Statistical Pruning for Maximum Likelihood Decoding

RADHIKA GOWAIKAR   BABAK HASSIBI

Department of Electrical Engineering

California Institute of Technology, Pasadena, CA 91125

{gowaikar,hassibi}@systems.caltech.edu

June 27, 2003

Abstract

In many communications problems, maximum-likelihood (ML) decoding reduces to finding the closest (skewed) lattice point in N-dimensions to a given point $x \in \mathcal{C}^N$. In its full generality, this problem is known to be NP-complete and requires complexity exponential in N. Recently, the expected complexity of the sphere decoder, a particular algorithm that solves the ML problem exactly, has been computed where it is shown that over a wide range of rates, SNRs and dimensions the expected complexity is polynomial in N. In this paper, we propose two algorithms that, for large N, offer substantial computational savings over the sphere decoder, while maintaining performance arbitrarily close to ML. Both methods statistically prune the search space to a subset that, with high probability, contains the optimal solution, thereby reducing the complexity of the search. Bounds and approximations for the performance and complexity of both algorithms are provided. Simulations are presented to compare the algorithms with each other as well as with the original sphere decoder.

1 Introduction

Multiple antenna communication systems have been shown to be capable of achieving high data rates. However, reliable decoding in these systems requires very high complexity. For a wide class of space-time transmission schemes (see e.g., [1, 2, 3]) maximum-likelihood (ML) decoding requires us to solve an Integer Least Squares problem. This is the problem of finding the closest (skewed) lattice point in N-dimensions to a given point $x \in \mathcal{C}^N$, which is known in general to be NP-hard. All practical communications systems employ approximations or heuristics and typically require $O(N^3)$ operations (since underlying all the methods is the calculation of a pseudo-inverse). Zero forcing cancellation, nulling and cancelling and nulling and cancelling with optimal ordering [1, 2, 4] are some of these. However, the bit error rate (BER) performance of these is vastly inferior to that of the exact methods.

Exact methods that search over the entire (finite) signal-space require exponential search. More sophisticated exact methods such as Kannan's algorithm [5], the KZ algorithm [6] and the sphere decoding algorithm of [7] attempt to reduce the search space. The branch and bound algorithm, popularly used to solve integer (usually linear) programming problems, could also be used [8]. However, branch and bound imposes additional constraints on the optimizing variables to reduce the size of the problem and also requires one to estimate upper and lower bounds for the objective function to prune the search tree. Hence it is not suitable for the ML decoding problem considered here.

In the sphere decoding algorithm we first determine all lattice points lying in a hypersphere centered at x and then determine the point closest to x. The complexity of the algorithm is therefore determined by the amount of work that is required to determine all lattice point inside a given hypersphere. (For some alternatives to sphere decoding see [6, 9, 10]). It can be shown that, both from a worst-case and from an average point of view, the sphere decoding algorithm requires exponential complexity (see e.g., [11]). In [12] an alternative viewpoint has been taken up where, since in communications problems the noise vector and the lattice-generating-matrix are random, the computational complexity is viewed as a random variable.

Analyzing the expected complexity of sphere decoding, as well as its second-order moment, [12] shows that, over a wide-range of rates, dimensions and SNRs, the algorithm yields polynomial-time (often cubic) complexity. While this is a very interesting result, for large enough N, the expected complexity of sphere decoding does become prohibitively large and non-polynomial.

The above result shows the power of the probabilistic viewpoint and we will continue to use it in this paper. The main point is to view the randomness underlying the problem as an opportunity and to take advantage of it. Thus, we will propose two different modifications to the sphere decoding algorithm that use statistical pruning to reduce the exponentially large search space to one that is much smaller yet contains the optimal solution with high probability. This causes a significant reduction in complexity, at the price of a slight increase in the bit error rate (BER). While both algorithms are straightforward modifications of the original sphere decoding algorithm, the difficulty is in analyzing the resulting performance (which requires finding the probability that the optimal solution is not in the pruned region) as well as the computational complexity (which requires computing the expected number of points that remain in the pruned search space). Thus, the lion share of the paper is devoted to developing exact expressions and, when not possible, bounds or approximations for these quantities. The end result, demonstrated through simulations and "semi-analytic" expressions, is that significant computational savings can be obtained while keeping the BER arbitrarily close to that of the ML decoder.

The remainder of the paper is organized as follows. In Sections 2 and 3 we introduce the Integer Least Squares problem and demonstrate that it arises in the maximum likelihood (ML) decoding of multiple antenna systems. In Section 4 the basic sphere decoding algorithm is explained and in Section 5 the notion of complexity is outlined. We introduce the statistics of the problem and propose two new algorithms viz. the Increasing Radii algorithm and the Ellipsoid Pruning algorithm that exploit them in Section 6. In Section 7 we bound the performance of these algorithms with respect to the optimal, or ML, performance and in Section 8 we analyze the complexity of both the proposed algorithms. We then present simulations in Section 9.

A qualitative discussion of the algorithms as well as ideas for future work and conclusions are to be found in Section 10.

2 Integer Least Squares Problem

The integer least squares problem is the following minimization problem.

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

where $x \in \mathcal{C}^{N \times 1}$ and $H \in C^{N \times M}$ are known and $\mathcal{Z}^{M \times 1}$ is the M-dimensional integer lattice. Often the search space is a finite subset of the infinite lattice, say $\mathcal{A} \subset \mathcal{Z}^{M \times 1}$, in which case the problem becomes

$$\min_{s \in \mathcal{A} \subset \mathcal{Z}^{M \times 1}} \|x - Hs\|^2$$

This problem arises in several situations in communications, cryptography etc. For a general H, it is known to be NP hard in the worst-case sense [13] as well as the average sense [14]. We now describe this problem in the context of ML decoding in a multiple antenna system.

3 System Model

We assume a discrete-time block-fading multiple antenna channel model with M transmit and N receive antennas, where the channel is known to the receiver. This is a reasonable assumption for communication systems where the signalling rate is much higher than the rate at which the propagation environment changes, so that the channel may be learned (perhaps by transmitting known training sequences). If $\mathcal{S}$ is the finite signal constellation, then during any channel use, the transmitted signal $\tilde{s} \in \mathcal{S}^{M \times 1}$ and the received signal

$x \in \mathcal{C}^{N\times 1}$ are related by

$$x = H\tilde{s} + v \tag{1}$$

where $H \in \mathcal{C}^{N\times M}$ is the known channel matrix with independent, identically distributed (i.i.d.) complex Gaussian entries of variance σ_h^2 i.e. $\mathcal{CN}(0, \sigma_h^2)$. We assume $N \geq M$. $v \in \mathcal{C}^{N\times 1}$ is the unknown additive noise vector, comprised of i.i.d. complex Gaussian entries of variance σ_v^2 i.e. $\mathcal{CN}(0, \sigma_v^2)$. We assume $\sigma_v^2 = 1$. With this setup we have $\sigma_h = \frac{\sigma_v}{\sigma_s}\sqrt{\frac{\rho}{M}}$ where ρ is the expected signal-to-noise ratio (SNR) and σ_s^2 is the average power of the signal constellation $\mathcal{S}$. Under the aforementioned assumptions the ML criterion requires us to find $s \in \mathcal{S}^{M\times 1}$ that minimizes $\|x - Hs\|^2$. This is equivalent to the Integer Least Squares problem mentioned in Section 2 where the search space, $\mathcal{A}$, is finite viz. $\mathcal{S}^{M\times 1}$ but has cardinality exponential in M.

This is different from the general Integer Least Squares problem in that H and v are random and hence the complexity of solving this problem is also a random variable. Therefore it is the various moments of the complexity that are of interest to us – we focus on the expected complexity in this paper.

Also, the underlying probability distributions tell us how to prune the search space in order to avoid the exponential complexity of the general Integer Least Squares problem while maintaining performance close to optimal.

In this paper we only consider L^2-QAM constellations with even L i.e.

$$\mathcal{S} = \left\{ a + jb \mid a, b \in \left\{ -\frac{L-1}{2}, -\frac{L-3}{2}, \cdots, \frac{L-3}{2}, \frac{L-1}{2} \right\} \right\} \tag{2}$$

It is then easy to show that $\sigma_s^2 = \frac{L^2-1}{6}$. This gives us $\sigma_h = \sqrt{\frac{6}{L^2-1}\frac{\rho}{M}}$.

4 Sphere Decoder

In sphere decoding we search only over lattice points that lie in a hypersphere of radius r around x, thus reducing the search space and the computation. Therefore we need to find all $s \in \mathcal{S}^{M\times 1}$ that satisfy

$$r^2 \geq \|x - Hs\|^2 \tag{3}$$

To this end, consider the QR decomposition of the channel matrix,

$$H = Q \begin{bmatrix} R \\ 0_{(N-M)\times M} \end{bmatrix}, \tag{4}$$

where R is an $M \times M$ upper triangular matrix with non-negative diagonal entries and Q is an $N \times N$ unitary matrix. Such a decomposition is unique. Let $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ where Q_1 is $N \times M$ and Q_2 is $N \times (N - M)$. We then have

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

Note that $x - Hs = H(\tilde{s} - s) + v$ and $Q_2^* x = Q_2^* v$. Define

$$z = Q^* x - \begin{bmatrix} R \\ 0 \end{bmatrix} s = Q^* v + \begin{bmatrix} R \\ 0 \end{bmatrix} (\tilde{s} - s), \qquad \lambda_i = |z_{N-i+1}|^2 \quad \text{for} \quad i = 1, \ldots, N. \tag{5}$$

(Observe that it is in the z_i variables that we are defining the sphere of (3).) Now finding all s that satisfy (3) amounts to finding all s that satisfy

$$\lambda_1 + \lambda_2 + \ldots + \lambda_N \leq r^2 \tag{6}$$

H and x are known to the receiver. Hence it can determine Q and R. Therefore $Q_2^* x = [z_{M+1}, \ldots, z_N]^T$ is known to the receiver. Hence, so are $\lambda_1, \ldots, \lambda_{N-M}$. Moreover, these are independent of s and $\tilde{s}$. Therefore it is enough to solve

$$\lambda_{N-M+1} + \cdots + \lambda_N \leq r'^2 \tag{7}$$

for a general r' or for $r'^2 = r^2 - (\lambda_1 + \cdots + \lambda_{N-M})$. Note that due to the upper-triangularity of R, λ_i depends only on the unknowns $s_M, \ldots, s_{N-i+1}$ for $i = (N-M+1), \ldots, N$. Therefore (7) can be solved by successively solving

$$\begin{aligned} \lambda_{1+N-M} &\leq r'^2 \\ \lambda_{1+N-M} + \lambda_{2+N-M} &\leq r'^2 \\ &\vdots \\ \lambda_{1+N-M} + \lambda_{2+N-M} + \cdots + \lambda_N &\leq r'^2 \end{aligned} \tag{8}$$

for $s_M, s_{M-1}, \ldots, s_1$. This works because the first condition gives an interval for s_M, and for any predetermined $s_M, \ldots, s_{M-i+2}$, the i-th condition gives an interval for s_{M-i+1}. Once all $s \in \mathcal{S}^{M\times 1}$ that satisfy (7) are known, we can find that s which minimizes $\|x - Hs\|^2$. If there are no solutions found, we increase r' and resolve the problem. For more on the sphere decoder see [12].

5 Computational Complexity

Computational complexity is defined as the number of arithmetical operations required before the decoder gives an output. Apart from the complexity of the QR factorization, the major computation involved in finding the closest point is in determining all points in each lower dimension. We see that the algorithm constructs a search tree where the branches at depth k in the tree correspond to the lattice points inside the hypersphere of radius r and dimension k. Clearly, the total computation involved depends on the *number* of points the decoder visits as it constructs the tree. For a point in the k-th dimension, the number of operations or flops required to process it turn out to be proportional to k. ($2k + 17$ in [12].) Therefore we have

$$C = \sum_{k=1}^{M} (\text{Expected \# of points in } k\text{-sphere of radius } r) \cdot (\text{flops/point}) \tag{9}$$

Thus, the complexity of the algorithm depends on the *size* of the search tree and the computation required at each dimension.

For the setup involving a real channel and 2-PAM as the signal space and the sphere decoder at the receiving end, [12] obtains the following complexity

$$C = \sum_{k=1}^{M} (2k + 17) \sum_{l=0}^{k} \binom{k}{l} \gamma \left(\frac{r^2}{2(1 + \frac{4\rho}{M} l)}, \frac{k + N - M}{2} \right) \tag{10}$$

[12] also has similar expressions for other constellations.

Notice that the radius r has to be chosen carefully. If r is too large, we obtain too many points, but if it is too small, we get no points in the hypersphere and have to redo the computation with a larger r. In [12], a choice of r based on the statistics of the noise is suggested. This r is proportional to N. If we were to choose r' directly, it would be proportional to M.

While the sphere decoding algorithm is one of the more efficient exact methods to solve the maximum likelihood problem with finite constellations (L-PAM, L^2-QAM etc.), it stops giving polynomial complexity at some N which is in the range of practical interest. The reason for this is understood as follows – because the radius r is proportional to N, the algorithm retains a very large fraction of the lattice points (in fact nearly all the points) upto some dimension k before it starts to prune the tree. For instance, if $N = 1000$, we have $r = \alpha N$ such that upto dimension $k = 100$ we keep nearly all the points of the lattice. This already gives us L^{100} points to search over and the complexity quickly becomes exponential.

6 Statistical pruning

With a view to decreasing the computational complexity we now propose modifications to the sphere decoding algorithm that reduce the size of the tree. We suggest two methods – the Increasing Radii Algorithm and the Ellipsoid Pruning Algorithm – both of which define a region around x, different from the hypersphere, to search in. Neither of these perform exact ML decoding but can give performance as close to ML as desired through the choice of certain parameters. Both rely heavily on the statistics of the problem (such as the distribution of λ_i) for performance as well as reduction in complexity.

6.1 Statistics

For any vector $s \in \mathcal{S}^{M\times 1}$ define $s^i \in \mathcal{S}^{i\times 1}$ as the lower length-i subvector of s i.e. the vector $[s_{M-i+1}, \ldots, s_M]^T$. Define $c_i = \frac{1}{\sigma_v^2+\sigma_h^2\|s^i-\tilde{s}^i\|^2}$ and $c_0 = \frac{1}{\sigma_v^2} = 1$. It is shown in Appendix A that the characteristic function for λ_i is

$$Ee^{j\alpha\lambda_i} = \begin{cases} \frac{1}{1-\frac{j\alpha}{c_0}} & \text{for} \quad i \le N-M \\ \frac{(1-\frac{j\alpha}{c_{i-1-N+M}})^{i-1}}{(1-\frac{j\alpha}{c_{i-N+M}})^i} & \text{for} \quad i > N-M \end{cases} \tag{11}$$

and the probability density function (pdf) is

$$p_{\lambda_i}(\lambda_i) = \begin{cases} c_o e^{-c_0\lambda_i} & \text{for} \quad i \le N-M \\ \frac{c_{i-N+M}^i}{c_{i-1-N+M}^{i-1}} e^{-c_{i-N+M}\lambda_i} \sum_{k=0}^{i-1} \binom{i-1}{k} \frac{\lambda_i^k}{k!} (c_{i-1-N+M} - c_{i-N+M})^k & \text{for} \quad i > N-M \end{cases} \tag{12}$$

Note that the $\lambda_i s$ are independent random variables for given vectors s and $\tilde{s}$. Define $\beta_{i,j} = \sum_{k=i}^{j} \lambda_{k+N-M}$ for $1 \le i \le j \le M$. We denote $\beta_{1,i}$ by β_i. It is easy to see that for the β_is we have

$$Ee^{j\alpha\beta_i} = \frac{1}{(1-\frac{j\alpha}{c_i})^{i+N-M}} \tag{13}$$

and

$$p_{\beta_i}(\beta_i) = c_i^i \frac{e^{-c_i \beta_i}}{(i-1)!} \beta_i^{i-1+N-M} \tag{14}$$

From this one can compute the means and variances of λ_i and β_i. We have

$$E\lambda_i = \begin{cases} \frac{1}{c_0} & \text{for} \quad i \leq N-M \\ \frac{i}{c_{i-N+M}} - \frac{(i-1)}{c_{i-1-N+M}} & \text{for} \quad i > N-M \end{cases}$$

$$\text{var}\lambda_i = \begin{cases} \frac{1}{c_0^2} & \text{for} \quad i \leq N-M \\ \frac{i}{c_{i-N+M}^2} - \frac{(i-1)}{c_{i-1-N+M}^2} & \text{for} \quad i > N-M \end{cases}$$

$$E\beta_i = \frac{i}{c_{i-N+M}}$$

$$\text{var}\beta_i = \frac{i}{c_{i-N+M}^2}$$

6.2 Increasing Radii Algorithm (IRA)

The sphere decoder gives exponential complexity for large N because the first several conditions of (8) are very loose. Thus, the tree of the points visited grows exponentially for the first several dimensions. This is also clear from the fact that the sums $\lambda_{1+N-M} + \cdots + \lambda_{k+N-M}$ which occur in (8) viz. β_k have monotonically increasing means while r' is typically chosen on the basis of the distribution of β_M. Therefore the first several conditions do not prune the search space as much as desired. Taking our cue from this, we propose a schedule of radii $r_1 \leq r_2 \leq \ldots \leq r_M$ and solve for

$$\begin{aligned} \lambda_{1+N-M} &\leq r_1^2 \\ \lambda_{1+N-M} + \lambda_{2+N-M} &\leq r_2^2 \\ &\vdots \\ \lambda_{1+N-M} + \lambda_{2+N-M} + \cdots + \lambda_N &\leq r_M^2 \end{aligned} \tag{15}$$

Denote by $\mathcal{D}_{IR,k}$ the region in $\mathcal{S}^{k\times 1}$ containing points that satisfy the first k inequalities of (15). (Note that these points have been determined by finding the values of s_M, s_{M-1} …, s_{M-k+1} that satisfy the first k conditions.) We refer to $\mathcal{D}_{IR,M}$ as $\mathcal{D}_{IR}$ in the following discussion. As in the sphere decoder we can determine all $s \in \mathcal{D}_{IR}$ by solving the inequalities in (15) successively.

6.3 Ellipsoid Pruning Algorithm (EPA)

To motivate this modification we first rewrite (7) as

$$\frac{\lambda_{1+N-M}}{r'^2} + \frac{\lambda_{2+N-M}}{r'^2} + \cdots + \frac{\lambda_N}{r'^2} \leq 1 \tag{16}$$

We note that the λ_is are independent but not identically distributed for $i = 1 + N - M, \ldots, N$. However the equation above – and the sphere decoder which it represents – treats them identically. If we modify the above equation to

$$\frac{\lambda_{1+N-M}}{r_1^2} + \frac{\lambda_{2+N-M}}{r_2^2} + \cdots + \frac{\lambda_N}{r_M^2} \leq 1 \tag{17}$$

the choice of r_is determines the extent of the search region for each λ_i. Note that this is an ellipsoid in the z_i dimensions defined in (5). The r_is allow us to search in a region different from the hypersphere of radius r that the sphere decoder suggests and through the choice of r_is we hope to reduce the search space. Like

(7), we can solve (17) by solving successively for

$$\begin{aligned}
\frac{\lambda_{1+N-M}}{r_1^2} &\leq 1 \\
\frac{\lambda_{1+N-M}}{r_1^2} + \frac{\lambda_{2+N-M}}{r_2^2} &\leq 1 \\
&\vdots \\
\frac{\lambda_{1+N-M}}{r_1^2} + \frac{\lambda_{2+N-M}}{r_2^2} + \cdots + \frac{\lambda_N}{r_M^2} &\leq 1
\end{aligned} \tag{18}$$

Denote by $\mathcal{D}_{EP,k}$ the region in $\mathcal{S}^{k\times 1}$ containing points that satisfy the first k inequalities of (18). (Note that these points have been determined by finding out values of s_M, s_{M-1} ..., s_{M-k+1} that satisfy the first k conditions.) The k-th inequality in (18) is an ellipsoid in dimension k. We refer to $\mathcal{D}_{EP,M}$ as $\mathcal{D}_{EP}$ in the following discussion.

For both algorithms, once the points within $\mathcal{D}_{IR}$ or $\mathcal{D}_{EP}$ are determined, we find that point in $\mathcal{D}_{IR}$ or $\mathcal{D}_{EP}$ which minimizes $\|x - Hs\|$ and declare it as the decoder output. Note that in the IRA the metric used is $\sum_{i=1}^{M} \lambda_{i+N-M} = \|Q_1^* x - Rs\|^2$ which is directly related to $\|x - Hs\|^2$ since $\|Q_1^* x - Rs\|^2 = \|x - Hs\|^2 - \|Q_2^* x\|^2$ and $\|Q_2^* x\|^2$ is a term known to the receiver. In the EPA, however, $\|x - Hs\|^2$ does *not* show up directly.

Note that the regions $\mathcal{D}_{IR}$ and $\mathcal{D}_{EP}$ are, in general, different from the hypersphere. Depending on the values of $r_1, \ldots, r_M$ they may include more or less points than the hypersphere of radius r. To reduce the complexity, we naturally try to reduce the number of points in $\mathcal{D}_{IR}$ or $\mathcal{D}_{EP}$. However, because of the 'asymmetry' of both regions it is possible that the lattice point closest to x does not lie in the search space. Thus, unlike the sphere decoder, we are *not* doing ML decoding and are, potentially, incurring a greater BER. What we get in return is reduced computational complexity. This is the tradeoff inherent in both modifications.

As with the sphere decoder, for both these algorithms, if $\mathcal{D}$ is empty, we increase the search region and run the decoder again.

6.4 Pseudocode

The algorithms are presented below in pseudocode. Both use a depth-first search to construct the tree. It is only in Step 3 of the function DECREASE that the difference between the two shows up. We use the vector r of size $M \times 1$ to denote the schedule $r_1, \ldots, r_M$ that we are using for the decoding. GETNEWSCHEDULE returns the new sequence of r_is with which we repeat the search when the region $\mathcal{D}$ is empty.

function DECODE(x, H, r)

1. $H = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$
2. $t = Q^* x$, $y = [t_1, \ldots, t_M]$
3. $\mathcal{D} = \emptyset$, $y^{''} = r' = s = 0_{M \times 1}$
4. **while** $\mathcal{D} = \emptyset$

    $r =$GETNEWSCHEDULE

    $\mathcal{D}$=DECREASE$(N, y, R, r, y^{''}, r', s, \mathcal{D})$
5. $s^* = \text{argmin}_{s \in \mathcal{D}} \|x - Hs\|^2$
6. **output** s^*

function DECREASE$(k, y, R, r, y^{''}, r', s, \mathcal{D})$

1. **if** $k = 0$

    $\mathcal{D} = \mathcal{D} \cup \{s\}$

    return
2. **elseif** $k = N$

    $r'_k = r_1, \qquad y^{''}_k = y_1$
3. **else**

    $y^{''}_k = y_k - \sum_{j=k+1}^{N} r_{k,j} s_j$

    if INCREASING RADII ALGORITHM

    $$r'_k = \left((r^2_{N-k+1} - r^2_{N-k}) + r'^2_{k+1} - (y^{''}_{k+1} - R_{k+1,k+1} s_{k+1})^2 \right)^{1/2}$$

    elseif ELLIPSOID PRUNING ALGORITHM

    $$r'_k = \left(\frac{r^2_{N-k+1}}{r^2_{N-k}} (r'^2_{k+1} - (y^{''}_{k+1} - R_{k+1,k+1} s_{k+1})^2) \right)^{1/2}$$
4. $LB = \left\lfloor \frac{r'_k + y^{''}_k}{r_{k,k}} \right\rfloor, \qquad UB = \left\lceil \frac{-r'_k + y^{''}_k}{r_{k,k}} \right\rceil - 1$
5. **for** $n = LB : UB$

$s_k = n$

$\mathcal{D} =$DECREASE$(k-1, y, R, r, y^{''}, r', s, \mathcal{D})$

6. **return**

Clearly, for all r_i being equal the modified algorithms are the same as the sphere decoder.

6.5 Comment on the notation

Since we present two algorithms, we use the subscripts IR and EP in the rest of the paper for the *I*ncreasing *R*adii Algorithm and the *E*llipsoid *P*runing Algorithm respectively. However, some of the analysis and insights are common to both. In these cases we drop the subscript. This is clear either from the context or is stated explicitly.

7 Probability of error

Both algorithms redo the search with a new sequence of r_is if the solution set of (15) or (17), viz. $\mathcal{D}$, is empty. Let $\mathcal{D}^i$ be the solution set at the i-th iteration. The algorithm terminates at the first i for which $\mathcal{D}^i \neq \emptyset$. We assume that $\mathcal{D}^{i-1} \subseteq D^i$ and $\mathcal{D}^{\infty} = \mathcal{S}^{M\times 1}$.

Recall that $\tilde{s}$ is the transmitted point. Define $\epsilon_i = P(\tilde{s} \notin \mathcal{D}^i)$. With probability $P(\text{error})$ we make an error by decoding to $s \neq \tilde{s}$.

$$
\begin{aligned}
P(\text{error}) &= \sum_{i=1}^{\infty} P(\text{error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset) \\
&= \sum_{i=1}^{\infty} P(\text{error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \in \mathcal{D}^i) + \sum_{i=1}^{\infty} P(\text{error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin D^i) \\
&= \sum_{i=1}^{\infty} P(\|x - Hs\|^2 \leq \|v\|^2 \text{for } s \in \mathcal{D}^i, s \neq \tilde{s}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \in \mathcal{D}^i) \\
&\qquad\qquad + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin D^i) \\
&\leq \sum_{i=1}^{\infty} P(\text{ML decoder error}, \mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset) + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin D^i) \\
&= P_e^{ML} + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin D^i) \qquad (19) \\
&\leq P_e^{ML} + \sum_{i=1}^{\infty} P(\mathcal{D}^i \neq \emptyset, \mathcal{D}^{i-1} = \emptyset, \tilde{s} \notin D^1) \\
&= P_e^{ML} + \epsilon_1 \qquad (20)
\end{aligned}
$$

where P_e^{ML} is the probability of error with ML decoding. We expect that (19) is a tight bound relating the probability of error of the modified algorithms to P_e^{ML}. This is because it takes into account all the successive schedules of r_i that the algorithms may go through. However it is not clear how to evaluate it exactly and hence we propose the simple bound of (20). This would be equal to (19) if we chose to use only one schedule of r_i and declared all bits to be in error if the corresponding $\mathcal{D}$ turned out to be empty, rather than increasing the r_i and running the decoder again. We now determine $\epsilon = P(\tilde{s} \notin \mathcal{D})$ for the two algorithms.

7.1 ϵ with Increasing Radii Algorithm

Lemma 1. *For the IRA, given a set of* $r_1, \ldots, r_M$, $\epsilon_{IR} = P(\tilde{s} \notin \mathcal{D}_{IR})$ *is given by*

$$\epsilon_{IR} = \sum_{k=1}^{M} e^{-r_k^2} J_{k-1} \tag{21}$$

where

$$J_k = \sum_{l=0}^{k-1} (-1)^{k-l+1} \frac{r_{l+1}^{2(k-l)}}{(k-l)!} J_l, \qquad J_0 = 1$$

Proof. If $s = \tilde{s}$, we have $z = Q^* v$. Since Q is unitary, $Q^* v$ has the same statistics as v i.e. i.i.d. entries distributed as $\mathcal{CN}(0,1)$. With $\lambda_i = |z_{N-i+1}|^2$, we have $p_{\lambda_i}(\lambda_i) = e^{-\lambda_i}$. $1 - \epsilon_{IR}$ is the probability that $\lambda_{1+N-M}, \ldots, \lambda_N$ satisfy (15). Because the λ_is are independent,

$$p_{\lambda_{1+N-M}, \lambda_{2+N-M}, \ldots, \lambda_N}(\lambda_{1+N-M}, \lambda_{2+N-M}, \ldots, \lambda_N) = e^{-(\lambda_{1+N-M} + \lambda_{2+N-M} + \cdots + \lambda_N)}$$

Call the reqion defined by this $\mathcal{P}$. Therefore

$$\begin{aligned} 1 - \epsilon_{IR} &= \int_{\mathcal{P}} e^{-(\lambda_{1+N-M} + \cdots + \lambda_N)} d\lambda_N \ldots d\lambda_{1+N-M} \\ &= \int_0^{r_1^2} \cdots \int_0^{r_M^2 - (\lambda_{1+N-M} + \cdots + \lambda_{N-1})} e^{-(\lambda_{1+N-M} + \cdots + \lambda_N)} d\lambda_N \ldots d\lambda_{1+N-M} \end{aligned}$$

Changing variables to $\mu_i = \sum_{j=1}^{i} \lambda_{j+N-M}$ for $i = 1, \ldots, M$

$$1 - \epsilon_{IR} = \int_0^{r_1^2} \int_{\mu_1}^{r_2^2} \cdots \int_{\mu_{M-1}}^{r_M^2} e^{-\mu_M} d\mu_M \ldots d\mu_1$$

If we call this integral I_M and integrate out μ_M we get

$$I_M = I_{M-1} - e^{-r_M^2} J_{M-1} \tag{22}$$

where $J_{M-1} = \int_0^{r_1^2} \int_{\mu_1}^{r_2^2} \cdots \int_{\mu_{M-2}}^{r_{M-1}^2} d\mu_{M-1} \ldots d\mu_1$. It can be shown that the J_i's satisfy the recurrence

$$J_k = \sum_{l=0}^{k-1} (-1)^{k-l+1} \frac{r_{l+1}^{2(k-l)}}{(k-l)!} J_l, \qquad J_0 = 1 \tag{23}$$

Thus $J_0, \ldots, J_{M-1}$ can be computed. We define $I_0 = 1$. Then, using (22) recursively, we get $I_M = 1 - \sum_{k=1}^{M} e^{-r_k^2} J_{k-1}$. We thus have

$$\epsilon_{IR} = \sum_{k=1}^{M} e^{-r_k^2} J_{k-1}$$

□

7.2 ϵ for Ellipsoid Algorithm

Lemma 2. *For the EPA, given a set of* $r_1, \ldots, r_M$, $\epsilon_{IR} = P(\tilde{s} \notin \mathcal{D}_{IR})$ *is given by*

$$\epsilon_{EP} = \sum_{i=1}^{M} \frac{e^{-r_i^2}}{\prod_{\substack{k=1 \\ k \neq i}}^{M} (1 - \frac{r_i^2}{r_k^2})} \tag{24}$$

Proof. Define $\Lambda = \frac{\lambda_{1+N-M}}{r_1^2} + \frac{\lambda_{2+N-M}}{r_2^2} + \cdots + \frac{\lambda_N}{r_M^2}$. For $s = \tilde{s}$, $Ee^{j\alpha\lambda_i} = \frac{1}{1-j\alpha}$. Hence

$$Ee^{j\alpha\Lambda} = \prod_{i=1}^{M} \frac{1}{1 - \frac{j\alpha}{r_i^2}}$$

We can find $p_\Lambda(\Lambda)$ by the Fourier inverse of the above with respect to α.

$$p_\Lambda(\Lambda) = \sum_{i=1}^{M} \frac{r_i^2 e^{-r_i^2 \Lambda}}{\prod_{\substack{k=1 \\ k \neq i}}^{M} (1 - \frac{r_i^2}{r_k^2})}$$

ϵ_{EP} is the probability that this Λ satisfies $\Lambda \geq 1$. Therefore

$$\begin{aligned} \epsilon_{EP} &= \int_1^\infty p_\Lambda(\Lambda) d\Lambda \\ &= \sum_{i=1}^M \frac{e^{-r_i^2}}{\prod_{\substack{k=1 \\ k \neq i}}^M (1 - \frac{r_i^2}{r_k^2})} \end{aligned}$$

□

7.3 Choice of ϵ

Thus we obtain exact expressions for ϵ in both cases. Once we decide how much worse than ML we are prepared to be, we can choose ϵ using the bound in (20). We can then use the expressions above to determine the radii $r_1, \ldots, r_M$. Note, however, that since both (21) and (24) give highly under-determined equations for the r_is there is an entire family of schedules of r_i that give a particular epsilon. But if we choose a functional form for the radii we can use the expressions obtained above to determine the r_is. For example, if we choose a linear schedule i.e. $r_i^2 = (\delta \log M + i)\sigma_v^2$, we may choose δ such that $\epsilon = 0.01$ etc. Thus we can stay as close to the ML performance as we desire through choice of $r_i s$.

8 Analysis of Computational Complexity

Recall the concept of computational complexity outlined in Section (5). In particular, we focus on the expression in (9). Since we are not searching over hyperspheres any more we have a modified expression for the complexity.

$$C = \sum_{k=1}^M (\text{Expected \# of points in } \mathcal{D}_k) \cdot (\text{flops/point}) \tag{25}$$

From the pseudocode of Section (6.4) we can determine that the flops/point is $8k + 32$.

s^k is as defined in Section 6.1. Define $P(s^k \in \mathcal{D}_k)$ to be the probability that the point s^k is in the search region at dimension k i.e. it satisfies the the first k equations of (15) or (18) according to whether we are using the IRA or the EPA. Clearly

$$\text{Expected \# of points in } \mathcal{D}_k = \sum_{s^k \in \mathcal{S}^{k\times 1}} P(s^k \in \mathcal{D}_k) \tag{26}$$

We now need to compute $P(s^k \in \mathcal{D}_k)$ for both algorithms and then do the sum in (26). Note that the number of terms in the sum is L^{2k} i.e. exponential in k. Naturally, we would like to evaluate the sum *without* having to explicitly evaluate $P(s^k \in \mathcal{D}_k)$ for each of the L^{2k} values of s^k. Whether this can be done or not depends on the functional form of $P(s^k \in \mathcal{D}_k)$. Therefore while determining $P(s^k \in \mathcal{D}_k)$ we also keep in mind the summation of (26).

For the IRA we present an upper bound as well as an approximation to $P(s^k \in \mathcal{D}_{IR,k})$. With the upper bound, the summation of (26) can be done without explicit enumeration, whereas it cannot with the approximation. For the EPA we present an upper bound on $P(s^k \in \mathcal{D}_{EP,k})$ that, after some manipulation, can be summed without explicit enumeration.

8.1 Complexity for the Increasing Radii Algorithm

For any $s^k \in \mathcal{S}^{k\times 1}$, the joint distribution of $\lambda_{1+N-M}, \ldots, \lambda_{k+N-M}$ determines $P(s^k \in \mathcal{D}_{IR,k})$. More specifically,

$$P(s^k \in \mathcal{D}_{IR,k}) = \int_0^{r_1^2} \cdots \int_0^{r_k^2 - (\lambda_{1+N-M} + \cdots + \lambda_{k-1+N-M})} p_{\lambda_{1+N-M},\ldots,\lambda_{k+N-M}}(\lambda_{1+N-M}, \ldots, \lambda_{k+N-M}) d\lambda_{k+N-M} \ldots d\lambda_{1+N-M} \tag{27}$$

We know the distribution of the λ_is from (12). Since the λ_i's are independent we have

$$p_{\lambda_{1+N-M},\ldots,\lambda_{k+N-M}}(\lambda_{1+N-M}, \ldots, \lambda_{k+N-M}) = \prod_{i=1}^{k} p_{\lambda_{i+N-M}}(\lambda_{i+N-M}) \tag{28}$$

Substituting (12) and (28) into (27), the integral for $P(s^k \in \mathcal{D}_{IR,k})$ can be obtained exactly. However, this integral is very involved, and, moreover, even if evaluated exactly, would not give an expression that can be summed easily in (26). Hence we now present one upper bound and one approximation to $P(s^k \in \mathcal{D}_{IR,k})$.

8.1.1 A Simple Upper Bound

Theorem 1. *For the Increasing Radii Algorithm the computational complexity is bounded as*

$$C \leq \sum_{k=1}^{M} (8k+32) \sum_{n=0}^{2k(L-1)^2} G_{L,k}[n] \sum_{l=0}^{N-M} \binom{N-M}{l} \left(1 - \frac{1}{1+\sigma_h^2 n}\right)^l (1+\sigma_h^2 n)^{l-N+M} \Gamma\left(\frac{r_k^2}{1+\sigma_h^2 n}, k+l\right) \tag{29}$$

where $G_{L,k}[n]$ *is the coefficient of* x^n *in* $\frac{1}{L^{2k}}\left(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2}\right)^{2k}$ *and* $\Gamma(x,a) = \int_0^x \frac{e^{-t}}{\Gamma(a)} t^{a-1} dt$

Proof. Recall that $\beta_{i,j} = \sum_{k=i}^{j} \lambda_{k+N-M}$. For any s, let B_i be the event that $\beta_{1,i} \leq r_i^2$ for $i = 1, \ldots, M$. Since the λ_is are independent, we have $Ee^{j\alpha\beta_{1,i}} = \frac{(1-\frac{j\alpha}{c_0})^{N-M}}{(1-\frac{j\alpha}{c_i})^{i+N-M}}$ and

$$P_{\beta_{1,i}}(\beta_{1,i}) = c_i^{i+N-M} e^{-c_i\beta_{1,i}} \sum_{l=0}^{N-M} \binom{N-M}{l} \frac{\beta_{1,i}^{i+l-1}}{(i+l-1)!}(1-c_i)^l.$$

since $c_0 = 1$.

$s^k \in \mathcal{D}_{IR,k}$ if it satisfies the first k conditions of (15). Therefore,

$$
\begin{aligned}
& P(s^k \in \mathcal{D}_{IR,k}) \\
= \;& P(B_1, \ldots, B_k) \\
\leq \;& P(B_k) \\
= \;& \int_0^{r_k^2} p_{\beta_{1,k}}(\beta_{1,k}) d\beta_{1,k} \\
= \;& \sum_{l=0}^{N-M} \binom{N-M}{l} (1-c_k)^l c_i^{N-M-l} \Gamma(c_k r_k^2, k+l) \\
= \;& \sum_{l=0}^{N-M} \binom{N-M}{l} \left(1 - \frac{1}{\sigma_v^2 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2}\right)^l (\sigma_v^2 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2)^{l-N+M} \Gamma\left(\frac{r_k^2}{\sigma_v^2 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2}, k+l\right)
\end{aligned}
$$

where $\Gamma(x,a) = \int_0^x \frac{e^{-t}}{\Gamma(a)} t^{a-1} dt$ is the incomplete gamma function.

We now need to evaluate the summation of (26) with this upperbound. From the definition of $\mathcal{S}$ in (2) it is evident that each entry in $s^k - \tilde{s}^k$ can only take values of the form $x + jy$ where $x, y \in \{-(L-1), -(L-2), \ldots, (L-2), (L-1)\}$. Therefore $\|s^k - \tilde{s}^k\|^2$ can take values in $\{0, \ldots, 2k(L-1)^2\}$. Denote by $r_k^L(n)$ the 'average' number of solutions to $\|s^k - \tilde{s}^k\|^2 = n$. More precisely

$$
r_k^L(n) = \frac{1}{L^{2k}} \sum_{\tilde{s}^k \in \mathcal{S}^{k\times 1}} \text{number of } s^k \in \mathcal{S}^{k\times 1} \text{ such that } \|s^k - \tilde{s}^k\|^2 = n
$$

We have assumed, without loss of generality, that all points are equally likely to be transmitted. Also $\sigma_v^2 = 1$. With this the summation of (26) becomes

$$
\begin{aligned}
& \sum_{s^k \in \mathcal{S}^{k\times 1}} P(s^k \in \mathcal{D}_{IR,k}) \\
\leq \;& \sum_{s^k \in \mathcal{S}^{k\times 1}} \sum_{l=0}^{N-M} \binom{N-M}{l} \left(1 - \frac{1}{1 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2}\right)^l (1 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2)^{l-N+M} \Gamma\left(\frac{r_k^2}{1 + \sigma_h^2 \|s^k - \tilde{s}^k\|^2}, k+l\right) \\
= \;& \sum_{n=0}^{2k(L-1)^2} r_k^L(n) \sum_{l=0}^{N-M} \binom{N-M}{l} \left(1 - \frac{1}{1 + \sigma_h^2 n}\right)^l (1 + \sigma_h^2 n)^{l-N+M} \Gamma\left(\frac{r_k^2}{1 + \sigma_h^2 n}, k+l\right)
\end{aligned}
\tag{30}
$$

It is shown in Appendix B that $r_k^L(n)$ is given by the coefficient of x^n in $G_L^k(x)$ where

$$G_L(x) = \frac{1}{L^2}\left(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2}\right)^2 \tag{31}$$

We denote $G_L^k(x)$ by $G_{L,k}(x)$ and the coefficient of x^n in this by $G_{L,k}[n]$. This gives us $r_k^L(n) = G_{L,k}[n]$. Using this in (30) and the expressions relating to complexity stated in (25) and (26), we get the upper bound in (29). $\square$

8.1.2 Approximate Analysis

It is useful to think of the Increasing Radii algorithm in the context of a random walk on the positive real line. If λ_{i+N-M} is the random variable denoting the length of the i-th step, we can think of the sums $\lambda_{1+N-M}+\lambda_{2+N-M}+\cdots+\lambda_{i+N-M}$ as the total distance covered in the first i steps. In the Increasing Radii algorithm, we are interested in the joint probability that at the end of the i-th step the distance covered is not more than r_i^2 *for all* $i = 1, \ldots, M$. Recall the definition of $\beta_{i,j}$ from the previous section. Note that since the λ_{i+N-M}s are independent (and not identical) and hence independent of $\beta_{1,i-1}$, a parallel interpretation for the sequence $\beta_{1,1}, \beta_{1,2}, \ldots, \beta_{1,M}$s is also that of a (non-stationary) Markov chain in discrete-time and continuous space i.e. $\beta_{1,1} \to \beta_{1,2} \to \ldots \to \beta_{1,M}$. More generally, if we start at some arbitrary step k, and stop at step $k+j$, $j \geq 0$ we still have a Markov chain given by $\beta_{k,k} \to \beta_{k,k+1} \to \ldots \to \beta_{k,k+j}$.

Theorem 2. *For any $s^k \in \mathcal{S}^{k\times 1}$, the probability that $s^k \in \mathcal{D}_{IR,k}$ is given by*

$$P(s^k \in \mathcal{D}_{IR,k}) = \prod_{i=1}^{k} \int_0^{X_i} p_{\lambda_{i+N-M}}(\lambda_{i+N-M})d\lambda_{i+N-M} + O\left(\left[\int_0^{X_k} p_{\lambda_{k+N-M}}(\lambda_{k+N-M})d\lambda_{k+N-M}\right]^3\right) \tag{32}$$

where $X_1 = r_1^2$ and $X_2, \ldots, X_k$ satisfy

$$\int_0^{r_i^2 - r_{i-1}^2 - X_i + X_{i-1}} p_{\lambda_{i+N-M}}(\lambda_{i+N-M})d\lambda_{i+N-M} = \frac{1}{2}\int_0^{X_{i-1}} p_{\lambda_{i+N-M}}(\lambda_{i+N-M})d\lambda_{i+N-M} \tag{33}$$

Instead, if we use the recursion given below for $X_2, \ldots, X_k$

$$X_i = r_i^2 - r_{i-1}^2 - \frac{1+\log 2}{2c_{i-1}} + \frac{X_{i-1}}{2} + \frac{1}{2c_{i-1}}\sqrt{(1+\log 2 + c_{i-1}X_{i-1})^2 - 4c_{i-1}X_{i-1}} \tag{34}$$

(32) continues to hold but the error term becomes $O\left(\left[\int_0^{X_k} p_{\lambda_{k+N-M}}(\lambda_{k+N-M})d\lambda_{k+N-M}\right]^2\right)$.

Note: The integral within the error term is a probability and hence less than 1. Therefore the approximations above are of interest.

Proof. Recall the preceeding discussion regarding Markov chains. In order to use this point of view, we first propose the following approximation, proved in Appendix C.1.

$$\int_0^{\Delta} f(x)g(x)dx = g(x^{'})\int_0^{\Delta} f(x)dx + O\left(\left[\int_0^{\Delta} f(x)dx\right]^2\right) \tag{35}$$

which holds for any $x^{'} \in [0, \Delta]$ and $\int_0^{\Delta} f(x)dx \ll 1$. This approximation is especially good when $x^{'}$ satisfies

$$\int_0^{x^{'}} f(t)dt = \frac{1}{2}\int_0^{\Delta} f(t)dt \tag{36}$$

since the error term then becomes $O\left(\left[\int_0^{\Delta} f(x)dx\right]^3\right)$.

In the Appendix C.2 we make use of this approximation to obtain (32) and (33).

Furthermore, in Appendix C.2 it is shown that if we solve (33) approximately we still get (32) but with a different error term. This approximate solution leads to (34) and is proven in Appendix C.4. □

From (12) we see that $\int_0^{X_k} p_{\lambda_{k+N-M}}(\lambda_{k+N-M})d\lambda_{k+N-M}$ can be evaluated exactly and will be a linear combination of some incomplete gamma functions. A faster but approximate method of evaluating this integral is

$$\int_0^{X_i} p_{\lambda_i}(\lambda_i)d\lambda_i \approx \left(Q(\sqrt{2c_i}(\mu - \sqrt{\pi X_i}/2)) - Q(\sqrt{2c_i}(\mu + \sqrt{\pi X_i}/2))\right)^2 \tag{37}$$

where $\mu = \sqrt{(i-1+N-M)\sigma_h^2|\tilde{s}_i - s_i|^2/2}$ and $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-x^2/2}dx$. The thinking behind this approximation is outlined in Appendix C.4. Although the approximation error has not been rigourously analyzed, we have found this to be a good approximation.

Using the preceeding Theorem, (25), (26) and (37)

$$C = \sum_{k=1}^{N}(8k+32)\sum_{s^k \in \mathcal{S}^{k\times 1}}\prod_{l=1}^{k}(Q(\sqrt{2c_i}(\mu - \sqrt{\pi X_i}/2)) - Q(\sqrt{2c_i}(\mu + \sqrt{\pi X_i}/2)))^2$$

The summation over $s^k \in \mathcal{S}^{k\times 1}$ is not easy to perform and hence we evaluate this expression by Monte Carlo simulations.

8.2 Complexity for the Ellipsoid Algorithm

We now compute $P(s^k \in \mathcal{D}_{EP,k})$. For $i = 1, \ldots, k$, define A_i as the event that s^k satisfies the i-th condition of (18). Observe that $A_k \Rightarrow A_{k-1} \Rightarrow \cdots \Rightarrow A_1$. Hence $P(s^k \in \mathcal{D}_{EP,k}) = P(A_1, \ldots, A_k) = P(A_k)$. To compute this it is sufficient to consider only the k-th condition of (18). Define Λ_k as follows

$$\Lambda_k = \frac{\lambda_{1+N-M}}{r_1^2} + \frac{\lambda_{2+N-M}}{r_2^2} + \cdots \frac{\lambda_{k+N-M}}{r_k^2} \tag{38}$$

Since the λ_is are independent we can use (11) to obtain

$$Ee^{j\alpha\Lambda_k} = \prod_{i=1}^{k}\frac{(1-\frac{j\alpha}{r_i^2 c_{i-1}})^{i-1+N-M}}{(1-\frac{j\alpha}{r_i^2 c_i})^{i+N-M}} \tag{39}$$

We now have

$$P(s^k \in \mathcal{D}_{EP,k}) = \int_0^1 p_{\Lambda_k}(\Lambda_k)d\Lambda_k$$

Although $p_{\Lambda_k}(\Lambda_k)$ is simply the Fourier inverse of the characteristic function given above it is difficult to compute. Therefore we use a Chernoff bound for the integral above.

8.2.1 Chernoff bounds

Theorem 3. *For any $s^k \in \mathcal{S}^{k\times 1}$, and $\|s^k - \tilde{s}^k\|^2 \geq m_k$ the probability that $s^k \in \mathcal{D}_{EP,k}$ is bounded as*

$$\begin{aligned}
P(s^k \in \mathcal{D}_{IR,k}) &\leq e^{k+N-M-r_k^2 c_k} \prod_{i=1}^{k} \frac{(1+\frac{k+N-M-r_k^2 c_k}{r_i^2 c_{i-1}})^{i-1+N-M}}{(1+\frac{k+N-M-r_k^2 c_k}{r_i^2 c_i})^{i+N-M}} && (40)\\
&\leq b_k \sum_{i=1}^{k-1} e^{-r_k^2 c_k} \exp\left(\frac{a_{i,k}}{\left(1+\frac{k+N-M-r_k^2 c_k}{c_i r_i^2}\right)}\right) c_k^{k+N-M} && (41)
\end{aligned}$$

where

$$\begin{aligned}
m_k &= \lceil((k+N-M)/r_k^2 - \sigma_v^2)/\sigma_h^2\rceil\\
b_k &= \frac{r_k^{2(k+N-M)} e^{k+N-M}}{(k-1)(k+N-M)^{(k+N-M)}} \prod_{i=1}^{k-1} \exp\left(-(i+N-M)\left(1-\frac{r_i^2}{r_{i+1}^2}\right)\right)\\
a_{i,k} &= (i+N-M)(k-1)\left(1-\frac{r_i^2}{r_{i+1}^2}\right)
\end{aligned}$$

Note: The bound in (40) holds for all k, that in (41) requires $k \geq 2$.

Proof. Recall Λ_k from (38). We now have

$$\begin{aligned}
P(s^k \in \mathcal{D}_{EP,k}) &= E(I(1-\Lambda_k)) && \text{where } I \text{ is the indicator function}\\
&\leq E(e^{\nu_k(1-\Lambda_k)}) && \text{for } \nu_k \geq 0\\
&= e^{\nu_k} E(e^{-\nu_k \Lambda_k})\\
&= e^{\nu_k} \prod_{i=1}^{k} \frac{(1+\frac{\nu_k}{r_i^2 c_{i-1}})^{i-1+N-M}}{(1+\frac{\nu_k}{r_i^2 c_i})^{i+N-M}} && (42)\\
&= f(\nu_k) && \text{say}
\end{aligned}$$

To find $\nu_k \geq 0$ that minimizes $f(\nu_k)$ we set $\frac{df(\nu_k)}{d\nu_k} = 0$.

$$1 + \sum_{i=1}^{k} \frac{i-1+N-M}{r_i^2 c_{i-1} + \nu_k} - \sum_{i=1}^{k} \frac{i+N-M}{r_i^2 c_i + \nu_k} = 0 \tag{43}$$

Since the upperbound holds for all $\nu_k \geq 0$ we solve this approximately by first rewriting it as

$$1 + \sum_{i=1}^{k-1} \left(\frac{i+N-M}{r_{i+1}^2 c_i + \nu_k} - \frac{i+N-M}{r_i^2 c_i + \nu_k} \right) - \frac{k+N-M}{r_k^2 c_k + \nu_k} = 0$$

and ignoring the terms inside the $\sum$. Thus we have

$$\nu_k = k + N - M - r_k^2 c_k \tag{44}$$

We need this to be positive which gives the condition that $c_k \leq (k+N-M)/r_k^2$ or $\|s^k - \tilde{s}^k\|^2 \geq ((k+N-M)/r_k^2 - \sigma_v^2)/\sigma_h^2$. Since all entries of $(s^k - \tilde{s}^k)$ are integers this becomes $\|s^k - \tilde{s}^k\|^2 \geq \lceil ((k+N-M)/r_k^2 - \sigma_v^2)/\sigma_h^2 \rceil = m_k$, say. The points that do not satisfy this are points that are 'sufficiently' close to $\tilde{s}^k$ in the l_2 sense. For such points we use an upper bound of 1 on $P(s^k \in \mathcal{D}_{EP,k})$.

Therefore we have

$$P(s^k \in \mathcal{D}_{EP,k}) \leq e^{k+N-M-r_k^2 c_k} \prod_{i=1}^{k} \frac{(1 + \frac{k+N-M-r_k^2 c_k}{r_i^2 c_{i-1}})^{i-1+N-M}}{(1 + \frac{k+N-M-r_k^2 c_k}{r_i^2 c_i})^{i+N-M}} \qquad \text{for } \|s^k - \tilde{s}^k\|^2 \geq m_k \tag{45}$$

This is (40).

We now need to use the above in (26). The current form of the upper bound would require explicit summation over the L^{2k} values of s^k or a Monte Carlo simulation. For $k = 1$ the explicit summation can be performed but to avoid it for $k \geq 2$ we propose the following upper bound on the expression of (42) that makes the actual summation over all $s^k \in \mathcal{S}^{k\times 1}, k \geq 2$ possible without exhaustive enumeration.

$$\begin{aligned} P(s^k \in \mathcal{D}_{EP,k}) &\leq e^{\nu_k} \prod_{i=1}^{k} \frac{(1+\frac{\nu_k}{r_i^2 c_{i-1}})^{i-1+N-M}}{(1+\frac{\nu_k}{r_i^2 c_i})^{i+N-M}} && \text{from (42)} \\ &\leq \frac{e^{\nu_k}}{\left(1+\frac{\nu_k}{c_k r_k^2}\right)^{k+N-M}} \prod_{i=1}^{k-1} \exp\left(-(i+N-M)\left(1-\frac{r_i^2}{r_{i+1}^2}\right)\right) \times && (46) \\ &\qquad \frac{1}{(k-1)} \sum_{i=1}^{k-1} \exp\left(\frac{(i+N-M)(k-1)\left(1-\frac{r_i^2}{r_{i+1}^2}\right)}{\left(1+\frac{\nu_k}{c_i r_i^2}\right)}\right) \end{aligned}$$

We prove (46) in Appendix D. Finally, substituting for ν_k from (44) and rearranging we have (41). □

8.2.2 Summing over $s^k \in \mathcal{S}^{k\times 1}$

Theorem 4. *For the Increasing Radii Algorithm the computational complexity is bounded as*

$$C \leq \sum_{k=1}^{M} (8k+32) \cdot A(\mathcal{D}_k) \tag{47}$$

where

$$A(\mathcal{D}_1) = \sum_{i=0}^{m_1-1} G_{L,1}[i] + b_1 \sum_{n=0}^{(L-1)^2} G_{L,1}[n] \frac{e^{-r_1^2/(\sigma_v^2+\sigma_h^2 n)}}{(\sigma_v^2+\sigma_h^2 n)^{k+N-M}} \left(1+\frac{\sigma_v^2(1+N-M)}{r_1^2} - \frac{\sigma_v^2 r_1^2}{\sigma_v^2+\sigma_h^2 n}\right)^{N-M} \tag{48}$$

with

$$b_1 = \frac{r_1^{2(1+N-M)} e^{1+N-M}}{(1+N-M)^{(1+N-M)}}, \qquad m_1 = \lceil ((1+N-M)/r_1^2 - \sigma_v^2)/\sigma_h^2 \rceil$$

and for $k \geq 2$

$$A(\mathcal{D}_k) = \sum_{i=0}^{m_k-1} G_{L,k}[i] + b_k \sum_{i=1}^{k-1} \left(\sum_{u_i=0}^{2i(L-1)^2} \sum_{v_i=\max\{0,m_k-u_i\}}^{2(k-i)(L-1)^2} p'(u_i,v_i) G_{L,i}[u_i] G_{L,k-i}[v_i] \right) \tag{49}$$

with

$$\begin{aligned}
p'(u_i, v_i) &= \frac{e^{-r_k^2/(\sigma_v^2+\sigma_h^2(u_i+v_i))}}{(\sigma_v^2+\sigma_h^2(u_i+v_i))^{k+N-M}} \exp\left(\frac{a_{i,k}}{\left(1+\frac{k+N-M-r_k^2/(\sigma_v^2+\sigma_h^2(u_i+v_i))}{r_i^2/(\sigma_v^2+\sigma_h^2 u_i)}\right)}\right) \\
m_k &= \lceil((k+N-M)/r_k^2-\sigma_v^2)/\sigma_h^2\rceil \\
b_k &= \frac{r_k^{2(k+N-M)} e^{k+N-M}}{(k-1)(k+N-M)^{(k+N-M)}} \prod_{i=1}^{k-1} \exp\left(-(i+N-M)\left(1-\frac{r_i^2}{r_{i+1}^2}\right)\right) \\
a_{i,k} &= (i+N-M)(k-1)\left(1-\frac{r_i^2}{r_{i+1}^2}\right)
\end{aligned}$$

As defined in Theorem 1, $G_{L,k}[n]$ is the coefficient of x^n in $\frac{1}{L^{2k}}\left(L+\sum_{j=1}^{L-1} 2(L-j)x^{j^2}\right)^{2k}$

Proof. We present the analysis for $k \geq 2$. In (41) note that only the c_is depend on s^k. Therefore, while using (41) in (26) it is enough to evaluate

$$\begin{aligned}
&\sum_{\substack{s^k \in \mathcal{S}^{k\times 1} \\ \|s^k-\tilde{s}^k\|^2 \geq m_k}} e^{-r_k^2 c_k} \exp\left(\frac{a_{i,k}}{\left(1+\frac{k+N-M-r_k^2 c_k}{c_i r_i^2}\right)}\right) c_k^{k+N-M} \\
= &\sum_{\substack{s^k \in \mathcal{S}^{k\times 1} \\ \|s^k-\tilde{s}^k\|^2 \geq m_k}} \frac{e^{-r_k^2/(\sigma_v^2+\sigma_h^2\|s^k-\tilde{s}^k\|^2)}}{(\sigma_v^2+\sigma_h^2\|s^k-\tilde{s}^k\|^2)^{k+N-M}} \exp\left(\frac{a_{i,k}}{\left(1+\frac{k+N-M-r_k^2/(\sigma_v^2+\sigma_h^2\|s^k-\tilde{s}^k\|^2)}{r_i^2/(\sigma_v^2+\sigma_h^2\|s^i-\tilde{s}^i\|^2)}\right)}\right)
\end{aligned}$$

for $i = 1, \ldots, k-1$.

Define $s^{i+1:k} = [s_{N-i}, \ldots, s_{N-k+1}]^T$. Denote $\|s^i - \tilde{s}^i\|^2$ by u_i and $\|s^{i+1:k} - \tilde{s}^{i+1:k}\|^2$ by v_i. With this, $\|s^k - \tilde{s}^k\|^2 = u_i + v_i$. The summand above can now be written as a function of u_i and v_i. Call this function $p'(u_i, v_i)$ i.e.

$$p'(u_i, v_i) = \frac{e^{-r_k^2/(\sigma_v^2+\sigma_h^2(u_i+v_i))}}{(\sigma_v^2+\sigma_h^2(u_i+v_i))^{k+N-M}} \exp\left(\frac{a_{i,k}}{\left(1+\frac{k+N-M-r_k^2/(\sigma_v^2+\sigma_h^2(u_i+v_i))}{r_i^2/(\sigma_v^2+\sigma_h^2 u_i)}\right)}\right) \tag{50}$$

Observe that we need $u_i + v_i \geq m_k$. As in Section 8.1.1 u_i can only take values in $\{0, \ldots, 2i(L-1)^2\}$ and v_i in $\{0, \ldots, 2(k-i)(L-1)^2\}$. Recall $r_k^L(n)$ as defined in Section 8.1.1. Now the summation can be written compactly as

$$\sum_{u_i=0}^{2i(L-1)^2} \sum_{v_i=\max\{0,m_k-u_i\}}^{2(k-i)(L-1)^2} p'(u_i,v_i) r_i^L(u_i) r_{k-i}^L(v_i)$$

Analogous to the analysis of Section 8.1.1 we have $r_i^L(u_i)r_{k-i}^L(v_i) = G_{L,i}[u_i] \cdot G_{L,k-i}[v_i]$. We are now set to put together (26) and (41) to get

$$\begin{aligned}
&\text{Expected \# of points in } \mathcal{D}_k \\
= \quad & \sum_{s^k \in \mathcal{S}^{k\times 1}} P(s^k \in \mathcal{D}_k) \\
\leq \quad & \sum_{i=0}^{m_k-1} G_{L,k}[i] + b_k \sum_{i=1}^{k-1} \left(\sum_{u_i=0}^{2i(L-1)^2} \sum_{v_i=\max\{0,m_k-u_i\}}^{2(k-i)(L-1)^2} p'(u_i,v_i) G_{L,i}[u_i] G_{L,k-i}[v_i] \right)
\end{aligned}$$

The first term comes from using an upper bound of 1 on $P(s^k \in \mathcal{D}_{EP,k})$ for $\|s^k - \tilde{s}^k\|^2 < m_k$. b_k is as defined in Theorem 3. This is (49).

For $k = 1$, (40) can be summed using similar techniques giving (48). This, with (25) and (26), completes the proof.

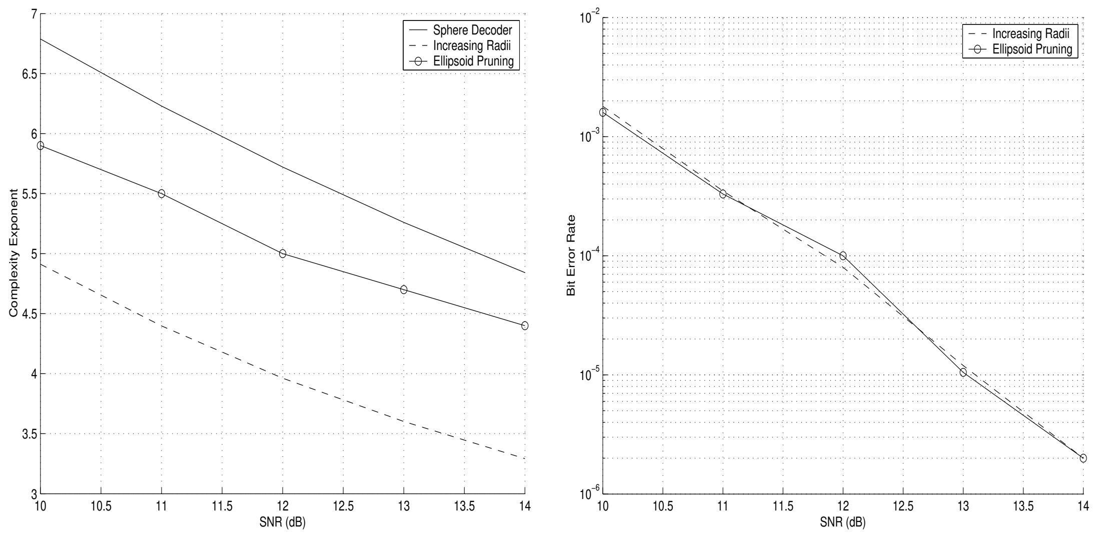
$\square$

9 Simulations

In this section we present the results of simulations for different system parameters. We have $M = N$ in these and $L = 2$. For different values of N and SNR we present the complexity and the BER. We note that since H and $\mathcal{S}$ are complex this amounts to solving $2N$ dimensional real problems. The computational complexity C is presented through the complexity exponent $C_E = \log C/\log N$. With this, a complexity exponent of C_E means that the complexity is N^{C_E}.

In all simulations for the Sphere Decoder we have used a value of r chosen to give a particular ϵ. For the Increasing Radii Algorithm we have used a linear schedule of radii i.e. we have $r_i = i + \delta \log N$ where δ is chosen with some value of ϵ_{IR} in mind. For the Ellipsoid Pruning Algorithm, the choice of r_i proved to be more tricky. We have used a schedule where $r_i = m(i-1)^{0.4} + b$ where $m = (N-1)^{0.6}$ and b is chosen to ensure that ϵ_{EP} takes some particular value.

9.1 Computational Complexity and BER



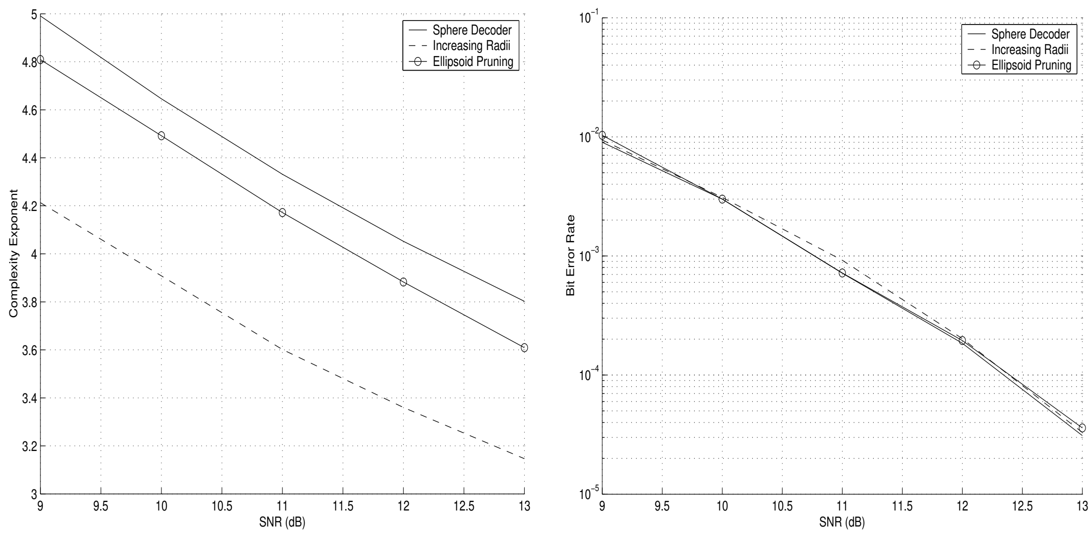
(a) Complexity Exponent

(b) BER

Figure 1: Complexity Exponent and BER for M=N=50 and 4-QAM.

In Figures 1 and 2 we compare the complexity exponent and BER for N=50 and N=20 with SNR ranging from 10dB to 14dB and 9dB to 13dB respectively and a QAM constellation. For all three algorithms we get a schedule as follows. For a particular received vector x and channel H we decode using a schedule designed to give ϵ= 0.1. If this returns an empty $\mathcal{D}$, we rerun the decoder with an increased set of radii that give ϵ=0.01 and so on.

In Fig 1(a) we see that the complexity exponent, C_E, can be reduced significantly while going from the sphere to one of the modified decoders. While the EPA can give a complexity that is upto $50^{1} = 50$ times less than the complexity of the sphere decoder, the computational savings from the IRA are much more



(a) Complexity Exponent

(b) BER

Figure 2: Complexity Exponent and BER for M=N=20 and 4-QAM.

dramatic. We see a complexity exponent that is upto 1.4 orders of magnitude smaller which means that the IRA can run $50^{1.4}$=240 times faster.

In Fig 1(b) we see the BER of the modified algorithms. Unfortunately, we have not been able to produce the BER plot for the sphere decoder for this dimension since it would take too long to give accurate Bit Error Rates.

For the BER comparison we present results of a smaller sized problem viz. N=20 in Fig 2. From Fig 2(a) and Fig 2(b) we see that with computational savings of 0.2 and 0.8 orders of magnitude (i.e. 2 and 11 times less computation) we get a BER that is very close to the optimal BER ensured by ML decoding.

Another observation to make from this is that (20) is a loose bound since for this setup it says that the proposed algorithms can give BERs that are as much as 0.1 above the optimal – a gross over-estimate.

9.2 Comparing complexities

From the previous section it is clear that the modified algorithms can be used to give complexities that are much lower than that of the sphere decoder while still giving BERs close to optimal. Therefore in this section we compare only the complexities of the three algorithms using radius schedules that give the same ϵ.

In Figure 3 we compare the complexity of the sphere decoder with the modified algorithms. For these simulations we have used a fixed schedule of radii for each algorithm that gives an ϵ of 0.1 and run the algorithms sufficiently many times to get a reliable estimate of the complexity exponent. For N taking values 10, 20, 30, 40, 50 and 60 we see the behaviour of the complexity exponent for all three algorithms.

In all cases the modified algorithms give significantly lower values of complexity for a wide range of SNRs. We also see that the Increasing Radii Algorithm gives a much larger decrease in complexity compared to the Ellipsoid Pruning Algorithm for the schedules we are using. We see that for N=60 the Increasing Radii Algorithm gives a reduction factor of about $60^{2.2}$=8165 compared to the sphere decoder.

9.3 Simulations for the Upper Bound and Approximate Analysis for the IRA

We now compare the actual complexity of the Increasing Radii Algorithm as obtained by simulations, with the upper bound and approximation derived in Theorem 1 and Theorem 2.

In Figure 4 we present curves that show the complexity exponent for the Increasing Radii Algorithm. For N being 10, 20, 30, 40, 50 and 60 and for a wide range of SNRs we show the complexity exponent obtained through simulation, by using the upperbound of Theorem 1 and the approximation of Theorem 2 with (37). We see that the upperbound is very good in this entire range. As for the approximation, we see that it is quite good for SNRs upto around 15dB and then starts to underestimate the complexity.

9.4 Simulations for the Upper Bound for the EPA

We now compare the actual complexity of the Ellipsoid Pruning Algorithm as obtained by simulations, with the upper bound obtained in Theorem 4.

In Figure 5 we present curves that show the complexity exponent for the Ellipsoid Pruning Algorithm. For N being 10, 20, 30, 40, 50 and 60 and for a wide range of SNRs we show the complexity exponent obtained through simulation, by using the upperbound of Theorem 3 and the summation technique of Theorem 4. We do not present the complexity values from the simulations for the low SNRs since it is very time-consuming to obtain these reliably as we start operating too close to the capacity of the system. We see that the upperbound is uniformly good this range of SNRs. Not surprisingly, for the high SNR region the upper bound becomes very tight. This is because the Chernoff bound as well as the AM-GM inequality used in the derivation of Theorem 3 become tight. This upperbound can be used as a good estimate of complexity for other purposes as will be explained in the following discussion.

10 Discussion

In this section, we try to get a qualitative feel for why the Increasing Radii Algorithm reduces complexity compared to the sphere decoder. We also investigate the question of what the optimal scheduling of r_l for the modified algorithms should be.

10.1 Savings from approximate algorithms

Consider the complexity expression for the sphere decoder for the case of S being the 4-QAM constellation. This is similar to that for the 2-PAM constellation given in (10). We assume $M = N$.

$$C = \sum_{k=1}^{N} (8k+32) \sum_{l=0}^{2k} \binom{2k}{l} \Gamma\left(\frac{r^2}{1+\frac{2\rho}{N}l}, k\right)$$

Consider the function $\Gamma(x,a) = \int_0^x \frac{e^{-t}}{\Gamma(a)} t^{a-1} dt$. Note that for large values of a this function behaves like $I(x-a)$ where I is the indicator function. Assume that $N \gg 1$ so that for 'most' expressions in the inner summation above we can use this approximation for $\Gamma(x,a)$. We now have

$$\begin{aligned} C &\approx \sum_{k=1}^{N}(8k+32)\sum_{l=0}^{2k}\binom{2k}{l} I\left(\frac{r^2}{1+\frac{2\rho}{N}l}-k\right) \\ &= \sum_{k=1}^{N}(8k+32)\sum_{l=0}^{\min(2k,\lfloor\frac{N}{2\rho}(\frac{r^2}{k}-1)\rfloor)}\binom{2k}{l} \end{aligned}$$

where the upper limit for the sum over l is obtained by solving $\frac{r^2}{1+\frac{2\rho}{N}l} = k$. We now propose to approximate the inner summation as follows. If in the inner summation l goes upto or beyonk k (i.e. if $\binom{2k}{k}$ appears in the summation) we approximate the sum to be 2^{2k}. If l does *not* go upto k we ignore the summation. With this, the expression becomes

$$C \approx \sum_{k=1}^{k^*}(8k+32)2^{2k}$$

where k^* is obtained by solving $\min(2k^*, \lfloor\frac{N}{2\rho}(\frac{r^2}{k^*}-1)\rfloor) = k^*$. Note that k^* has the interpretation that it is the largest sub-dimension for which all the points in that dimension are in the search region i.e. for all sub-dimensions upto k^* no pruning has occurred. Also, with this approximation, the complexity of the algorithm will be $\approx 2^{2k^*}$. If we assume $r^2 = N + \delta \log N$ we have

$$\begin{aligned} k^* &= \frac{N}{4\rho}\left(-1+\sqrt{1+8\rho+\frac{8\rho\delta\log N}{N}}\right) \\ &\approx \frac{N}{4\rho}\left(-1+\sqrt{1+8\rho}\left(1+\frac{1}{2}\frac{8\rho\delta\log N}{N(1+8\rho)}\right)\right) \\ &= \frac{N}{4\rho}(\sqrt{1+8\rho}-1)+\frac{\delta\log N}{\sqrt{1+8\rho}} \end{aligned}$$

where, for the approximation, we use the (reasonable) assumption that $\frac{8\rho\delta\log N}{N}$ is small compared to $(1+8\rho)$ for large N. With this, the complexity exponent is $\approx \frac{N}{4\rho}(\sqrt{1+8\rho}-1)$. This is always smaller than N and represents the computational savings of the sphere decoder over decoding by exhaustive search. However, for large enough N and fixed ρ, it still represents an exponential search.

Let us now look at the upper bound on the expression for the complexity of the IRA for $N = M$ and $L = 2$. From Theorem 1 we have

$$C \leq \sum_{k=1}^{N}(8k+32)\sum_{l=0}^{2k}\binom{2k}{l}\Gamma\left(\frac{r_k^2}{1+\frac{2\rho}{N}l}, k\right)$$

With approximations similar to the preceeding we get

$$C \approx \sum_{k=1}^{k^*}(8k+32)2^{2k}$$

where k^* now is obtained by solving $\min(2k^*, \lfloor \frac{N}{2\rho}(\frac{r_{k^*}^2}{k^*} - 1)\rfloor) = k^*$. Note that k^* here has the same interpretation as before. If we assume $r_k^2 = k + \delta \log N$ it follows that

$$k^* = \sqrt{\frac{N\delta \log N}{2\rho}}$$

which gives a complexity exponent of $\approx \sqrt{\frac{N\delta \log N}{2\rho}}$.

Although this is not a polynomial time complexity, by comparing the complexity exponents for the two algorithms it is evident that the Increasing Radii Algorithm offers huge computational savings compared to the sphere decoder. This is inspite of the fact that the values of δ in the two analyses may differ somewhat since, in both cases, δ is chosen to make $\epsilon = \epsilon_0$.

It would be interesting to obtain a similar approximate expression for complexity exponent of the Ellipsoid Pruning Algorithm.

10.2 Finding optimum values of r_i

In Section 9, where we presented the simulations, we used a fixed functional for the radii viz. $r_i = i + \delta \log N$ for IRA and $r_i = m(i-1)^{0.4} + b$ for EPA. However, for a fixed value of ϵ there is an entire family of schedules satisfying (21) or (24). Each of these schedules will have a particular complexity associated with it and one (or more) of them will have the lowest possible complexity. By predetermining a functional form we are probably using a schedule that is not optimal in the sense of complexity although it does give BER as close to optimal as we desire.

Therefore, for a fixed ϵ, we are interested in finding a schedule of r_i that minimizes the complexity. Since we do not have exact expressions for the complexity of either algorithm, we propose to minimize the expression for the upperbound or the approximation. In particular, for the Ellipsoid Pruning Algorithm, we attempt to minimize the upper bound obtained using (40) viz.

$$f(r_1, \ldots, r_M) = \sum_{k=1}^{M} (8k + 32) \sum_{s^k \in \mathcal{S}^{k \times 1}} e^k \prod_{i=1}^{k} \frac{(1 + \frac{k}{r_i^2 c_{i-1}})^{i-1}}{(1 + \frac{k}{r_i^2 c_i})^i} \tag{51}$$

We now observe that f is a convex function in the $1/r_i^2$ variables. Define $l_i = 1/r_i^2$. The optimization problem now becomes

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{k=1}^{M} (8k+32) \sum_{s^k \in \mathcal{S}^{k\times 1}} e^k \prod_{i=1}^{k} \frac{(1+\frac{kl_i}{c_{i-1}})^{i-1}}{(1+\frac{kl_i}{c_i})^i} \\ \text{subject to} \quad & \epsilon_{EP} \le \epsilon_0 & (52) \\ & -l_i \le 0 \quad \forall i & (53) \end{aligned}$$

where $\epsilon_{EP} = \sum_{i=1}^{M} \frac{e^{-1/l_i}}{\prod_{k=1, k\neq i}^{N} (1 - \frac{l_k}{l_i})}$.

Were it not for the first constraint, this would a convex problem. We can find a local minimum by a gradient descent technique or a stochastic search technique like simulated annealing. However, it would be more interesting to find a formulation of this problem that can be solved exactly by numerical or analytical methods. This may be possible by using a different constraint entirely or by using approximations to ϵ that are convex in l_i.

10.3 Future Work

While the preceeding section gives some idea of how to go about finding an optimum schedule for the EPA, finding an optimal schedule for the IRA seems to be more challenging since the complexity expressions we have (the upper bound as well as the approximation) are not analytically tractable.

In the context of finding optimal schedules, more "analytic" expressions for the complexity as well as the BER would be of interest since these might lead to analytic expressions for the optimal schedules and help quantify more satisfactorily the trade-off between performance and complexity.

The sphere decoding technique can be used for joint detection and decoding of block codes [15]. By analogy, the modified algorithms are also applicable in this context. Analysis of performance and complexity in this scenario is interesting.

Another question of interest is the matter of choosing radii based on the known H. Clearly, the smallest region around x that contains the closest point depends on H as well as v and the current choice of r_i only takes the statistics of v into consideration and is independent of H. However, this strategy might make the problem much harder to tackle.

10.4 Conclusions

In this paper we have looked at the Integer Least Squares problem in a probabilistic setting. Because of this the complexity of decoding is a random variable. Also, because of the statistics of the problem we are in a position to prune the search space so that we reduce the complexity while still keeping the transmitted point in the search region with high probability.

We have proposed two methods of doing this pruning and obtained "semi-analytic" expressions for the complexity and the probability of error of both. These give some insight about the optimal schedule.

The algorithms give significant computational savings relative to the sphere decoder while still maintaining BERs close to optimal. For example, for a real problem in 100 dimensions we can decode with upto 240 times less computation.

We believe that this approach to the problem demonstrates promise and that further work to analyze and optimize these statistical techniques will be of practical and theoretical interest.

A Derivation of (11) and (12)

For any $u \in \mathcal{C}^{T\times 1}$, we define $u^i = [u_{T-i+1}, \ldots, u_T]^T$. Consider the $H = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$ decomposition where H is $N \times M$ with i.i.d. $\mathcal{CN}(0, \sigma_h^2)$ entries, Q is unitary of size $N \times N$ and R is upper triangular of size $M \times M$. It can be shown that the non-diagonal entries of R are i.i.d. $\mathcal{CN}(0, \sigma_h^2)$ and the diagonal element $R(i,i)$ is a scaled χ-square distributed random variable. (Refer [16].) More specifically, $2R(i,i)/\sigma_h^2$ is χ-square with $2(N-i+1)$ degrees of freedom. This means that it is the sum of squares of $2(N-i+1)$ i.i.d. standard real Gaussian random variables i.e. variables having a $\mathcal{N}(0,1)$ distribution.

Therefore a right lower square submatrix of $\begin{bmatrix} R \\ 0 \end{bmatrix}$ of size $(i+N-M) \times i$, say R_i, is statistically similar to it i.e. it can be thought of as having arisen from the QR decomposition of an $(i+N-M) \times i$ matrix H_i having i.i.d. $\mathcal{CN}(0, \sigma_h^2)$ entries (not necessarily a submatrix of H) i.e. $H_i = Q_i \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix}$ where Q_i is unitary of size $(i+N-M) \times (i+N-M)$.

Recall z from Section 4. We have $z = Q^* x - \begin{bmatrix} R \\ 0 \end{bmatrix} s = \begin{bmatrix} R \\ 0 \end{bmatrix} (\tilde{s} - s) + Q^* v$. Define $w = Q^* v$. Clearly, w has the same statistics as v i.e. i.i.d. $\mathcal{CN}(0,1)$ entries. Introduce $v_i = Q_i w^{i+N-M}$. Now v_i is of length $(i+N-M)$. (It is not necessarily a sub-vector of v.) Moreover, v_i will also have i.i.d. $\mathcal{CN}(0,1)$

entries. We can now write w^{i+N-M} as $Q_i^* v_i$.

Define $\gamma_i = \sum_{j=1}^{i+N-M} \lambda_j$ for $i = 1, \ldots, M$. Note that γ_i is the squared norm of z^{i+N-M}. Also, we have s^i and $\tilde{s}^i$ as the lower length-i subvectors of s and $\tilde{s}$ respectively. From the above arguments, we have $z^{i+N-M} = \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix} (\tilde{s}^i - s^i) + Q_i^* v_i$. Therefore

$$\begin{aligned}
\gamma_i = \|z^{i+N-M}\|^2 &= \left\| \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix} (\tilde{s}^i - s^i) + Q_i^* v_i \right\|^2 \\
&= \left\| Q_i \begin{bmatrix} R_i \\ 0_{N-M,i} \end{bmatrix} (\tilde{s}^i - s^i) + Q_i Q_i^* v_i \right\|^2 \\
&= \left\| H_i(\tilde{s}^i - s^i) + v_i \right\|^2
\end{aligned}$$

But it is clear that the vector $H_i(\tilde{s}^i - s^i) + v_i$ has i.i.d. $\mathcal{CN}(0, \sigma_v^2 + \sigma_h^2 \|s^i - \tilde{s}^i\|^2)$ i.e. $\mathcal{CN}(0, 1/c_i)$ entries. Therefore γ_i is a scaled χ-square distributed random variable. More specifically, $2c_i\gamma_i$ is χ-square with $2i$ degrees of freedom. This means that it is the sum of squares of $2i$ i.i.d. standard real Gaussian random variables i.e. variables having a $\mathcal{N}(0,1)$ distribution. The expressions for the characteristic function of these are standard and we have $Ee^{j\alpha\gamma_i} = \frac{1}{(1 - \frac{j\alpha}{c_i})^{i+N-M}}$.

For λ_i where $i > (N-M)$, note that $\gamma_{i-N+M} = \lambda_i + \gamma_{i-1-N+M}$. Moreover, since the λ_is are independent, so are λ_i and $\gamma_{i-1-N+M}$. Therefore $Ee^{j\alpha\gamma_{i-N+M}} = Ee^{j\alpha\lambda_i + j\alpha\gamma_{i-1-N+M}} = Ee^{j\alpha\lambda_i} Ee^{j\alpha\gamma_{i-1-N+M}}$. Thus

$$\begin{aligned}
Ee^{j\alpha\lambda_i} &= \frac{Ee^{j\alpha\gamma_{i-N+M}}}{Ee^{j\alpha\gamma_{i-1-N+M}}} \\
&= \frac{(1 - \frac{j\alpha}{c_{i-1-N+M}})^{i-1}}{(1 - \frac{j\alpha}{c_{i-N+M}})^{i}}
\end{aligned} \tag{A.1}$$

For $i \leq (N-M)$ it is easy to see that λ_i is the squared norm of the the $(N-i+1)$-th entry of $Q^* v$. $Q^* v$

has the same statistics as v i.e. i.i.d. entries, each with distribution $\mathcal{CN}(0,1)$. With this the characteristic function of λ_i is clearly $\frac{1}{1-\frac{j\alpha}{c_0}}$.

This is (11). Fourier inversion of this gives us (12).

B Derivation of generating function in (31)

Theorem 5. *For $s, \tilde{s} \in \mathcal{S}^{k\times 1}$, the 'average' number of solutions to $\|s^k - \tilde{s}^k\|^2 = n$ (as defined by $r_k^L(n)$ in Section 8.1.1) is given by the coefficient of x^n in $(G_L(x))^k$ where $G_L(x) = \frac{1}{L^2}\left(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2}\right)^2$. Recall that $\mathcal{S} = \left\{a + jb \mid a, b \in \left\{-\frac{L-1}{2}, -\frac{L-3}{2}, \cdots, \frac{L-3}{2}, \frac{L-1}{2}\right\}\right\}$.*

Proof. For any complex vector r of length k define the vector r_b as a real vector of length $2k$ where $r_b(2j-1) = \Re(r(j))$ and $r_b(2j) = \Im(r(j))$ for $j = 1, \ldots, k$.

Let $r = s^k - \tilde{s}^k$ where $s^k, \tilde{s}^k \in \mathcal{S}^{k\times 1}$ and $\mathcal{S}$ is as defined in (2). Then define r_b as above. Also define $\mathcal{S}_b = \left\{-\frac{L-1}{2}, -\frac{L-3}{2}, \cdots, \frac{L-3}{2}, \frac{L-1}{2}\right\}$.

Consider an arbitrary entry of r_b, say $r_b(j)$. For a fixed $\tilde{s}^k$, $\tilde{s}_b^k(j)$ is known. Say $\tilde{s}_b^i(j) = t \in \mathcal{S}_b$, then $r_b(j)$ takes all values in $\mathcal{S}_t = \mathcal{S}_b - t$. Define $q_t = \sum_{j\in\mathcal{S}_t} x^{j^2}$ $\forall t \in \mathcal{S}_b$. Associate with a fixed vector $\tilde{s}^k$ the product $q(\tilde{s}^k) = \prod_{j=1}^{2k} q_{\tilde{s}_b^k(j)}$. Clearly, for this fixed $\tilde{s}^k$, the number of solutions to $\|s^k - \tilde{s}^k\|^2 = n$ is the coefficient of x^n in $q(\tilde{s}^k)$.

Since all the L^{2k} possible $\tilde{s}^k \in \mathcal{S}^{k\times 1}$ are assumed equally likely, the 'average' number of solutions to

$\|s^k - \tilde{s}^k\|^2 = n$ is given by the coefficient of x^n in

$$\begin{aligned}
\frac{1}{L^{2k}} \sum_{\tilde{s}^k \in \mathcal{S}^{k\times 1}} q(\tilde{s}^k) &= \frac{1}{L^{2k}} \sum_{\tilde{s}^k \in \mathcal{S}^{k\times 1}} \prod_{j=1}^{2k} q_{\tilde{s}_b^k(j)} \\
&= \frac{1}{L^{2k}} \sum_{\sum_{t\in\mathcal{S}_b} \alpha_t = 2k} \binom{2k}{\{\alpha_t | t \in \mathcal{S}_b\}} \prod_{t\in\mathcal{S}_b} q_t^{\alpha_t} \\
&= \frac{1}{L^{2k}} \left(\sum_{t\in\mathcal{S}_b} q_t \right)^{2i} \\
&= \frac{1}{L^{2k}} \left(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2} \right)^{2i}
\end{aligned}$$

where $\binom{2k}{\{\alpha_t | t\in\mathcal{S}_b\}}$ is the multinomial coefficient

$$\binom{2i}{\alpha_{\frac{-(L-1)}{2}}, \alpha_{\frac{-(L-3)}{2}}, \cdots, \alpha_{\frac{L-3}{2}}, \alpha_{\frac{L-1}{2}}}$$

Finally, we define $G_L(x) = \frac{1}{L^2}\left(L + \sum_{j=1}^{L-1} 2(L-j)x^{j^2}\right)^2$. □

We note here that this is closely related to the problem of representing integers as a sum of squares. For more on this refer [12].

C Derivations for Section 8.1.2

C.1 Proof of (35) and (36)

Let $I(\Delta) = \int_0^\Delta f(x)g(x)dx$. Let $U(x) = \int_0^x f(t)dt$. Let $h(U(x)) = x$. Then,

$$\begin{aligned}
I(\Delta) &= \int_0^{U(\Delta)} g(h(U))dU \\
&= G(U(\Delta)) - G(0)
\end{aligned}$$

where $\frac{dG(x)}{dx} = g(h(x))$. Using Taylor expansions for $G(U(\Delta))$ and $G(0)$ around $\frac{1}{2}U(\Delta)$ we have

$$\begin{aligned} I(\Delta) &= U(\Delta)\frac{dG(U(x))}{dU(x)}\Big|_{U(x)=\frac{1}{2}U(\Delta)} + O([U(\Delta)]^3) \\ &= g(x^{'})\int_0^{\Delta} f(x)dx + O\Big(\Big[\int_0^{\Delta} f(x)dx\Big]^3\Big) \end{aligned}$$

where $x^{'}$ is obtained by solving

$$\int_0^{x^{'}} f(t)dt = \frac{1}{2}\int_0^{\Delta} f(t)dt$$

In fact, for any $x^{'} \in [0, \Delta]$ the following holds

$$I(\Delta) = g(x^{'})\int_0^{\Delta} f(x)dx + O\Big(\Big[\int_0^{\Delta} f(x)dx\Big]^2\Big)$$

For $\int_0^{\Delta} f(x)dx \ll 1$ this implies

$$I(\Delta) \approx g(x^{'})\int_0^{\Delta} f(x)dx \tag{C.2}$$

C.2 Proof of (32) and (33)

Define $\beta_{i,j} = \sum_{k=i}^{j} \lambda_{k+N-M}$ for $1 \le i \le j \le M$ where λ_i is as defined in Section(4).

$$P(s^k \in \mathcal{D}_{IR,k}) = P(\beta_{1,1} \le r_1^2, \ldots, \beta_{1,k} \le r_k^2) = \int_0^{\infty} P(\beta_{1,1} \le r_1^2, \ldots, \beta_{1,k} \le r_k^2|\beta_{1,1})p_{\beta_{1,1}}(\beta_{1,1})d\beta_{1,1} \tag{C.3}$$

Note that $\beta_{i,i} \to \beta_{i,i+1} \to \ldots \to \beta_{i,j}$ with $1 \le i \le j \le M$ is a Markov chain. Hence

$$P(s^k \in \mathcal{D}_{IR,k}) = \int_0^{\infty} P(\beta_{1,1} \le r_1^2|\beta_{1,1})p_{\beta_{1,1}}(\beta_{1,1}) \cdot P(\beta_{1,2} \le r_1^2, \ldots, \beta_{1,k} \le r_k^2|\beta_{1,1})d\beta_{1,1}$$

With $P(\beta_{1,1} \leq r_1^2|\beta_{1,1})p_{\beta_{1,1}}(\beta_{1,1})$ as $f(\beta_{1,1})$ and $P(\beta_{1,2} \leq r_1^2, \ldots, \beta_{1,k} \leq r_k^2|\beta_{1,1})$ as $g(\beta_{1,1})$ we use (C.2) to get

$$\begin{aligned} P(s^k \in \mathcal{D}_{IR,k}) &\approx P(\beta_{1,1} \leq r_1^2)P(\beta_{1,2} \leq r_2^2, \ldots, \beta_{1,k} \leq r_k^2|\beta_{1,1} = \beta_{1,1}^{'}) \\ &= P(\beta_{1,1} \leq r_1^2)P(\beta_{2,2} \leq r_2^2 - \beta_{1,1}^{'}, \ldots, \beta_{2,k} \leq r_k^2 - \beta_{1,1}^{'}) \end{aligned}$$

where the optimum $\beta_{1,1}^{'}$ is obtained by solving $\int_0^{\beta_{1,1}^{'}} p_{\beta_{1,1}}(\beta_{1,1})d\beta_{1,1} = \frac{1}{2}\int_0^{r_1^2} p_{\beta_{1,1}}(\beta_{1,1})d\beta_{1,1}$ and the error is $O\left(\left[P(\beta_{2,2} \leq r_2^2 - \beta_{1,1}^{'}, \ldots, \beta_{2,k} \leq r_k^2 - \beta_{1,1}^{'})\right]^3\right)$.

Observe that the second term on the RHS is of the same form as the expression for $P(s^k \in \mathcal{D}_{IR})$ in (C.3) and can be similarly approximated. In order to formalize this recursion, define $W_i = P(\beta_{i,j} \leq r_i^2 - \sum_{l=1}^{j-1}\beta_{l,l}^{'}$ for $j = i, \ldots, k)$ and $q_i = P(\beta_{i,i} \leq r_i^2 - \sum_{l=1}^{i-1}\beta_{l,l}^{'})$ where the $\beta_{i,i}^{'}$s are obtained successively by solving

$$\int_0^{\beta_{i,i}^{'}} p_{\beta_{i,i}}(\beta_{i,i})d\beta_{i,i} = \frac{1}{2}\int_0^{r_i^2 - \sum_{l=1}^{i-1}\beta_{l,l}^{'}} p_{\beta_{i,i}}(\beta_{i,i})d\beta_{i,i} \tag{C.4}$$

The general recursion we have is

$$W_{t-1} = q_{t-1}W_t + O(W_t^3) \qquad \text{for} \quad t = 2, \ldots, k$$

where we start with $W_k < 1$. It is easy to see that we get $W_1 = q_1 \cdot q_2 \cdots q_{k-1}W_k + O(W_k^3)$. But $W_k = q_k$. Therefore $P(s^k \in \mathcal{D}_k) = W_1 = q_1 \cdot q_2 \cdots q_k + O(q_k^3)$. Note that if we do not solve (C.4) exactly, this approximation still holds, but with error being $O(q_k^2)$.

To summarize

$$P(s^k \in \mathcal{D}_k) = \prod_{i=1}^{k} P\left(\beta_{i,i} \leq r_i^2 - \sum_{l=1}^{i-1}\beta_{l,l}^{'}\right) + O\left(\left[P(\beta_{k,k} \leq r_k^2 - \sum_{l=1}^{k-1}\beta_{l,l}^{'})\right]^3\right)$$

where the $\beta_{i,i}^{'}$s are as obtained in (C.4). Define $X_i = r_i^2 - \sum_{l=1}^{i-1}\beta_{l,l}^{'}$. Note that $\beta_{i,i} = \lambda_{i+N-M}$. This gives us (32).

C.3 Solving (33) approximately to get (34)

Define $X_i' = r_i^2 - r_{i-1}^2 - X_i + X_{i-1}$. We rewrite (33) as

$$E_{\lambda_{i+N-M}} I(X_i' - \lambda_i) = \frac{1}{2} E_{\lambda_{i+N-M}} I(X_{i-1} - \lambda_i)$$

To solve this approximately we use Chernoff bounds on both sides.

$$E_{\lambda_{i+N-M}} e^{\nu(X_i' - \lambda_i)} = \frac{1}{2} E_{\lambda_{i+N-M}} e^{\nu(X_{i-1} - \lambda_i)}, \qquad \nu \geq 0 \tag{C.5}$$

Note that we use the same parameter ν on both sides. To determine the optimum value of ν we try to minimize the LHS.

$$\begin{aligned}
\frac{d}{d\nu} E_{\lambda_{i+N-M}} e^{\nu(X_i' - \lambda_i)} &= \frac{d}{d\nu} e^{\nu X_i'} \frac{(1 + \frac{\nu}{c_{i-1}})^{i-1+N-M}}{(1 + \frac{\nu}{c_i})^{i+N-M}} \\
&= \left(X_i' + \frac{i-1+N-M}{\nu + c_{i-1}} - \frac{i+N-M}{\nu + c_i} \right) e^{\nu X_i'} \frac{(1 + \frac{\nu}{c_{i-1+N-M}})^{i-1}}{(1 + \frac{\nu}{c_i})^{i+N-M}} \\
&\approx \left(X_i' + \frac{i-1+N-M}{\nu + c_{i-1}} - \frac{i+N-M}{\nu + c_{i-1}} \right) e^{\nu X_i'} \frac{(1 + \frac{\nu}{c_{i-1}})^{i-1+N-M}}{(1 + \frac{\nu}{c_i})^{i+N-M}} \\
&= \left(X_i' - \frac{1}{\nu + c_{i-1}} \right) e^{\nu X_i'} \frac{(1 + \frac{\nu}{c_{i-1+N-M}})^{i-1}}{(1 + \frac{\nu}{c_{i+N-M}})^{i}}
\end{aligned}$$

Now $\frac{d}{d\nu} E_{\lambda_{i+N-M}} e^{\nu(X_i' - \lambda_{i+N-M})} = 0$ gives us $\nu = \frac{1}{X_i'} - c_{i-1}$.

Substitution in (C.5) gives

$$c_{i-1} X_i'^2 - (1 + \log 2 + c_{i-1} X_{i-1}) X_i' + X_{i-1} = 0$$

This is a quadratic in $X_i^{'}$ with two real roots. We require $X_i^{'} \geq 0$ and hence take the greater root of the two roots. This gives us the recursion in (33). With this value of $X_i^{'}$ it is easy to check that $\nu \geq 0$ is satisfied.

C.4 Approximation in (37)

The following is a non-rigourous derivation of the approximation in (37).

$$
\begin{aligned}
Ee^{j\alpha\lambda_{i+N-M}} &= \frac{(1-\frac{j\alpha}{c_{i-1}})^{i-1+N-M}}{(1-\frac{j\alpha}{c_i})^{i+N-M}} \\
&= \frac{1}{1-\frac{j\alpha}{c_i}}\left(1+\frac{j\alpha\sigma_h^2|\tilde{s}_i-s_i|^2}{1-\frac{j\alpha}{c_i}}\right)^{i-1+N-M} \\
&\approx \frac{1}{1-\frac{j\alpha}{c_i}}\exp\frac{j\alpha(i-1+N-M)\sigma_h^2|\tilde{s}_i-s_i|^2}{1-\frac{j\alpha}{c_i}}
\end{aligned}
$$

But this is the characteristic function of a non-central χ-square distributed random variable i.e. $\lambda = \alpha_{i,1}^2+\alpha_{i,2}^2$ where are $\alpha_{i,1}$ and $\alpha_{i,2}$ are i.i.d. real gaussians $\mathcal{N}(\sqrt{(i-1+N-M)\sigma_h^2|\tilde{s}_i-s_i|^2/2}, 1/2c_i) = \mathcal{N}(\mu, 1/2c_i)$ say. We now have

$$
\begin{aligned}
\int_0^{X_i} p_{\lambda_i}(\lambda_i)d\lambda_i &= E(I(\lambda_i \leq X_i)) \\
&\approx E(I(\alpha_{i,1}^2+\alpha_{i,2}^2 \leq X_i)) && \text{(C.6)} \\
&\approx E(I(|\alpha_{i,1}| \leq \sqrt{\pi X_i}/2)I(|\alpha_{i,2}| \leq \sqrt{\pi X_i}/2)) && \text{(C.7)} \\
&= E(I(|\alpha_{i,1}| \leq \sqrt{\pi X_i}/2))E(I(|\alpha_{i,2}| \leq \sqrt{\pi X_i}/2)) && \text{(C.8)} \\
&= (Q(\sqrt{2c_i}(\mu-\sqrt{\pi X_i}/2)) - Q(\sqrt{2c_i}(\mu+\sqrt{\pi X_i}/2)))^2
\end{aligned}
$$

where $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-x^2/2}dx$ is the standard Q-function. Here, (C.6) comes from the preceeding discussion, (C.7) is an approximation obtained by replacing the disc $\alpha_{i,1}^2+\alpha_{i,2}^2 \leq X_i$ by a square with the same center and area as the disc, viz. $|\alpha_{i,1}| \leq \sqrt{\pi X_i}/2, |\alpha_{i,2}| \leq \sqrt{\pi X_i}/2$ and (C.8) uses independence of $\alpha_{i,1}$ and $\alpha_{i,2}$.

D Proof of upperbound in (46)

For $k \geq 2$, we have the following sequence of inequalities

$$\begin{aligned}
P(s^k \in \mathcal{D}_{EP,k}) &\leq e^{\nu_k} \prod_{i=1}^{k} \frac{(1+\frac{\nu_k}{r_i^2 c_{i-1}})^{i-1+N-M}}{(1+\frac{\nu_k}{r_i^2 c_i})^{i+N-M}} && \text{from (42)} \\
&= \frac{e^{\nu_k}}{\left(1+\frac{\nu_k}{c_k r_k^2}\right)^{k+N-M}} \prod_{i=1}^{k-1} \left(\frac{1+\frac{\nu_k}{c_i r_{i+1}^2}}{1+\frac{\nu_k}{c_i r_i^2}} \right)^{i+N-M} \\
&= \frac{e^{\nu_k}}{\left(1+\frac{\nu_k}{c_k r_k^2}\right)^{k+N-M}} \prod_{i=1}^{k-1} \left(1 - \frac{\frac{\nu_k}{c_i}\left(\frac{1}{r_i^2} - \frac{1}{r_{i+1}^2}\right)}{1+\frac{\nu_k}{c_i r_i^2}} \right)^{i+N-M} \\
&\leq \frac{e^{\nu_k}}{\left(1+\frac{\nu_k}{c_k r_k^2}\right)^{k+N-M}} \prod_{i=1}^{k-1} \exp\left(\frac{-(i+N-M)\frac{\nu_k}{c_i}\left(\frac{1}{r_i^2} - \frac{1}{r_{i+1}^2}\right)}{1+\frac{\nu_k}{c_i r_i^2}} \right) && \text{(D.9)} \\
&= \frac{e^{\nu_k}}{\left(1+\frac{\nu_k}{c_k r_k^2}\right)^{k+N-M}} \prod_{i=1}^{k-1} \exp\left(-(i+N-M)\left(1 - \frac{r_i^2}{r_{i+1}^2}\right)\right) \times && \text{(D.10)} \\
&\qquad \prod_{i=1}^{k-1} \exp\left(\frac{(i+N-M)\left(1-\frac{r_i^2}{r_{i+1}^2}\right)}{\left(1+\frac{\nu_k}{c_i r_i^2}\right)} \right) \\
&\leq \frac{e^{\nu_k}}{\left(1+\frac{\nu_k}{c_k r_k^2}\right)^{k+N-M}} \prod_{i=1}^{k-1} \exp\left(-(i+N-M)\left(1 - \frac{r_i^2}{r_{i+1}^2}\right)\right) \times && \text{(D.11)} \\
&\qquad \frac{1}{(k-1)} \sum_{i=1}^{k-1} \exp\left(\frac{(i+N-M)(k-1)\left(1-\frac{r_i^2}{r_{i+1}^2}\right)}{\left(1+\frac{\nu_k}{c_i r_i^2}\right)} \right)
\end{aligned}$$

Here (D.9) is obtained using the simple inequality $(1-x)^i \leq e^{-ix}$ for $x \in [0,1]$, $i \in \mathcal{Z}^+$ and (D.11) is obtained using the standard inequality stating that Geometric Mean $\leq$ Arithmetic Mean for positive real numbers (AM-GM inequality) on the second product of (D.10).

References

[1] 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.

[2] 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.

[3] M. O. Damen, A. Chkeif, and J.-C. Belfiore, “Lattice code decoder for space-time codes,” *IEEE Comm. Let.*, pp. 161–163, May 2000.

[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] R. Kannan, “Improved algorithms on integer programming and related lattice problems,” *Proc. 15th Annu. ACM Symp. on Theory of Computing*, pp. 193–206, 1983.

[6] 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.

[7] 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.

[8] J. Gross and J. Yellen, *Graph Theory and its Applications*. CRC Press, 1998.

[9] A. Banihashemi and A. Khandani, “On the complexity of decoding lattices using the Korkin-Zolotarev reduced basis,” *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 162–171, 1998.

[10] E. Agrell, A. Vardy, and K. Zeger, “Closest point search in lattices,” *submitted to IEEE Transactions on Information Theory*, 2002.

[11] M. Ajtai, "Generating hard instances of lattice problems," *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 99–108, 1996.

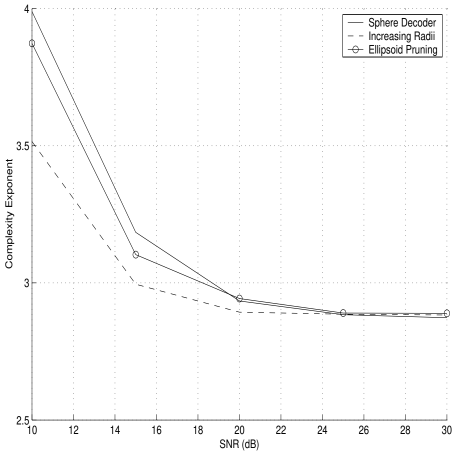
[12] B. Hassibi and H. Vikalo, "On sphere decoding algorithm. I. Expected complexity II. Variance, Generalizations, and Applications to Communications," *Submitted to the IEEE Transactions on Signal Processing*, 2003.

[13] M. Grotschel, L. Lovasz, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag, 2nd ed., 1993.

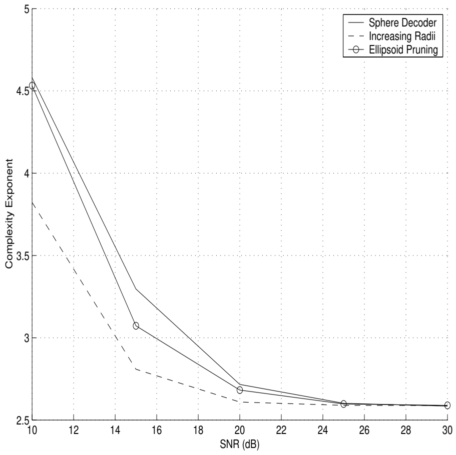
[14] M. Ajtai, "The shortest vector problem in L_2 is NP-hard for randomized reductions," *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 10–19, 1998.

[15] H. Vikalo and B. Hassibi, "On joint ML detection and decoding for linear block codes," in *Proc. IEEE ISIT*, 2003.

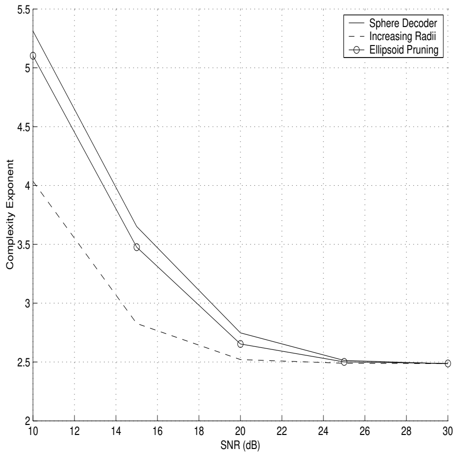
[16] A. Edelman, *Eigenvalues and Condition Numbers of Random Matrices*. PhD Thesis, MIT, Dept. of Math., 1989.



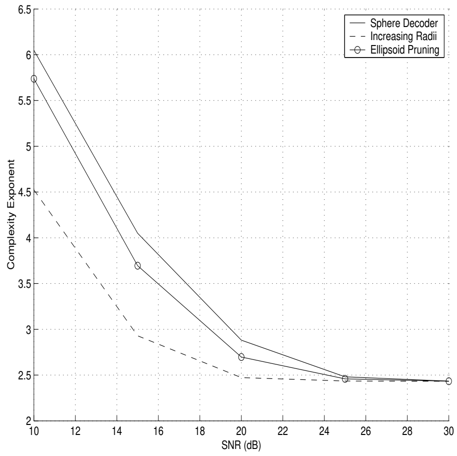
(a) N=10



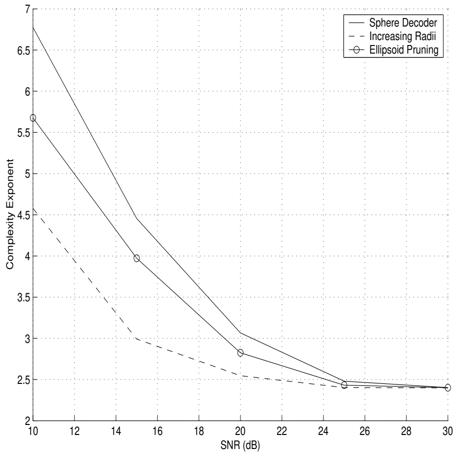
(b) N=20



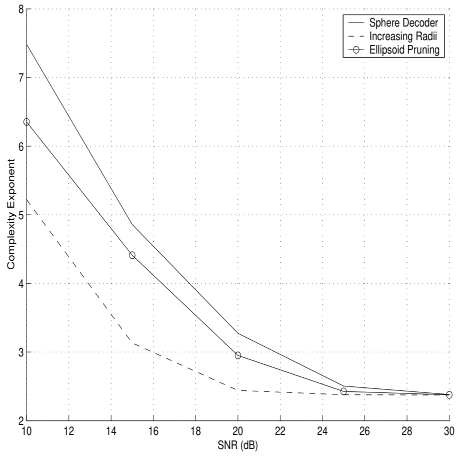
(c) N=30



(d) N=40

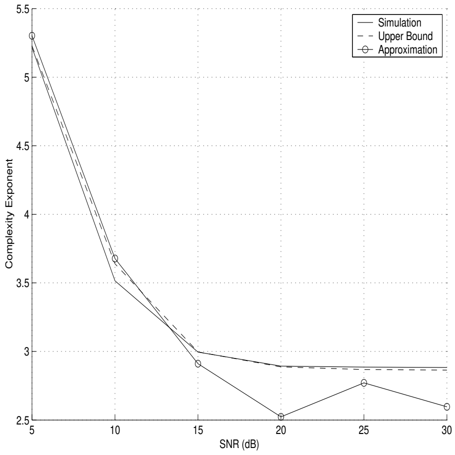


(e) N=50

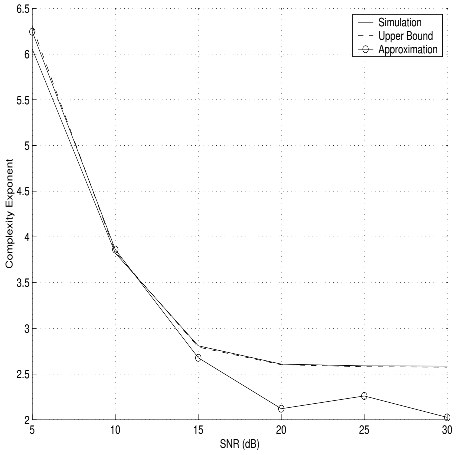


(f) N=60

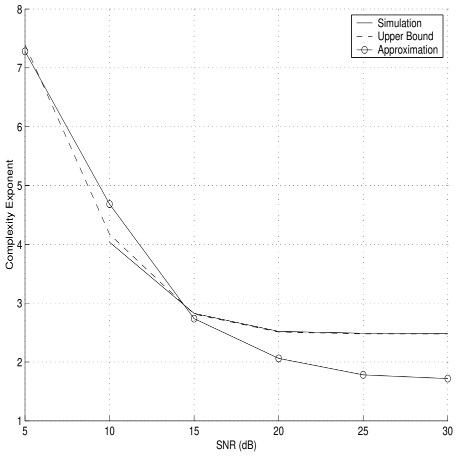
Figure 3: Complexity Exponents for three algorithms.



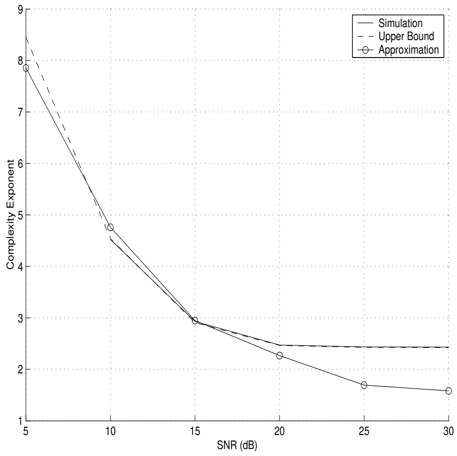
(a) N=10



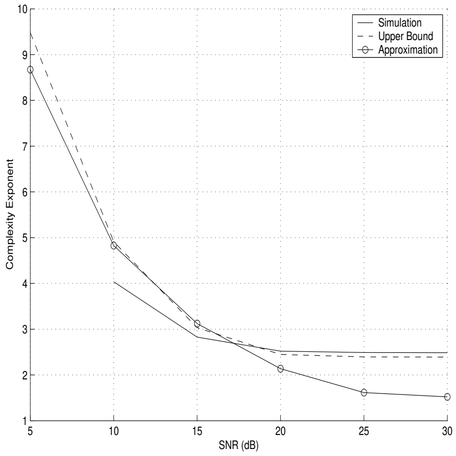
(b) N=20



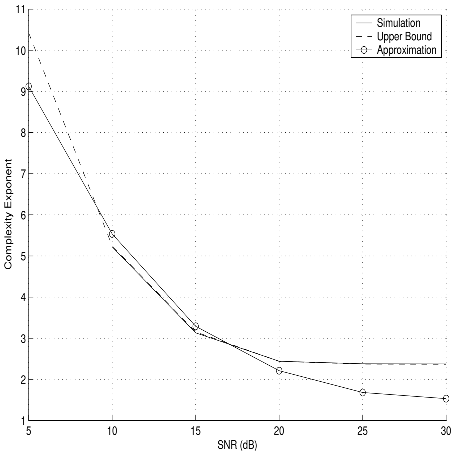
(c) N=30



(d) N=40

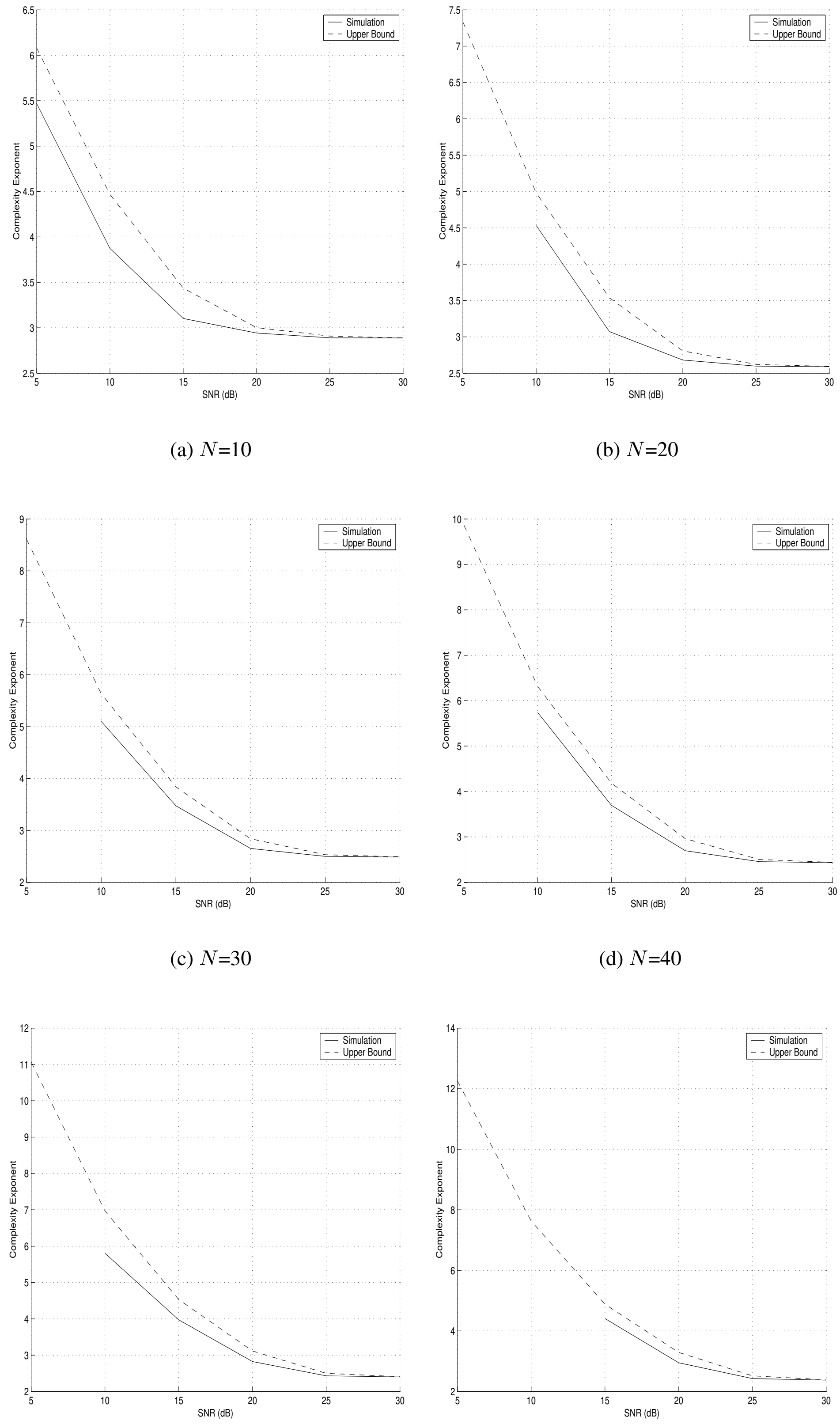


(e) N=50



(f) N=60

Figure 4: Complexity Exponent for the Increasing Radii Algorithm – simulated, upperbound and approximation.



(a) N=10

(b) N=20

(c) N=30

(d) N=40

(e) N=50

(f) N=60

Figure 5: Complexity Exponent for the Ellipsoid Pruning Algorithm – simulated and upperbound.