $p_Q(Q) = K_1\delta(QQ^* - I)$ shows that Θ is isotropically random since $p_Q(\Theta Q) = p_Q(Q\Omega) = p_Q(Q)$ for all unitary Θ, Ω. (A proof of the fact that the distribution of an isotropically random unitary Q is given by (A.3) can be found in [25].)

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 submatrices. 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, 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

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

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

$$Q_1^*H=\begin{bmatrix} R_{m-k,m-k} & \bar{H}_{m-k,k} \\ 0 & \bar{H}_{n-m+k,k}\end{bmatrix}. \tag{A.4}$$

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 $H_{n-m+k,k}$.[6] 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.4), 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

$$H=Q_1\begin{bmatrix} I & 0 \\ 0 & Q_2\end{bmatrix}\begin{bmatrix} R_{m-k,m-k} & \bar{H}_{m-k,k} \\ 0 & R_{k,k} \\ 0 & 0\end{bmatrix}. \tag{A.5}$$

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 $\bar{H}_{k,k}$).[7] 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.

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 $r_k(l)$.[8] 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

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

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 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 [24, Ch. 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 $k\le 24$. 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 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, Kac and Wakimoto conjectured formulas for $\theta^k(x)$ when $k=4m^2$ and $k=4m(m+1)$ [26]. In 1996, these conjectures were proved by Milne using Jacobi's elliptic functions, Hankel determinants and continued fractions [27]. For an expository review of this and subsequent results, the interested reader is referred to [28].

This exhausts known closed-form solutions for $r_k(l)$. There exist many asymptotic results (in both k and l) (see, e.g., [29,

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

[7]Thus, for example, $\bar{H}_{m-k,k}=R_{m-k,k}$.

[8]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.

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.

Ch. 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]` [30].

REFERENCES

[1] M. Grotschel, L. Lovász, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*, 2nd ed. New York: Springer-Verlag, 1993.

[2] M. Ajtai, "The shortest vector problem in L_2 is NP-hard for randomized reductions," in *Proc. 30th Ann. ACM Symp. Theory Comput.*, 1998, pp. 10–19.

[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," *IEEE Trans. Signal Process.*, submitted for publication.

[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 Commun. Lett.*, vol. 4, no. 5, pp. 161–163, May 2000.

[7] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1804–1824, Jul. 2002.

[8] R. Kannan, "Improved algorithms on integer programming and related lattice problems," in *Proc. 15th Annu. ACM Symp. Theory Comput.*, 1983, pp. 193–206.

[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," *Math. Comput.*, vol. 44, pp. 463–471, Apr. 1985.

[13] A. Hassibi and S. Boyd, "Integer parameter estimation in linear models with applications to GPS," *IEEE Trans. Signal Process.*, vol. 46, no. 11, pp. 2938–2952, Nov. 1998.

[14] H. Vikalo, "Sphere decoding algorithms for digital communications," Ph.D. dissertation, Stanford Univ., Stanford, CA, 2003.

[15] J. Conway and N. Sloane, *Sphere Packings, Lattices and Graphs*. New York: 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 J. Comput.*, vol. 15, pp. 285–286, 1986.

[18] Y. Gurevich, "Average case completeness," *J. Comput. Syst. Sci.*, vol. 42, no. 3, pp. 346–398, 1991.

[19] E. Viterbo and J. Boutros, "A universal lattice decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 1639–1642, Jul. 1999.

[20] E. Viterbo and E. Biglieri, "A universal decoding algorithm for lattice codes," in *Proc. Quatorzieme Colloque GRETSI*, 1993, pp. 611–614.

[21] M. L. Mehta, *Random Matrices*, 2nd ed. New York: Academic, 1991.

[22] R. Muirhead, *Aspects of Multivariate Statistical Theory*. New York: Wiley, 1982.

[23] R. Edelman, "Eigenvalues and condition numbers of random matrices," Ph.D. dissertation, Dept. Math., Mass. Inst. Technol., Cambridge, MA, 1989.

[24] G. Hardy, *Ramanujan: Twelve Lectures*. New York: Chelsea, 1940.

[25] B. M. Hochwald and T. L. Marzetta, "Unitary space-time modulation for multiple-antenna communication in Rayleigh flat-fading," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 543–564, Mar. 2000.

[26] V. Kac and M. Wakimoto, "Integrable highest weight modules over affine super-algebras and Appell's function," *Commun. Math. Phys.*, vol. 215, no. 3, pp. 631–682, 2001.

[27] 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.

[28] I. Peterson, "Surprisingly square," *Sci. News*, vol. 159, pp. 383–383, Jun. 2001.

[29] M. Knopp, *Modular Functions in Analytic Number Theory*. Chicago, IL: Markham, 1970.

[30] S. Wolfram, *The Mathematica Book*, 4th ed. Cambridge, U.K.: Cambridge Univ. Press, 1999.

Babak Hassibi was born in Tehran, Iran, in 1967. He received the B.S. degree from the University of Tehran in 1989 and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1993 and 1996, respectively, all in electrical engineering.

From October 1996 to October 1998, he was a Research Associate with the Information Systems Laboratory, Stanford University, and from November 1998 to December 2000, he was a Member of the Technical Staff in the Mathematical Sciences Research Center, Bell Laboratories, Murray Hill, NJ. Since January 2001, he has been with the Department of Electrical Engineering, California Institute of Technology, Pasadena, where he is currently an Associate Professor. He has also held short-term appointments at Ricoh California Research Center, Menlo Park, CA, the Indian Institute of Science, Bangalore, India, and Linköping University, Linköping, Sweden. His research interests include wireless communications, robust estimation and control, adaptive signal processing, and linear algebra. He is the coauthor of the books *Indefinite Quadratic Estimation and Control: A Unified Approach to H^2 and H^∞ Theories* (Philadelphia, PA: SIAM, 1999) and *Linear Estimation* (Englewood Cliffs, NJ: Prentice Hall, 2000).

Dr. Hassibi is a recipient of an Alborz Foundation Fellowship, the 1999 O. Hugo Schuck Best Paper Award of the American Automatic Control Council, the 2002 National Science Foundation Career Award, the 2002 Okawa Foundation Research Grant for Information and Telecommunications, the 2003 David and Lucille Packard Fellowship for Science and Engineering, and the 2003 Presidential Early Career Award for Scientists and Engineers (PECASE). He has been a Guest Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY special issue on space-time transmission, reception, coding, and signal processing and is currently an Associate Editor for Communications of the IEEE TRANSACTIONS ON INFORMATION THEORY.

Haris Vikalo was born in Tuzla, Bosnia and Herzegovina. He received the B.S. degree from the University of Zagreb, Zagreb, Croatia, in 1995, the M.S. degree from Lehigh University, Bethlehem, PA, in 1997, and the Ph.D. degree from Stanford University, Stanford, CA, in 2003, all in electrical engineering.

He held a short-term appointment at Bell Laboratories, Murray Hill, NJ, in the summer of 1999. From January 2003 to July 2003, he was a Postdoctoral Researcher, and since July 2003, he has been an Associate Scientist at the California Institute of Technology, Pasadena. His research interests include wireless communications, signal processing, estimation. and genomic signal and information processing.

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on June 11,2020 at 06:38:01 UTC from IEEE Xplore. Restrictions apply.